



universität  
wien

# DIPLOMARBEIT / DIPLOMA THESIS

Titel der Diplomarbeit / Title of the Diploma Thesis

## **Computerspielentwicklung mit Unity an der Sekundarstufe 2**

**Welche digitalen Kompetenzen werden vorausgesetzt  
und welche können erworben werden?**

verfasst von / submitted by

Mario Winkler

angestrebter akademischer Grad / in partial fulfilment of the requirements for  
the degree of

Magister der Naturwissenschaften (Mag.rer.nat.)

Wien, 2020 / Vienna, 2020

---

Studienkennzahl lt. Studienblatt /  
degree programme code as it appears on  
the student record sheet:

UA 190 313 884

---

Studienrichtung lt. Studienblatt /  
degree programme as it appears on  
the student record sheet:

Lehramtsstudium Unterrichtsfach Geschichte,  
Sozialkunde und Politische Bildung,  
Unterrichtsfach Informatik und  
Informatikmanagement

---

Betreut von / Supervisor:  
Mitbetreut von / Co-Supervisor:

Univ.-Prof. Dipl.-Ing. Dr. Renate Motschnig  
Mag. Dr. techn. Oswald Comber



## Computerspielentwicklung mit Unity an der Sekundarstufe 2

Welche digitalen Kompetenzen werden vorausgesetzt und welche können erworben werden?

### 1 Inhalt

<b>1</b>	<b>Inhalt.....</b>	<b>3</b>
<b>2</b>	<b>Einleitung und Motivation.....</b>	<b>11</b>
<b>3</b>	<b>Forschungsrahmen .....</b>	<b>14</b>
	3.1 Projekt „Learn to proGrAME“ .....	14
	3.2 Bereich der Forschung und verwandte Forschung .....	15
	3.3 Forschungsfragen .....	19
	3.4 Vorhaben und methodische Vorgehensweise .....	20
<b>4</b>	<b>IT-Skills und -Kompetenzen.....</b>	<b>21</b>
	4.1 Definitionen .....	21
	4.1.1 Wissen.....	21
	4.1.2 Fähigkeiten.....	24
	4.1.3 Fertigkeiten.....	25
	4.1.4 Hard und Soft Skills.....	27
	4.1.5 Kompetenzen .....	29
	4.1.6 Qualifikationen.....	32
	4.1.7 KONZEPT-DUMMIE: Von der Fähigkeit zur Kompetenz.....	35
	4.2 Acht Schlüsselkompetenzen der EU (DigComp2.1) .....	38
	4.2.1 Die Idee der acht Schlüsselkompetenzen .....	38
	4.2.2 Empfehlung zu Schlüsselkompetenzen aus 2006.....	39
	4.2.3 Empfehlung zu Schlüsselkompetenzen aus 2018.....	39
	4.2.4 DigComp2.1 (European Digital Competence Framework).....	42
	4.2.5 DigComp2.2AT (österreichische Version) .....	43
	4.3 Kompetenzkonstrukt im Bildungswesen (digi.komp.at).....	50
	4.3.1 Entstehung und Ziel der digi.komp-Modelle .....	50
	4.3.2 Lehrpläne .....	52
	4.3.3 digi.komp12.....	52
	4.3.4 digi.komp12 inklusive Kompetenzstufen.....	54
	4.3.5 digi.komp8.....	55
	4.4 Alles unter einen Hut bringen .....	57
	4.4.1 Berufliche und allgemeine Bildung (DigComp vs. digi.komp) .....	58
	4.4.2 Bausteine der Kompetenzmodelle (Wissen, Fähigkeit, Fertigkeit) .....	61
	4.4.3 Programmierspezifisches: Wissen und Fertigkeiten.....	67
	4.4.4 dpKa (digitales, programmierspez. Kompetenzaufbau)-Modell .....	69

<b>5</b>	<b>Unterrichtskonzept</b> .....	<b>76</b>
5.1	Verwendete Tools .....	76
5.2	Aufbau des Unterrichtskonzeptes .....	77
5.3	Pädagogische Perspektive .....	82
<b>6</b>	<b>Empirischer Teil</b> .....	<b>85</b>
6.1	Methodisches Vorgehen.....	85
6.2	Fragebogen (quantitative Hinweise) .....	87
6.2.1	Fragebogen Teil 1 (Erwerb von Kompetenzbausteinen) .....	89
6.2.2	Fragebogen Teil 2 (Vergleich von Unterrichtskonzepten).....	94
6.3	Interview (qualitative Forschung) .....	100
6.3.1	Kategorienbildung .....	100
6.3.2	Interviewleitfaden .....	106
6.3.3	Interview- und Transkriptionsdurchführung .....	111
6.3.4	Auswertung .....	112
6.4	Ergebnisdarstellung.....	134
6.4.1	Verantwortung, Datenschutz und Datensicherheit .....	136
6.4.2	Technische Grundlagen und Funktionsweisen .....	137
6.4.3	Betriebssysteme und Software .....	138
6.4.4	Produktion digitaler Medien .....	139
6.4.5	Suche, Auswahl und Organisation von Information .....	141
6.4.6	Algorithmen, Datenstrukturen und Programmierung .....	142
6.5	Deutung und Diskussion der Ergebnisse .....	147
6.5.1	Forschungsfrage 1: Voraussetzungen für Unity .....	147
6.5.2	Forschungsfrage 2: Kompetenzerwerb in Unity.....	150
6.5.3	Ausblick und Reflexion der Empirie .....	155
<b>7</b>	<b>Zusammenfassung und Schlussbetrachtung</b> .....	<b>158</b>
<b>8</b>	<b>Abbildungsverzeichnis</b> .....	<b>160</b>
<b>9</b>	<b>Literaturverzeichnis</b> .....	<b>163</b>
<b>10</b>	<b>Anhang</b> .....	<b>171</b>
10.1	DigComp2.2AT – ergänzt die europ. Version 2.1 (25 Punkte) ...	171
10.2	Digi.komp12 – Wahlpflichtfach Informatik (81 Punkte).....	177
10.3	Digi.komp8 – inkl. schulauton. Vertiefung (105 Punkte).....	184
10.4	Digi.komp4 – Kompetenzmodell (49 Punkte).....	192
10.5	Fragebogen.....	195
10.6	Interviewvereinbarung .....	198
10.7	Interviews .....	199
10.7.1	Interview 01.....	199
10.7.2	Interview 02.....	225
10.7.3	Interview 03.....	262

## Eidesstattliche Erklärung

---

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Weiter habe ich mich bemüht, sämtliche Inhaber der Bildrechte zu identifizieren und ihre Zustimmung zur Verwendung der Bilder in dieser Arbeit eingeholt. Sollte dennoch eine Urheberrechtsverletzung bekannt werden, ersuche ich um Rückmeldung bei mir.

Diese Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und sie wurde auch nicht veröffentlicht.

Wien, Juli 2019

Ort, Datum

\_\_\_\_\_.

Unterschrift



## Danksagung

---

An dieser Stelle möchte ich mich bei allen bedanken, die mich während meines bisherigen Werdeganges in irgendeiner Weise unterstützt haben und/oder für ein wenig Freude im Leben gesorgt haben. Es sind doch auch so viele kleine Dinge im Leben, die einem zu etwas Besonderem verhelfen.

Ganz besonders bedanken möchte ich mich bei meinen Eltern Erich und Erika Winkler, meinem Bruder Bernd Winkler und meiner Freundin Sandra Thek, die immer an mich geglaubt haben.

Ein weiterer besonderer Dank gilt meinen beiden Diplomarbeitbetreuern Frau Univ.-Prof. Dipl.-Ing. Dr. Renate Motschnig und Herrn Mag. Dr. techn. Oswald Comber für die großartige Zusammenarbeit, sowohl fachlich als auch menschlich.

Außerdem möchte ich mich bei den Personen bedanken, die mir ein Stück ihrer Freizeit geschenkt haben, um mir die Interviews zu geben.





# Abstract

---

This survey looked for requirements in form of digital competence components for a successive game development in Unity and found them. In addition it examined the game development in Unity about the acquisition of competence components of three competence models, namely the digi.komp12, the digi.komp8 and the DigComp2.2AT, in a programming specific view. The survey was leaded by the following questions:

1. Which digital competencies, skills and knowledge do adolescent pupils need, to develop games in the lessons successfully with Unity?
2. Which digital competencies, skills and knowledge gets acquired through game development with Unity?

To answer the survey questions, the most important terms, like knowledge, skill and competence were defined for this work. After that, the focus was directed to the competence models, like the DigComp2.2AT, which is a derivative of the digital competencies of the EU and the digi.komp models “digi.komp8” (lower secondary education) and “digi.komp12” ((upper) secondary education), which are designed for the schooling in Austria. Then the most important competence components will be merged in a newly elaborated “digitales, programmierspezifischen Kompetenzaufbau-Modell” (dpKa-Modell), what means to be a deductive category formation. With this model the teaching concept, a questionnaire and the interviews of experts were examined. For the evaluation of the materials, the qualitative content analysis of Mayring was used. With the found requirements, the teaching concept can be adapted in the area of the preparing measures for Unity, so that the focus can be directed to the requirements and the teaching concept can be shaped more effectively in terms of timing. The examination of the game development in Unity about acquired competence components shows, how far the competence models are covered by the teaching concept.

## Kurzfassung

---

Mit dieser Studie wurden Voraussetzungen in Form von digitalen Kompetenzbausteinen für eine erfolgreiche Spieleentwicklung mit Unity untersucht und gefunden. Weiter wurden die Kompetenzbausteine von drei Kompetenzmodellen, nämlich dem des digi.komp12, des digi.komp8 und des DigComp2.2AT, im Bereich der Programmierung auf einen Erwerb innerhalb der Spielentwicklung mit Unity untersucht. Die Studie ging den Fragen nach:

1. Welche digitalen Kompetenzen und Fertigkeiten und welches Vorwissen benötigen jugendliche Schülerinnen und Schüler, um im Unterricht erfolgreich Computerspiele mit Unity zu entwickeln?
2. Welche digitalen Kompetenzen, Fertigkeiten und welches Wissen werden in der Computerspieleentwicklung mit Unity erworben?

Zur Beantwortung dieser Forschungsfragen wurden die wichtigsten Begriffe, wie Wissen, Fertigkeit und Kompetenz für diese Arbeit definiert. Danach wurde der Blick auf Kompetenzmodelle gerichtet, nämlich den DigComp2.2AT, der eine Weiterentwicklung der digitalen Kompetenzen der EU darstellt und auf die beiden digi.komp-Modelle „digi.komp8“ (Sekundarstufe I) und „digi.komp12“ (Sekundarstufe II), die für die österreichische Schulbildung konzipiert sind. Anschließend wurden die für diese Arbeit wichtigsten Kompetenzbausteine in einem neu erarbeiteten digitalen, programmierspezifischen Kompetenzaufbau-Modell (dpKa-Modell) zusammengefügt, wodurch eine deduktive Kategorienbildung stattfand. Damit wurde das Unterrichtskonzept, ein Fragebogen und die Experteninterviews untersucht. Ausgewertet wurde anhand der qualitativen Inhaltsanalyse nach Mayring. Mit den herausgearbeiteten Voraussetzungen kann man das Unterrichtskonzept im Bereich der für Unity vorbereitenden Maßnahmen dahingehend adaptieren, dass man sich gezielt auf die Voraussetzungen konzentriert und damit den Unterricht bezüglich der mangelnden Unterrichtszeit effektiver gestaltet. Die Untersuchung der Spieleentwicklung mit Unity auf darin erworbene Kompetenzbausteine zeigt auf, inwiefern die Kompetenzmodelle dadurch abgedeckt sind.

---

## 2 Einleitung und Motivation

Der Prozess des Lernens kann etwas so Wundervolles sein, doch leider hat man während der Schullaufbahn meistens eher das Gefühl, dass man mit Dingen gequält wird, die einen weder interessieren noch Spaß machen. Da in unserem Bildungssystem ein interessensbasiertes Lernen und damit meine ich, dass man sich seine Fächer bis auf die Hauptgegenstände Mathe, Deutsch und Englisch beispielsweise aussuchen kann, zumindest bis zum Tertiärbereich nicht etabliert ist, kann man gegen das Interessensproblem nichts tun. Doch wenn einem ein Fach nicht interessiert, dann soll es doch wenigstens Spaß machen, was wiederum dazu führen könnte, dass langsam ein Interesse dafür entsteht. Menschen spielen gerne Spiele und wollen Spaß empfinden, weshalb auch Lernbücher vermehrt so aufgebaut werden, dass sie möglichst viel Spaß machen, wie zum Beispiel Hauke Fehr extra betont: *„In diesem Buch möchte ich zeigen, dass es auch einen einfachen Weg gibt, [...] bei dem Kreativität und Spaß am Programmieren im Vordergrund stehen [...]“* (Fehr 2017, S. 14).

Ich entschied mich für den Fachbereich EDVO (Elektronische Daten-Verarbeitung und Organisation) an der Höheren Technischen Lehranstalt, weil mir erzählt wurde, dass man Spiele programmiert. Daher bin ich auch mit einer sehr positiven Einstellung an das Fach Programmieren herangegangen, doch die Enttäuschung kam sehr schnell. Es wurden einem gefühlt irgendwelche unnötigen Codeteile erklärt und ständig wurde von irgendwelchen abstrakten Dingen gesprochen, ohne einem ein erstrebenswertes Ziel wie beispielsweise eine Aussicht auf die Programmierung eines Spieles zu geben. Wie gerne hätte ich Spaß am Programmieren in der HTL gehabt, doch stattdessen begann ich eine Abneigung für dieses Fach zu empfinden und habe daher auch immer schlechtere Noten geschrieben. Nach Beendigung der Schule durfte ich in die Arbeitswelt eintauchen. Aufgrund meines HTL Abschlusses im Bereich Informatik, bewarb ich mich, trotz vieler negativer Erfahrungen beim Erlernen des Programmierens, als Softwareentwickler, da mich die Programmierung selbst immer noch faszinierte. Also stieg ich bei einer Firma in Wien als

Softwareentwickler ein und merkte wie mein Interesse und der Spaß am Programmieren wieder anstieg, und dass, obwohl keine Spiele programmiert wurden. Stattdessen motivierte mich das Ziel für Kunden etwas zu programmieren, das diese tatsächlich über lange Zeit in Verwendung haben werden, sowie die Teamarbeit bei der Entwicklung von etwas Großem, welches man alleine nicht schaffen würde und der Gedanke, dass man als Programmierer so viele nützliche Dinge für unseren Alltag erschaffen kann.

Wenig später entschied ich mich auf Teilzeit umzustellen und nebenbei zu studieren. Ich wählte unter anderem das Lehramtsfach Informatik, um irgendwann als Informatiklehrer Unterrichtsstunden zu halten, die Spaß machen und junge Menschen dazu anreizen, sich für ein zukunftsorientiertes Fach zu interessieren. Mittlerweile bin ich bei der Diplomarbeit angelangt und habe mich dazu entschlossen etwas im Bereich der Gamification oder angrenzenden Bereichen zu erforschen, da ich im Laufe meines Lebens sehr gute Erfahrungen mit spielerischem Lernen gemacht habe, wie zum Beispiel mit der von „Coktel Vision“ entwickelten Computer-Lernsoftware „Addy“ – der „Teen“-Version zum Englisch lernen –, mit der man durch Lerneinheiten Punkte bekommen und damit Spiele freischalten kann (Wikipedia, Eintrag: "Addy"). Genauso großartig und hilfreich fand ich das mittlerweile nicht mehr erhältliche Computerspiel „Memorary“, bei dem man über das allseits bekannte Spiel „Memory“ Englisch lernt. Und grundsätzlich bin ich für so manche Geschichten erzählende und stark kompetitive Computerspiele zu begeistern.

Im Zuge der konkreten Themenfindung, wurden meine eigenen Themenvorschläge von dem äußerst interessanten Sparkling Science (Bundesministerium für Bildung, Wissenschaft und Forschung) Projekt „Learn to proGrAME – Programmieren lernen durch Computerspielentwicklung“ überschattet. Nach kurzer Vorstellung des Projektes durch Herrn Mag. Dr. Oswald Comber, sah ich ein riesiges Potenzial darin und war sehr begeistert davon, einen Teil dazu beizutragen. Das Projekt zielt darauf ab, SchülerInnen das Programmieren mit der Game-Engine „Unity“ und dem damit verbundenen

Visual Studio Express 2010 beizubringen, um den Programmierunterricht interessanter, projektartiger und zielgerichteter zu gestalten. Es wird der Fokus auf die spezifischen Bedingungen, durch die Computerspielentwicklung das Programmieren lernen verbessern kann, gesetzt. Bei „Unity“ handelt es sich um eine mächtige Spielentwicklungsumgebung, die viele Plattformen unterstützt, nämlich Computer mit Microsoft Windows, macOS und Linux, sowie die Konsolen Microsoft Xbox360, XboxOne, Sony Playstation 3 und Playstation 4, Nintendo Wii und Nintendo U, aber auch mobile Plattformen wie Apple iOS, Google Android, Windows phone usw. Diese Game-Engine wurde dahingehend konzipiert, dass es möglichst einfach ist, Spiele zu entwickeln (vgl. Gregory 2014, S. 29).

Anhand von Unity und dem konkreten Ziel innerhalb relativ kurzer Zeit eigene Spiele zu programmieren, kann man in SchülerInnen eine intrinsische Motivation wecken, die zu Spaß am Lernen führt und genau das will man im Unterricht erreichen. Es freut mich daher riesig an einem Projekt mitzuarbeiten, deren Ergebnisse für ProgrammierlehrerInnen in Zukunft von enormen Nutzen sein können.

## 3 Forschungsrahmen

### 3.1 Projekt „Learn to proGrAME“

Das Projekt „Learn to proGrAME – Programmieren lernen durch Computerspielentwicklung“ ist Teil des Forschungsprogrammes „Sparkling Science“, welches durch das Bundesministerium für Bildung, Wissenschaft und Forschung ins Leben gerufen wurde und seit 2007 einen unkonventionellen und in Europa einzigartigen Weg der wissenschaftlichen Nachwuchsförderung geht. Durch das Forschungsprogramm werden Projekte gefördert, in welchen SchülerInnen aller Schulstufen aktiv in Forschungsprozesse einbezogen werden. Die SchülerInnen unterstützen dabei also die Forschenden bei wissenschaftlichen Arbeiten. Ziel der Initiative ist der Abbau struktureller Barrieren zwischen dem Bildungs- und Wissenschaftssystem in Österreich. Außerdem will man einen schnelleren Transfer wissenschaftlicher Erkenntnisse in das Bildungssystem und die Öffentlichkeit unterstützen (vgl. Bundesministerium für Bildung, Wissenschaft und Forschung).

Das Forschungsprogramm „Sparkling Science“ wird von der OeAD-GmbH, der österreichischen Agentur für internationale Mobilität und Kooperation in Bildung, Wissenschaft und Forschung abgewickelt. Die OeAD-GmbH berät, bewirbt und unterstützt die strategische Entwicklung und begleitet Umsetzungsmaßnahmen. Sie bauen Brücken zwischen österreichischen Bildungsinstitutionen und Einrichtungen der ganzen Welt (vgl. Österreichische Austauschdienst GmbH).

Der Grundgedanke des Projekts „Learn to proGrAME“ ist, dass SchülerInnen das eigene Programmieren lernen erforschen und so über Prozesse Erkenntnisse gewinnen, die Forschenden, die schon Programmieren können, nicht mehr zugänglich sind. Die SchülerInnen nehmen daher nicht nur die gewöhnliche Rolle von Lernenden ein, sondern reflektieren ihr Lernen im Austausch mit anderen und tragen dadurch zur Generierung von neuem Wissen bei. Wobei die übergeordnete Forschungsfrage des Projekts wie folgt lautet: „Unter welchen spezifischen Bedingungen kann

Computerspielentwicklung das Programmieren lernen verbessern?“. Eines der wissenschaftlichen Hauptziele des Projekts ist es, im Bereich der Informatik-Didaktik den Einsatz von Computerspielentwicklung im Informatikunterricht mit fünfzehnjährigen SchülerInnen zu untersuchen. Die Auswirkungen auf die Vermittlung von IT-Skills und -kompetenzen und die Motivation beim Programmieren lernen sollen erforscht werden (Motschnig et al. 2016). Genau hier, bei den IT-Skills und -Kompetenzen setzt diese Diplomarbeit an, indem die Kompetenzen beforscht werden, die nötig sind, um mit Unity erfolgreich Spiele im Unterricht zu programmieren und die durch dessen Verwendung angeeignet werden. Dabei wird ein Bezug zu den „digitalen Kompetenzen“, deren Erlernung das Ziel des Informatikunterrichts in Österreich ist, hergestellt. Die EU hat diese digitalen Kompetenzen als eine der acht Schlüsselkompetenzen für das lebensbegleitende Lernen erklärt, da diese im 21. Jahrhundert einfach unverzichtbar sind. Auf den Webseiten [www.fit4internet.at](http://www.fit4internet.at) (EU: berufliche Bildung) und [www.digikomp.at](http://www.digikomp.at) (allgemeine/schulische Bildung) sind diese Kompetenzen und deren Aufbau durch unterschiedliche Kompetenzmodelle vorstellbar, vermittelbar und umsetzbar dargestellt.

---

### 3.2 Bereich der Forschung und verwandte Forschung

Die Beforschung der IT-Skills und -Kompetenzen hat, so wie das übergeordnete Projekt „Learn to proGrAME“ selbst, mit mehreren Forschungsfeldern Berührungspunkte, nämlich dem der „Informatik“, „Didaktik der Informatik“, „Lernforschung“, „Lernpsychologie“, „Computerspiele im Unterricht“ und „Medienpädagogik“. Es mag sich dabei zwar in der Nähe des Forschungsbereiches „Gamification“ aufhalten, unterscheidet sich jedoch davon, da bei der Gamification die Motivationssteigerung durch Ranglisten und/oder Belohnungen beispielsweise hervorgerufen wird, im Projekt „Learn to proGrAME“ allerdings aus der Spielentwicklung selbst eine intrinsische Motivation entstehen soll. Genauso muss man es zum Bereich der „Serious

Games“ abgrenzen, da darin über das Computerspiel lernrelevante Inhalte vermittelt werden (Motschnig et al. 2016).

Dieses Thema bzw. Forschungsgebiet wurden unter anderem darum gewählt, da es mit einer lerntechnisch sehr wichtigen Komponente zusammenhängt, die diese bedeutende intrinsische Motivation weckt, nämlich dem Spielen. Wir Menschen lieben es etwas zu spielen und bevorzugen dementsprechend auch Lernvarianten die spielerisches Lernen beinhalten. Daher müssen wir versuchen, diese spielerischen Elemente in allen möglichen Formen in den Unterricht einzubauen. Das Projekt „learn2proGrAME“ schafft das unter anderem dadurch, dass es das Ziel hat, Computerspiele selbst zu programmieren und diese klarerweise anschließend zu spielen. Spiele haben auf uns eine anziehende Wirkung, weshalb in der heutigen Zeit der Smartphones *„Computerspiele [...] nicht (mehr) nur als ein Unterhaltungsmedium, sondern auch – und das in einem zunehmenden Maße – als ein Massenmedium und damit auch als ein Kulturgut zu verstehen“* (Wimmer 2013, S. 7) sind. Hier liegt schließlich auch das Interesse der Wissenschaft, die die Vorteile dieses entstandenen Massenmediums analysieren, die wichtigen Teile davon abstrahieren und für andere Bereiche, wie zum Beispiel die Bildung oder Wirtschaft – zur Jobgestaltung im konkreten Fall – nutzen will.

Bei der Beforschung der Auswirkungen der Computerspielentwicklung auf die Motivation und die Steigerung der IT-Skills im Projekt „Learn to proGrAME“, gehen die Forschenden folgenden Fragen auf den Grund:

- Unter welchen Bedingungen kann Computerspielentwicklung das Programmieren lernen verbessern?
- Vor welchen Herausforderungen stehen junge Lernende, die mit professionellen Werkzeugen zur Spieleentwicklung Programmieren lernen?
- Wie wirkt sich Computerspielentwicklung im Unterricht auf die Motivation aus?



Die beiden letzten Fragen konnten bereits beantwortet werden. So bestätigten die Umfrageergebnisse der durchgeführten Fragebögen, sowie die aufgezeichneten Experteninterviews die Hypothese, dass die Entwicklung von Computerspielen im Informatikunterricht mit gesteigerter Motivation einhergeht (Comber 2018). Als große Herausforderungen bei der Computerspielentwicklung mit der Game-Engine Unity stellten sich die hohe Komplexität der zu erstellenden Quellcodes (Schleifen, Vererbung, ...) und die Benutzeroberfläche von Unity dar (vgl. Mayer 2018). Um jedoch die Frage, unter welchen Bedingungen die Computerspielentwicklung das Programmieren lernen verbessern kann, zu beantworten, müssen weitere Bereiche untersucht bzw. weitere Fragen beantwortet werden. Dementsprechend sind folgende wissenschaftliche Emissionen geplant oder bereits in Arbeit (Comber 2018):

- Game Development in Schools with Unity – Learning Design and Challenges
- Which skills does Game Development with professional tools like Unity promote?
- Challenges and Requirements for Developing Video Games with Unity in Secondary Schools
- Dealing with Complexity in Game-Development with Unity (dabei den Vorteil herausarbeiten, dass die sehr interessierten SchülerInnen besonders weit kommen können, wodurch eine Begabungsförderung erfolgt)

Außerdem wurden bzw. werden folgende wissenschaftliche Arbeiten in diesem Zusammenhang stehend betreut (Comber 2018):

- Cirri, A. (2018) Spieleprogrammierung im Unterricht: Aufschlüsselung und Implementierung eines Spielkonzepts in Scratch und Unity.
- Eisinger, T. (2018) Einführung in die Spieleprogrammierung mithilfe der Spiel-Engine Unity (Diplomarbeit)
- Fellingner, (laufend) N. Mehr Selbstständigkeit im Informatikunterricht - Der Einsatz eines Moodle Kurses zur selbstständigen Aneignung von Spieleprogrammierung mit Unity.

- Hörbe, (laufend) M. Analyse des Unterrichtskonzeptes Learn to proGrAME mit Unity (Bachelorarbeit)
- Issae, A. (laufend) The benefits and challenges of teamwork and pair-programming when learning to code in schools (PhD-Thesis)
- Janevski, (2018) F. Theoretische und praktische Methoden zur Gestaltung einer Unterrichtseinheit im Informatikunterricht (Bachelorarbeit)
- Mayr, H. (2018) Welche Probleme haben Schülerinnen und Schüler beim Programmierenlernen mit Unity im Rahmen des ProGrAME Projektes? (Bachelorarbeit)

### 3.3 Forschungsfragen

Wie der Titel dieser Diplomarbeit schon sagt, möchte ich herausfinden welche Skills Voraussetzung sind, damit man mit der Game-Engine „Unity“ erfolgreich Spiele im Unterricht programmieren kann. Weiter werde ich die Kompetenzen, die man durch die Computerspielentwicklung mit Unity, im Speziellen durch die Unterrichtseinheiten des Tutorials „learn2proGrAME“ (<https://learn2progame.github.io/learn2proGrAME-Tutorial/>) erwirbt, herausfiltern und analysieren. Abschließend sollen Schlüsse gezogen werden, inwiefern man die Unterrichtseinheiten bzw. die Tutorials zum Beispiel verbessern kann, sodass sie evtl. mehr Kompetenzen abdecken, oder weniger Kompetenzen abdecken und diese dafür genauer behandeln. Daraus ergeben sich folgende konkrete Forschungsfragen:

3. Welche digitalen Kompetenzen und Fertigkeiten und welches Vorwissen benötigen jugendliche Schülerinnen und Schüler, um im Unterricht erfolgreich Computerspiele mit Unity zu entwickeln?
4. Welche digitalen Kompetenzen, Fertigkeiten und welches Wissen werden in der Computerspieleentwicklung mit Unity erworben?

Um die Hauptfrage des Projekts „Unter welchen Bedingungen kann Computerspielentwicklung das Programmieren lernen verbessern?“ beantworten zu können, muss man unter anderem die IT-Skills und -kompetenzen, die erlernt werden bzw. ab einem bestimmten Zeitpunkt bereits vorhanden sein sollen, untersuchen. Immerhin will die Wissenschaft Errungenschaften vereinheitlicht messen und vergleichen, um den Wert einer Unterrichtsvariante, in diesem Fall im Hinblick auf vereinheitlichte IT-Kompetenzen einschätzen zu können. Dafür wird hier Bezug auf die „digitalen Kompetenzen“, welche die EU als eine der acht Schlüsselkompetenzen für das lebensbegleitende Lernen erklärt hat, sowie die österreichischen, allgemeinbildenden digi.komp-Modelle genommen.

### 3.4 Vorhaben und methodische Vorgehensweise

Zuallererst werden die wichtigsten Begriffe, wie Wissen, Fertigkeit und Kompetenz für diese Arbeit definiert. Danach sollen die existierenden Kompetenzmodelle als Ganzes und aus programmierspezifischer Perspektive unter die Lupe genommen werden. Dabei fällt der Schwerpunkt auf den DigComp2.2AT, der eine Weiterentwicklung der digitalen Kompetenzen der EU darstellt und auf die beiden digi.komp-Modelle „digi.komp8“ (Sekundarstufe I) und „digi.komp12“ (Sekundarstufe II), die vor allem für die schulische bzw. allgemeine Bildung konzipiert sind. Weiter werden die Fertigkeiten und Konzepte herausgearbeitet, die für die Arbeit eines Programmierers benötigt werden, denn diese sind für den Unterricht, der sich mit der Programmierung auseinandersetzt, ganz besonders interessant. Anschließend werden die Kompetenzen, die man sich durch die planmäßige Erlernung des Programmierens in der Schule aneignen sollte, mit den herausgearbeiteten Fertigkeiten verglichen und in einem neu erarbeiteten, sogenannten digitalen, programmierspezifischen Kompetenzaufbau-Modell (dpKa-Modell) auf einen Nenner gebracht und zusammengefügt. Danach fällt der Blick auf das aktuelle Unterrichtskonzept bzw. die aktuellen Lerneinheiten, die auf die Aneignung von IT-Skills und -Kompetenzen untersucht werden. Aus den dabei gemachten Notizen wird das notwendige Wissen, die notwendigen Fertigkeiten und Kompetenzen die SchülerInnen dadurch erwerben könnten abgeleitet. Aufbauend auf diesen erarbeiteten Grundlagen sollen über empirische Erhebungen, in diesem Fall qualitativ durch Experteninterviews, die Forschungsfragen geklärt werden. Da für die Komplexität der Untersuchung teilstrukturierte Experteninterviews als geeignet erscheinen, wird ein Interviewleitfaden entwickelt. Schließlich werden die Interviews durchgeführt und anhand der qualitativen Inhaltsanalyse nach Mayring ausgewertet. Dafür erfolgt, ausgehend vom erarbeiteten dpKa-Modell eine deduktive Kategorienbildung. Die Ergebnisse werden abschließend diskutiert und die Forschungsfragen damit beantwortet.

## 4 IT-Skills und -Kompetenzen

Was sind denn eigentlich IT-Skills und IT-Kompetenzen? Inwiefern sind sie durch die digitalen Kompetenzen laut [digi.komp.at](http://digi.komp.at) festgeschrieben und abgedeckt? Wie sind die Kompetenzen im Lehrplan abgebildet und was sehen die Lehrpläne für den Unterricht vor? Wie kann man die Kompetenzen und Skills mit den Programmier-Konzepten, wie beispielsweise Objektorientierung verbinden? Diese Fragen sollen hier geklärt werden.

---

### 4.1 Definitionen

Zuerst müssen ein paar Begriffe definiert werden, damit ein möglichst exaktes Verständnis für diese Begriffe vorhanden ist, da der Rest dieser Arbeit darauf aufbaut. Folgende Begriffe werden nun definiert: Wissen, Fähigkeit, Fertigkeit, Kompetenz. Man muss dabei bedenken, dass bestimmte Begriffe von Grund auf schon schwer zu definieren bzw. abzustecken sind, sich gegenseitig unterschiedlich stark überschneiden können und immer wieder mal einen Bedeutungswandel durchlaufen können, womit überhaupt eine neue Definition erforderlich wird (vgl. Bechmann 2016, S. 169–252). Außerdem werden stets Wörter aus anderen Sprachen eingedeutscht oder in den Wortschatz der „Deutschen Sprache“ aufgenommen – wie zum Beispiel der Begriff Skills – die dann allerdings meist ihren Platz in unserem Wortschatz erst finden müssen. Es gibt in unterschiedlichen Sprachen jedoch oftmals auch Wörter die in der jeweils anderen Sprache gar nicht existieren, oder nicht genau dieselbe Bedeutung haben. Daher gibt es und wird es wohl stets wissenschaftliche Diskussionen zu den Definitionen bestimmter Begriffe geben, wodurch man den einen oder anderen Begriff nicht exakt definieren kann. Trotzdem wollen wir uns für diese Arbeit auf die folgenden Definitionen einigen.

---

#### 4.1.1 Wissen

Wissen ist definiert als die „Gesamtheit der Kenntnisse, die jemand (auf einem bestimmten Gebiet) hat“ (Duden Wörterbuch 2019, Eintrag: "Wissen") und

bedeutet „etw. infolge eigener Erfahrung, Wahrnehmung, durch Lernen, Studium, durch Mitteilung von anderen im Gedächtnis, Bewusstsein haben und wiedergeben können, von etw. Kenntnis haben, über etw. unterrichtet sein“ (DWDS 2019, Eintrag: "Wissen" im Wörterbuch der deutschen Gegenwartssprache (WDG)). Zusätzlich führen einige Wörterbücher auch noch den Aspekt des Bewusstseins über das Wissen an, zu sehen an folgender Definition: „bewusste Kenntnisse eines Sachverhaltes“ (PONS Wörterbuch 2019, Eintrag: "Wissen"). Im englischen Wortschatz wird das Wort „knowledge“ für den Wissensbegriff verwendet, wobei deren Definition ebenso gleichzusetzen ist: „the fact or condition of knowing something with familiarity gained through experience or association“ / „the fact or condition of being aware of something“ (Merriam-Webster Dictionary 2019, Eintrag: "knowledge"), „understanding of or information about a subject that you get by experience or study, either known by one person or by people generally“ (Cambridge Dictionary 2019, Eintrag: "knowledge"), „awareness or familiarity gained by experience of a fact or situation“ (Lexico Dictionaries 2019, Eintrag: "knowledge"). Zusammenfassend wird der Begriff Wissen für diese Arbeit wie folgt definiert:

Es handelt sich beim Wissen um den Bestand von Fakten, Regeln und Theorien, über deren Verfügung sich eine Person bewusst ist und deren Gewissheit sich durch den höchstmöglichen Grad auszeichnet, sodass man von deren Wahrheit bzw. Gültigkeit ausgehen kann.

Auf diese Definition werden wir schließlich zurückgreifen, auch wenn in der Philosophie bzw. in den Wissenschaften weiterhin keine Einigkeit über deren korrekte Bestimmung herrscht. Um es von den naheliegenden Begriffen „Information“ und „Daten“, die später unter anderem in den Konzepten der Programmierung vorkommen, abzugrenzen, wird hier auch noch auf diese eingegangen. Daten sind Informationseinheiten wie Symbole und Zeichen, die erst durch eine bestimmte Struktur zu Informationen werden und somit eine Bedeutung und einen Zweck bekommen. Informationen ermöglichen, durch die

Auseinandersetzung mit ihnen, jemandem die Generierung von Wissen. Das Wissen selbst ist also etwas Subjektives und entsteht erst durch die Interpretation, Verarbeitung, Organisation, Strukturierung bzw. durch das in Aktion treten der Informationen und ist somit als Information in einem intelligenten Netzwerk zu betrachten. Damit man sich das Ganze besser vorstellen kann, folgt eine Abbildung der DIKW-Pyramide (Data-Information-Knowledge-Wisdom-Pyramide).

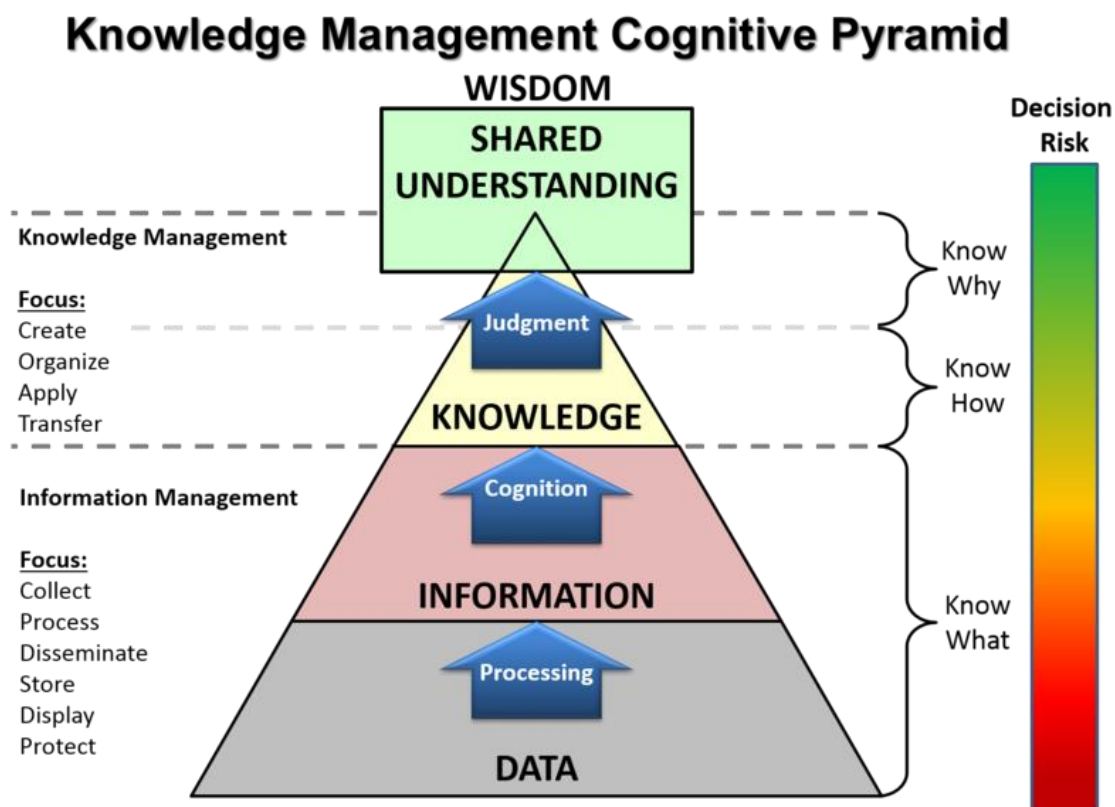


Abbildung 1: DIKW-Pyramide (Matthew.viel)

Quelle: [https://commons.wikimedia.org/wiki/File%3AKM\\_Pyramid\\_Adaptation.png](https://commons.wikimedia.org/wiki/File%3AKM_Pyramid_Adaptation.png)

Der Weisheitsteil an der Spitze der Pyramide wird erreicht, wenn man nicht nur weiß wie etwas ist, sondern auch warum etwas so ist, sodass man das Wissen auch richtig einzusetzen versteht. Je näher man am Bereich der Weisheit dran ist, umso geringer wird auch das Entscheidungsrisiko, das auftritt, wenn man auf Grund des Vorliegenden eine Entscheidung trifft. Handelt es sich beim Vorliegenden um Daten (unstrukturierten Zeichen und Symbolen) so trägt dies nicht als Hilfe für eine Entscheidung bei, handelt es sich hingegen um Wissen,

so weiß man nicht nur was vor einem liegt, sondern bereits wie man das Vorliegende einsetzen kann, damit man eine gute Entscheidung treffen kann.

---

### 4.1.2 Fähigkeiten

Die Fähigkeit findet folgende Definitionen vor: Fähigkeit ist „das Können auf einem bestimmten Gebiet“ / „die Eigenschaft etwas zu können“ (PONS Wörterbuch 2019, Eintrag "Fähigkeit"), / die „besondere Eigenschaft, die den Menschen befähigt, auf einem bestimmten Gebiet etwas zu leisten“ (DWDS 2019, Eintrag "Fähigkeit" im Wörterbuch der deutschen Gegenwartssprache (WDG)), / die „geistige, praktische Anlage, die zu etwas befähigt“ / die „durch bestimmte Anlagen, Eigenschaften geschaffene Möglichkeit, gewisse Funktionen zu erfüllen, gewissen Anforderungen zu genügen, etwas zu leisten“ / das „Vermögen, etwas zu tun“ (Duden Wörterbuch 2019, Eintrag "Fähigkeit"). Der englische Begriff „ability“ ist dem Begriff der „Fähigkeit“ gleichzusetzen: „ability“ ist definiert als „the quality or state of being able“ (Merriam-Webster Dictionary 2019, Eintrag "ability") / „the physical or mental power or skill needed to do something“ (Cambridge Dictionary 2019, Eintrag "ability"). Eine Fähigkeit beschreibt in der Regel also etwas Angeborenes. Nichtsdestotrotz können sich Fähigkeiten einer Person im Laufe ihres Lebens langsam verändern, und zwar auch willentlich. Allerdings handelt es sich dabei um einen langwierigen und schwierigen Prozess. Beispielsweise ist die Disziplin langfristig veränderbar/erreichbar, aber nicht einfach schnell zu erlernen.

Wir definieren uns die Fähigkeit als körperliche bzw. geistige Voraussetzungen einer Person, um gewissen Anforderungen zu genügen, wobei diese von der genetischen Veranlagung der Person und allem, was sie sich in ihrem Leben bis dahin angeeignet hat, abhängt.

Doch Vorsicht, denn dieser Begriff wird nicht nur auf eine simple Variante wie die folgende verwendet. Dadurch dass wir Menschen Hände und Füße haben, ist es uns möglich bzw. haben wir die Fähigkeit uns fortzubewegen, womit das Gehen oder Kriechen, eine durch Übung der jeweiligen speziellen



Fortbewegungsart, erworbene Fertigkeit ist. Später werden wir aber über weniger simple Fähigkeiten wie die Teamfähigkeit stolpern, die nicht mehr nur einfache, elementare Fähigkeiten sind, sondern bereits komplexe Kompetenzen darstellen, die selbst wiederum aus grundlegendere Fähigkeiten, Fertigkeiten und Wissen bestehen. Das ganze Konzept rund um Fähigkeiten und Kompetenzen ist dadurch auch teilweise verschachtelt, wird bei näherem Betrachten aber verständlich. Trotz allem wird bei obigen Wortkombinationen der Fähigkeitsbegriff per Definition noch richtig verwendet, da man ja zur Ausübung der jeweiligen Kompetenz sozusagen fähig ist. Beispielsweise steht Teamfähigkeit für die Voraussetzungen der Person unter anderem in Gruppen zu arbeiten, Meinungen zu akzeptieren und kooperativ weiter zu entwickeln und gemeinsam an Ideen zu arbeiten. Wenn man also diese Voraussetzungen mitbringt, so ist man teamfähig. Es handelt sich dabei jedoch nicht mehr um eine der primitiven Grundfähigkeiten, wie die oben erwähnte Fortbewegung. Der Fähigkeitsbegriff hat also eine relativ große Spanne an Verwendungsmöglichkeiten.

Im Personalwesen häufig genannte Fähigkeiten sind: Konfliktlösungsfähigkeit, Entscheidungsfähigkeit und Organisationsfähigkeit, Kommunikationsfähigkeit, Konzentrationsfähigkeit, Kritikfähigkeit, Analysefähigkeit, Reaktionsfähigkeit, Anpassungsfähigkeit, und Lernfähigkeit. Aber auch Begriffe, die nicht das Wort Fähigkeit beinhalten zählen zu eben jenen, nämlich: Zuverlässigkeit, Verantwortungsbewusstsein, Durchsetzungsvermögen, Belastbarkeit, Kreativität, Empathie und räumliches Vorstellungsvermögen.

Fähigkeiten stellen schließlich die Basis für den Geschicklichkeitserwerb in der jeweiligen Fähigkeit dar, sind also Grundvoraussetzung für das Erwerben einer Fertigkeit.

---

### 4.1.3 Fertigkeiten

Fertigkeit ist wie folgt definiert: „bei der Ausführung bestimmter Tätigkeiten erworbene Geschicklichkeit; Routine, Technik“ (Duden Wörterbuch 2019,

Eintrag "Fertigkeit") / „durch Übung erworbene Gewandtheit, Geschicklichkeit im Ausführen bestimmter Tätigkeiten“ (DWDS 2019, Eintrag "Fertigkeit" im Wörterbuch der deutschen Gegenwartssprache (WDG)). Im englischen Sprachraum existiert der Begriff „skill“, der grundsätzlich mit dem deutschen Begriff „Fertigkeit“ zu übersetzen ist und auch bedeutungsmäßig diesem im Großen und Ganzen gleichkommt. Das Wort „skill“ wird jedoch teilweise mit dem Kompetenzbegriff gleichgesetzt, was wohl auch daran liegt, dass dieses Wort in den deutschen Wortschatz aufgenommen wurde, sich darin allerdings vor allem im umgangssprachlichen Einsatz noch nicht zu hundert Prozent eingeordnet hat. Die Definitionen von „skill“ lauten: „the ability to use one’s knowledge effectively and readily in execution or performance“ / „dexterity or coordination especially in the execution of learned physical tasks“ / „a learned power of doing something competently“ / „a developed aptitude or ability“ (Merriam-Webster Dictionary 2019, Eintrag "skill") / „an ability to do an activity or job well, especially because you have practised it“ (Cambridge Dictionary 2019, Eintrag "skill"). Für diese Arbeit sei Fertigkeit so definiert:

Eine Fertigkeit ist eine durch Übung entwickelte bzw. erworbene Geschicklichkeit für die Ausführung einer bestimmten Tätigkeit.

Es kommt also zum geschickten Einsatz einer Fähigkeit oder eines Zusammenspiels aus mehreren Fähigkeiten. Das heißt, dass die Fähigkeit die Voraussetzung für den Erwerb einer Fertigkeit darstellt und die Fertigkeit schließlich die durch Übung/Training fein geschliffene Fähigkeit ist. Um eine Fertigkeit zu erwerben benötigt man in jedem Fall Wissen, da man mit der Übung absichtlich oder beiläufig etwas dazu lernt und dadurch Wissen generiert.

Im konkreten Beispiel ist eine musikalische Veranlagung eine Fähigkeit und wenn diese besonders gut ausgeprägt ist, so spricht man sogar von einem Talent bzw. einer Begabung in diesem Bereich. Das Beherrschen des Musikinstrumentes hingegen ist eine erlernte Fertigkeit. Man muss hier sehr vorsichtig sein damit man nichts verdreht, da das ganze Konzept mit den

Fähigkeiten und Fertigkeiten sehr komplex sein kann, vor allem durch die Möglichkeit der Verschachtelung. Eine Fertigkeit kann aus einer oder mehreren Fähigkeiten bestehen, aber auch aus einer oder mehreren grundlegenden Fertigkeiten, sowie aus einer Kombination aus sozusagen ungeschliffenen Fähigkeiten und geschliffenen Fertigkeiten. Wenn das Beherrschen eines Musikinstruments als Fertigkeit gilt, so setzt diese auf der grundlegenden Fähigkeit auf, dass man keine körperlichen Einschränkungen hat, sodass man zum Spielen des Instrumentes fähig ist. Aber auch die musikalische Veranlagung ist eine Fähigkeit, die theoretisch zur Beherrschung eines Instrumentes beiträgt. Man muss also stets bedenken, was man unter der Beherrschung eines Musikinstrumentes verstehen soll, denn man könnte sie nur als Fingerfertigkeit im Umgang mit dem Instrument betrachten, womit die musikalische Veranlagung als Fähigkeit mit dieser Fertigkeit nichts mehr zu tun hätte.

Typische Fertigkeiten sind das Lesen, Schreiben, Sprechen und Rechnen, die auch als Grundfertigkeiten gelten, da deren Beherrschung von allen Menschen erwartet wird. Weitere klassische Fertigkeiten sind zum Beispiel Schwimmen, Fußball spielen, Fahrradfahren, Athletik, Musizieren, Handwerk, Freundlichkeit, sowie der geschickte Umgang mit Kunden. In der Literatur wird übrigens immer wieder Mal der Begriff Können verwendet, der stets auf die Kombination aus Fähigkeiten und deren damit verbundene, erworbene Fertigkeiten hinweist.

---

### 4.1.4 Hard und Soft Skills

Es sei hier auf die Verwendung der Begrifflichkeiten „hard skills“ und „soft skills“ einzugehen, bei denen man ganz besonders gut sieht, warum bzw. wie der Begriff der „Skills“ mit dem Kompetenzbegriff in Verbindung steht.

Hard Skills, auch als technische Fertigkeiten, fachliche Qualifikationen, oder Fachkompetenzen bezeichnet, können durch Ausbildung, Studium und praktische Erfahrung erworben, relativ genau gemessen und kategorisiert und durch Zeugnisse, Zertifikate und Bescheinigungen nachgewiesen werden. Sie

sind im Gegensatz zu den Soft Skills meist nur für ein Spezialgebiet relevant. Zu den Hard Skills gehören zum Beispiel Programmierkenntnisse, Projektmanagement, Sprachkenntnisse, Betriebswirtschaft, sowie die Digitalkompetenz (vgl. Miljković und Merten 2017, S. 16).

Soft Skills hingegen haben einen stark subjektiven Charakter und sind somit in unserer Persönlichkeit verankert, wodurch sie nicht einfach messbar sind. Sie sind nicht mit der „sozialen Kompetenz“ gleichzusetzen, da sie mehr beinhalten, darunter folgendes: Kreativität, Teamfähigkeit, Problemlösefähigkeit, Empathie, gesunder Menschenverstand, Vertrauen in die eigenen Fähigkeiten, Resilienz, Kommunikation und interkulturelle Kompetenzen (vgl. Miljković und Merten 2017, S. 16).

Der Kompetenzbegriff ist grundsätzlich ein Oberbegriff zu Fertigkeiten respektive Skills. Wie man sieht, beinhalten die Hard und Soft Skills, die ja eigentlich Fertigkeiten sind, auch bestimmte Kompetenzen und wären damit den Kompetenzen eigentlich sogar übergeordnet, zumindest aber gleichgestellt. An dieser Stelle wird die Schwierigkeit einer scharfen Begriffsabgrenzung deutlich. Doch führen wir uns nochmal vor Augen, dass eine Fertigkeit eigentlich nur ein geübtes Zusammenspiel von Fähigkeiten und/oder grundlegenderen Fertigkeiten ist. Gleichzeitig gibt es auch elementarere Kompetenzen die einen Menschen zu etwas befähigen, was wiederum zu einer Fertigkeit geschliffen werden kann und dann als Hard oder Soft Skill aufgezählt wird.

Es ist also kontextabhängig, was mit Skills und Kompetenz gemeint und welcher dieser beiden als Oberbegriff zu verstehen ist. Zu merken ist jedoch, dass das Konzept der Skills nicht so breit wie das der Kompetenzen aufgestellt ist und somit Skills das Fertigkeitensausmaß betreffen, wo man jederzeit eine Kompetenz darüberstülpen kann.

#### 4.1.5 Kompetenzen

Wenn von Kompetenzen die Rede ist, so wird in Österreich stets folgender Satz von Weinert zitiert, der auch als Bezugspunkt der österreichischen Bildungsstandards gilt (vgl. Bundesinstitut BIFIE 2019): Kompetenzen sind

*„... die bei Individuen verfügbaren oder durch sie erlernbaren kognitiven Fähigkeiten und Fertigkeiten, um bestimmte Probleme zu lösen, sowie die damit verbundenen motivationalen, volitionalen [Anm.: bewusst, willentlich] und sozialen Bereitschaften und Fähigkeiten, um die Problemlösungen in variablen Situationen erfolgreich und verantwortungsvoll nutzen zu können.“ (Weinert 2002, S. 27).*

Diese Definition hat sich im österreichischen Bildungswesen mittlerweile verankert (vgl. Beer 2011, S. 4), so beispielsweise auch konkret in der Geschichts- und Politikdidaktik (vgl. Kühberger 2015, S. 13) und wird in weiterer Folge auch für diese Diplomarbeit verwendet. Allerdings wird mit dieser Definition von Kompetenz ein relativ großer Bereich beschrieben, sodass es einem schwer fällt den Begriff genau zu erfassen.

Um den Kompetenzbegriff besser zu verstehen, grenzen wir ihn von einigen anderen Begriffen ab, die in der Literatur teilweise synonym dafür verwendet werden, wie beispielsweise die Folgenden: Talent, Fähigkeit, Fertigkeit, Leistung, Lernen, Wissen und Intelligenz. Leistung ist ein Maßstab zur Bewertung eines Handlungsergebnisses, für das geistige und körperliche Fähigkeiten eingesetzt wurden. Der Begriff „Lernen“ wird als ein Prozess mit kognitiver Veränderung verstanden. Im Gegensatz dazu betrachtet der Kompetenzbegriff individuelle Unterschiede in latenten Verhaltensbereitschaften. Wissen gilt als mögliches Produkt des Lernens, wohingegen Kompetenz als die Einheit der Kombination aus Wissen, Fähigkeit und Fertigkeit angesehen werden kann, sodass Wissen einen Teil von Kompetenz darstellt. Bei der Abgrenzung zum Intelligenzbegriff ist bedeutend, dass Kompetenz eher auf Domänen fokussiert ist und Intelligenz-Konstrukte als relativ zeitstabilere und weniger als erwerbbar Dispositionen gelten (vgl.

Wilhelm und Nickolaus 2013), wobei man Dispositionen als Persönlichkeitseigenschaften bzw. Wahrscheinlichkeiten, dass sich ein bestimmtes Verhalten einer Person in ähnlichen Situationen erneut zeigt, zu verstehen sind. Man stuft die Kompetenz also als veränderlich bzw. erlern- und trainierbar ein, wodurch es sich vom Fähigkeitskonzept unterscheidet. Schlussfolgernd kann Kompetenz als ein Resultat eines Lernprozesses betrachtet werden, wobei es zum Begriff des Wissens klare Grenzen gibt, da man für Kompetenz auch Fähigkeiten und Fertigkeiten – also Können – benötigt. Ein Talent wiederum ist lediglich eine untrainierte, aber besonders gut ausgeprägte Fähigkeit.

Schauen wir uns das Ganze an einem Beispiel im IT-Kontext an. Damit man eine effiziente App programmieren kann benötigt man einmal grundsätzlich gute analytische, logische und interpretative Fähigkeiten, das Wissen rund um das Projekt und die Programmierung selbst, und die Fertigkeit, um die App in der vorgesehenen Programmiersprache zu implementieren. Sind diese Bestandteile aus den drei Elementen Wissen, Fähigkeiten und Fertigkeiten gut genug verknüpft, um handlungsfähig zu sein und man hat auch noch die Motivation bzw. Einstellung diese Herausforderung zu meistern, sprich man ist handlungsbereit, so verfügt man über die Kompetenz, um eine effiziente App zu programmieren.

Die Kompetenzorientierung zeigt auch ihre Vorteile bei der Setzung von Zielen in der Bildung und bei der Überprüfung dieser Ziele. Vor allem im Bereich der Pädagogik hat man beim Kompetenzbegriff diese starken Vorzüge gegenüber dem Begriff der Lernziele bemerkt, die zum Beispiel beim Überprüfen des Erreichens zum Vorschein kommen: Lernziele sind einzelne Wissens- oder Könnens-Elemente, Kompetenzen hingegen zeigen eine koordinierte Anwendung verschiedener Einzelleistungen zur Lösung eines neuen Problems. Weiter orientiert sich der Kompetenzbegriff stets an lebensweltlichen Bezügen, womit man nicht für die Schule, sondern fürs Leben lernt. Schließlich achtet der

kompetenzorientierte Unterricht mehr auf die SchülerInnen als der am Stoff ausgerichtete Unterricht.

Erziehungswissenschaftlich betrachtet umfasst der Kompetenzbegriff die Elemente Disposition, Erlernbarkeit, Situationsbezug, Motivation und Handlungsfähigkeit – als Kombination aus Wissen, Fähigkeit und Fertigkeit – mit dem pädagogischen Ziel der Mündigkeit und Autonomie. Diese Elemente lassen sich wie folgt erklären (vgl. Roth 1971, S. 180):

- Disposition: Auf eine Persönlichkeitseigenschaft bezogen, die das Vermögen darstellt, etwas zu tun bzw. etwas zu erleiden (darauf wird etwas später noch näher eingegangen).
- Erlernbarkeit: Da Kompetenzen erlernbar sind, kann man sie von angeborenen Eigenschaften abgrenzen und sie durch pädagogische Maßnahmen beeinflussen.
- Situationsbezug: Man kann Kompetenzen durch das Lösen bestimmter Aufgaben erwerben und diese Kompetenzen anschließend in ähnlichen Situationen bzw. Aufgaben anwenden. Sie sind daher funktional auf Situationen bezogen, also kontextspezifisch.
- Motivation: Eine Kompetenz erfordert für deren Anwendung Motivation, also die Bereitschaft zum Handeln.
- Handlungsfähigkeit: Dies stellt die Verknüpfung von Wissen und Können dar, wobei Können sich wiederum aus Fähigkeiten und Fertigkeiten zusammensetzt.

### 4.1.6 Qualifikationen

Da man SchülerInnen in der Schule vor allem für den beruflichen Lebensweg etwas mitgeben will, muss man sich die Anforderungen des Arbeitsmarktes an sie ansehen. Die Wirtschaft und speziell das Personalwesen verwendet den Begriff „Qualifikation“ als Ausdruck der Eignung einer Person für eine bestimmte Aufgabe oder einen Beruf und definiert diesen als Zusammensetzung von Schlüsselqualifikation, Fachkompetenz (Hard Skills) und Sozialkompetenz (ähnlich den Soft Skills):

- **Fachkompetenz (Hard Skills):** Zeugt von der Fähigkeit berufstypische Aufgaben, die ein bestimmtes Fachwissen und Fertigkeiten erfordern, selbständig und eigenverantwortlich lösen zu können und das Ergebnis zu beurteilen. Das Fachwissen und die Fertigkeiten wurden dabei durch die Schul- und Berufsausbildung erworben und können durch Abschlüsse, Nachweise und Zeugnisse nachgewiesen werden (vgl. Kultusministerkonferenz 2018, S. 15).
- **Sozialkompetenz (ähnlich den Soft Skills):** Als einheitliches Konstrukt betrachtet, handelt es sich dabei um einen Kompromiss zwischen Anpassungsfähigkeit und Durchsetzungsfähigkeit. Ist man also sozial kompetent, so schafft man es, die eigenen Interessen innerhalb sozialer Interaktionen zu verwirklichen, ohne dass man die Interessen der Interaktionspartner verletzt. Definiert wird es durch die Gesamtheit von Wissen, Fähigkeit und Fertigkeit einer Person, die die Qualität des Sozialverhaltens fördert. (vgl. Kanning 2009, S. 15)
- **Schlüsselqualifikation:** Diese besagt, wie schnell und einfach eine Person Änderungen der Umweltzustände am Arbeitsplatz bewältigen kann. Als Umweltzustände gelten dabei die zu bewältigenden Aufgaben, sowie die benötigten sozialen Interaktionen. Eine Qualifikation ist etwas Objektives, eine Kompetenz hingegen eine individuelle Eigenschaft. Mit der Schlüsselqualifikation ist also die Fähigkeit zur Adaption bzw. zum Transfer von Fach- und Sozialkompetenz gemeint (Bildungskommission NRW 1995). Man hat irgendwann schließlich versucht, die bis dahin über



850 genannten Schlüsselqualifikationen wie folgt zu ordnen. Dabei hat man unterschieden ob sie die Einstellung, die Verhaltensweisen oder die Fähigkeiten des Auszubildenden betreffen. In der folgenden Abbildung werden exemplarisch einige Schlüsselqualifikationen benannt:

<b>Modell exemplarischer Schlüssel- qualifikationen für den betrieblichen Bereich</b>		
<b>1. Einstellungen</b>	<b>2. Verhaltensweisen</b>	<b>3. Fähigkeiten</b>
1.1 Sorgfältigkeit	2.1 Ausdauer	3.1 Kommunikationsfähigkeit
1.2 Ordnungssinn	2.2 Eigeninitiative	3.2 Kooperationsfähigkeit
1.3 Freundlichkeit	2.3 Offenheit	3.3 Teamfähigkeit
1.4 Leistungsbereitschaft	2.4 Toleranz	3.4 Durchsetzungsvermögen
1.5 Mobilität	2.5 Zielstrebigkeit	3.5 Kreativität
1.6 Flexibilität	2.6 Zuverlässigkeit	3.6 Entscheidungsfähigkeit
		3.7 Lernfähigkeit
		3.8 Organisationsfähigkeit
		3.9 Problemlösungsfähigkeit

Abbildung 2: Schlüsselqualifikationen (Reibold und Regier 2009, S. 91–92)

In der beruflichen Bildung wird häufig der Begriff Schlüsselqualifikation synonym zu Kompetenz verwendet. Kompetenzen sind im Vergleich zum allgemeineren Begriff Qualifikation jedoch nicht so auf Anforderungen von Tätigkeiten oder Berufen bezogen, sondern auf allgemeine Dispositionen (Persönlichkeitseigenschaften), – also Wahrscheinlichkeiten, dass sich ein bestimmtes Verhalten einer Person in ähnlichen Situationen erneut zeigt (vgl. Asendorpf und Neyer 2012, S. 3) –, von Menschen, die für die Bewältigung lebensweltlicher Anforderungen nötig sind. Persönlichkeitseigenschaften gelten als zeitlich überdauernde, und somit relativ stabile Bereitschaften von Individuen, die situationsbedingt bestimmte Aspekte ihres Verhaltens beschreiben. Die Bekanntesten davon sind die sogenannten „Big Five“ (im Englischen als OCEAN-Modell bezeichnet): Offenheit (Aufgeschlossenheit = konservativ / erfinderisch), Gewissenhaftigkeit (Perfektionismus = unbekümmert

/ organisiert), Extraversion (Geselligkeit = reserviert / gesellig), Verträglichkeit (Empathie = wettbewerbsorientiert / kooperativ), Neurotizismus (emotionale Labilität = selbstsicher / emotional). Diese fünf Faktoren gelten als sehr stabil, unabhängig und weitgehend kulturstabil. Heute gelten die Big Five als universelles Standardmodell in der Persönlichkeitsforschung (vgl. John et al. 2008, S. 114–117). Wenn im Personalwesen von Schlüsselqualifikationen die Rede ist, so spricht man jedoch nicht nur von den Big Five, sondern eher von konkreteren Kompetenzen, oder/und von den in obiger Grafik teilweise aufgelisteten Schlüsselqualifikationen, die einem die Adaption bzw. den Transfer von Sach- und Sozialkompetenz ermöglichen oder erleichtern und dadurch in Kombination mit der Fachkompetenz und Sozialkompetenz eine Qualifikation darstellen. Da mit dem Qualifikationsbegriff überwiegend die Eignung einer Person im Personalwesen beschrieben wird, existiert ein zu beachtender Unterschied zur Eignung im Bereich der Pädagogik, wo der Kompetenzbegriff verwendet wird und dieser die Dimension der Eignung, also die Handlungsfähigkeit um die Bereitschaft zu handeln erweitert und ganz besonders dadurch eher einen Personenbezug statt Aufgabenbezug hat.

4.1.7 KONZEPT-DUMMIE: Von der Fähigkeit zur Kompetenz

Um die ganzen Begrifflichkeiten in einem Verhältnis zueinander darzustellen, habe ich eine Grafik namens „Konzept-Dummie“ angefertigt. Sie stellt auf der linken Seite den Aufbau der Begriffe dar und zeigt die Abhängigkeiten voneinander auf und soll diese auf der rechten Seite mit dem Beispiel in der Sprechblase verständlich machen.

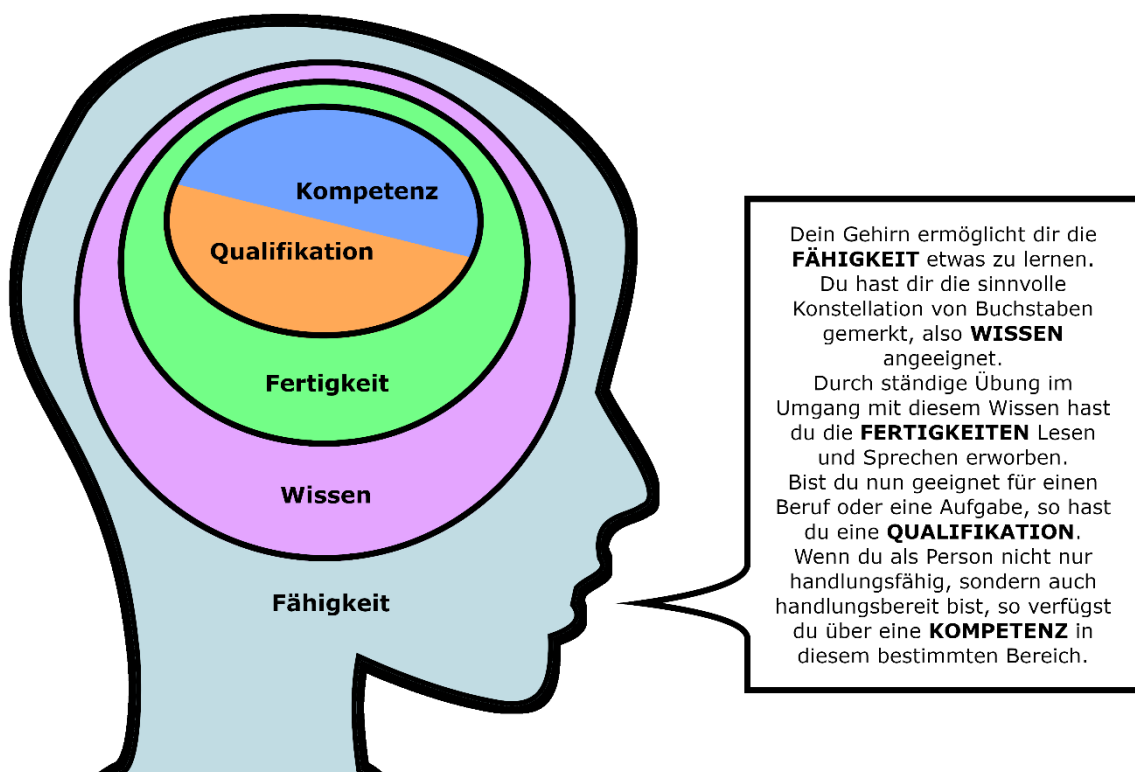


Abbildung 3: Konzept-Dummie (Eigene Darstellung)

Ich erkläre nun den Aufbau der Begriffe auf der linken Seite kurz. Die Fähigkeit stellt die elementarste und flächendeckendste Begrifflichkeit dar und ist somit die Grundvoraussetzung für alle weiteren Konzepte. Sie beschreibt noch lediglich die benötigte Voraussetzung, um einer Anforderung zu genügen und nimmt deshalb die meiste Fläche bzw. auch die Grundfläche von allem anderen ein, weil ein Mensch über sehr viele Fähigkeiten verfügt und alle weiteren Begrifflichkeiten respektive Konzepte darauf aufbauen. Wissen als die erste große konzeptuelle Spezifizierung baut auf dem Einsatz unseres Gehirnes auf, welches uns die Fähigkeit gibt bzw. ermöglicht, Informationen zu interpretieren

und vernetzt abzuspeichern, also Wissen zu generieren. Die Wissensebene liegt damit, wie im Bild angedeutet auf der nächsthöheren Ebene. Man kann sich die Begriffe in der Abbildung also wie Ebenen einer nicht viereckigen, aber runden Pyramide vorstellen, wobei die jeweilige vorhergehende Ebene bereits erreicht sein muss, um auf die nächste Ebene steigen zu können. Die Fertigkeit sei die nächste Spezifizierung, die auf Fähigkeiten und Wissen aufbaut und anhand der durchzuführenden Übungen für den Geschicklichkeitserwerb eben weiter spezifiziert. Egal wie man zu einer Fertigkeit gelangt, also ob bewusst oder unbewusst, in jedem Fall hat man durch die Übung an Erfahrungen gewonnen, sozusagen Wissen aufgebaut, sodass man bei einer Anwendung nun geschickter ist. Ob man sich über dieses Wissen bewusst ist oder nicht und ob man das Wissen bewusst einsetzen kann oder nicht, spielt dabei vorerst keine Rolle. Spätestens jedoch um die höchste Stufe, nämlich die Kompetenz zu erreichen, spielt das Bewusstsein über ein Wissen eine bedeutende Rolle, da Kompetenz nach einer Reflektion verlangt. Zuletzt geben uns die Qualifikation und die Kompetenz weitere Spezialisierungsmöglichkeiten, die beide auf den bisherigen drei Spezifikationen aufbauen und diese so zusammenspielen lassen, sodass man über eine Eignung eine Aussage treffen kann. Bei der Qualifikation geht es um die Eignung für eine Aufgabe oder einen Beruf. Die Kompetenz beschreibt dagegen die persönliche Eignung in einem bestimmten Bereich, also inklusive der persönlichen Handlungsbereitschaften.

Die Begriffe Fähigkeit, Wissen, Fertigkeit sind recht einfach und klar voneinander abzugrenzen, doch zwischen Qualifikation und Kompetenz gibt es problematische Hürden, um einen eindeutigen Aufbau aufeinander zu erkennen. Die Qualifikation beinhaltet innerhalb vieler ihrer Definitionen bereits zwei Kompetenzen, nämlich die Fachkompetenz und die Sozialkompetenz, wodurch sie einerseits der Kompetenz übergeordnet wäre, ihr fehlt als Gesamtkonstrukt in jedem Fall jedoch die Handlungsbereitschaft, wodurch sie andererseits der Kompetenz untergeordnet wäre. Betrachtet man die Fachkompetenz eher als Hard Skills und die Sozialkompetenz eher als Soft Skills, mit jeweils angepassten Inhalten, so kann man die Qualifikation jedoch

der Kompetenz unterordnen und genauso betrachten wir jedenfalls in weiterer Folge die Qualifikation. Allenfalls ist festzuhalten, dass eine Qualifikation auf Aufgaben und Berufe, eine Kompetenz hingegen auf die Person selbst bezogen ist. Es gibt keinen erkennbaren Grund warum diese beiden Konstrukte nicht auf Augenhöhe für unterschiedliche Ziele existieren sollten, weshalb sie auch in der Zeichnung so dargestellt sind.

Im Internet hingegen ist eine Darstellung von den Herrn Heyse und Erpenbeck recht häufig anzutreffen, nach der die Qualifikation ebenso die untergeordnete Rolle zur Kompetenz einnimmt, da zur Qualifikation noch Normen, Werte und Regeln, also mehr oder weniger die Handlungsbereitschaft hinzukommt. Zu sehen an der Abbildung nebenan (siehe Heyse und Erpenbeck 2010). Doch diese Darstellung berücksichtigt nicht die Bausteine einer Qualifikation, die nach der Fachkompetenz und der Sozialkompetenz fragt, weshalb für diese Arbeit die Grafik „Konzept-Dummie“ als Basis für alles Weitere verwendet wird.



Abbildung 4: Wissen, Qualifikation, Kompetenz. Screenshot aus dem pdf-Dokument im Internet. Quelle: [http://www.ciando.com/img/books/extract/3799263675\\_lp.pdf](http://www.ciando.com/img/books/extract/3799263675_lp.pdf) (Heyse und Erpenbeck 2010, XI)

Jedenfalls könnte es darauf auslaufen, dass die Definition der Qualifikation sich irgendwann dahingehend entwickelt, dass man sie nicht mehr mit den Bauteilen der Fachkompetenz und Sozialkompetenz, sondern mit denen der Hard- und Soft-Skills oder sogar noch exakter mit Fertigkeiten definiert, da wie bereits angesprochen, die Begriffe Hard- und Soft-Skills derzeit nicht eindeutig als Fertigkeiten verwendet werden.

### 4.2 Acht Schlüsselkompetenzen der EU (DigComp2.1)

In diesem Kapitel soll konkret auf das digitale Kompetenzmodell der Europäischen Union eingegangen werden. Dazu wird zuerst erklärt, woher die Ideen dazu kommen und wie das ganze Konzept entstanden ist. Abschließend werden wir uns spezifisch auf die österreichische Version konzentrieren.

---

#### 4.2.1 Die Idee der acht Schlüsselkompetenzen

Die EU bzw. der Europäische Rat hat sich selbst mit der Bildung und den Kompetenzen der EU-BürgerInnen befasst und kam im Jahr 2000 zu dem Schluss, dass die Grundfertigkeiten, die man durch lebensbegleitendes Lernen erwerben soll, in einem europäischen Rahmen festgelegt werden sollten. Damit soll eine Reaktion auf die Globalisierung und den Übergang zu wissensbasierten Volkswirtschaften gezeigt werden. Außerdem wird die umfassende Rolle der Bildung betont, da diese zum Erlernen sozialer und bürgerlicher Werte wie Gleichheit, Respekt und Toleranz beiträgt und von besonderer Bedeutung ist, wenn man bedenkt, dass alle Mitgliedstaaten sich mit der Frage konfrontiert sehen, wie man mit der kulturellen und sozialen Verschiedenheit umgeht. Sie stärkt auch den sozialen Zusammenhalt, indem sie Menschen befähigt, erwerbstätig zu werden und zu bleiben. Am 18. Dezember 2006 brachte die Europäische Union dazu ein Amtsblatt mit dem Titel „Empfehlung des europäischen Parlaments und des Rates zu Schlüsselkompetenzen für lebensbegleitendes Lernen“ heraus. Das Ziel dieser Empfehlung sind die Maßnahmen der Mitgliedstaaten mit Hilfe des darin beschriebenen Referenzrahmens, der nationale Reformen und eine weitere Zusammenarbeit unter den Mitgliedstaaten fördert, zu unterstützen. Man hat sich entschlossen diese Empfehlung auf Gemeinschaftsebene zu verwirklichen, da man sie zielführend auf Ebene der Mitgliedstaaten nicht ausreichend verwirklichen kann. Der Referenzrahmen definiert acht Schlüsselkompetenzen, die in einer Wissensgesellschaft für Beschäftigungsfähigkeit, sozialen Zusammenhalt und aktive Bürgerschaft notwendig sind und dient als Referenzinstrument für politische Entscheidungsträger, Bildungsanbieter und

Ausbildungsträger, um das Streben nach gemeinsam vereinbarten Zielen auf nationaler und europäischer Ebene zu unterstützen.

---

### 4.2.2 Empfehlung zu Schlüsselkompetenzen aus 2006

In diesem in Kraft stehendem Dokument aus 2006 ist der Begriff „Kompetenz“ als eine Kombination aus Wissen, Fähigkeit und Einstellung definiert, wobei diese drei Elemente der Kombination nicht näher erläutert werden. Mit Schlüsselkompetenzen hervorgehoben werden diejenigen Kompetenzen, die alle Menschen benötigen, um sich persönlich zu entfalten, sozial zu integrieren, den Bürgersinn wahrzunehmen und um einer Beschäftigung im Sinne eines Jobs nachzugehen. Nun sind darin folgende acht, als Schlüsselkompetenzen ausgezeichnete Kompetenzen aufgelistet, deren Bedeutung als gleich betrachtet wird (vgl. Rat und Parlament der Europäischen Union 2006):

1. Muttersprachliche Kompetenz
2. Fremdsprachliche Kompetenz
3. Mathematische Kompetenz und grundlegende naturwissenschaftlich-technische Kompetenz
4. Computerkompetenz
5. Lernkompetenz
6. Soziale Kompetenz und Bürgerkompetenz
7. Eigeninitiative und unternehmerische Kompetenz
8. Kulturbewusstsein und kulturelle Ausdrucksfähigkeit

---

### 4.2.3 Empfehlung zu Schlüsselkompetenzen aus 2018

Am 22. Mai 2018 kam eine aktualisierte Empfehlung zu den acht Schlüsselkompetenzen heraus, bei der der Kompetenzbegriff als Kombination von Kenntnissen, Fertigkeiten und Einstellungen definiert wird und diese wie folgt beschrieben werden (vgl. Rat der Europäischen Union 2018):

- Kenntnisse bestehen aus Zahlen und Fakten, Konzepten, Ideen und Theorien, die das Verständnis eines bestimmten Bereichs fördern.

- Fertigkeiten bezeichnen die Fähigkeit, vorhandenes Wissen einzusetzen und Prozesse auszuführen, um damit Ergebnisse zu erzielen.
- Einstellungen beinhalten die Bereitschaft zu handeln, sowie auf Personen, Ideen und Situationen zu reagieren, als auch entsprechende Mindsets.

Man kann hier mit ruhigem Gewissen behaupten, dass das oben genannte Element „Kenntnisse“ mit dem Begriff Wissen deckungsgleich ist, denn wenn man von etwas Kenntnis hat, so besitzt man das Wissen darüber. Der Kombinationsteil der Fertigkeiten umfasst die verbesserten bzw. weiter trainierten Fähigkeiten und komplettiert somit die drei Bestandteile der Handlungsfähigkeit einer Kompetenz, nämlich Wissen, Fähigkeit und Fertigkeit. Zusätzlich wird hier dem Kompetenzbegriff wieder das Element der Einstellungen angehängt, welches schließlich die Handlungsbereitschaft ausdrückt. Menschen beobachten ihre Umwelt nicht neutral, sondern bewerten und handeln bzw. reagieren entsprechend. Man wird wohl niemanden finden, der beispielsweise keine Meinung zu Spinnen hat. Dieses Element besteht wiederum aus drei Komponenten, die gemeinsam die Gesamteinstellung zu etwas ausdrücken (vgl. Aronson et al. 2008, S. 194):

- Affektiv basierte Einstellung (Gefühle und Emotionen): Sie besteht aus der emotionalen Reaktion auf das Einstellungsobjekt. Am Beispiel der Spinne: Angst, Wut, ...
- Kognitiv basierte Einstellung (Annahmen und Überzeugungen): Bestehend aus den Überzeugungen und Gedanken über das Einstellungsobjekt. Am Beispiel der Spinne: Der Faden, mit dem sie ein Spinnennetz erzeugt, ist verhältnismäßig stärker/reißfester als ein Stahlseil, was für uns Menschen bewundernswert ist.
- Behavioral basierte Einstellung (Verhaltensweisen): Diese besteht aus dem beobachtbaren Verhalten und den Handlungen im Hinblick auf das Einstellungsobjekt. Am Beispiel der Spinne: Man könnte die Spinne vorsichtig aus dem Haus tragen oder töten.



Es werden im Referenzrahmen von 2006 als auch im Aktualisierten von 2018 Fertigkeiten erwähnt, die als fester Bestandteil aller Schlüsselkompetenzen gelten, nämlich kritisches Denken, Teamwork, Kreativität, Problemlösung, analytische Fähigkeiten, Kommunikations- und Verhandlungskompetenz und interkulturelle Kompetenz. Schließlich wurde nicht nur die Definition des Kompetenzbegriffes auf die heutige Zeit angepasst, sondern auch die acht Schlüsselkompetenzen, die nun wie folgt lauten (vgl. COMM/DG/UNIT 2019; vgl. Rat der Europäischen Union 2018):

1. Lese- und Schreibkompetenz
2. Mehrsprachenkompetenz
3. Mathematische Kompetenz und Kompetenz in Naturwissenschaften, Informatik und Technik
4. Digitale Kompetenz
5. Persönliche, soziale und Lernkompetenz
6. Bürgerkompetenz
7. Unternehmerische Kompetenz
8. Kulturbewusstsein und kulturelle Ausdrucksfähigkeit

Von Bedeutung ist hier der vierte Punkt, der die digitale Kompetenz beschreibt:

*„Digitale Kompetenz umfasst die sichere, kritische und verantwortungsvolle Nutzung von und Auseinandersetzung mit digitalen Technologien für die allgemeine und berufliche Bildung, die Arbeit und die Teilhabe an der Gesellschaft. Sie erstreckt sich auf Informations- und Datenkompetenz, Kommunikation und Zusammenarbeit, Medienkompetenz, die Erstellung digitaler Inhalte (einschließlich Programmieren), Sicherheit (einschließlich digitales Wohlergehen und Kompetenzen in Verbindung mit Cybersicherheit), Urheberrechtsfragen, Problemlösung und kritisches Denken.“ (Rat der Europäischen Union 2018)*

Weiter werden die wesentlichen Kenntnisse, Fertigkeiten und Einstellungen, die mit der digitalen Kompetenz in Zusammenhang stehen beschrieben, wie zum

Beispiel die Fähigkeit, digitale Inhalte aufzurufen, zu filtern, nutzen, beurteilen, erstellen, programmieren und zu teilen, als Fertigkeit angeführt wird. Es wird in der Empfehlung des Referenzrahmens schließlich noch darauf hingewiesen, dass die Mitgliedstaaten die Entwicklung von Schlüsselkompetenzen fördern und dabei ein besonderes Augenmerk auf die Verbesserung der sogenannten Grundkompetenzen (Lesen, Schreiben, Rechnen und digitale Grundkompetenzen) legen sollen (vgl. Rat der Europäischen Union 2018).

---

### 4.2.4 DigComp2.1 (European Digital Competence Framework)

Die Europäische Kommission versucht die Kompetenzorientierung für die Mitgliedsstaaten weiter zu vereinfachen, indem sie das „European Digital Competence Framework for Citizens“, auch DigComp genannt, als Referenzrahmen erstellt hat, worin erklärt wird, was es bedeutet digital kompetent zu sein. Der DigComp wurde vom Joint Research Centre – Service der Europäischen Kommission für Wissenschaft und Wissen – entwickelt und 2013 unter der Version 1.0 veröffentlicht. Mittlerweile sind wir bei Version 2.1 angelangt, die 2017 veröffentlicht wurde und die heute aktuelle Version darstellt. Die „Digitale Kompetenz“ wird darin mit vier Dimensionen – eigentlich fünf Dimensionen, doch eine wird im Laufe der Versionsentwicklung wieder herausgenommen – beschrieben. Die erste Dimension stellen die fünf Kompetenzbereiche dar (vgl. Carretero et al. 2017):



Abbildung 5: Fünf Kompetenzbereiche der "digitalen Kompetenz". Screenshot aus dem pdf-Dokument im Internet.

Quelle: [https://publications.jrc.ec.europa.eu/repository/bitstream/JRC110624/dc\\_guide\\_may18.pdf](https://publications.jrc.ec.europa.eu/repository/bitstream/JRC110624/dc_guide_may18.pdf)  
(12.04.2020)

Die oben angeführten Kompetenzbereiche bestehen aus 21 weiter spezialisierten Kompetenzen mit Beschreibungen (Dimension 2), bei denen jeweils acht Beherrschungslevels (Dimension 3) zu meistern sind. Die 4. Dimension, die als Kombination aus Wissen, Fertigkeiten und Einstellungen besteht, wird darin nicht angeführt. Zuletzt kommen noch Beispiele der Anwendung (Dimension 5) zum Vorschein (vgl. Carretero et al. 2017). Die erste Dimension der fünf Kompetenzbereiche findet man auch auf der Webseite des Europass (europäischen Lebenslaufes) vor. Allerdings werden dabei nur zwischen jeweils drei Beherrschungslevels unterschieden, nämlich der elementaren, selbstständigen und kompetenten Verwendung, die jedoch genauer beschrieben sind (siehe European Centre for the Development of Vocational Training 2020). Viel weiter soll hier auf die europäische Vorlage „DigComp2.1“ aber nicht eingegangen werden, da die „Digitale Kompetenz“ von einigen Mitgliedstaaten wie Österreich zum Beispiel jeweils noch an den Staat angepasst und im Detail beschrieben bzw. definiert wurde.

---

### 4.2.5 DigComp2.2AT (österreichische Version)

Der „DigComp2.1“ wurde vom Bundesministerium für Digitalisierung und Wirtschaftsstandort auf Deutsch übersetzt und ergänzend überarbeitet, sodass wir in Österreich das digitale Kompetenzmodell „DigComp2.2 AT“ verwenden können/sollen. Bereits in der Einleitung wird dabei auf die Bedeutung von Programmierkompetenzen und die Verbindung von allgemeiner und beruflicher Bildung hingewiesen:

*„Mit dem Digitalen Kompetenzmodell für Österreich - DigComp 2.2 AT wird die notwendige Verbindung der beruflichen Bildung mit informatischer Bildung und der Medienkompetenzvermittlung sinnvoll weiterentwickelt, etwa durch die Aspekte der Gestaltung der digitalen Identität oder des geforderten Aufbaus von Programmierkompetenzen und Modellierfähigkeiten. Vor diesem Hintergrund verbindet Digitale Grundbildung für alle Bürgerinnen und Bürger allgemeine und berufliche Bildung“ (Nárosy et al. 2018, S. 6).*

### *4.2.5.1 Kompetenzbereiche und Kompetenzen*

Das Besondere an der österreichischen Version ist der vorangestellte Kompetenzbereich 0 namens „Grundlagen und Zugang“, der aus drei Kompetenzen besteht und für jene BürgerInnen eingeführt wurde, für die die Digitalisierung noch ein Fremdwort ist (vgl. Bundesministerium für Digitalisierung und Wirtschaftsstandort 2018). Außerdem wurde beim Kompetenzbereich Sicherheit die Kompetenz „Sich vor Betrug und Konsumentenrechtsmissbrauch schützen“ hinzugefügt. Dadurch existieren nun sechs Kompetenzbereiche (Dimension 1) mit insgesamt 25 einzelnen Kompetenzen (Dimension 2), die in folgender Tabelle inklusive einem Vergleich mit der Vorlage des europäischen DigComp2.1 dargestellt sind:

DigComp 2.1 (2017)	DigComp 2.2 AT (2019)
	<b>0. Grundlagen und Zugang</b> 0.1. Konzepte der Digitalisierung verstehen 0.2. Digitale Geräte bedienen 0.3. Inklusive Formen des Zugangs zu digitalen Inhalten nutzen und bereitstellen
<b>1. Information and data literacy</b> 1.1 Browsing, searching and filtering data, information and digital content 1.2 Evaluating data, information and digital content 1.3 Managing data, information and digital content	<b>1. Umgang mit Informationen und Daten</b> 1.1. Daten, Informationen und digitale Inhalte recherchieren, suchen und filtern 1.2. Daten, Informationen und digitale Inhalte <b>kritisch</b> bewerten und interpretieren 1.3. Daten, Informationen und digitale Inhalte verwalten
<b>2. Communication and collaboration</b> 2.1 Interacting through digital technologies 2.2 Sharing through digital technologies 2.3 Engaging in citizenship through digital technologies 2.4 Collaborating through digital technologies 2.5 Netiquette 2.6 Managing digital identity	<b>2. Kommunikation und Zusammenarbeit</b> 2.1. Mithilfe digitaler Technologien kommunizieren 2.2. Mithilfe digitaler Technologien Daten und Informationen teilen <b>und zusammenarbeiten</b> 2.3. Digitale Technologien für die gesellschaftliche Teilhabe verwenden 2.4 Ein- und Verkäufe durchführen 2.5. <b>Angemessene Ausdrucksformen verwenden</b> 2.6. Die digitale Identität gestalten
<b>3. Digital content creation</b> 3.1 Developing digital content 3.2 Integrating and re-elaborating digital content 3.3 Copyright and licences 3.4 Programming	<b>3. Kreation digitaler Inhalte</b> 3.1. Digitale Inhalte entwickeln 3.2. Digitale Inhalte integrieren und neu erarbeiten 3.3. Werknutzungsrecht und Lizenzen 3.4. Programmieren <b>und Abläufe automatisieren</b>
<b>4. Safety</b> 4.1 Protecting devices 4.2 Protecting personal data and privacy 4.3 Protecting health and well-being 4.4 Protecting the environment	<b>4. Sicherheit</b> 4.1. Geräte schützen 4.2. Personenbezogene Daten und Privatsphäre schützen 4.3. Gesundheit und Wohlbefinden schützen 4.4. <b>Sich vor Betrug und Konsumentenrechtsmissbrauch schützen</b> 4.5. Umwelt schützen
<b>5. Problem solving</b> 5.1 Solving technical problems 5.2 Identifying needs and technological responses 5.3 Creatively using digital technologies 5.4 Identifying digital competence gaps	<b>5. Problemlösen und Weiterlernen</b> 5.1. Technische Probleme lösen 5.2. Bedürfnisse und technologische Antworten darauf erkennen 5.3. Kreativ mit digitalen Technologien umgehen 5.4. Digitale Kompetenzlücken erkennen

Abbildung 6: Kompetenzbereiche inkl. Kompetenzen. Screenshot aus dem pdf-Dokument im Internet. Quelle: [https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz\\_Arbeitsmarkt.html](https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz_Arbeitsmarkt.html) (12.04.2020)

Beim digitalen Kompetenzmodell DigComp2.2AT handelt es sich im Gegensatz zum europäischen DigComp2.1, welches fünf Dimensionen hat (siehe weiter oben), um ein dreidimensionales Modell, bestehend aus den Dimensionen Kompetenzbereiche, Kompetenzen und Kompetenzstufen. Das Entwicklerteam des DigComp2.2AT hat auf die Übertragung detaillierter Kompetenzausprägungen, also die Beispiele der Anwendung respektive die Dimension 5 des DigComp2.1 verzichtet, da der Erkenntniswert dem Team als „kaum der Rede wert“ erschien. Gleichsam wird, wie auch im DigComp2.1 auf die Ausformulierung einer möglichen Dimension 4, die konkreter die Einzelteile der Kompetenz, nämlich Wissen, Fertigkeiten und Einstellungen beschreiben würde, verzichtet (vgl. Nárosy et al. 2018). Die Dimension 3, genannt Kompetenzstufen, beschreibt die Ausprägung der einzelnen Kompetenzen mit acht Stufen, die sich auf die Merkmale von Schwierigkeit, Komplexität, Fähigkeits- und Wissensumfang beziehen (siehe Abbildung rechts):

### 4.2.5.2 Kompetenzstufen

Durch diesen Stufenaufbau ist das Modell mit dem Europäischen Qualifikationsrahmen EQR und den davon abgeleiteten Nationalen Qualifikationsrahmen NQR, die beide genauso auf acht Stufen basieren, abgleichbar. Diese sogenannten Qualifikationsrahmen dienen als Instrument zum Vergleich von Qualifikationen unterschiedlicher Bildungssysteme. Der EQR gilt dabei als Metarahmen (vgl. Europäische Kommission 2019), über den die einzelnen NQR verglichen werden können und somit eine Qualifikation, die



Abbildung 7: dreidimensionales Würfel-Modell des DigComp 2.2 AT. Screenshot aus dem pdf-Dokument im Internet. Quelle: <https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz-Arbeitsmarkt.html> (12.04.2020)

man in einem Land erworben hat, auf eine Art und Weise in ein anderes Land übertragen und vergleichen kann. Dabei sollte jedes Land einen eigenen Qualifikationsrahmen entwickeln, indem die nationalen Qualifikationen, angefangen bei der beruflichen Bildung, über die Hochschulbildung, bis hin zur Weiterbildung, einem Niveau zugeordnet werden können, basierend natürlich auf den Lernergebnissen. Geläufiger ist der Qualifikationsrahmen für den europäischen Hochschulraum, für den die vereinheitlichten Qualifikationen Bachelor, Master und Ph.D. (Doctor of Philosophy) verwendet werden (vgl. Nationale Koordinierungsstelle für den NQR in Österreich 2019):

### Nationaler Qualifikationsrahmen Österreich



Abbildung 8: Nationaler Qualifikationsrahmen Österreich. Screenshot von der Website.  
Quelle: <https://www.qualifikationsregister.at/> (12.04.2020)

Dieses Konzept von Kompetenzstufen entstammt der Entwicklung einer Taxonomie für Lernziele aus dem Jahre 1956 vom Pädagogen Benjamin Bloom, die man auch auf Kompetenzen anwenden und somit als kognitive Kompetenzgrade betrachten kann. Bloom spricht von sechs Leveln: 1. Wissen, 2. Verstehen, 3. Anwendung, 4. Analyse, 5. Evaluation, 6. Synthese. Diese Taxonomie wurde von seinen Studenten 2001 überarbeitet und sieht nun wie folgt aus (vgl. Anderson et al. 2001, S. 67–68):

# Bloom's Taxonomy

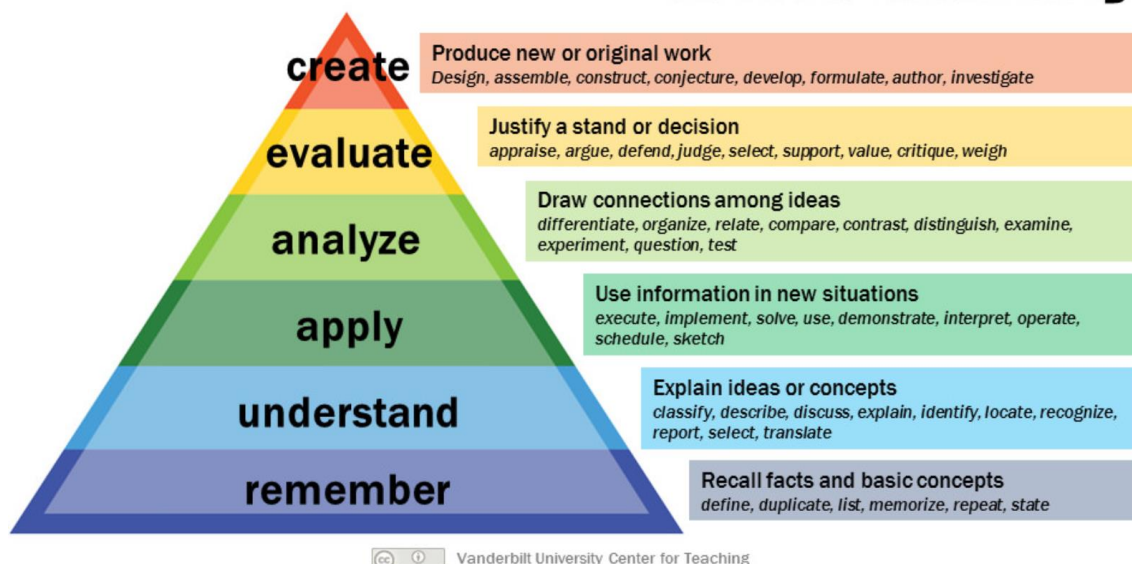


Abbildung 9: Bloom's Taxonomy.. Screenshot von der Website. Quelle: <https://tophat.com/blog/blooms-revised-taxonomy-pyramid/> (Vanderbilt University Center for Teaching) (12.04.2020)

Der DigComp2.2AT orientiert sich an der kognitiven Domäne dieser überarbeiteten Taxonomie von Bloom und ist wie folgt beschrieben:

DigComp-Referenzrahmen	Stufe	Komplexität der Aufgabe	Autonomie	Kognitive Prozessdimension <sup>15</sup>
Grundlegend	Stufe 1	Einfache Aufgabe	Mit Anleitung	Erinnern
	Stufe 2	Einfache Aufgabe	Selbstständig bzw. mit Anleitung, wo erforderlich	Erinnern
Selbständig	Stufe 3	Klar definierte und alltägliche Aufgabe	Selbstständig	Verstehen
	Stufe 4	Aufgaben, sowohl klar definiert als auch nicht alltägliche Probleme	Unabhängig und am eigenen Bedarf orientiert	Verstehen
Fortgeschritten	Stufe 5	Aufgaben und Probleme jeglicher Art	Dazu auch andere anleitend	Anwenden
	Stufe 6	Die bestmögliche Lösung für Aufgabenstellungen und Probleme	Dazu auch fähig zur Anpassung am Bedarf anderer in einem komplexen Kontext	Evaluiieren



DigComp-Referenzrahmen	Stufe	Komplexität der Aufgabe	Autonomie	Kognitive Prozessdimension
Hoch spezialisiert	Stufe 7	Komplexe Aufgaben, für die es nur eingeschränkte Lösungsmöglichkeiten gibt	Trägt zur professionellen Praxis der Berufsgemeinschaft bei und leitet andere dabei an	Neu schaffen
	Stufe 8	Komplexe Aufgaben unter Einfluss vieler interagierender Faktoren	Schlägt neue Ideen und Prozesse in der jeweiligen Domäne vor	Neu schaffen
Erklärung: Kognitive Prozessdimension - Die „cognitive domain“ orientiert sich an der <a href="#">Revised Bloom's Taxonomy</a>				

Abbildung 10: DigComp 2.2 AT - Kompetenzstufen. Screenshot aus dem pdf-Dokument im Internet. Quelle: [https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz\\_Arbeitsmarkt.html](https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz_Arbeitsmarkt.html) (12.04.2020)

Laut „DigComp2.2AT“ sollten Erwachsene zumindest folgende Kompetenzstufen erreicht haben:

*„Erwachsene als per definitionem selbstständige Personen sollten ihre digitalen Kompetenzen zumindest auf den Kompetenzstufen 3 bis 4 verorten können. Dabei ist weniger bedeutsam, ob man nun bei der einen oder anderen Kompetenz auf Stufe 3, Stufe 4 oder vielleicht sogar Stufe 5 zu liegen kommt (Eltern sollten ja durchaus in der Lage sein, ihren Kindern auch die eine oder andere digitale Kompetenz zu vermitteln – und dabei handelt es sich definitionsgemäß um eine Aufgabe auf Kompetenzstufe 5).“ (Nárosy et al. 2018, S. 35)*

Wir spannen nun den Bogen bis zu den einzelnen Kompetenzen zurück, wo unter anderem folgende Kompetenz genannt wird:

- Programmieren und Abläufe automatisieren

Entsprechend dem vorherigen Zitat sollten Erwachsene also in dieser Kompetenz die Kompetenzstufe 3 bis 4, eventuell sogar 5 erreichen. Das heißt, dass sie beim Programmieren klar definierte und alltägliche Aufgaben bis hin zu Aufgaben und Probleme jeglicher Art von selbstständig über unabhängig, bis eventuell sogar anleitend lösen können. Um die Aufgabe zu lösen, muss man laut „DigComp2.2AT“ eine Abfolge verständlicher Anweisungen für ein

Computersystem planen und entwickeln können. Damit befindet man sich beim Programmieren im Hinblick auf Blooms Taxonomy zumindest in der kognitiven Dimension des Verstehens, wenn nicht sogar des Anwendens, wobei man wie oben erwähnt in der Stufe des Verstehens bereits mindestens selbstständig klar definierte Aufgaben durch tatsächliche Programmierung lösen kann. Hier kann man sehr gut die Bedeutung des Programmierens für die Zukunft herauslesen. Und die Zukunft sind unsere Kinder. Also sollten wir uns in der Schule darum kümmern, dass die Kinder bzw. SchülerInnen sich diese Kompetenzen aneignen, wodurch auch ein Grundstein dafür gelegt ist, dass sie später als Erwachsene noch über diese Kompetenzen verfügen.

---

### 4.3 Kompetenzkonstrukt im Bildungswesen (digi.komp.at)

Sehen wir uns das Kompetenzgebilde nun in den österreichischen Schulen, sozusagen im Bereich der Bildung an. Dazu werfen wir zuerst einen kurzen Blick auf die Entstehung von digi.komp.at, danach auf die Lehrpläne und gehen dann auf die Kompetenzmodelle „digi.komp“ ein. Für diese Arbeit betrachten wir das Kompetenzmodell digi.komp8, das für die Sekundarstufe 1 (NMS, AHS Unterstufe) und das Modell digi.komp12, das für die Sekundarstufe 2 (AHS Oberstufe, BHS) gilt. Später wird auch noch auf den Zusammenhang zwischen den Kompetenzmodellen der „digi.komp“-Plattform und des „DigComp2.2AT“ eingegangen und über die allgemeine und berufliche Bildung diskutiert.

---

#### 4.3.1 Entstehung und Ziel der digi.komp-Modelle

Das Projekt digi.komp mit den Verantwortlichen Helmut Stemmer, Günther Schwarz und Peter Micheuz (vgl. Krisper-Ullyett 2013, S. 2) hat im Auftrag des BMUKK/BMBF seit 2011 an für Schulstufen entsprechenden Referenzrahmen für die digitalen Kompetenzen gearbeitet und schließlich diese in Form von Kompetenzmodellen auf der Website „digi.komp.at“ veröffentlicht (vgl. Micheuz 2011). Zuerst wurde die Grundstruktur für ein Kompetenzmodell für die Sekundarstufe I, bestehend aus 4 Kompetenzfeldern und 16 Inhaltsbereichen von Fachdidaktikern aus der Schweiz und Deutschland 2010 gemeinsam

festgelegt. Diese Grundstruktur übernahm eine vom damaligen Bundesministerium für Unterricht, Kunst und Kultur eingesetzte Arbeitsgruppe, entwickelte deren Inhalt weiter und bezeichnete es als digi.komp8 – die 8 weist auf die Schulstufe hin, nach der die SchülerInnen über diese Kompetenzen verfügen sollten. Bei dieser Arbeitsgruppe waren VertreterInnen von Pädagogischen Hochschulen und Universitäten, eLearning-ExpertInnen der VS und NMS, sowie ArbeitsgemeinschaftsleiterInnen für Informatik im Einsatz. Diese ministerielle Arbeitsgruppe diskutierte die Deskriptoren in mehreren Meetings und verabschiedete letzten Endes das Modell. Es handelt sich also um ein österreichisches Kompetenzmodell, das später auch einer Evaluierung von deutschen ExpertInnen durchlief. Mit diesem Modell hat man etwas geschaffen, das in ähnlicher Form von der EU mit dem DigComp-Modell erst eineinhalb Jahre später – nämlich im Oktober 2013 – publiziert wurde.

Zielsetzung des digi.komp-Projektes ist, die digitalen Kompetenzen für die Schule zu konkretisieren und ins schulische Umfeld zu integrieren bzw. sie sogar verbindlich zu machen. Nun ist es der digi.komp-Initiative auch gelungen, für den gesamten allgemeinbildenden Bereich ein einheitliches Modell zu entwickeln. Man erkannte aber, dass es nötig sein werde ein konkretes Fach für die Herausbildung digitaler Kompetenzen einzuführen und festzuschreiben (vgl. Krisper-Ullyett 2013). Dies wurde mehr oder weniger mit dem Kompetenzmodell digi.komp8 inzwischen auch umgesetzt, indem der Lehrplan mit der Verpflichtung ergänzt wurde, die verbindliche Übung „Digitale Grundbildung“ integriert in anderen Fächern oder in speziellen Stunden zu vermitteln. Doch ursprünglich entwickelte man das digi.komp8-Modell, welches heute schließlich als digi.komp4-Modell (für die 4. Schulstufe) existiert und für das digi.komp12-Modell etwas verändert wurde, damit es für den Abschluss der zwölften Schulstufe angemessen ist. Das digi.komp8-Modell lebt in dieser damaligen Form nicht mehr, sondern wurde mittlerweile völlig neu entworfen und in den Lehrplänen mit der oben angesprochenen Übung „Digitale Grundbildung“ verankert, weshalb es zumindest derzeit auf der Website [digi.komp.at](http://digi.komp.at) noch gar nicht dargestellt ist.

### 4.3.2 Lehrpläne

Die Digitalen Kompetenzen bzw. die „informatische Bildung“ ist in den österreichischen Lehrplänen festgeschrieben. Für die Sekundarstufe 1 (NMS, AHS) wurde dies kürzlich als verbindliche Übung „Digitale Grundbildung“ in den Lehrplänen verankert (vgl. Bundesministerium für Digitalisierung und Wirtschaftsstandort 2020). Bei der Sekundarstufe 2 hingegen gibt es ein etwas anderes Kompetenzmodell, welches ebenso im Lehrplan enthalten ist, aber vor allem für die AHS Oberstufe gilt, da im berufsbildenden Schulwesen die Kompetenzmodelle häufig in den Fachlehrplänen integriert sind. Das Bundesministerium für Bildung, Wissenschaft und Forschung hat die den Lehrplänen entsprechenden Kompetenzmodelle für die unterschiedlichen Schulstufen auf der Website „<https://digikomp.at/>“ „vorstellbar, vermittelbar und umsetzbar“ dargestellt. Es wird dort auch auf das europäische Modell „DigComp2.1“ verwiesen, jedoch nur die Bedeutung der Digitalen Kompetenzen davon erwähnt und nicht darauf eingegangen, inwiefern diese miteinander zu tun haben oder auch nicht. Vorgestellt werden jedenfalls die Kompetenzmodelle digi.komp4, digi.komp8, digi.komp12 und digi.kompP. Die Zahl soll in diesem Fall andeuten, um welche Schulstufe es sich handelt, nach der sie die entsprechenden Kompetenzen erworben haben sollen. Als Beispiel sollten sie die Kompetenzen des digi.komp4 nach Abschluss der 4. Schulstufe erworben haben. Beim digi.kompP handelt es sich allerdings um ein Modell zur Weiterbildung von PädagogInnen, weshalb hier ein P angehängt wurde (vgl. Bundesministerium für Bildung, Wissenschaft und Forschung 2020).

---

### 4.3.3 digi.komp12

Es wird hier auf digi.komp12, der in zwei Versionen existiert eingegangen. Nachstehende digitale Kompetenzen aus insgesamt vier Kompetenzbereichen sollen laut dem digi.komp12-Kompetenzmodell (Bundesministerium für Bildung, Wissenschaft und Forschung 2020) (Vorbereitung auf das Absolvieren der Reife- und Diplomprüfung mit Schwerpunkt Informatik) erlernt werden:

1. Informationstechnologie, Mensch und Gesellschaft
  - 1.1. Bedeutung von Informatik in der Gesellschaft
  - 1.2. Verantwortung, Datenschutz und Datensicherheit
  - 1.3. Geschichte der Informatik
  - 1.4. Berufliche Perspektiven
2. Informatiksysteme
  - 2.1. Technische Grundlagen und Funktionsweisen
  - 2.2. Betriebssysteme und Software
  - 2.3. Netzwerke
  - 2.4. Mensch-Maschine-Schnittstelle
3. Angewandte Informatik
  - 3.1. Produktion digitaler Medien
  - 3.2. Kalkulationsmodelle und Visualisierung
  - 3.3. Suche, Auswahl und Organisation von Information
  - 3.4. Kommunikation und Kooperation
4. Praktische Informatik
  - 4.1. Konzepte der Informationsverarbeitung
  - 4.2. Algorithmen, Datenstrukturen und Programmierung
  - 4.3. Datenmodelle und Datenbanksysteme
  - 4.4. Intelligente Systeme

Da man Kompetenzen nicht unmittelbar sehen kann, hat man in sogenannten Bildungsstandards Verhaltensweisen beschrieben, die als Indikatoren für die dahinterliegenden bzw. zu erreichenden Kompetenzen dienen. Diese Bildungsstandards oder Indikatoren sind für die praktikable Anwendung in Form von „Kann-Beschreibungen“ formuliert (vgl. Beer 2011, S. 10). Hilbert Meyer beschreibt die Bildungsstandards als *„bildungspolitisch gewollte, anhand von landesweit oder auch international geeichten Messinstrumenten kontrollierbare Kompetenzniveaus, die die Schülerinnen und Schüler auf der Grundlage eines differenzierten Bildungsangebots der Schule in einem bestimmten Alter erreicht haben sollen.“* Indikatoren dienen hierbei als konkret und beobachtbare Beschreibung um ein Phänomen eines theoretisches Konstruktes in der

Wirklichkeit festzumachen (Meyer 2011, S. 21). Die „Kann-Beschreibungen“ beziehungsweise Indikatoren werden wie beispielsweise im Unterpunkt 4.2 „Algorithmen, Datenstrukturen und Programmierung“ eingesetzt und lauten hier (siehe Bundesministerium für Bildung, Wissenschaft und Forschung 2020):

- Ich kann den Algorithmusbegriff erklären.
- Ich kann einfache Algorithmen nachvollziehen und erklären.
- Ich kann die Umsetzung von Algorithmen mit einem Computer erklären.
- Ich kann einfache Aufgaben mit Mitteln der Informatik modellieren.
- Ich kann einfache Algorithmen entwerfen, diese formal darstellen, implementieren und testen.
- Ich kann an Hand von einfachen Beispielen die Korrektheit von Programmen bewerten.

---

### 4.3.4 digi.komp12 inklusive Kompetenzstufen

Dieses Kompetenzmodell digi.komp12 existiert zusätzlich, wie zu Beginn angesprochen, auf digikomp.at in einer detaillierteren Version, wo zudem auch noch Kompetenzstufen – in Form von Handlungsdimensionen – berücksichtigt werden, nämlich für das Wahlpflichtfach Informatik. Die Gliederung der Handlungsdimension sieht so aus, dass es pro Kompetenzbereichsunterpunkt jeweils drei Stufen gibt: 1. Wissen und Verstehen, 2. Anwenden und Gestalten, 3. Reflektieren und Bewerten. Beim angesprochenen digi.komp12 für das Wahlpflichtfach Informatik sieht der Unterpunkt 4.2 „Algorithmen, Datenstrukturen und Programmierung“ daher wie folgt aus (siehe Bundesministerium für Bildung, Wissenschaft und Forschung 2020):

- Kompetenzstufe 1 „Wissen und Verstehen“:
  - Ich kann den Algorithmusbegriff erklären.
  - Ich kann Aufgaben und Problemstellungen algorithmisch und formalsprachlich in geeigneten Datenstrukturen beschreiben.

- Ich kann wesentliche Aspekte und Methoden der Softwareentwicklung und des Softwareprojektmanagements erklären.
- Ich kann wesentliche Aspekte der Prozeduralen, Funktionalen und Objektorientierten Programmierung nennen und an Beispielen erläutern.
- Kompetenzstufe 2 „Anwenden und Gestalten“:
  - Ich kann Aufgaben mit Mitteln der Informatik modellieren.
  - Ich kann Algorithmen entwerfen, diese formal darstellen, implementieren und testen.
  - Ich kann ein Softwareprojekt planen und durchführen
- Kompetenzstufe 3 „Reflektieren und Bewerten“:
  - Ich kann die Schritte der Softwareentwicklung reflektieren.
  - Ich kann die Angemessenheit der Entwicklungswerkzeuge grob einschätzen.
  - Ich kann die Effizienz von Algorithmen bewerten.
  - Ich kann gezielt nach Programmfehlern suchen und diese korrigieren.

Diese Kompetenzstufen stellen unter Anbetracht des Aufbaues der Kompetenz einen äußerst wichtigen Teil dar, da sie sich auf die Bausteine einer Kompetenz ummünzen lassen. Dazu wird aber später noch genauer eingegangen.

---

### 4.3.5 digi.komp8

Nun zum digi.komp8, bei dem die SchülerInnen durch die im Lehrplan festgeschriebene, verbindliche Übung „digitale Grundbildung“ innerhalb von vier Jahren Kompetenzen aus folgenden acht Bereichen erwerben sollen (Bundesministerium für Bildung, Wissenschaft und Forschung 2020):

- Gesellschaftliche Aspekte von Medienwandel und Digitalisierung
  - Digitalisierung im Alltag
  - Chancen und Grenzen der Digitalisierung
  - Gesundheit und Wohlbefinden

- Geschichtliche Entwicklung (nur als Vertiefungslehrstoff)
- Informations-, Daten- und Medienkompetenz
  - Suchen und finden
  - Vergleichen und bewerten
  - Organisieren
  - Teilen
- Betriebssysteme und Standard-Anwendungen
  - Grundlagen des Betriebssystems
  - Textverarbeitung
  - Präsentationssoftware
  - Tabellenkalkulation
- Mediengestaltung
  - Digitale Medien rezipieren
  - Digitale Medien produzieren
  - Inhalte weiterentwickeln
- Digitale Kommunikation und Social Media
  - Interagieren und kommunizieren
  - An der Gesellschaft teilhaben
  - Digitale Identitäten gestalten
  - Zusammenarbeiten
- Sicherheit
  - Geräte und Inhalte schützen
  - Persönliche Daten und Privatsphäre schützen
- Technische Problemlösung
  - Technische Bedürfnisse und entsprechende Möglichkeiten identifizieren
  - Digitale Geräte nutzen
  - Technische Probleme lösen
- Computational Thinking
  - Mit Algorithmen arbeiten
  - Kreative Nutzung von Programmiersprachen



Wir fokussieren uns in dieser Arbeit vor allem auf Unterpunkte aus dem Punkt 8 „Computational Thinking“, der so strukturiert ist:

- 8.1 Mit Algorithmen arbeiten: SchülerInnen
  - nennen und beschreiben Abläufe aus dem Alltag,
  - verwenden, erstellen und reflektieren Codierungen (z.B. Geheimschrift, QR-Code),
  - vollziehen eindeutige Handlungsanleitungen (Algorithmen) nach und führen diese aus,
  - formulieren eindeutige Handlungsanleitungen (Algorithmen) verbal und schriftlich.
- 8.2 Kreative Nutzung von Programmiersprachen: SchülerInnen
  - Erstellen einfache Programme oder Webanwendungen mit geeigneten Tools, um ein bestimmtes Problem zu lösen oder eine bestimmte Aufgabe zu erfüllen,
  - Kennen unterschiedliche Programmiersprachen und Produktionsabläufe.

---

### 4.4 Alles unter einen Hut bringen

Inwiefern diese bisherigen Kompetenzmodelle berücksichtigt werden und was die Besonderheiten der einzelnen Modelle sind, wird hier beantwortet. Ein besonderes Augenmerk fällt dabei auf die Berufliche und Allgemeine Bildung. Danach wird über den Aufbau der Kompetenzmodelle und den Zusammenhang zwischen Wissen, Fertigkeiten und Kompetenzen mit den Kompetenzmodellen diskutiert. Weiter sollen programmierspezifische Fertigkeiten und eventuell dafür konzipierte Modelle hier Einzug finden. Die Kompetenzen des digi.komp8 werden schließlich mit denen des digi.komp12 und mit jenen des DigComp2.2AT auf einen Nenner gebracht um eine Basis für die Beantwortung der Forschungsfragen zu haben, die nach Kompetenzen als Voraussetzung und als zu erwerbend fragen.

### 4.4.1 Berufliche und allgemeine Bildung (DigComp vs. digi.komp)

Vergleicht man die Kompetenzmodelle digi.komp8, digi.komp12 und DigComp2.2AT, so fallen einem sofort die gravierenden Unterschiede auf. Doch woran liegt das, wenn doch die Seite der digi.komp-Modelle auf das DigComp-Modell verweist, zumindest auf die europäische Version? Dazu sind die Entstehungsgeschichten der einzelnen Modelle interessant, um deren Zusammenhänge und Ziele zu verstehen.

Das DigComp-Modell ist, wie weiter oben beschrieben als konkretes Modell für die digitalen Kompetenzen, die eine der acht Schlüsselkompetenzen aus dem Referenzrahmen 2006 der Europäischen Kommission darstellen, entstanden. Die erste Version des DigComp-Modelles (Version 1.0) kam erst im Jahr 2013 heraus und wurde vom Joint Research Centre, das sich um Wissenschafts- und Wissensservices der Europäischen Union kümmert, entwickelt. Das digi.komp-Projekt hat im Auftrag des BMUKK/BMBWF bereits 2011 das digi.komp8-Modell veröffentlicht (vgl. Micheuz 2011). Die österreichischen digi.komp-Modelle sind also vor dem europäischen DigComp-Modell und unabhängig davon entstanden. Sie weisen dennoch auf das DigComp2.1-Modell hin, das sich immerhin für alle Mitgliedsstaaten der EU als Referenzrahmen ausweist. Der Grund dafür ist offenbar, darauf aufmerksam zu machen, dass man früher oder später deren Umsetzung in den digi.komp-Modellen und deren Festschreibung in den Lehrplänen verwirklichen will, dies aber scheinbar noch in Arbeit ist (vgl. Swertz 2019, S. 10). Aber nicht nur das digi.komp4- und das digi.komp12-Modell, sondern auch das frisch überarbeitete und im Lehrplan festgeschriebene digi.komp8-Modell unterscheidet sich drastisch vom DigComp-Modell. Wirft man einen Blick auf die dahinterliegenden Institutionen, so wird schnell klar, warum die Allgemeine (/Schulische) und die Berufliche Bildung heterogene Modelle verwenden. Hinter dem digi.komp-Modell steht das Bundesministerium für Bildung, Wissenschaft und Forschung, das sich für den Erwerb der Digitalen Kompetenzen im schulischen Umfeld, also in der Allgemeinbildung bemüht und dazu die Website „<https://www.digikomp.at/>“

einsetzt. Am anderen Ufer hingegen will das Bundesministerium für Digitalisierung und Wirtschaftsstandort den Erwerb der Digitalen Kompetenzen in der Beruflichen Bildung etablieren und hat dazu die Gründung des Vereins „fit4internet“ getriggert (vgl. Swertz 2019, S. 9). Dieser Verein bietet auf seiner Website „<https://www.fit4internet.at/>“ Kurse und Selbstüberprüfungen zum Kompetenzmodell DigComp2.2AT an und setzt sich die Steigerung der digitalen Kompetenzen von ÖsterreicherInnen zum Ziel (vgl. fit4internet 2019). Der Verein selbst stellt sich als überparteiliche und unabhängige Initiative vor. Das Bundesministerium für Digitalisierung und Wirtschaftsstandort hat diesem Verein das Mandat erteilt, eine sogenannte „DigiComp Task Force“ zu installieren, um DigComp2.2AT weiterzuentwickeln (vgl. Swertz 2019, S. 8–12). Zudem wurde der Verein mit finanziellen Mitteln in Höhe von € 2.500.000 vom Bundesministerium und von beteiligten Wirtschaftsunternehmen gefördert (vgl. Wimmer 2019). Dies zeigt unter anderem das enorme politische und ökonomische Interesse an diesem Projekt. Der Verein fit4internet etabliert sozusagen ein Kompetenzmodell, das ganz allgemein für BürgerInnen und für Unternehmen gelten soll, wohingegen digi.komp den schulischen Bereich anspricht. Man versucht damit sinnvollerweise alles außerhalb der schulischen Bildung abzudecken und kann die fit4internet-Initiative daher als Bereichsergänzung mit klar unterschiedlicher Zielsetzung betrachten (vgl. Swertz 2019, S. 9).

Für einen gemeinsamen Nenner der Kompetenzmodelle des digi.komp8 und des DigComp2.2AT sollte man die jeweiligen Kompetenzpunkte aus den beiden Modellen einander zuordnen, so wie es Herr Christian Swertz mit zwei weiteren Personen durchgeführt hat und in seinem Zeitschriftenaufsatz in der Zeitschrift „Medienimpulse“ aufzeigt. Er weist darin auf eine zurückhaltende Interpretation dieser Zuordnung hin, da der Lehrplan des digi.komp8 101 Punkte (mittlerweile 105 Punkte) enthält und damit deutlich umfassender als das DigComp2.2AT-Modell mit seinen 25 Punkten ist (vgl. Swertz 2019, S. 13). Außer der gegenseitigen Zuordnung der Punkte der Kompetenzmodelle, ordnet er die ganzen Punkte noch den Unterdimensionen des Medienkompetenzbegriffs von

Baacke (siehe Baacke 1997) zu, um einen umfassenderen quantitativen Vergleich anzustellen. Wir legen unseren Fokus jedoch weder auf den direkten Vergleich noch auf eine prozentuelle Abdeckung bestimmter Bereiche der Medienkompetenz. Stattdessen suchen wir die Gemeinsamkeiten, die offensichtlich bedeutende Punkte darstellen – auch im Sinne des lebenslangen Lernens – und ergänzen diese um bedeutende, modellspezifische Punkte aus den jeweiligen Modellen um ein Gesamtkonstrukt von Kompetenzen rund um die Programmierung zu erhalten. Dazu wurden die Kompetenzpunkte wie am Beispiel von Herrn Swertz einander zugeordnet (siehe Anhang). Die Zuordnung des DigComp2.2AT mit dem Digi.komp8 wurde von Herrn Swertz eins zu eins übernommen. Die Zuordnung zwischen dem DigComp2.2AT und Dig.komp12, sowie die Zuordnung zwischen Dig.komp8 und Dig.komp12 wurden neu gemacht.

So wie auch die Herkunft der Modelle mit ihren unterschiedlichen dahinterstehenden Institutionen zeigt auch die Zuordnung der Kompetenzpunkte, dass das DigComp2.2AT-Modell deutliche Spuren einer beruflichen Bildung und das digi.komp8-Modell der einer allgemeinen Bildung trägt (vgl. Swertz 2019, S. 17–20). Besonders interessant ist der Punkt 5.4 „Digitale Kompetenzlücken erkennen“ aus dem DigComp2.2AT, der auf lebenslanges Lernen und eine Reflexion der eigenen digitalen Kompetenzen abzielt. Dieser Punkt kommt in keiner der allgemeinbildenden digi.komp-Modelle vor, was aber auch nicht notwendig ist, da diese nur für eine bestimmte Phase im Leben konzipiert sind, nämlich der Schullaufbahn. Weiter wird auf den Vergleich zwischen DigComp2.2AT und digi.komp8 hier aber nicht eingegangen, da dieser genauer im Artikel von Herrn Swertz nachzulesen ist. Das digi.komp12-Modell hingegen geht seinen ganz eigenen Weg, indem sowohl eindeutige, tiefgreifende Aspekte der beruflichen als auch der allgemeinen Bildung darin vorkommen. Das sieht man beispielsweise am Punkt 1.3 „Geschichte der Informatik“, der im sozusagen beruflichen Kompetenzmodell DigComp2.2AT überhaupt nicht vorkommt und im digi.komp8 nur innerhalb der Kompetenzstufe „Wissen und Verstehen“ behandelt wird.

Dem stehen beispielsweise die Punkte 2.3 „Netzwerke“, 4.3 „Datenmodelle und Datenbanksysteme“ und 4.4 „Intelligente Systeme“ gegenüber, die in den beiden anderen Kompetenzmodellen nicht vorzufinden sind. Der Punkt der Netzwerke wird bestenfalls über Bereiche der Zusammenarbeit und Kommunikation oder der Informations-, Daten- und Medienkompetenz in elementarster Weise in den anderen Modellen angeschnitten. Diese drei Punkte zeigen auch wie sehr fachspezifisch der digi.komp12 wird. Der digi.komp12 geht somit viel stärker ins Detail als DigComp2.2AT was die berufliche Bildung betrifft, was auch verständlich ist, da der digi.komp12 jenen SchülerInnen dient, die sich beruflich gezielt in der Informatikschiene weiterbewegen wollen, wohingegen der DigComp2.2AT ein Modell für jeden österreichischen Staatsbürger darstellen soll und daher auch keine dermaßen starke Spezialisierung sinnvoll bzw. möglich ist.

Obwohl unser Hauptaugenmerk auf dem digi.komp12 liegt, da wird uns mit der Sekundarstufe II beschäftigen, werden wir stets einen Blick auf die beiden anderen Modelle werfen, um zum Einen auch der beruflichen Bildung gerecht zu werden und zum Anderen eine kompetenzmäßig fundierte Basis für eventuell benötigte Voraussetzungen zu haben.

---

### 4.4.2 Bausteine der Kompetenzmodelle (Wissen, Fähigkeit, Fertigkeit)

Es wurden in dieser Arbeit bereits auf einige Begriffe, die mit Kompetenzen zusammenhängen näher eingegangen, nämlich Wissen, Fähigkeit und Fertigkeit. Diese Begriffe stellen die Grundbausteine einer Kompetenz dar. Nun stellt sich die Frage, inwiefern die Kompetenzmodelle mit diesen Begriffen hantieren. Es sei erwähnt, dass Fertigkeiten sozusagen geschliffene Fähigkeiten sind, weshalb im weiteren Verlauf hauptsächlich von Wissen und Fertigkeiten gesprochen wird. Außerdem haben wir von den Kompetenzstufen und den Prozessdimensionen der überarbeiteten Taxonomie von Bloom gesprochen. Daher sehen wir uns an, wie solche Levels in den Kompetenzmodellen Verwendung finden.

### 4.4.2.1 Bausteine des DigComp2.2AT

Wir haben über das DigComp2.2AT-Modell bereits gehört, dass hier, wie auch schon beim europäischen Vorlagenmodell DigComp2.1 auf eine Spezifizierung der Kompetenzen auf der Wissens- und Fertigkeitsebene verzichtet wurde. Stattdessen findet man kurze Sätze zur genaueren Beschreibung der Kompetenzen vor, die die Kompetenz aber natürlich deutlich weniger genau skizzieren als konkrete Wissens- und Fertigkeitsbausteine. Allerdings wird die Entwicklung und Ausprägung der Kompetenzen durch acht Kompetenzstufen, die sich auf die kognitiven Prozessdimensionen der überarbeiteten Taxonomie von Bloom beziehen (Erinnern, Verstehen, Anwenden, Evaluieren und Neu schaffen), wobei die Reflexion nicht miteinbezogen wird, dafür aber manche Prozessdimensionen (Erinnern, Verstehen und Neu erschaffen) doppelt vorkommen, ausgedrückt. Diese acht Kompetenzstufen werden durch die Komplexität der Aufgabenstellung und durch die Autonomie des Betroffenen genau definiert (vgl. Nárosy et al. 2018, S. 33–34). Sehen wir uns das Ganze an einem Beispiel an. Dazu nehmen wir aus dem Kompetenzbereich 0 „Grundlagen und Zugang“ die Kompetenz 0.2 „Digitale Geräte bedienen“ her und befassen uns darin mit dem im Dokument des Bundesministeriums angegebenen Beispiels, ein Smartphone bedienen zu lernen. Die Kompetenz 0.2 „Digitale Geräte bedienen“ ist wie folgt beschrieben: „Sich unterschiedliche Bedienkonzepte von digitalen Geräten aneignen und verwenden können.“. Das Beispiel lautet so:

***Stufe 1 – Aufgabe: Smartphone bedienen lernen:** Damit jemand überhaupt in der Lage ist, digitale Kompetenzen zu erwerben, muss ein erster Einstieg erfolgen. Eine einfache Aufgabe (Handy einschalten oder eine App aufrufen) wird gestellt, als Anleitung vorgezeigt und von der lernenden Person nachgeahmt. Und im Verlauf des Lernprozesses erinnert man sich nach mehreren Wiederholungen immer besser, wie „das“ funktioniert. Rasch ist man vielleicht in der Lage, nur mehr von Fall*

*zu Fall Unterstützung zu brauchen (Stufe 2) bzw. eine Aufgabe verlässlich selbstständig erledigen zu können (Stufe 3). (Nárosy et al. 2018, S. 34)*

Stufe 1 und 2 befinden sich auf der kognitiven Prozessdimension „Erinnern“ und Stufe 3 auf der des „Verstehens“. Damit sind die Kompetenzstufen klar vorstellbar aufgezeigt. Die Kompetenzstufen zeigen also wie sehr eine Kompetenz ausgeprägt ist. Man kann laut dem DigComp2.2AT-Modell somit insgesamt 25 Kompetenzen über jeweils acht Kompetenzstufen meistern.

---

### 4.4.2.2 Bausteine des digi.komp8

Das digi.komp8-Modell beschreibt jeden seiner 105 Punkte durch einen Satz. Diesen Punkten sind allerdings nicht wie den 25 Kompetenzen des DigComp2.2AT jeweils acht Kompetenzstufen zugeschrieben. Sie sind auch nicht auf dem gleichen Level im Sinne von Kompetenzstufen bzw. Prozessdimensionen, wie man an folgenden zitierten Punkten sieht:

- 1.2.1 SchülerInnen kennen wichtige Anwendungsgebiete der Informationstechnologie und informationstechnologische Berufe
- 3.2.1 SchülerInnen geben Texte zügig ein
- 1.1.2 SchülerInnen reflektieren die eigene Medienbiografie sowie Medienerfahrungen im persönlichen Umfeld

Beim Punkt 1.2.1 muss man etwas lediglich „kennen“, womit man sich in der kognitiven Prozessdimension des „Wissens“ aufhält. Der Punkt 3.2.1 spricht von einer Texteingabe, die sich auf dem Level des „Anwendens“ befindet und der Punkt 1.1.2 schließlich erfordert eine Reflexion, was die Ebene des „Reflektierens“ benötigt. Es existiert also keine Zuweisung oder Reihung bezüglich möglicher Kompetenzstufen und die Punkte sind auch nicht so verallgemeinert, dass man die acht Kompetenzstufen auf jeden einzelnen Punkt anwenden könnte. Man sollte diese 105 Punkte also nicht unbedingt als einzelne Kompetenzen betrachten, sondern eher als Anhaltspunkte, die einem zeigen, ob man in einem Bereich, wie zum Beispiel dem der „Digitalisierung im Alltag“ mehr oder weniger kompetent ist. In etwa so als wäre die Anzahl der

erreichten Anhaltspunkte eines Bereiches im Vergleich zu dessen maximaler Anzahl der Ausprägungsfaktor der Kompetenz. Zudem gibt es keinen Hinweis auf eine Spezifizierung auf Wissen und Fertigkeiten.

---

### 4.4.2.3 Bausteine des digi.komp12

Das digi.komp12-Modell mit seinen 81 Punkten in der Version für das Wahlpflichtfach Informatik hat zwar keine explizit definierten Kompetenzstufen, ist dafür aber außer der Inhaltsdimension auch in eine Handlungsdimension eingeteilt, die aus drei Teilen besteht:

1. Wissen und Verstehen
2. Anwenden und Gestalten
3. Reflektieren und Bewerten

Diese drei Teile bestehen aus allen sechs kognitiven Prozessdimensionen Blooms. Nimmt man sie genauer unter die Lupe, so erkennt man auch eine grobe Reihung, die zwar nicht jener der überarbeiteten Taxonomie von Bloom entspricht, dafür aber eine Einteilung in die Bausteine (Wissen, Fertigkeit, Handlungsbereitschaft) einer Kompetenz zulässt. Der 1. Teil „Wissen und Verstehen“ lässt sich als der Wissensbaustein, der 2. Teil „Anwenden und Gestalten“ als der Fertigkeitsbaustein und der 3. Teil „Reflektieren und Bewerten“ als die Handlungsbereitschaft bezeichnen. Denn wenn ich etwas weiß oder verstehe beschreibe ich einen Wissenszustand, der vorhanden ist oder nicht. Man kann eine Anwendung oder Gestaltung „geschickt oder weniger geschickt“ vollziehen, aber in jedem Fall lernen, um sie geschickter durchzuführen, genauso wie eine Fertigkeit. Für die Reflexion und Bewertung benötige ich schließlich Regeln, Normen und Werte, damit ich in den entsprechenden Situationen auch handlungsbereit bin. Das bedeutet, dass die einzelnen, in Kann-Beschreibungen formulierten Punkte selbst wieder nicht als Kompetenzen betrachtet werden können, sondern nur einen Teil einer Kompetenz ausformulieren. Je nachdem in welchem der drei Handlungsdimensionen sich der Punkt befindet, handelt es sich um Wissen,



eine Fertigkeit, oder um eine Handlungsbereitschaft. Daher erwirbt man hier erst durch die Erreichung eines Punktes des dritten Teils (Reflektieren und Bewerten) der Handlungsdimension in einem Bereich wie dem des Punktes 1.1 „Bedeutung von IT in der Gesellschaft“ eine Kompetenz dafür. Sehen wir uns das an diesem Punkt 1.1 als Beispiel an:

### 1.1 Bedeutung von IT in der Gesellschaft (3 Punkte)

Wissen und Verstehen (BAUSTEIN: Wissen):

- **1.1.1** Ich kann Beispiele für den Einsatz von Informatiksystemen und ihre gesellschaftlichen und wirtschaftlichen Auswirkungen beschreiben.

Anwenden und Gestalten: (BAUSTEIN: Fertigkeit)

- **1.1.2** Ich kann Wissen über Informatiksysteme im digitalen privaten und schulischen Umfeld zielgerichtet anwenden und nutzen.

Reflektieren und Bewerten: (BAUSTEIN: Handlungsbereitschaft)

- **1.1.3** Ich kann den Einfluss von Informatiksystemen auf meinen Alltag, auf die Gesellschaft und Wirtschaft einschätzen und an konkreten Beispielen Vor- und Nachteile abwägen.

Also erst wenn man den Punkt 1.1.3 beherrscht, hat man die Kompetenz für die „Bedeutung von IT in der Gesellschaft“ erworben. Verfügt man nur über die Punkte 1.1.1 und 1.1.2, so ist man nur handlungsfähig, aber noch nicht handlungsbereit. Das Modell besteht also aus 81 Punkten, aber nicht 81 Kompetenzen. Im Modell wird zwar nicht dezidiert von Wissen und Fertigkeiten gesprochen, aber man kann diese Bausteine, wie man an obigen Beispiel sieht, nahtlos den Handlungsdimensionen zuordnen. Über dieses System lassen sich zudem auch noch die acht Kompetenzstufen des DigComp2.2AT darüberlegen, da dort die Stufen durch die Komplexität der Aufgabe und durch die Autonomie der Person in Anwendung der Kompetenz angegeben ist und die Kompetenzstufen daher einen Aufgabenbezug haben:

## KAPITEL: IT-Skills und -Kompetenzen

DigComp-Referenzrahmen	Stufe	Komplexität der Aufgabe	Autonomie	Kognitive Prozessdimension <sup>15</sup>
Grundlegend	Stufe 1	Einfache Aufgabe	Mit Anleitung	Erinnern
	Stufe 2	Einfache Aufgabe	Selbstständig bzw. mit Anleitung, wo erforderlich	Erinnern
Selbständig	Stufe 3	Klar definierte und alltägliche Aufgabe	Selbstständig	Verstehen

Abbildung 11: DigComp 2.2 AT - Kompetenzstufen. Screenshot aus dem pdf-Dokument im Internet. Quelle: [https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz\\_Arbeitsmarkt.html](https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz_Arbeitsmarkt.html) (12.04.2020)

Die kognitiven Prozessdimensionen würde man damit theoretisch sogar noch ein zweites Mal darüberstülpen, was so aber nicht ganz stimmt, da es sich um wortwörtlich „kognitive“ Prozessdimensionen der gesamten Kompetenz bzgl. einer Aufgabe handeln würde. An dieser versuchten Verschachtelung sieht man, dass beim DigComp2.2AT die kognitiven Prozessdimensionen als rein geistige Dimension mit Aufgabenbezug, beim digi.komp12 hingegen diese als verhaltensorientierte Dimension mit reinem Kompetenzbezug integriert sind. Man bemerkt also auch hier die unterschiedliche Entwicklung, die zwischen der beruflichen und der allgemeinen Bildung zum Ausdruck kommt. Die kognitiven Prozessdimensionen dienen beim DigComp2.2AT als Einstufung auf die Anwendung der erworbenen Kompetenz, was einer Bewertung der Kompetenz im Sinne einer Qualifikation für eine Aufgabe gleichkommt. Das digi.komp12-Modell verwendet die kognitiven Prozessdimensionen stattdessen um den Fortschritt im Erwerbsprozess einer Kompetenz zu skizzieren.

---

### 4.4.2.4 Vergleich der Modelle

Zusammenfassend betrachtet ist das DigComp2.2AT-Modell mit seinen 25 beschriebenen Kompetenzen speziell für die berufliche Bildung aufgebaut, da es nicht auf Wissen und Fertigkeiten bzgl. dem Kompetenzerwerb heruntergebrochen ist und seine Kompetenzstufen sich auf die Anwendung der jeweils bereits erworbenen Kompetenzen bezieht. Das scheinbar

umfangreichste digi.komp8-Modell mit seinen 105 Punkten besitzt zwar thematisch die größte Spannweite, wirkt aber wie ein unfertiges, oder nicht vollständig durchdachtes Kompetenzmodell, da die einzelnen Punkte jeweils unterschiedliche Dinge in unterschiedlicher Ausprägung, also mit unterschiedlichen Kompetenzbausteinen (Wissen, Fertigkeit, Handlungsbereitschaft) bzw. Handlungsdimensionen (Wissen und Verstehen, Anwenden und Gestalten, Reflektieren und Bewerten) beschreiben. Obwohl dieses Modell das neueste von allen ist, scheint es als hätte man hier lediglich Lernziele aber keine Kompetenzen ausgearbeitet. Beim digi.komp12-Modell wiederum handelt es sich um ein absolut ausgereiftes Kompetenzmodell, ganz besonders im Hinblick auf die allgemeine Bildung. Hierin sind die 81 Punkte innerhalb der jeweiligen Kompetenz aufgeteilt auf die Kompetenzbausteine bzw. Handlungsdimension, behandeln dabei aber jeweils die gleiche Thematik. Somit wird der Erwerbsprozess der Kompetenzen sichtbar.

---

### 4.4.3 Programmierspezifisches: Wissen und Fertigkeiten

Wir setzen uns zwar mit den Kompetenzen auseinander, doch ist es wichtig, programmierspezifisches Wissen und programmierspezifische Fertigkeiten zu beachten, da diese Teile der Kompetenzen sind. Matthias Kramer, Peter Hubwieser und Torsten Brinda haben dazu ein Kompetenzstrukturmodell für objektorientierte Programmierung („competency structure model of introductory object-oriented programming“) entwickelt, das explizit die Wissens- und Fertigkeitsbereiche im Punkt 1 „OOP knowledge and skills“ auflistet (Kramer et al. 2016):

1. OOP knowledge and skills
  - 1.1. data structure (graph, tree, array)
  - 1.2. class & object structure (object, attribute, association)
  - 1.3. algorithmic structure (loops, conditional statement)
  - 1.4. notional machine (data, working memory, processor, statement, program, automation)

2. Mastering representation (language, syntax, semantics)
3. Cognitive Process
  - 3.1. Problem solving stage (understanding the problem, determine how to solve the problem, translating the problem into a computer language program, testing and debugging the program)
  - 3.2. Cognitive Process Type
    - 3.2.1. Interpreting (Remember, Understand, Analyze, Evaluate)
    - 3.2.2. Producing (Apply, Create)
4. Metacognitive process

Unter Punkt 2 „Mastering representation“ ist die Programmiersprache als solche selbst beschrieben und unter Punkt 3 „Cognitive Process“ findet man wieder die Kompetenzstufen Wissen, Verstehen, Analyse, Evaluierung, Anwendung und Gestaltung, aber auch den Unterpunkt 3.1 „Problem solving stage“. Dieser Unterpunkt wurde extra ins Modell integriert, da die Literatur vermehrt Programmierprozesse generell als Problemlöseprozesse charakterisiert und dieser sich auf mehrere Phasen aufteilen lässt (vgl. Kramer et al. 2016). Der 4. Punkt „Metacognitive process“ repräsentiert metakognitive Faktoren wie Freiwilligkeit, Motivation, Selbstwirksamkeit und Werte und stellt damit die Handlungsbereitschaft des Kompetenzmodelles dar. Dieses Modell als Ganzes soll einen mehrdimensionalen Kompetenzraum darstellen, sodass jeder sogenannte Kompetenzkandidat durch einen bestimmten Unterraum abgebildet werden kann. So deckt der Kompetenzkandidat „fähig zu sein, Felder in Java zu programmieren“ einen Unterraum ab, der durch die folgenden Werte abgegrenzt ist: „1.1 Arrays, 2 Java, 3.1 Translating, 3.2.1 Understand, 3.2.2 Apply. Dieses Modell wurde bereits mehrmals untersucht und mit Curricula mehrerer Länder/Staaten (Deutschland, Kanada, Indien, Vereinigtes Königreich, USA) und Institutionen bzw. Empfehlungen (Code.org, ACM/IEEE curriculum, Computing At School Curriculum, ...) abgecheckt und stellte sich als funktionierendes Modell heraus, vor allem dadurch, dass es alle Fälle rund um die Programmierung durch einen Unterraum innerhalb des Modelles abdecken kann (vgl. Kramer et al. 2016).

Von diesem Modell nehmen wir uns den Punkt 1 „OOP knowledge und skills“ und ordnen später sofern möglich die darin enthaltenen Unterpunkte den zu beforschenden Kompetenzteilen zu.

Warum hier ein objektorientiertes Modell und nicht ein Modell für das imperative, funktionale, logisch-deklarative, ereignisgesteuerte, oder konkurrierende Programmierparadigma gewählt wurde, liegt daran, dass das objektorientierte Paradigma das am häufigsten verwendete Paradigma in kommerziellen Kreisen ist (vgl. Üçoluk und Kalkan 2012, S. 9).

---

#### 4.4.4 dpKa (digitales, programmierspez. Kompetenzaufbau)-Modell

Es soll nun alles in einem sogenannten dpKa-Modell (digitales, programmierspezifisches Kompetenzaufbau-Modell) unter einen Hut gebracht werden. Doch was bedeutet alles? Es geht beim Projekt proGrAME in erster Linie darum, das Programmieren zu erlernen. Das hängt mit dem Erlernen einer konkreten Programmiersprache zusammen, in diesem Fall mit der Programmiersprache C#. Um etwas zu programmieren muss man auf der Ebene des Wissens und Verstehens grundlegende Programmierstrukturen (Verzweigungen, Schleifen, Prozeduren) kennen, um dann auf der nächsten Ebene des Anwendens und Gestaltens durch Übung, wie mit jeder anderen Fertigkeit auch, geschickt im Umgang damit werden. Man soll auch fähig werden operational, algorithmisch und abstrakt zu denken, was einem zum Beispiel bei der gezielten Suche und Korrektur von Programmierfehlern hilft, das sich auf der höchsten Handlungsdimension, nämlich der des Reflektierens und Bewertens befindet. Es ist beim Programmieren laut Fehr daher jedenfalls von Vorteil, wenn einem logisches Denken naheliegt und man sich beispielsweise Reihen und Größenordnungen im Kopf vorstellen kann. Außerdem ist es wichtig zu verstehen, mit welchen Mitteln stetig wiederkehrende, typische Aufgaben gelöst werden können, man sich also Muster einprägen und wiedererkennen kann (vgl. Fehr 2017, S. 15).

Bevor man sich jedoch an eine Programmiersprache heranwagt, benötigt man schon im Vorhinein weitere, aber elementarere Dinge, wie natürlich grundlegende Computerkenntnisse, um beispielsweise mit einzelnen Dateien und einem Editor umzugehen. Dazu gehört der Umgang mit dem Windows-Explorer, inklusive Kenntnisse über Dateien und Verzeichnisstrukturen, sowie deren sinnvolle Gestaltung. Ein technisches Grundverständnis also, was Dateien sind und deren Varianten wie Text-, Grafik- und Videodateien unterscheiden kann und wie man die Dateien in Verzeichnissen verschieben, finden und kopieren kann.

Ist man nach den grundlegenden Dingen schließlich beim Programmieren selbst angekommen, trifft man häufig auf Aufgabenstellungen und Problemsituationen, für die man sich Hilfe im Internet holen kann. Dabei befinden wir uns in einem weiteren für die Programmierung wichtigen Bereich, nämlich dem der Informationsbeschaffung. Viele dieser Informationen und Hilfestellungen sind jedoch auf Englisch, weshalb Hauke Fehr im Zusammenhang mit dem Programmieren-Lernen daher auch Englischkenntnisse als Vorteil nennt, da es einem dadurch leichter fällt, Diskussionen und Forenbeiträge rund um das Thema Programmieren im Internet zu lesen. An diesem Punkt muss die Fokussierung aber begrenzt werden, da Englischkenntnisse nicht mehr in das Feld der digitalen Kompetenzen gehören. Es wären noch weitere Kompetenzen für die Programmierung sehr interessant, wie zum Beispiel Teamwork als Teil der sozialen Kompetenz usw., doch wie gesagt, gehen wir nicht über die digitalen Kompetenzen hinaus, weshalb unser Fokus auf den Kompetenzmodellen bleibt.

Im sogenannten dpKa-Modell (digitales, programmierspezifisches Kompetenzaufbau-Modell) werden nun alle digitalen, programmierrelevanten Wissensbausteine, Fertigkeiten und Handlungsbereitschaften bzw. Kompetenzen zusammengefügt. Es handelt sich also um eine Auswahl an Wissen, Fertigkeiten und Kompetenzen der drei Kompetenzmodelle DigComp2.2AT, digi.komp8 und digi.komp12. Die Kompetenzen wurden dabei

grundsätzlich vom digi.komp12 übernommen und um einige Punkte des digi.komp8 ergänzt. Sämtliche Punkte des digi.komp8 wurden entsprechend einer verlaufsmäßigen Reihenfolge in die jeweilig passende Handlungsdimension hinzugefügt. Hauptaugenmerk waren natürlich der Bereich 4.2 „Algorithmen, Datenstrukturen und Programmierung“ des digi.komp12 und der Bereich 8 „Computational Thinking“ des digi.komp8, die konkret die Kompetenzen der Programmierung abgreifen. Die restlichen Kompetenzen halte ich aufgrund der im vorigen Absatz angeführten Überlegungen, die zum Teil auch aus meiner mehrjährigen Erfahrung als Programmierer stammen, für sehr wichtig für das Programmieren, weshalb diese ebenso berücksichtigt werden. Bei diesen zusätzlichen Kompetenzen handelt es sich meiner Ansicht nach um einen Grundstock für die Programmierung. Dadurch sind Punkte die als Voraussetzung fürs Programmieren mit Unity nützlich sein können und auch diejenigen die man sich dadurch aneignen kann in dieser Auswahl enthalten. In eckigen Klammern sind die inhaltlich zur Arbeit passenden Punkte aus dem DigComp2.2AT [DC ...] und dem digi.komp8 [Dk8 ...] angegeben, die mit dem jeweiligen Punkt deckungsgleich bzw. überwiegend überschneidend sind, wobei bei denjenigen des digi.komp8 auch der Text (in roter Schrift) hinzugefügt wurde. Die Kompetenzmodelle mit der vollständigen Zuordnung der jeweils anderen Modellpunkte befinden sich im Anhang. Diese Auswahl wurde zudem um Punkte, die nicht im digi.komp12-, aber im digi.komp8-Modell vorkommen erweitert (● rote Punkte). Das DigComp2.2AT-Modell hingegen bietet inhaltlich keine sinnvollen Ergänzungen an. Trotzdem kann man durch die angeführten Punkte des DigComp2.2At-Modells schlussendlich sehen, welche Kompetenzen davon abgedeckt werden. Zuletzt sind auch noch die Wissens- und Fertigkeitselemente des Kompetenzstrukturmodells für objektorientierte Programmierung [KsmOOP] aus dem vorigen Kapitel den Punkten zugeordnet, oder es wurden dafür, wo nötig neue Punkte erstellt (in blau).

### Dk12 – 1.2 Verantwortung, Datenschutz und Datensicherheit

#### Wissen und Verstehen (Wissen):

- **1.2.1** Ich kann meine Rechte und Pflichten in der Nutzung von Informatiksystemen beschreiben und wesentliche Aspekte des Datenschutzes und der Datensicherheit erklären. [DC 2.5, 3.3, 4.1, 4.2], [Dk8 2.4.2] SchülerInnen kennen die Grundzüge des Urheberrechts sowie des Datenschutzes (insb. das Recht am eigenen Bild) ...
- **Dk8 2.4.3** SchülerInnen kennen Lizenzmodelle, insb. Offene (Creative Commons, Open Educational Resources). [DC 3.3]

#### Anwenden und Gestalten (Fertigkeiten):

- **1.2.2** Ich kann beim Einsatz von Informatiksystemen mein Wissen um Pflichten und Rechte in Bezug auf meine Person und meine Arbeitsumgebung, auf persönliche und fremde Daten verantwortungsbewusst anwenden. [DC 2.5, 4.1, 4.2] [Dk8 2.4.2] ... und wenden diese Bestimmungen [bzgl. Urheberrecht und Datenschutz] an.

### Dk12 – 2.1 Technische Grundlagen und Funktionsweisen

#### Wissen und Verstehen (Wissen):

- **2.1.1** Ich kann Komponenten von Informatiksystemen beschreiben und ihre Funktionsweise und ihr Zusammenwirken erklären. [Dk8 7.1.1] SchülerInnen kennen die Bestandteile und Funktionsweise eines Computers und eines Netzwerks

#### Anwenden und Gestalten (Fertigkeiten):

- **2.1.3** Ich kann ein Computersystem samt Peripheriegeräten sachgerecht nutzen. [DC 0.2, 5.2] [Dk8 3.1.1] SchülerInnen nutzen die zum Normalbetrieb notwendigen Funktionen eines Betriebssystems einschließlich des Dateimanagements sowie der Druckfunktion. [Dk8 7.2.3] SchülerInnen nutzen unterschiedliche digitale Geräte entsprechend ihrer Einsatzmöglichkeiten

#### Reflektieren und Bewerten (Handlungsbereitschaft):

- **2.1.5** Ich kann unterschiedliche digitale Endgeräte bzw. Informatiksysteme in Bezug auf ihre technischen Eigenschaften und ihre Leistungsfähigkeit bewerten. [DC 5.2]
- **2.1.6** Ich kann einfache Fehler diagnostizieren und beheben. [DC 5.1] [Dk8 7.3.1] SchülerInnen erkennen technische Probleme in der Nutzung von digitalen Geräten und melden eine konkrete Beschreibung des Fehlers an die richtigen Stellen.

### Dk12 – 2.2 Betriebssysteme und Software

#### Wissen und Verstehen (Wissen):

- **2.2.2** Ich kann Kategorien von Software nennen und deren Anwendung beschreiben. [DC 5.2] [Dk8 7.1.2] SchülerInnen kennen gängige proprietäre und offene Anwendungsprogr. und zugehörige Dateitypen.

#### Anwenden und Gestalten (Fertigkeiten):



- **2.2.4** Ich kann mich in die Bedienung für mich neuer Software selbstständig einarbeiten. [DC 0.2]
- **Dk8 3.2.1** SchülerInnen geben Texte zügig ein, [DC 0.2]
- **Dk8 3.2.3** SchülerInnen führen Textkorrekturen durch (inkl. Zuhilfenahme von Überarbeitungsfunktionen, Rechtschreibprüfung). [DC 0.2]
- **Dk8 7.3.2** SchülerInnen nutzen Hilfesysteme bei der Problemlösung,
- **Dk8 7.3.3** SchülerInnen führen Datensicherungen und -Wiederherstellungen aus.

### Dk12 – 3.1 Produktion digitaler Medien

#### Wissen und Verstehen (Wissen):

- **3.1.1** Ich kann gängige Medienformate und ihre Eigenschaften beschreiben. [DC 3.1]

#### Anwenden und Gestalten (Fertigkeiten):

- **3.1.3** Ich kann digitale Medien in Form von Text, Ton, Bildern und Filmen sachgerecht bearbeiten, produzieren und publizieren. [DC 3.1], [Dk8 4.2.2] SchülerInnen gestalten digitale Medien mittels aktueller Technologien, ggf. unter Einbeziehung anderer Medien: Texte, Präsentationen, Audio-, Video-beiträge sowie multimediale Lernmaterial.

#### Reflektieren und Bewerten (Handlungsbereitschaft):

- **3.1.4** Ich kann digitale Produkte (Artefakte) in Bezug auf inhaltliche Relevanz, Wirkung und Design einschätzen und bewerten.
- **Dk8 4.2.1** SchülerInnen erleben sich selbstwirksam, indem sie digitale Technologien kreativ und vielfältig nutzen,
- **Dk8 4.2.5** SchülerInnen setzen Wissen über Techniken und Ästhetiken populärer Medienkulturen eigenverantwortlich um, [DC 3.1]

### Dk12 – 3.3 Suche, Auswahl und Organisation von Information

#### Wissen und Verstehen (Wissen):

- **3.3.1** Ich kann wichtige Informationsquellen im Internet anführen, die für meine schulischen und privaten Informationsbedürfnisse nützlich und notwendig sind. [DC 5.2, 5.3]
- **3.3.2** Ich kann lokal und in Netzwerken Methoden der Informationsgewinnung und -organisation benennen. [DC 1.1, 5.3]
- **3.3.3** Ich kann Möglichkeiten grundlegenden digitalen Wissensmanagements beschreiben.

#### Anwenden und Gestalten (Fertigkeiten):

- **Dk8 2.1.1** SchülerInnen formulieren ihre Bedürfnisse für die Informationssuche, [DC 1.1]
- **3.3.4** Ich kann unter Verwendung passender Dienste und Angebote und Wahl geeigneter Suchmethoden Informationen und Medien gezielt suchen und auswählen. [DC 1.1, 5.3], [Dk8 2.1.2] SchülerInnen planen zielgerichtet und selbstständig die Suche nach Informationen, Daten und

digitalen Inhalten mit Hilfe geeigneter Strategien und Methoden (z. B. Suchbegriffe), passender Werkzeuge bzw. nützlicher Quellen.

- **3.3.5** Ich kann im Rahmen persönlichen Lernmanagements Informationen und Medien strukturiert speichern und verfügbar halten., **[DC 1.3]**, **[Dk8 2.3.1]** SchülerInnen speichern Informationen, Daten und digitale Inhalte sowohl im passenden Format als auch in einer sinnvollen Struktur, in der diese gefunden und verarbeitet werden können.

### Dk12 – 4.2 Algorithmen, Datenstrukturen und Programmierung

#### Wissen und Verstehen (Wissen):

- **[KsmOOP 1]** Ich kann unterschiedliche Datenstrukturen beschreiben **5.1 data structure (graph, tree, array)**
- **4.2.1** Ich kann den Algorithmus-begriff erklären.
- **4.2.2** Ich kann Aufgaben und Problemstellungen algorithmisch und formalsprachlich in geeigneten Datenstrukturen beschreiben. **[DC 3.4]**, **[Dk8 8.1.4]** SchülerInnen formulieren eindeutige Handlungsanleitungen (Algorithmen) verbal und schriftlich [siehe auch 4.2.6] **5.3 algorithmic structure (loops, conditional statement)**
- **4.2.3** Ich kann wesentliche Aspekte und Methoden der Softwareentwicklung und des Softwareprojektmanagements erklären. **5.4 notional machine (data, working memory, processor, statement, program, automation)**
- **4.2.4** Ich kann wesentliche Aspekte der *Prozeduralen, Funktionalen und Objektorientierten* [sic!] Programmierung nennen und an Beispielen erläutern. **5.2 class & object structure (object, attribute, association)**
- **Dk8 8.2.2** SchülerInnen kennen unterschiedliche Programmiersprachen und Produktionsabläufe. **6 Mastering representation (language, syntax, semantics)**

#### Anwenden und Gestalten (Fertigkeiten):

- **Dk8 8.1.2** SchülerInnen verwenden, erstellen [und reflektieren] Codierungen (z. B. Geheimschrift, QR-Code), **[DC 0.1]** **6 Mastering representation (language, syntax, semantics)**
- **4.2.5** Ich kann Aufgaben mit Mitteln der Informatik modellieren.
- **4.2.6** Ich kann Algorithmen entwerfen, diese formal darstellen, implementieren und testen. **[DC 3.4]**, **[Dk8 8.1.4]** SchülerInnen formulieren eindeutige Handlungsanleitungen (Algorithmen) verbal und schriftlich [siehe auch 4.2.2] **[Dk8 8.2.3]** SchülerInnen beherrschen grundlegende Programmierstrukturen (Verzweigung, Schleifen, Prozeduren). **5.3 algorithmic structure (loops, conditional statement)**
- **[KsmOOP 2]** Ich kann Datenstrukturen, sowie Klassen- und Objektstrukturen in einem Algorithmus verwenden **5.1 data structure (graph, tree, array)**, **5.2 class & object structure (object, attribute, association)**
- **4.2.7** Ich kann ein Softwareprojekt planen und durchführen. **[Dk8 8.2.1]** SchülerInnen erstellen einfache Programme oder Webanwendungen mit

geeigneten Tools, um ein bestimmtes Problem zu lösen oder eine bestimmte Aufgabe zu erfüllen, [DC 5.3] 5.4 notional machine (data, working memory, processor, statement, program, automation)

### Reflektieren und Bewerten (Handlungsbereitschaft):

- **4.2.8** Ich kann die Schritte der Softwareentwicklung reflektieren.
- **4.2.9** Ich kann die Angemessenheit der Entwicklungswerkzeuge grob einschätzen.
- **Dk8 8.1.5** SchülerInnen entdecken Gemeinsamkeiten und Regeln (Muster) in Handlungsanleitungen, [DC 0.1]
- **Dk8 8.1.6** SchülerInnen erkennen die Bedeutung von Algorithmen in automatisierten digitalen Prozessen (z. B. automatisiertes Vorschlagen von potenziell interessanten Informationen).
- **4.2.10** Ich kann die Effizienz von Algorithmen bewerten.
- **4.2.11** Ich kann gezielt nach Programmfehlern suchen und diese korrigieren. [DC 5.1]

In diesem dpKa-Modell haben wir nun alle digitalen, programmierrelevanten Wissensbausteine, Fertigkeiten und Handlungsbereitschaften bzw. Kompetenzen zusammengefügt, sodass wir nun damit das Unterrichtskonzept untersuchen können, eine Grundlage für den empirischen Teil haben und schließlich die Forschungsfragen beantworten können.

### 5 Unterrichtskonzept

Dieses Kapitel geht auf das derzeit existierende Unterrichtskonzept, mit dem dafür ausgearbeitetem Tutorial ein. Hier wollen wir aus dem Unterrichtskonzept bereits mögliche behandelte Bereiche des dpKa-Modells herausfiltern. In den Klammern sind die entsprechend abgedeckten Punkte des dpKa-Modells angegeben.

#### 5.1 Verwendete Tools

Das aktuelle Tutorial auf der Moodle-Seite des GRG 16, Maroltingergasse sowie das eigens für den Unterricht erstellte learn2proGrAME-Tutorial (<https://learn2progame.github.io/learn2proGrAME-Tutorial/>) für den Einsatz von Unity stellen sozusagen die Lerneinheiten dar. Allerdings ist nicht jede Lerneinheit und jedes damit verbundene Tool direkt für das Erlernen einer Programmiersprache von Nutzen. Folgende Tools inklusive ihrer entsprechenden Lerneinheiten finden in den Tutorials jedenfalls Verwendung: GIMP, Scratch online (<https://scratch.mit.edu/>), Visual Studio Express 2010, Unity, evtl. Audacity (**2.2.2W**). Das Bildbearbeitungsprogramm GIMP sowie das Soundbearbeitungsprogramm Audacity dienen uns, um das ganze Projekt rund um die eigene Spieleprogrammierung abzurunden und die SchülerInnen zusätzlich zu motivieren. Mit GIMP beispielsweise soll eine eigene Spielfigur erstellt werden und mit Audacity eigene Sounds (**3.1.1W, 3.1.3A**). Scratch online hingegen ist eine 2D-Spiele-Engine für Nichtprogrammierer, mit einer grafischen Programmiersprache, bestehend aus zusammensteckbaren Codeblöcken – kann man sich wie Legosteine vorstellen –, die eine großartige Möglichkeit darstellt, um jungen Menschen Programmierkonzepte wie Konditionen, Schleifen und Eventbasierende Programmierung beizubringen (**4.2.2W, 4.2.4W, 4.2.6A**) (vgl. Gregory 2014, S. 31). Die integrierte Entwicklungsumgebung Visual Studio Express 2010 in Verbindung mit der Spiele-Engine Unity bieten einem schließlich die Möglichkeit, das

Programmieren in seiner reinsten Form bzw. eine konkrete Programmiersprache zu erlernen.

Wie man unter anderem anhand der Spielfigur-Erstellung mit GIMP sehen konnte, benötigt man mehr als nur Programmierkenntnisse um ein eigenes Computerspiel zu erstellen. Für die Entwicklung benötigt man ein ganzes Team aus Spezialisten mit besonderen Kenntnissen aus unterschiedlichen Bereichen. Ein typisches Spielentwicklungsteam besteht normalerweise aus den folgenden fünf Hauptdisziplinen, wobei jede weitere Unterdisziplinen beherbergt: engineers (Softwareentwickler/Techniker), artists (Künstler), game designers (Spieldesigner), producers (Produzenten) und anderes Management- bzw. Supportpersonal (Marketing, Rechte, IT, ...) (vgl. Gregory 2014, S. 5). Wobei alle Personen innerhalb eines Spielentwicklungsteams zumindest zwischenzeitlich in die Rolle des Game Designers schlüpfen, indem sie alle höchstwahrscheinlich die eine oder andere Entscheidung hinsichtlich des Spielkonzepts selbstständig treffen (müssen) (vgl. Schell und Feilen 2012, S. 25). Im Zuge des Tutorials von proGrAME schlüpft man in mehrere Rollen, wie in die des Künstlers, wenn man sich seine Spielfigur erstellt, oder in die des Managementpersonals wenn man über die Werknutzungsrechte der eigenen Spielfigur und ergänzenden Bildmaterialien aus dem Internet informiert wird, aber klarerweise auch in die des Softwareentwicklers selbst. **(4.2.3W)**

---

### 5.2 Aufbau des Unterrichtskonzeptes

Schon zu Beginn des Tutorials, also noch bevor irgendwelche Tools verwendet werden, gibt es bereits Übungsaufgaben zum Thema des Dateimanagements **(2.1.3A)**:

# Dateimanagement

Für Teilnehmer/innen verborgen



Dateien und Ordner suchen erstellen, umbenennen, verschieben und löschen.  
ZIP-Dateien erstellen und entpacken.



Aufgabe - Ordnerstruktur und Dateien



Grundgerüst für eine HTML-Datei



Wie verpacke ich mehrere Dateien in eine ZIP-Datei mit 7-Zip?

Abbildung 12: Dateimanagement. Screenshot vom Moodle-Kurs des GRG16 Maroltingergasse (Comber, O. et. al., 2019)

Nach dem Dateimanagement findet man die Einführung in die Bildbearbeitung mit GIMP, wo schließlich auch die Spielfigur erstellt wird und eigens dafür erstellte Video-Tutorials auf Youtube zum Einsatz kommen (**3.3.1W**).

## Bildbearbeitung mit GIMP



GIMP ist eine freie, kostenlose Bildbearbeitungssoftware.

[Download GIMP](#)



GIMP Einzelfenstermodus



GIMP - Benutzerhandbuch



GIMP Tutorials



Die GIMP Hauptansicht



Aufgabe - GIMP Fotocollage



GIMP Datei - Elefant



Tutorial Videos: Spielfigur



InfQuiz - GIMP



Bilder verbessern

Verbessere die Bilder, die in sich in diesem Ordner befinden!

Abbildung 13: GIMP. Screenshot vom Moodle-Kurs des GRG16 Maroltingergasse (Comber, O. et. al., 2019)

Danach folgt die Einführung ins Programmieren mit Scratch. Auch hier findet man wieder eine Tutorial-Aufgabe und anschließend fünf weitere Übungsaufgaben, wo man sich mit folgenden Themen auseinandersetzt:

Variablen setzen, Benutzereingaben/Benutzerausgaben, Events und Trigger, Schleifen (while/do-while) und Bedingungen (if) (**4.2.2W**).

## Einführung ins Programmieren mit Scratch

Für Teilnehmer/innen verborgen



Erste Schritte mit Scratch

1. Starte Scratch [http://scratch.mit.edu/projects/editor/?tip\\_bar=getStarted](http://scratch.mit.edu/projects/editor/?tip_bar=getStarted)
2. Sieh dir das Video an und mache das Tutorial durch.
3. Wenn du fertig bist, speichere deine Projekte (*mit Datei/Herunterladen auf deinen Computer*) auf deinem Netzlaufwerk im Ordner "Scratch". Wähle geeignete, aussagekräftige Dateinamen.

Aufgabe - Scratch Tutorial Projekt

Aufgabe - Scratch Geschichte

→ **Scratch - veröffentlichte Beispielprojekte** (Lösungen der nachfolgenden Aufgaben)

Aufgabe - Scratch Fangspiel

Aufgabe - Scratch Pfeiltastensteuerung mit Animation

Aufgabe - Scratch Gelsenplage

Aufgabe - Scratch "Scratch die Kurve"

Abbildung 14: Scratch. Screenshot vom Moodle-Kurs des GRG16 Maroltingergasse (Comber, O. et. al., 2019)

Hat man die Scratch-Übungen einmal hinter sich gebracht, landet man bei der Programmierung mit Microsoft Visual Studio Express 2010 in der Programmiersprach C# (**KsmOOP 1, Dk8 8.2.2W, Dk8 8.1.2A, 4.2.6A, bzgl. Texteingabe: Dk8 3.2.1A, Dk8 3.2.3A**).

## Programmieren mit C# - Einführung Konsoleanwendung



Einführung in der Programmieren mit C# und Visual Studio 2010 Express.

[Download Microsoft Visual Studio Express 2010](#)

C# Sharp Tutorial - Csharpme.de

C# Sharp Tutorial - SoloLearn

Erstes Programm schreiben (Hello World!)

Was ist C#? Installation am eigenen PC!

Abbildung 15: Microsoft Visual Studio Express 2010. Screenshot vom Moodle-Kurs des GRG16 Maroltingergasse (Comber, O. et. al., 2019)

Dabei werden einem zum Beispiel Punkte wie Variablen, Datentypen, die for-Schleife, die sleep-Funktion und der Random-Mechanismus beigebracht.

## KAPITEL: Unterrichtskonzept

---

Nach Abschluss aller bisherigen Tutorial-Teile kommt man zur Spieleentwicklung mit Unity, für das es ein ganz eigenes, externes, von Herrn Oswald Comber erstelltes Tutorial gibt (<https://learn2progame.github.io/learn2proGrAME-Tutorial/>).

### Spieleentwicklung mit Unity



Unity ist eine Entwicklungsumgebung für 2D- und 3D-Spiele, die auch von zahlreichen professionellen Entwicklern genutzt wird.

Downloadlink Unity (inkl. Visual Studio Community Edition): <https://unity3d.com/de/get-unity/download/archive>

In der Schule arbeiten wir mit Version 5.6.2.



→ [Link zum allgemeinen Tutorial](#)

→ [Direkt zum Bouncy-Fant Tutorial](#)



Kostenlose freie Grafiken

Kostenlose freie Sprites, Tilesets und mehr auf <https://www.gameart2d.com/freebies.html>. Lizenz für die kostenlosen freien Sprites und Tilesets: <https://www.gameart2d.com/license.html>

Kostenloses Tileset Regenwald ([Download](#))



Freie GamePinS Tiles and Sprites

Diese Tiles und Sprites sind von Oswald Comber erstellt und stehen unter nachfolgender Lizenz zur freien Verwendung zur Verfügung:



Creative Commons Namensnennung 4.0 International Lizenz.



Gemeinsame Sammlung freier Assets (Grafiken und Sounds)



Video Tutorials - YouTube Channel learn to proGrAME

→ [Youtube Channel](#)



→ [Youtube Channel - learn to proGrAME](#)

Abbildung 16: Unity. Screenshot vom Moodle-Kurs des GRG16 Maroltingergasse (Comber, O. et. al., 2019)

In diesem großen Kapitel lernen die SchülerInnen nicht nur Algorithmen in C# zu entwickeln (**4.2.2W**, **Dk8 8.2.2W**, **Dk8 8.1.2A**, **4.2.6A**), sondern auch Daten-, Klassen- und Objektstrukturen kennen und einzusetzen, indem sie Spiele programmieren (**KsmOOP1W**, **4.2.4W**, **KsmOOP2A**). Die Spiele-Tutorials



haben dabei bereits Softwareprojekt-Charakter, da man letzten Endes auch lernt, fertige Applikationen zu erstellen (**4.2.3W**).

Folgt man dem Link „Link zum allgemeinen Tutorial“ kommt man auf die Website des Tutorials, auf der sogar beschrieben ist, was man lernen wird:

Was wirst du in diesem Tutorial lernen?

In diesem Tutorial lernst du die Grundlagen zur Spieleprogrammierung in Unity mit der Programmiersprache C#.

- Dabei wirst du zuerst die [Bedienoberflächen von Unity und Visual Studio](#) erkunden.
- Danach schreibst du dein [erstes Programm in C#](#).
- Wie gehen Programmiererinnen und Programmierer mit Fehlern um? Darüber erfährst du mehr im Kapitel zu [De-Bugging](#).
- Anschließend lernst du über [Variablen](#) und [Operatoren](#), um dein erstes Spiel, das allseits beliebte [Krapfenklicker](#), zu entwickeln.
- Nachdem du die Unity Spielobjektkomponente [Transform](#) kennengelernt hast, kannst du eine Spielerweiterung für [Krapfenklicker](#) schreiben.
- Beim nächsten selbstgeschriebenen Spiel - [Ellis' großes Abenteuer](#) - lernst du, mit [Verzweigungen](#) Spielzustände zu erzeugen und Figuren in einem Grafikprogramm namens [GIMP](#) zu zeichnen.
- Im ersten Teil des Tutorials [Space-Asteroids](#) lernst du den Umgang mit [Funktionen](#) und [Zugriffsmodifikatoren](#). Weiters lernst du das Konzept der Kollisionsabfrage kennen. Danach lernst du [Partikeleffekte](#) kennen und wirst damit coole Spezialeffekte in das Spiel einbauen.
- Als Vorbereitung auf das Jump'n'Run artige Arcade Spiel [BouncyFant](#) wird letztlich die [Schleife](#) - insbesondere die [Spielschleife](#) - als wichtiger Aspekt der Spieleprogrammierung vorgestellt.
- Letztlich werden die [Grundlagen objekt-orientierter Programmierung](#) vorgestellt.

Abbildung 17: Unity-Tutorial (Comber, O. et. al., 2019) Screenshot von der Website. Quelle: <https://learn2programe.github.io/learn2proGrAME-Tutorial/> (14.04.2020)

Nach den Spiele-Tutorials in Unity, bekommen die SchülerInnen die Aufgabe eigene Spiele zu entwickeln. Dadurch lernt man teilweise auch in kleinem Rahmen ein Softwareprojekt zu planen und durchzuführen (**4.2.7A**). Spätestens nach diesem Kapitel hat man mehrere Entwicklungswerkzeuge kennengelernt und sollte deren Angemessenheit für unterschiedliche Projekte grob einschätzen können (**4.2.9R**). Außerdem lernt man, insofern man sich das oben aufgelistete Tutorial-Kapitel „De-Bugging“ ansieht, gezielt nach Programmfehlern zu suchen und diese zu korrigieren (**2.1.6R**, **4.2.11R**).

### 5.3 Pädagogische Perspektive

Um Kompetenzen, die wie wir gehört haben, personenbezogen sind, zu vermitteln, sollte man der Selbstentfaltung bzw. dem Individuum als Ganzes möglichst viel Freiraum geben, aber trotzdem das Interesse an der Aneignung der Kompetenzen fördern. Denn die Chance auf erfolgreiches Lernen und somit auf den Erwerb von Kompetenzen, ist dann am höchsten, wenn die Lernanreize möglichst nahe am Übergangspunkt zwischen Wissen und Nicht-Wissen gesetzt werden. Je tiefer sich die Lernanreize im Bereich des Nicht-Wissens befinden, desto wahrscheinlicher führt man Lernende in den Zustand der Überforderung, da Lernvoraussetzungen fehlen. Setzt man die Lernanreize zu weit in den Bereich des Wissens, so wird hingegen der Zustand der Unterforderung immer wahrscheinlicher, wodurch Lernpotenziale nicht ausgeschöpft werden (vgl. Olechowski und Bábosik 2003, S. 215). Aufgrund dessen, dass man in einer Klasse nicht nur mit einer lernenden Person zusammensitzt, sondern oftmals ca. 25 SchülerInnen vor sich hat, gestaltet sich die Aufgabe die Lernanreize richtig zu setzen, als sehr schwer zu meisternde Aufgabe. Und trotzdem muss man sich dieser stellen. Johann Friedrich Herbart bezeichnete bereits vor über 200 Jahren die Ignoranz der Differenzierung, also das nicht beachten der Unterschiedlichkeit einzelner SchülerInnen, als einen Grundfehler, indem er auf die Verschiedenheit der Köpfe hindeutet, die das große Hindernis aller Schulbildung seien (vgl. Herbart in Jank und Meyer 2009, S. 78). Darauf achtend hat das Bundesministerium für Unterricht, Kunst und Kultur den dafür verwendeten Begriff „Individualisierung“ im Rahmen des Unterrichts präzisiert: „Unter Individualisierung verstehen wir die Gesamtheit aller unterrichtsmethodischen und lern- / lehrorganisatorischen Maßnahmen, die davon ausgehen, dass das Lernen eine ganz persönliche Eigenaktivität jeder einzelnen Schülerin bzw. jedes einzelnen Schülers selbst ist, und die darauf abzielen, die Schülerinnen und Schüler dabei gemäß ihrer Persönlichkeit, ihrer Lernvoraussetzungen und Potenziale bestmöglich zu fördern und zu fordern.“ (Bundesministerium für Unterricht, Kunst und Kultur 2007). Diese Individualisierung verlangt entsprechende Lehr- und Lernmittel,

sowie Lernumgebungen und einer Förderung des Individuums durch ein sowohl angemessenes Verhalten und als auch einer angemessenen Haltung der LehrerInnen. Diese sollen also in wertschätzender Art und Weise die SchülerInnen beim Lernen begleiten, auf sie abgestimmte Lernangebote und Informationen bereitstellen und ihnen die Bedeutsamkeit der Inhalte näherbringen. Dabei sollen die Lernenden zu eigenverantwortlichem Lernen ermutigt werden und es sollen ihnen stets bewertungsfreie Bereiche zur Entfaltung individueller Neigungen gegeben werden (Radnitzky und Westfall-Greiter 2009).

Diesen riesigen Spagat, die Lernanreize richtig zu setzen unter Beachtung der Individualisierung von bis zu 25 SchülerInnen schafft das Projekt „learn to proGrAME“ durch ein von Kopf bis Fuß durchdachtes und hervorragend aufbereitetes Tutorial. Die SchülerInnen können dadurch selbstständig arbeiten und können alle benötigten Informationen direkt im Tutorial jederzeit nachlesen (**3.3.1W, 3.3.2W, 3.3.4A**). Die Lehrperson hat somit den Freiraum, die SchülerInnen in wertschätzender Art zu begleiten. Auf die Frage ob und wie die Lehrperson diesen Freiraum nutzt, hat man keinen Einfluss und das wird auch von Lehrkraft zu Lehrkraft unterschiedlich sein. Weiter wird den SchülerInnen zum Beispiel mit der Möglichkeit eine Spielfigur zu zeichnen und der Entwicklung eines eigenen Spieles (**3.1.3A**), ein konkreter bewertungsfreier Bereich zur Entfaltung gegeben (**Dk8 4.2.1R**). Damit wird dem Konzept der Zwang zu notenbedachter Leistung zumindest teilweise entzogen, wodurch sie sich bei der Spielfigur-Gestaltung selbstwirksam erleben können. Trotz des durchdachten Tutorials und möglicher Hilfen durch die Lehrperson handelt es sich beim Programmieren um ein Tätigkeitsgebiet, das sich rund um „Versuch und Irrtum“ bewegt, was sehr frustrierend sein kann. Damit die Motivation trotzdem nicht vor dem Erreichen der schönen Erfolgserlebnisse, die man beim ersten erfolgreichen Ausführen des Programmes hat, verloren geht, sollte man Ihnen schon zu Beginn des Programmierunterrichts bewusst machen, dass ein Programm selten vom ersten Moment an perfekt und fehlerfrei funktioniert, sondern dass es meistens durch Trial and Error entsteht, also probieren, irren

und korrigieren (vgl. Fehr 2017, S. 20). Auch das fand Einzug in das Tutorial, in dem diese Problematik bereits zu Beginn unter der Überschrift „Vorwort und Motivation“ angesprochen wird:

*„Anfänger, Fortgeschrittene und Meister haben eines gemeinsam: Jeder macht mal Fehler. Nur der Umgang damit unterscheidet sich. Viele Fehler werden bereits vom Compiler erkannt, aber manche Fehler bleiben verborgen und treten nur unter bestimmten Bedingungen zu Tage. Diese Fehler werden auch als Bugs bezeichnet. Aber davon wollen wir uns nicht einschüchtern lassen, sondern den Schritt vom "Anwender" zum "Programmierer" wagen!“ (Comber 2020)*

Damit ist ein weiterer wichtiger Bereich des Programmierens angesprochen, nämlich der der Fehlersuche und -korrektur (**2.1.6R, 4.2.11R**).

Insgesamt betrachtet hat dieses Tutorial bzw. Unterrichtskonzept ein riesiges Potenzial, um im Sinne der Allgemeinbildung kompetente Programmierer aus den SchülerInnen zu machen. Daher sollte es in seiner Grundform auf keinen Fall verändert werden. Sehr wohl aber sind hier und da Verbesserungen möglich. Wir untersuchen das Ganze nun in Bezug auf die Vermittlung von Kompetenzen anhand von Experteninterviews und versuchen unter anderem, mit den Ergebnissen kompetenzbezogene Stärken und Schwächen ans Tageslicht zu bringen. Dadurch sollten sich schließlich mögliche Verbesserungen herausstellen.

Einen Überblick dazu, was man hier herauslesen konnte und was tatsächlich erworben wird, gibt die Vergleichsmatrix im Kapitel „Ergebnisdarstellung“.

---

## 6 Empirischer Teil

In diesem Kapitel wird zuerst das methodische Vorgehen besprochen. Danach findet die Auswertung des Fragebogens statt, der für eine andere Studie konzipiert wurde und jeweils am Ende des durchgeführten Unterrichtsprojektes oder am Schuljahresende von den SchülerInnen ausgefüllt wurde bzw. wird, aber auch für diese Studie sehr interessante Ergebnisse liefert. Anschließend auf das Interview eingegangen, der Interviewleitfaden präsentiert und die Ergebnisse des Interviews ausgewertet. Zum Schluss der Arbeit folgen eine Ergebnisdarstellung, eine Diskussion und die Deutung der Ergebnisse.

---

### 6.1 Methodisches Vorgehen

Hier nochmal ein Blick auf die Forschungsfragen, die hier beantwortet werden sollen:

5. Welche digitalen Kompetenzen und Fertigkeiten und welches Vorwissen benötigen jugendliche Schülerinnen und Schüler, um im Unterricht erfolgreich Computerspiele mit Unity zu entwickeln?
6. Welche digitalen Kompetenzen, Fertigkeiten und welches Wissen werden in der Computerspieleentwicklung mit Unity erworben?

Die erste Frage, die nach den Voraussetzungen für die Spieleprogrammierung mit Unity fragt, sucht nicht nur nach Kompetenzen, sondern auch nach kleineren Dingen wie den Voraussetzungsbestandteilen, nämlich Wissen und Fertigkeiten. Die zweite Frage fordert ebenso tatsächlich erworbene Kompetenzen, Fertigkeiten und Wissen als Antwort, jedoch mit Blick auf die Spieleentwicklung mit Unity selbst. Wie man an den Forschungsfragen bereits erkennen kann, ist der Forschungsgegenstand dieser Studie die Kompetenz bzw. auch deren Bausteine, nämlich Wissen, Fertigkeit und Handlungsbereitschaft.

Es wurde bereits erfolgreich versucht, aus dem Unterrichtskonzept potenziell zu erwerbende digitale Kompetenzen und Voraussetzungen herauszulesen. Auf diesen soll nun aufgebaut werden.

Im Laufe einer anderen Studie wurde bereits ein Fragebogen beantwortet, der teilweise quantitativ Aussagen der SchülerInnen wiedergeben kann, welche digitalen Kompetenzen (Kategorien) man laut ihnen durch das Projekt erwirbt. Daher wird dieser Fragebogen zuerst unter die Lupe genommen. Allerdings wurde dieser Fragebogen nicht für die hier durchgeführte Studie konzipiert, weshalb nur auf ein paar digitale Kompetenzen in quantitativer Weise hingewiesen werden kann. Diese Fragebogenuntersuchung liefert jedenfalls nur Hinweise auf Kompetenzen, da eine Aussage von SchülerInnen als Selbsteinschätzung keine Kompetenz nachweisen kann. Sehr wohl aber ist dies durch Kompetenztestungen oder Experteninterviews möglich. Bei den Experteninterviews berichten Experten aus ihrem Alltag und können anhand von Erfahrungen bzw. Beispielen, wo man die Aneignung einer Kompetenz von SchülerInnen miterlebt hat, den Kompetenzerwerb nachweisen.

Nach der Fragebogenauswertung wird daher versucht, mit Hilfe eines Experteninterviews die Forschungsfragen möglichst vollständig zu beantworten. Dabei fiel die Entscheidung, die Forschungsfragenbeantwortung mit einem Experteninterview abzuwickeln, unter anderem mit der Begründung, dass die Experten nicht nur Unity, sondern das ganze Unterrichtskonzept mit all seinen Facetten kennen und das nötige Verständnis für die Entwicklung von Kompetenzen innerhalb dieses Unterrichtskonzeptes haben. Bei den Experten handelt es sich um Lehrpersonen, die dieses Projekt in ihren Schulklassen bereits durchgeführt haben. Als Methode zur Auswertung der Interviews wurde die qualitative Inhaltsanalyse nach Mayring gewählt, da für diese Studie weder die Gefühle noch der Dialekt oder Sonstiges, sondern nur der Inhalt selbst für die Auswertung bedeutend sind.

---

## 6.2 Fragebogen (quantitative Hinweise)

Am Ende dieses Unterrichtsprojekts oder auch am Schulende werden die SchülerInnen jeweils gebeten, einen Fragebogen zum Informatikunterricht auszufüllen (siehe Anhang). Darunter waren folgende, für diese Studie im Sinne von quantitativen Hinweisen, interessante Fragen bzw. einzustufende Statements zu finden, wobei in Klammer wieder jeweils die abgedeckten Punkte des dpKa-Modelles angegeben sind:

1. Durch die Spieleprogrammierung ...
  - a. verstehe ich besser wie Programme ablaufen! (4.2.2W)
  - b. habe ich gelernt besser zu programmieren! (Dk8 8.1.2A, 4.2.6A)
  - c. verstehe ich besser was ein Algorithmus ist! (4.2.1W)
  - d. verstehe ich besser wie Softwareprojekte ablaufen! (4.2.3W)
  - e. welche Probleme beim Entwickeln von Software auftreten! (4.2.11R)
2. Spieleprogrammierung hat im Vergleich zum herkömmlichen Informatik-Unterricht mir in folgenden Bereichen "mehr" oder "weniger" gebracht:
  - a. Teamwork (**wird hier nicht behandelt: keine digitale Kompetenz**)
  - b. Projektmanagementfähigkeiten (4.2.3W, 4.2.7A)
  - c. Verwenden von Anwendungsprogrammen (**keine Zielkompetenz**)
  - d. Programmieren (Dk8 8.1.2A, 4.2.6A, 4.2.11R)
  - e. Verstehen von Datenstrukturen (KsmOOP 1W)
  - f. Verständnis von Algorithmen (4.2.1W, 4.2.2W)
  - g. Einblick in die Logik von Programmen (Dk8 8.1.6R)

Der erste Teil dieser Fragen, fragt nach konkreten Verbesserungen in bestimmten Bereichen und ist somit als Vorinformation für die Experteninterviews sehr wertvoll. Der zweite Teil dieser Fragen untersucht hingegen die Unterschiede zwischen dem Unterricht der Spieleentwicklung mit Unity und dem herkömmlichen Unterricht, also welcher mehr oder weniger in unterschiedlichen Bereichen bringt. Dadurch entsteht keine konkrete Aussage darüber, ob Kompetenzerwerbe von den SchülerInnen erzielt werden. Diese Problematik wird später noch genauer erläutert.

Es wird nun auf die Ergebnisse der Fragen und den mit den Fragen verbundenen Kompetenzen eingegangen. Die Fragebogenergebnisse können aber ihre jeweils angeknüpften Kompetenzen nicht nachweisen, sondern stellen nur einen Hinweis darauf dar, denn um den Kompetenzerwerb der SchülerInnen nachzuweisen, benötigt man Kompetenztestungen oder eben Experteninterviews. In den folgenden Diagrammen sind die Zustimmungen auf das entsprechende Statement bzw. die entsprechende Aussage in grün strichliert und die Verneinungen in rot punktiert dargestellt. Die Balken für keine Antwort und „weder noch“ sind mit hellgrau voll ausgefüllt. Zwischen den Antwortmöglichkeiten es „trifft nicht zu“ und es „trifft wenig zu“ gibt es natürlich einen Unterschied, der aber vor allem dann interessant wird, wenn starke Extrema auftreten oder eine relative Ausgeglichenheit zwischen Zustimmung und Verneinung herrscht. Dazu wird dann aber explizit bei dem jeweiligen Ergebnis diskutiert. Außerdem habe ich dafür eine Gewichtung vorgenommen, sodass „trifft völlig zu“ den Wert 2 und „trifft eher zu“ den Wert 1 bekommt, mit der jeweiligen Antwortanzahl multipliziert und anschließend die beiden Ergebnisse zusammengezählt werden, sodass man einen, in diesem Fall positiven Ausschlag erhält. Von diesem Ergebnis für den positiven Ausschlag wird dann das Ergebnis des negativen Ausschlags, der entsprechend dem positiven Ausschlag berechnet wird, abgezogen, wodurch man eine Tendenz erhält. Je weiter im Positiven die Tendenz liegt, desto stärker die Zustimmung und je weiter sie im Negativen liegt, desto stärker ist die Verneinung der Aussage oder Frage. Diese Berechnung zeige ich beim ersten Diagramm als konkretes Beispiel vor, sodass hier keine Unklarheiten bleiben.



6.2.1 Fragebogen Teil 1 (Erwerb von Kompetenzbausteinen)

Frage 1a:

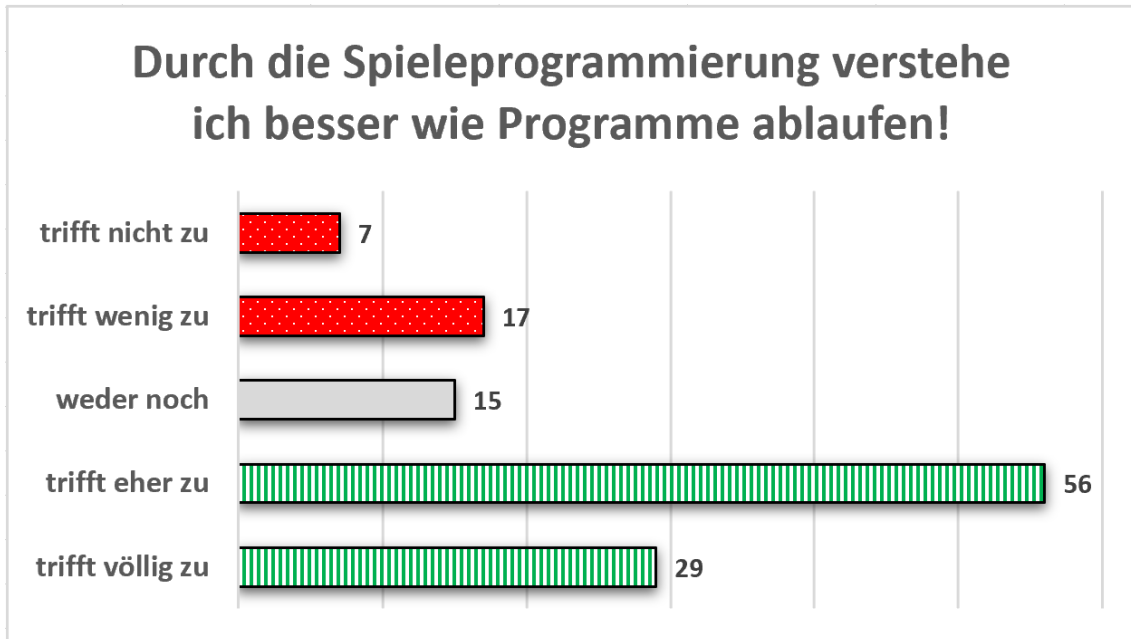


Abbildung 18: Verständnis der Lernenden des Ablaufs von Programmen (N=124) (Eigene Darstellung)

Diese Frage deckt den Wissensbaustein 4.2.2 des dpKa-Modells ab, laut dem man Aufgaben und Problemstellungen algorithmisch und formalsprachlich in geeigneten Datenstrukturen beschreiben kann. Wie man bereits am Diagramm sehen kann, herrscht hier eine überwiegende Zustimmung diesen Wissensbaustein erworben zu haben. Wie stark diese Zustimmung ist möchte ich nun mit der oben angesprochenen Berechnung der Tendenz verdeutlichen:

ANTWORT	GEWICHTUNG	ANZAHL	ERGEBNIS
trifft völlig zu	2	29	58
trifft eher zu	1	56	56
<b>POSITIVER AUSSCHLAG</b> (Summe der Zustimmungen):			<b>114</b>
trifft nicht zu	2	7	14
trifft wenig zu	1	17	17
<b>NEGATIVER AUSSCHLAG</b> (Summe der Verneinungen):			<b>31</b>
<b>TENDENZ-Wert</b> (positiver minus negativer Ausschlag):			<b>83</b>

Wie man anhand des Tendenzwertes sehen kann, überwiegt die Zustimmung sehr deutlich, weshalb sich eine klare und eindeutige Aussage treffen lässt.

Frage 1b:

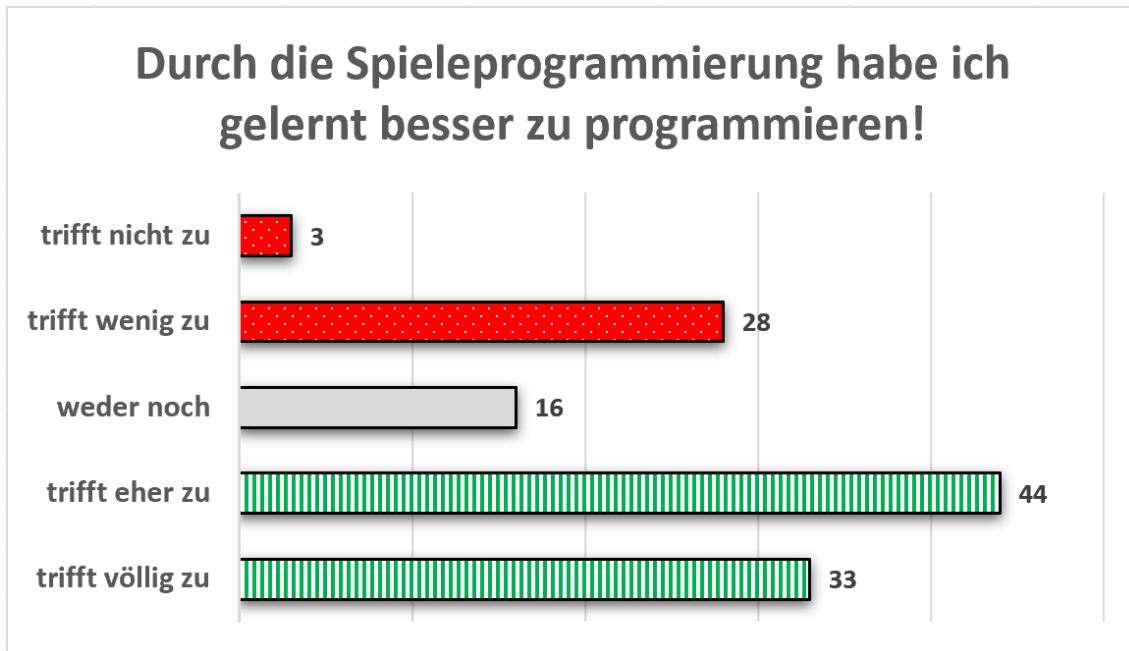


Abbildung 19: Erwerb der Fertigkeit Programmieren (N=124) (Eigene Darstellung)

Auch bei diesem Statement zeigt sich bereits im Diagramm die klar stärker vertretene Zustimmung. Der Tendenzwert  $(33 \cdot 2 + 44 - 28 - 3 \cdot 2)$  beträgt 76, was wiederum eine eindeutige Bejahung ausdrückt. Der kurze „weder noch“-Balken spricht für eine polarisierende Meinung dazu und in der Kombination, dass niemand „keine Antwort“ gegeben hat, möglicherweise auch für ein einfach verständliches Statement für die SchülerInnen. Bei dem Statement werden jedenfalls die Fertigkeitsbausteine Dk8 8.1.2A, 4.2.6A des dpKa-Modells angesprochen. Dk8 8.1.2A lautet: SchülerInnen verwenden und erstellen Codierungen, dem der KsmOOP-Modell-Punkt 6 „Mastering representation (language, syntax, semantics)“ zugeordnet ist. 4.2.6A lautet: Ich kann Algorithmen entwerfen, diese formal darstellen, implementieren und testen. Diesem Punkt wurde der KsmOOP-Modell-Punkt 5.3 „algorithmic structure (loops, conditional statement)“ zugeordnet. Alle diese Punkte sind somit durch dieses Fragebogen-Statement quantitativ, durch SchülerInnenaussagen als wahrscheinlich erworben angesprochen.

## Frage 1c:

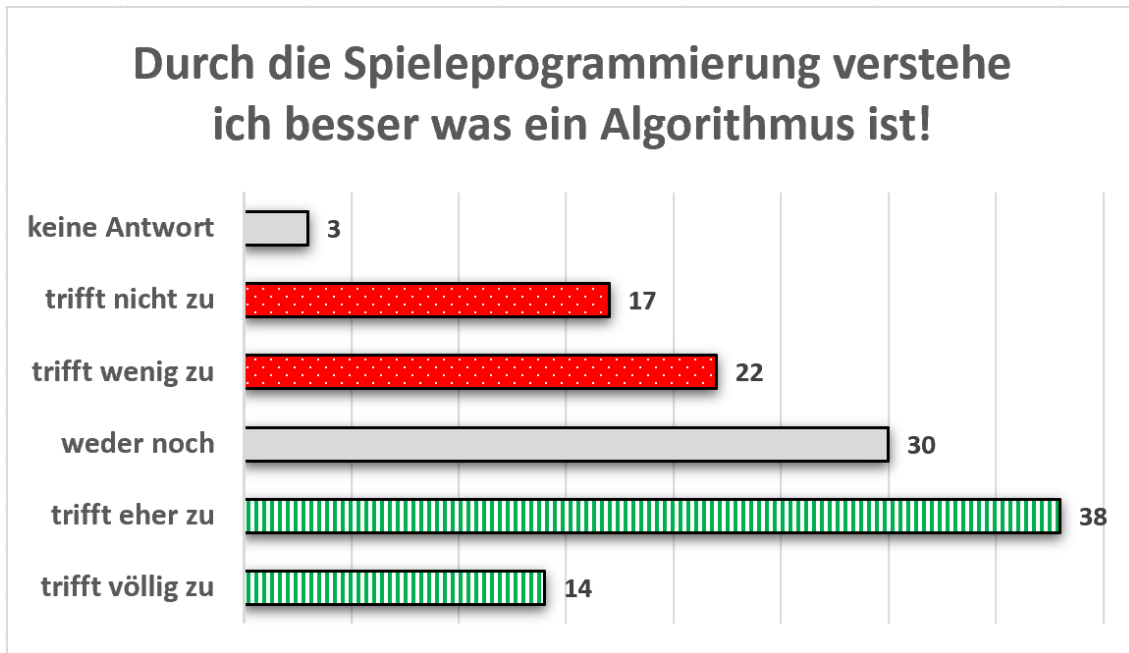


Abbildung 20: Verständnis des Algorithmusbegriffes (N=124) (Eigene Darstellung)

Mit dieser Frage wird quasi eins zu eins der Punkt 4.2.1W des dpKa-Modells abgefragt, nämlich ob man den Algorithmus-begriff erklären kann. Man sieht hier in der Grafik bereits eine gleichmäßigere Verteilung im Vergleich zu den ersten beiden Fragen, obwohl trotzdem noch die Zustimmung die Mehrheit hat. Der Tendenzwert ( $14 \cdot 2 + 38 - 22 - 17 \cdot 2$ ) beträgt hier nur 10 und liegt damit nicht weit weg von der neutralen 0, weshalb man meiner Meinung nach, aufgrund der relativ kleinen Stichprobe von 124 Befragten keine zu aussagekräftige Beurteilung bezüglich einer Zustimmung treffen kann. Vor allem auch dadurch, dass die Extrema auch noch relativ geringe Werte haben, ganz besonders bei der Zustimmung, wo die Masse mit „trifft eher zu“ geantwortet hat. Stattdessen stupe ich dieses Ergebnis eher als neutral ein, bzw. als minimale Verbesserung des Verständnisses eines Algorithmus. Dass dieser Punkt also durch das Unterrichtskonzept abgedeckt ist, lässt sich anhand dieser quantitativen SchülerInnenaussagen nicht wirklich sagen. Was hierzu das Ergebnis der Experteninterviews sagt und ob es diese geringe Zustimmung stützt, sieht man bei der Auswertung der Interviews.

Frage 1d:

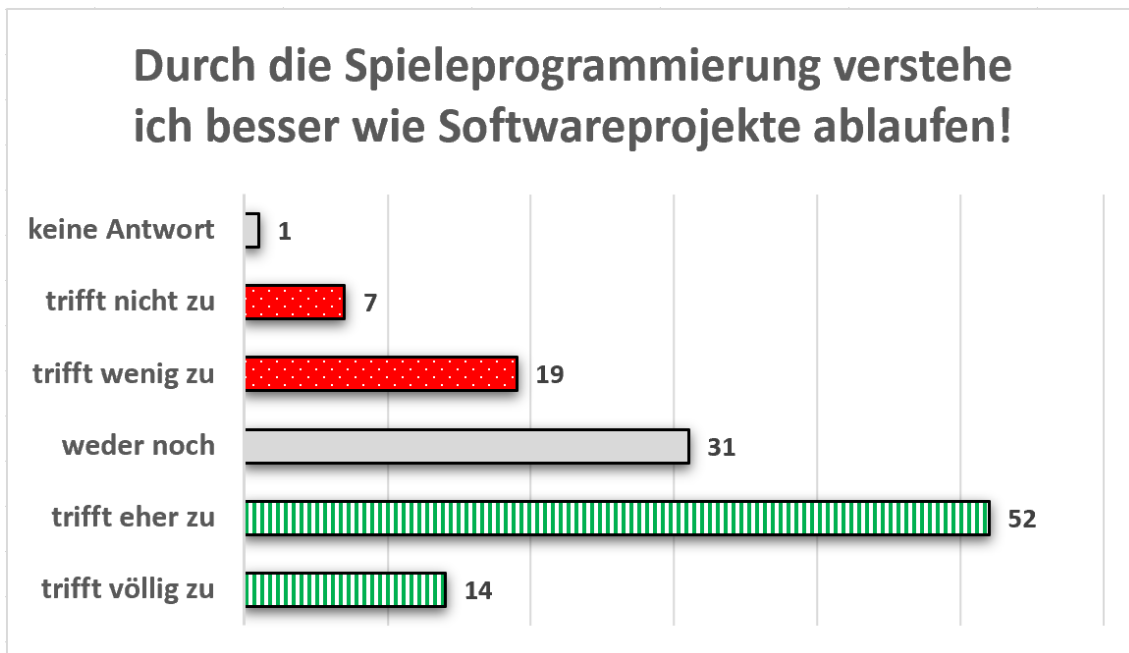


Abbildung 21: Verständnis der Lernenden von Softwareprojekten (N=124) (Eigene Darstellung)

Diese Frage greift den Punkt 4.2.3W ab, der besagt, dass SchülerInnen wesentliche Aspekte und Methoden der Softwareentwicklung und des Softwareprojektmanagements erklären können. Das Diagramm zeigt eine Mehrheit im Bereich der Zustimmung auf, indem sowohl mehr Stimmen für „trifft eher zu“ als für „trifft wenig zu“ und auch mehr für „trifft völlig zu“ als für „trifft nicht zu“ ausfallen. Dadurch liegt auch der Tendenzwert mit  $(14 \cdot 2 + 52 - 19 - 7 \cdot 2 =) 47$  wieder klarer im positiven Bereich. Dass dieser Punkt sehr wahrscheinlich erfüllt ist, ließ sich bereits im Kapitel „Verwendete Tools“, wo auf die Aufgaben eines Spieleentwicklerteams eingegangen wurde und im Kapitel „Aufbau des Unterrichtskonzeptes“, wo von den erstellten Spielen als kleine Softwareprojekte die Rede ist, herauslesen und erahnen.

## Frage 1e:

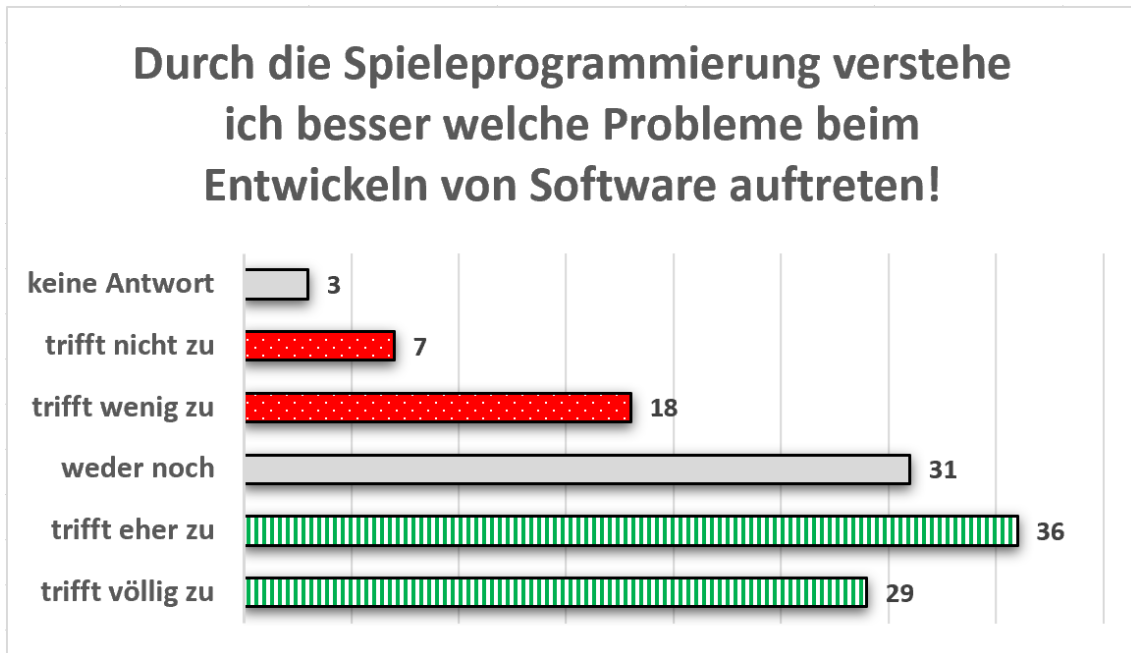


Abbildung 22: Verständnis der Lernenden von Problemen bei der Entwicklung (N=124) (Eigene Darstellung)

Dieses Statement beschreibt das Verständnis von Softwareentwicklungsproblemen. Nimmt man das Wort Verständnis her, befinden wir uns in der niedrigsten Ebene des Kompetenzaufbaues, nämlich beim Wissensbaustein. Trotzdem wurde dem Statement der dpKa-Modell-Punkt 4.2.11R zugeschrieben, der besagt, dass man gezielt nach Programmfehlern suchen und diese korrigieren kann. Denn genau das ermöglicht das Verständnis von Problemen bei der Entwicklung von Software. Damit erreicht dieses Statement die dritte und höchste Ebene des Kompetenzaufbaues, nämlich den Baustein der Handlungsbereitschaft und zeigt, aufgrund der im Diagramm eindeutig ersichtlichen Zustimmung, dass die SchülerInnen durch das Unterrichtskonzept höchstwahrscheinlich eine Kompetenz erwerben, indem sie alle drei erforderlichen Ebenen einer Kompetenz im Bereich „Algorithmen, Datenstrukturen und Programmierung“ erklimmen, von der Wissensebene angefangen, über die Fertigkeitsebene, bis zur Handlungsbereitschaftsebene. Dies sieht man an den Zustimmungen zu den Fragen 1a (Wissen), 1b (Fertigkeit) und hier zu 1e (Handlungsbereitschaft). Der Tendenzwert bekräftigt die Zustimmung durch den klar positiven Wert von 62 ( $= 29 \cdot 2 + 36 - 18 - 7 \cdot 2$ ).

### 6.2.2 Fragebogen Teil 2 (Vergleich von Unterrichtskonzepten)

Nun zum zweiten Teil der Fragen bzw. Fragebogenstatements. Diese sind mit etwas Vorsicht zu genießen, da sie bezüglich potenziell zu erwerbender Kompetenzbausteine einen Vergleich zwischen herkömmlichem Informatikunterricht und dem Unterricht mit Unity abfragen. Also ob der Unterricht im Vergleich zu herkömmlichem Unterricht mehr oder weniger in einem Bereich gebracht hat. Es geht in erster Linie um den Vergleich zwischen unterschiedlichen Unterrichtskonzepten, wodurch man zwar ableiten kann, dass die SchülerInnen etwas erworben haben, es sich aber um keine konkrete Antwort bezüglich eines Erwerbs handelt. Denn auch wenn eine klare Zustimmung herrscht, laut der sie eindeutig viel mehr als durch herkömmlichen Informatikunterricht in einem Bereich erlernt haben, sagt das noch nicht wieviel sie in diesem Bereich insgesamt gelernt haben. Wenn ich beispielsweise zu zwei Nachhilfelehrern unabhängig voneinander gehe, damit sie mir jeder auf seine Art und Weise die Objektorientierung erklären und ich habe von dem einen Lehrer mehr mitgenommen, heißt das noch immer nicht, dass ich die Objektorientierung verstanden habe. Wenn ich nämlich beim einen Lehrer nichts und beim anderen nur gelernt habe, dass es dabei Objekte gibt, dann habe ich zwar verhältnismäßig mehr von einem Lehrer im Bereich der Objektorientierung mitgenommen, weiß aber trotzdem nicht was Objektorientierung bedeutet, auch wenn ich nun ein bisschen mehr davon weiß. Auf der anderen Seite könnte man auch von beiden sehr viel gelernt haben und versteht es nach jedem seiner Erklärung, empfindet aber die eine Erklärung besser. Außerdem weiß man nicht, was die SchülerInnen unter einem herkömmlichen Informatikunterricht verstehen, also welchen sie sonst überhaupt kennen. Das heißt, dass diese Ergebnisse für unsere Verwendung, also der Interpretation von Aneignungen der Kompetenzbausteine, eher eine Wahrscheinlichkeit darstellen, als eine klare Aussage machen, oder besser gesagt einen klaren Hinweis darauf geben.

## Frage 2b:

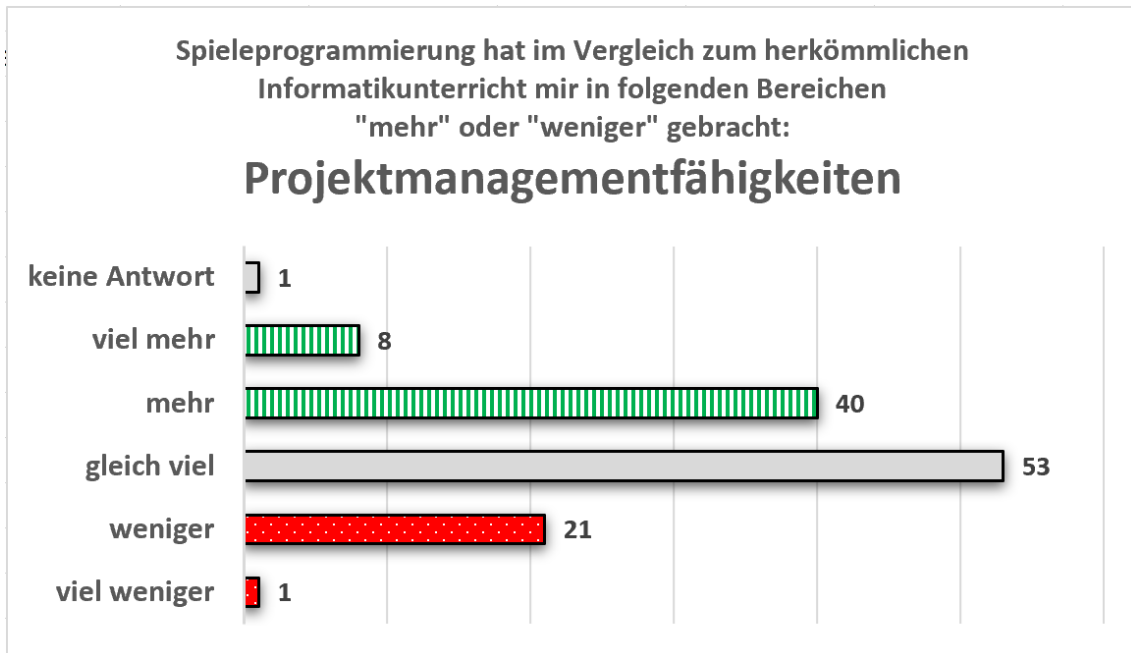


Abbildung 23: evtl. Verbesserung von Projektmanagementfähigkeiten (N=124) (Eigene Darstellung)

Im Bereich der Projektmanagementfähigkeiten hat den SchülerInnen der Unterricht mit Unity mehr als herkömmlicher Unterricht gebracht, da sowohl die starke Zustimmung (viel mehr) als auch die schwache Zustimmung (mehr) ihre jeweiligen Gegenüber offensichtlich übersteigen. Der Tendenzwert ( $8 \cdot 2 + 40 - 21 - 1 \cdot 2$ ) unterstreicht dies mit dem Wert 33. Dieser Frage sind die dpKa-Modell-Punkte 4.2.3W und 4.2.7A zugeordnet. Der Punkt 4.2.3W handelt vom Wissensbaustein, der mir ermöglicht, dass ich wesentliche Aspekte und Methoden der Softwareentwicklung und des Softwareprojektmanagements erklären kann. Da Projektmanagementfähigkeiten neben dem benötigten Wissen auch die Fertigkeitsebene ansprechen, wurde ihnen auch der Punkt 4.2.7A zugeteilt, laut dem man ein Softwareprojekt planen und durchführen kann. Und auch wenn die Betrachtung des Unterrichtskonzeptes in vorigen Kapiteln ebenso darauf schließen lässt, so muss man wie gesagt, dieser Zustimmung mit Vorsicht begegnen.

Frage 2d:

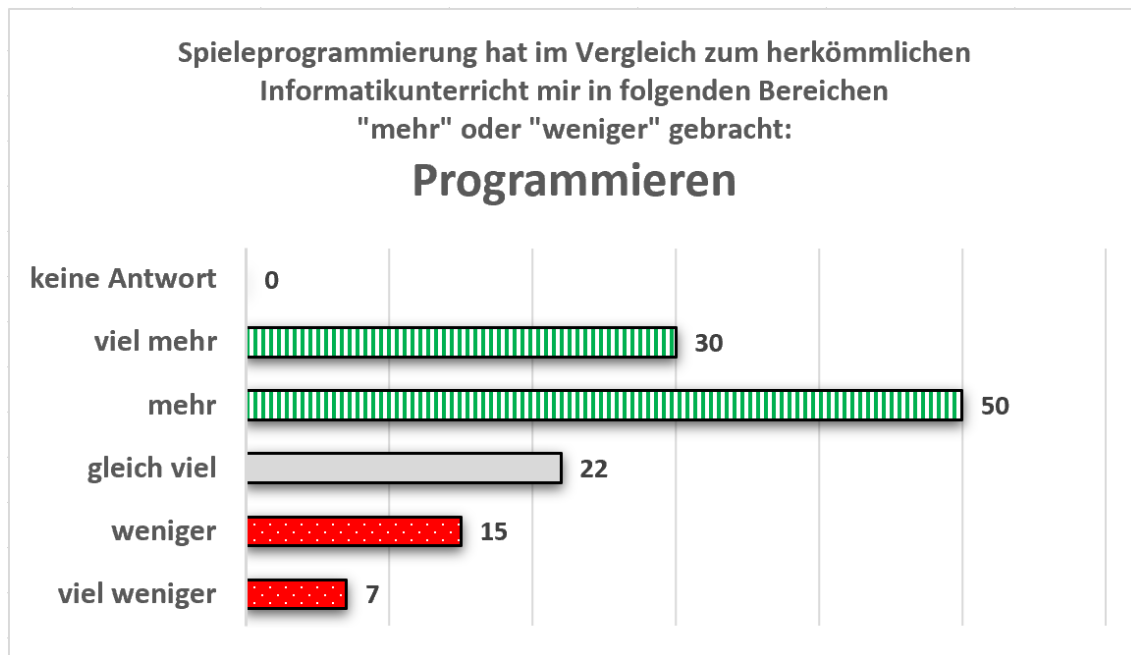


Abbildung 24: evtl. Erwerb bzw. Verbesserung der Programmierfertigkeit (N=124) (Eigene Darstellung)

Auf die hier zu sehende Frage gibt es nicht nur eine eindeutige Antwort, sondern diese Frage wurde in unserem Interpretationssinne bereits mit der Frage 1b beantwortet. Vom Tendenzwert ( $30 \cdot 2 + 50 - 15 - 7 \cdot 2$ ) her, spricht das Ergebnis mit einem Wert von 81 eine klare Sprache. Die Frage beinhaltet die Punkte Dk8 8.1.2A „SchülerInnen verwenden und erstellen Codierungen“, 4.2.6A „Ich kann Algorithmen entwerfen, diese formal darstellen, implementieren und testen“ und 4.2.11R „Ich kann gezielt nach Programmfehlern suchen und diese korrigieren“. Wie gesagt, da diese Frage bereits mit der für uns besser geeigneten Frage 1b beantwortet wurde, gibt es hier nicht mehr zu sagen.



## Frage 2e:

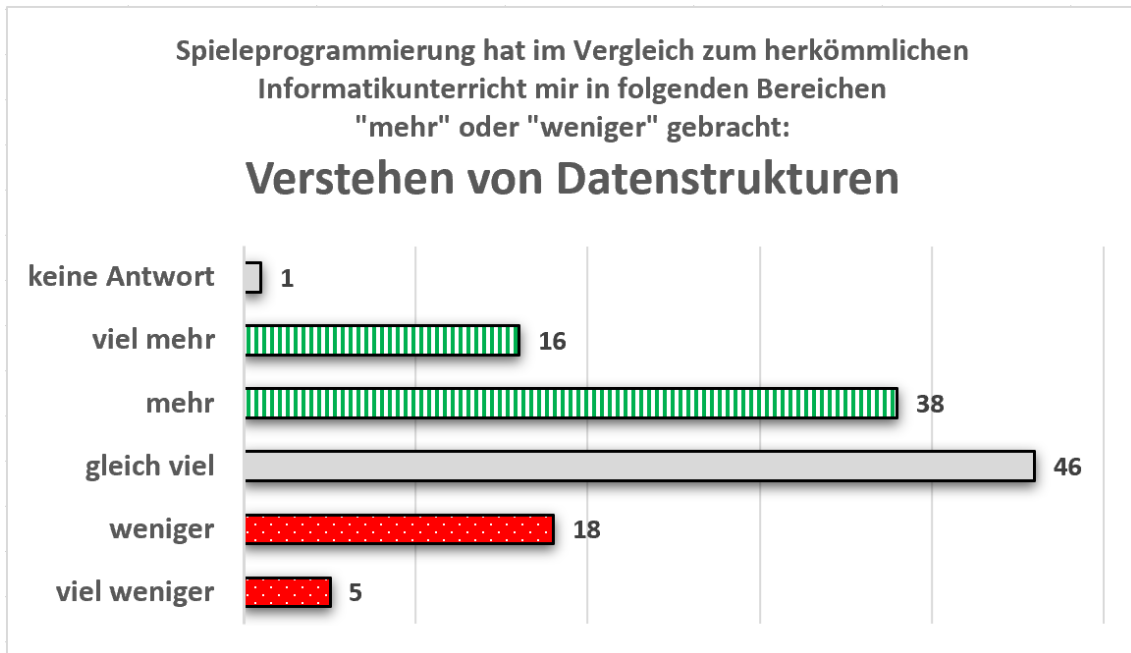


Abbildung 25: evtl. Verständnis von Datenstrukturen (N=124) (Eigene Darstellung)

Hat die Spieleprogrammierung mit Unity mehr für das Verständnis von Datenstrukturen im Vergleich zu herkömmlichen Unterricht gebracht, ist eine Frage die den dpKa-Modell-Punkt KsmOOP1, lautend auf „Ich kann unterschiedliche Datenstrukturen beschreiben“ umfasst. Die Grafik zeichnet ein Bild einer zustimmenden Mehrheit, sodass die beiden Zustimmungswerte ihre jeweiligen Kontrahenten mit jeweils mehr als der doppelten Anzahl überragen. Dementsprechend sieht auch die Tendenz mit dem Wert 42 ( $16 \cdot 2 + 38 - 18 - 5 \cdot 2$ ) aus. Trotz der klaren Zustimmung sei dieses Ergebnis aufgrund der hier forschungsfremden Intention der Frage mit Bedacht zu verwenden.

Frage 2f:

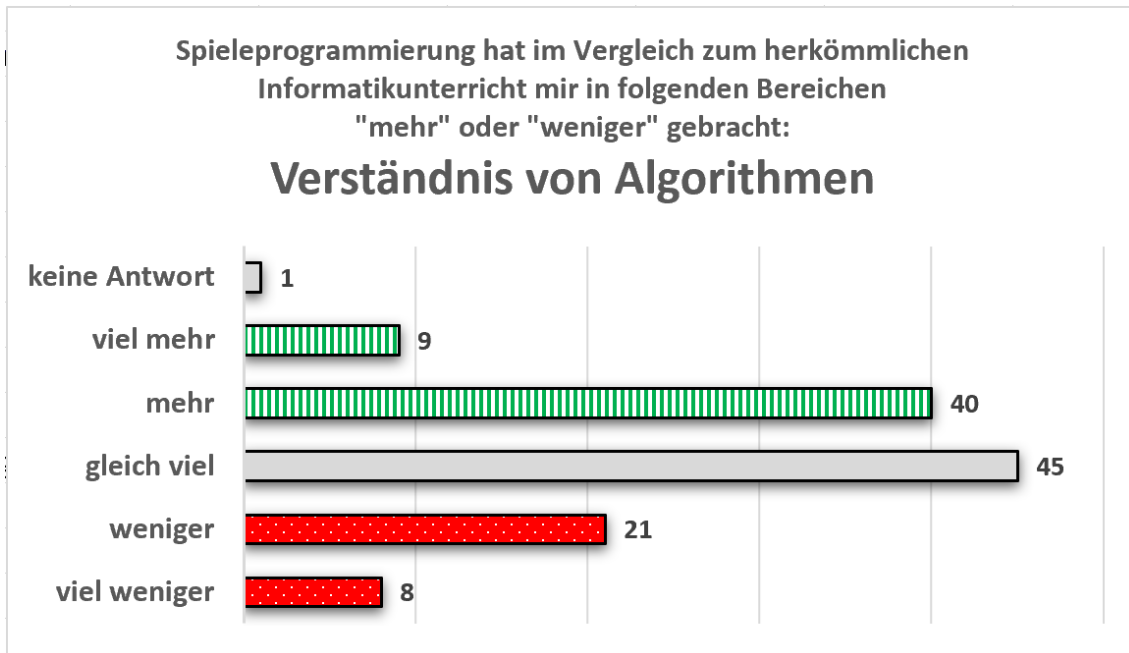


Abbildung 26: evtl. Verständnis von Algorithmen (N=124) (Eigene Darstellung)

Dieses Statement wurde ebenso bereits von einer anderen, deutlich geeigneteren Frage behandelt, nämlich der Frage 1c. Zugeordnet sind ihr die dpKa-Modell-Punkte 4.2.1W und 4.2.2W. Man kann den Algorithmus-begriff erklären besagt Punkt 4.2.1W und dass man Aufgaben und Problemstellungen algorithmisch und formalsprachlich in geeigneten Datenstrukturen beschreiben kann, beschreibt der Punkt 4.2.2W. Wie schon bei der Frage 1c sind die Antworten hier im Vergleich zu anderen Fragen gleichmäßiger verteilt. Doch es gibt wiederum ein kleines Plus auf der Zustimmungsseite und so auch einen positiven Tendenzwert ( $9 \cdot 2 + 40 - 24 - 8 \cdot 2$ ) von 18.

## Frage 2g:

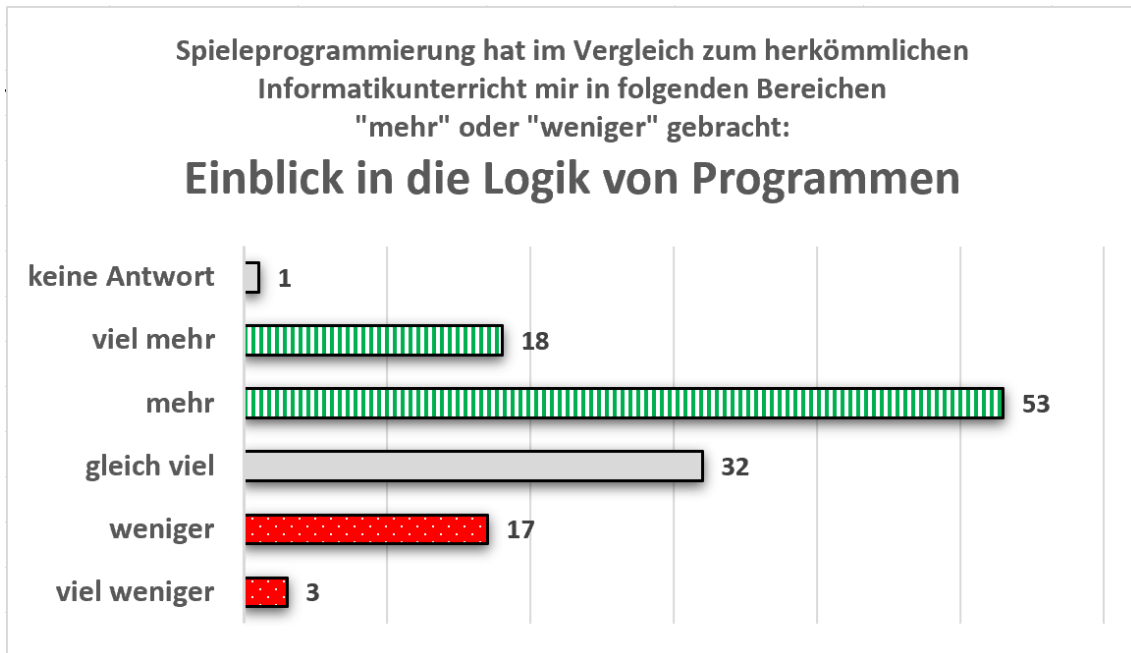


Abbildung 27: evtl. Einblick in die Logik von Programmen (N=124) (Eigene Darstellung)

Bei dieser Frage handelt es sich um einen bisher noch nirgends zu sehenden dpKa-Modell-Punkt Dk8 8.1.6R, laut dem die SchülerInnen die Bedeutung von Algorithmen in automatisierten Prozessen erkennen. Sobald man über einen Einblick in die Logik von Programmen verfügt, hat man auch die Bedeutung von Algorithmen erkannt, denn die Logik eines Programmes beschreibt, was es im Hintergrund tut und somit welchen Handlungsanleitungen es folgt und wenn man das aus einem Programm herauslesen kann, erkennt man gleichzeitig auch die Bedeutung des Algorithmus dahinter. Aus dem Diagramm kann man die sehr klare Zustimmung herauslesen. Die Tendenz  $(18 \cdot 2 + 53 - 17 - 3 \cdot 2)$  nimmt hier den Wert 66 an. Das entspricht zwar einem deutlichen Wert, doch da die Fragestellungen eigentlich etwas anderes erfragt, werten wir das Ergebnis nur als einen kleinen Hinweis darauf, dass der Einblick in die Logik von Programmen durch den Unterricht mit Unity verbessert wird.

Diese Hinweise auf Kompetenzbausteinabdeckungen des Fragebogens fließen, nach den nun folgenden Experteninterviews gemeinsam mit eben diesen zum Schluss in die Beantwortung der Forschungsfragen ein. Einen Überblick dazu gibt schließlich die Vergleichsmatrix im Kapitel „Ergebnisdarstellung“.

### 6.3 Interview (qualitative Forschung)

Dieses Kapitel beinhaltet alles rund um das Interview, also angefangen von der Kategorienbildung und dem Interview inklusive Interviewleitfaden, über die Transkriptionsdurchführung, bis zur Auswertung. Dazu habe ich mich am Dokument „Forschen mit Leitfadeninterviews und qualitativer Inhaltsanalyse“ von Frau Stefanie Vogt und Frau Melanie Werner orientiert (Vogt und Werner 2014), die darin den gesamten Ablauf eines solchen Interviews beschreiben und ihre Entscheidungen bzw. Methodenauswahl sehr gut begründen.

---

#### 6.3.1 Kategorienbildung

Es erfolgte durch die Entwicklung des dpKa-Modells bereits eine deduktive Kategorienbildung. Trotzdem wird zusätzlich noch eine induktive Kategorienbildung in Betracht gezogen, insofern sich das aus der Interviewauswertung ergibt. Den Kategorien werden jetzt noch sprechende Codes bzw. Tags zugeordnet, sodass sich ein klareres Kategoriensystem daraus bildet, wodurch die Codierung der Interviews vereinfacht wird. Die Codes entsprechen einer im Vergleich zu Mayrings bzw. Vogt und Werners Vorlage adaptierten Kategorienbildung, da uns mit dem dpKa-Modell ein Kategoriensystem vorliegt, das auch gleiche Bereiche mit unterschiedlicher Ausprägung, also unterschiedlicher Kompetenzstufe beschreibt. Nehmen wir als Beispiel die Softwareentwicklung her, die sowohl in der Kompetenzstufe „Wissen und Verstehen (Wissen)“, als auch in der Kompetenzstufe „Anwenden und Gestalten (Fertigkeiten)“ existiert. Diese Unterkategorie „Software“, wie sie später genannt wird, gibt es somit zwei Mal, jedoch mit verschiedener Ausprägung. Die Codes setzen sich schließlich aus mehreren Teilen zusammen, nämlich aus einer sprechenden Bezeichnung (Oberkategorie) für die Kompetenzkategorie, der Kennung der Kompetenzstufe (Zwischenkategorie) und einer sprechenden Bezeichnung (Unterkategorie) für die Beschreibung des Kompetenzbausteines (Definition) bzw. der Bausteinnummer. Der Text bzw. die „Kann-Beschreibung“ des Kompetenzbausteines entspricht der „Definition“ in Mayrings Kategoriensystem.

Zum Beispiel deutet beim Code „Prog.W.Algorithmus“ der Teil „Prog“ auf die Kompetenzkategorie (Oberkategorie) „Algorithmen, Datenstrukturen und Programmierung“, der Teil „W“ auf die Kompetenzstufe „Wissen und Verstehen (Wissen)“, der Teil „Algorithmus“ als sprechende Bezeichnung auf die gesamte Beschreibung des Kompetenzbausteines inklusive der dazugehörigen Bausteinnummer „4.2.1“ selbst hin. Zur besseren Übersicht ein Beispiel in einer Tabelle:

	<b>dpKa-Modell</b>	<b>Code / Tag</b>
<b>Kompetenzkategorie (OBERKATEGORIE)</b>	Algorithmen, Datenstrukturen und Programmierung	Prog
<b>Kompetenzstufe (ZWISCHENKATEGORIE)</b>	Wissen und Verstehen (Wissen)	W
<b>Bausteinnummer (UNTERKATEGORIE)</b>	4.2.1	Algorithmus
<b>Beschreibung des Kompetenzbausteines / Bausteintext (DEFINITION)</b>	Ich kann den Algorithmus-begriff erklären.	Ich kann den Algorithmus-begriff erklären.

Die sprechenden Code-Bezeichnungen für die Kompetenzkategorien (Oberkategorien) und die Beschreibungen der Kompetenzbausteine (Unterkategorien) sind so gewählt, dass sie die Kategorie und die Beschreibung möglichst repräsentativ darstellen. Die Kompetenzstufen können außer dem Wert „W“ für „Wissen und Verstehen (Wissen)“ zudem noch die Werte „A“ für „Anwenden und Gestalten (Fertigkeiten)“ und „R“ für „Reflektieren und Bewerten (Handlungsbereitschaft)“ annehmen.

## Dk12 – 1.2 Verantwortung, Datenschutz und Datensicherheit

### Wissen und Verstehen (Wissen):

- **Recht.W.Daten-1.2.1** Ich kann meine Rechte und Pflichten in der Nutzung von Informatiksystemen beschreiben und wesentliche Aspekte des Datenschutzes und der Datensicherheit erklären. [DC 2.5, 3.3, 4.1, 4.2], [Dk8 2.4.2] Grundzüge des Urheberrechts sowie Datenschutzes
- **Recht.W.Lizenz-Dk8 2.4.3** SchülerIn kennt Lizenzmodelle, [DC 3.3]

### Anwenden und Gestalten (Fertigkeiten):

- **Recht.A.Daten-1.2.2** Ich kann beim Einsatz von Informatiksystemen mein Wissen um Pflichten und Rechte in Bezug auf meine Person und meine Arbeitsumgebung, auf persönliche und fremde Daten verantwortungsbewusst anwenden. [DC 2.5, 4.1, 4.2] [Dk8 2.4.2]

## Dk12 – 2.1 Technische Grundlagen und Funktionsweisen

### Wissen und Verstehen (Wissen):

- **Technik.W.Computer-2.1.1** Ich kann Komponenten von Informatiksystemen beschreiben und ihre Funktionsweise und ihr Zusammenwirken erklären. [Dk8 7.1.1]

### Anwenden und Gestalten (Fertigkeiten):

- **Technik.A.Computer-2.1.3** Ich kann ein Computersystem samt Peripheriegeräten sachgerecht nutzen. [DC 0.2, 5.2] [Dk8 3.1.1] Dateimanagement, [Dk8 7.2.3]

### Reflektieren und Bewerten (Handlungsbereitschaft):

- **Technik.R.Computer-2.1.5** Ich kann unterschiedliche digitale Endgeräte bzw. Informatiksysteme in Bezug auf ihre technischen Eigenschaften und ihre Leistungsfähigkeit bewerten. [DC 5.2]
- **Technik.R.Fehler-2.1.6** Ich kann einfache Fehler diagnostizieren und beheben. [DC 5.1] [Dk8 7.3.1] erkennen technischer Probleme

## Dk12 – 2.2 Betriebssysteme und Software

### Wissen und Verstehen (Wissen):

- **Software.W.Arten-2.2.2** Ich kann Kategorien von Software nennen und deren Anwendung beschreiben. [DC 5.2] [Dk8 7.1.2]

### Anwenden und Gestalten (Fertigkeiten):

- **Software.A.Arten-2.2.4** Ich kann mich in die Bedienung für mich neuer Software selbstständig einarbeiten. [DC 0.2]
- **Software.A.Text-Dk8 3.2.1** SchülerIn gibt Texte zügig ein, [DC 0.2]
- **Software.A.Korrektur-Dk8 3.2.3** SchülerInnen führen Textkorrekturen durch (inkl. Überarbeitungsfunktionen, Rechtschreibprüfung). [DC 0.2]
- **Software.A.Hilfe-Dk8 7.3.2** SchülerIn nutzt Hilfesysteme
- **Software.A.Sicherung-Dk8 7.3.3** SchülerInnen führen Datensicherungen und -Wiederherstellungen aus.

### Dk12 – 3.1 Produktion digitaler Medien

#### Wissen und Verstehen (Wissen):

- **Medien.W.Formate-3.1.1** Ich kann gängige Medienformate und ihre Eigenschaften beschreiben. [DC 3.1]

#### Anwenden und Gestalten (Fertigkeiten):

- **Medien.A.Formate-3.1.3** Ich kann digitale Medien in Form von Text, Ton, Bildern und Filmen sachgerecht bearbeiten, produzieren & publizieren. [DC 3.1], [Dk8 4.2.2]

#### Reflektieren und Bewerten (Handlungsbereitschaft):

- **Medien.R.Formate-3.1.4** Ich kann digitale Produkte (Artefakte) in Bezug auf inhaltliche Relevanz, Wirkung und Design einschätzen und bewerten.
- **Medien.R.Selbstwirksamkeit-Dk8 4.2.1** SchülerInnen erleben sich selbstwirksam, indem sie digitale Technologien kreativ & vielfältig nutzen
- **Medien.R.Techniken-Dk8 4.2.5** SchülerInnen setzen Wissen über Techniken und Ästhetiken populärer Medienkulturen eigenverantwortlich um, [DC 3.1]

### Dk12 – 3.3 Suche, Auswahl und Organisation von Information

#### Wissen und Verstehen (Wissen):

- **Info.W.SucheQuellen-3.3.1** Ich kann wichtige Informationsquellen im Internet anführen, die für meine schulischen und privaten Informationsbedürfnisse nützlich und notwendig sind. [DC 5.2, 5.3]
- **Info.W.SucheMethoden-3.3.2** Ich kann lokal und in Netzwerken Methoden der Informationsgewinnung und -organisation benennen. [DC 1.1, 5.3]
- **Info.W.Wissensmanagement-3.3.3** Ich kann Möglichkeiten grundlegenden digitalen Wissensmanagements beschreiben.

#### Anwenden und Gestalten (Fertigkeiten):

- **Info.A.SucheBedarf-Dk8 2.1.1** SchülerInnen formulieren ihre Bedürfnisse für die Informationssuche, [DC 1.1]
- **Info.A.Suche-3.3.4** Ich kann unter Verwendung passender Dienste und Angebote und Wahl geeigneter Suchmethoden Informationen und Medien gezielt suchen und auswählen. [DC 1.1, 5.3], [Dk8 2.1.2]
- **Info.A.Wissensmanagement-3.3.5** Ich kann im Rahmen persönlichen Lernmanagements Informationen und Medien strukturiert speichern und verfügbar halten., [DC 1.3], [Dk8 2.3.1]

## Dk12 – 4.2 Algorithmen, Datenstrukturen und Programmierung

### Wissen und Verstehen (Wissen):

- **Prog.W.Datenstruktur-[KsmOOP 1]** Ich kann unterschiedliche Datenstrukturen beschreiben **5.1 data structure (graph, tree, array)**
- **Prog.W.Algorithmus-4.2.1** Ich kann den Algorithmus-begriff erklären.
- **Prog.W.Modellierung-4.2.2** Ich kann Aufgaben und Problemstellungen algorithmisch und formalsprachlich in geeigneten Datenstrukturen beschreiben. **[DC 3.4], [Dk8 8.1.4], 5.3 algorithmic structure (loops, conditional statement)**
- **Prog.W.Software-4.2.3** Ich kann wesentliche Aspekte und Methoden der Softwareentwicklung und des Softwareprojektmanagements erklären. **5.4 notional machine (data, working memory, processor, statement)**
- **Prog.W.Aspekte-4.2.4** Ich kann wesentliche Aspekte der *Prozeduralen, Funktionalen und Objektorientierten* [sic!] Programmierung nennen und an Beispielen erläutern. **5.2 class & object structure (object, attribute, association)**
- **Prog.W.Sprache-Dk8 8.2.2** SchülerInnen kennen unterschiedliche Programmiersprachen und Produktionsabläufe. **6 Mastering representation (language, syntax, semantics)**

### Anwenden und Gestalten (Fertigkeiten):

- **Prog.A.Sprache-Dk8 8.1.2** SchülerInnen verwenden, erstellen [und reflektieren] Codierungen, **[DC 0.1] 6 Mastering representation**
- **Prog.A.Modellierung-4.2.5** Ich kann Aufgaben mit Mitteln der Informatik modellieren.
- **Prog.A.Algorithmus-4.2.6** Ich kann Algorithmen entwerfen, diese formal darstellen, implementieren und testen. **[DC 3.4], [Dk8 8.1.4], [Dk8 8.2.3], 5.3 algorithmic structure**
- **Prog.A.Datenstruktur-[KsmOOP 2]** Ich kann Datenstrukturen, sowie Klassen- und Objektstrukturen in einem Algorithmus verwenden **5.1 data structure 5.2 class & object structure**
- **Prog.A.Software-4.2.7** Ich kann ein Softwareprojekt planen und durchführen. **[Dk8 8.2.1], [DC 5.3] 5.4 notional machine**

### Reflektieren und Bewerten (Handlungsbereitschaft):

- **Prog.R.Software-4.2.8** Schritte der Softwareentwicklung reflektieren.
- **Prog.R.SoftwareWerkzeug-4.2.9** Ich kann die Angemessenheit der Entwicklungswerkzeuge grob einschätzen.
- **Prog.R.AlgorithmusMuster-Dk8 8.1.5** SchülerInnen entdecken Regeln und Muster in Handlungsanleitungen, **[DC 0.1]**
- **Prog.R.AlgorithmusBedeutung-Dk8 8.1.6** SchülerInnen erkennen die Bedeutung von Algorithmen in automatisierten digitalen Prozessen (z. B. automatisiertes Vorschlagen von potenziell interessanten Informationen).
- **Prog.R.AlgorithmusEffizienz-4.2.10** Ich kann die Effizienz von Algorithmen bewerten.
- **Prog.R.SpracheAlgorithmus-4.2.11** Ich kann gezielt nach Programmfehlern suchen und diese korrigieren. **[DC 5.1]**



Bei dieser Codeentwicklung wurde darauf geachtet, dass Punkte, die denselben Bereich ansprechen, sich aber in unterschiedlichen Kompetenzstufen befinden, außer dem Kompetenzstufenwert (W/A/R) denselben Code bekommen. Dadurch sieht man nun grob welche Bereiche in allen drei Kompetenzstufen abgebildet sind und dass der Erwerb dieser gleich codierten Kompetenzbausteine den vollständigen Erwerb einer Kompetenz beschreibt. Manche Punkte teilen einen Bereich in mehrere Unterbereiche auf, wie man bei den Punkten „Prog.R.AlgorithmusMuster“, „Prog.R.AlgorithmusBedeutung“ und „Prog.R.AlgorithmusEffizienz“ sieht, oder sie stellen Aspekte eines Bereiches zusätzlich noch extra dar, so wie sich das anhand der Punkte „Prog.R.Software“ und „Prog.R.SoftwareWerkzeug“ zeigen lässt. Denn wenn ich über die Schritte der Softwareentwicklung reflektiere (Prog.R.Software), so kann ein Teil davon sein, dass ich darüber reflektiere, ob ich die richtigen Entwicklungswerkzeuge eingesetzt habe und könnte daher auch fähig sein, die Angemessenheit von Entwicklungswerkzeugen einschätzen zu können. Immerhin stellen alle diese Punkte einen ganzen Bereich (Unterkategorien / grobe Definitionen) dar und sind nicht auf der kleinsten Mikroebene beschrieben. Es gibt auch Punkte wie „Software.A.Text“, die hier nur auf der zweiten Kompetenzstufe „Anwenden und Gestalten (Fertigkeit)“ existieren, da der dazugehörige Wissensbaustein zu trivial ist, der nämlich wäre, dass man für eine zügige Texteingabe wohl nur über die Handhabung der Tastatur (Zehnfingersystem) und über die Tastenbelegung der Tastatur Bescheid wissen muss. Aufgrund dieser Trivialität dieses Wissensbausteines ist er höchstwahrscheinlich auch in den Kompetenzmodellen digi.komp4 bis digi.komp12 nicht abgebildet. In der Kompetenzkategorie „Algorithmen, Datenstrukturen und Programmierung“ gibt es jedenfalls keine Unterkategorien, die in der jeweils vorhergehenden Kompetenzstufe (Zwischenkategorie) nicht abgebildet sind.

Es wird sich bei den Interviews jedenfalls zeigen, ob eine Erweiterung durch eine induktive Kategorienbildung nötig ist und dann noch Kompetenzbausteine zum dpKa-Modell hinzugefügt werden müssen und dadurch sogar auf eine eventuelle Unvollständigkeit eines Kompetenzmodelles hinweisen könnten.

### 6.3.2 Interviewleitfaden

Der Interviewleitfaden besteht aus zwei **Clustern** mit Fragen, wobei jeder Cluster den Schwerpunkt auf eine der beiden Forschungsfragen legt. Es wurde ein Dutzend Erzählimpulse verwendet, da es sich bei den meisten Fragen um konkrete Ja/Nein-Fragen handelt. Erst durch das Nachhaken (Steuerungsfragen) nach Beispielen, woran man die Expertenantwort erkennen kann, entsteht der wahre Erzählimpuls, der aber im Verhältnis zu allgemeineren Fragen sicherlich schwächer ausfällt, da nach konkreten Bausteinen gefragt wird. Vor allem im ersten Cluster, wo es um die erworbenen Kompetenzen im Bereich „Algorithmen, Datenstrukturen und Programmierung“ geht, wird sehr gezielt nach den einzelnen Bausteinen gefragt, wohingegen der zweite Cluster deutlich allgemeiner ausfällt, indem nach eventuellen Wissens- und Fertigungs- und Handlungsbereitschaftsbausteinen aus ganzen Kompetenzbereichen gefragt wird. Es sind auch einige noch allgemeinere Fragen enthalten, um das Interview nicht zu sehr auf die Kompetenzbausteine zu fokussieren, sodass mehr aus dem Alltag berichtet wird. Beim zweiten Cluster wird zum Beispiel nach den Schwierigkeiten der SchülerInnen bei der Spieleentwicklung mit Unity gefragt. Vor allem hier kommen auch mehrere unterschiedliche Fragetypen zum Einsatz, die die Perspektive der Befragten darauf verändern sollen.

Den Fragen wurde jeweils in eckiger Klammer ein Tag zugewiesen, der aus einer fortlaufenden Nummer, einem „F“ als Kennzeichnung für eine Frage (/Erzählimpuls) und einem für die Frage sprechenden Kürzel besteht. Beispiel: **[01F-Erfahrung]**. Diese Tags werden später, neben den dpKa-Modell-Tags/Codes, für die Codierung der Interviews verwendet. Außerdem wurde den Fragen jeweils der abgefragte dpKa-Modell-Punkt in runder Klammer hinzugefügt. Beispiel: **(4.2.1W)**. Gesprochenes ist *kursiv* und alles andere **fett** geschrieben.

Der Interviewleitfaden:

---

*Ich bedanke mich und es freut mich, dass Sie an diesem Interview als Experte teilnehmen.*

*Ihre Antworten sind freiwillig und Sie können das Interview jederzeit abbrechen. Bezüglich der Nutzungsrechte habe ich eine Vereinbarung mitgebracht, mit deren Unterschrift Sie mit der Aufzeichnung des Interviews einverstanden sind, Sie mir die Nutzungsrechte für die dadurch entstehenden Dokumente überlassen und der Verwendung für ausschließlich wissenschaftliche Zwecke in anonymisierter Form zustimmen.*

*Dieses Experteninterview ist Teil meiner Diplomarbeit. Es untersucht Wissen, Fertigkeiten und Kompetenzen rund um die Spieleentwicklung mit Unity. Mehr will ich an dieser Stelle nicht sagen, da Sie unvoreingenommen auf die Fragen antworten sollen. Gerne erzähle ich Ihnen aber nach dem Interview mehr darüber. Da Sie das Unterrichtsprojekt „learn to proGrAME“ mit mehr als einer Ihrer Klassen durchgeführt haben, erlaube ich mir, Sie als Experte für mein Interview heranzuziehen. Wenn Sie bereit sind, beginne ich nun mit der Aufnahme. Sind Sie bereit?*

**[01F-Erfahrung] - Erzählaufforderung (Ice breaker):** *Beschreiben Sie bitte ihre Erfahrungen mit dem Unterrichtsprojekt „learn to proGrAME“.*

### **1. CLUSTER**

**[02F-guterProgrammierer] - Erzählimpuls 1:** *Wodurch zeichnet sich Ihrer Meinung nach ein guter Programmierer aus?*

**Steuerungs- und Aufrechterhaltungsfragen (in weiterer Folge durch „S:“ gekennzeichnet):**

- **[02F-guterProgrammierer] - S:** *Wo sieht man das in Bezug auf digitale Kompetenzen?*

- **[03F-BeitragUnity]** - **S:** *Wo kann dabei die Spieleentwicklung mit Unity einen Beitrag leisten?*
- **[04F-MusterschülerJob]** - **S:** *zirkuläre F.:* *Denken Sie an eine/n MusterschülerIn im Projekt. Was denken Sie, nahm die-/derjenige für einen etwaigen Job als ProgrammiererIn mit?*
- **[05F-NutzenUnity]** - **S:** *Was zeichnet die Spieleentwicklung mit Unity in Bezug auf die Programmierung aus, bzw. wo sehen Sie den größten Nutzen?*

**[06F-VerständnisAlgo]** - **Erzählimpuls 2 (4.2.1W):** *Hat die Spieleentwicklung mit Unity einen Einfluss auf das Verständnis der SchülerInnen eines Algorithmus? S:* *Wenn ja, welchen Einfluss? Waren Änderungen feststellbar? Wenn ja, wodurch?*

**[07F-Aspekte]** - **Erzählimpuls 3 (4.2.4W):** *Können die SchülerInnen nach der Spieleentwicklung mit Unity wesentliche Aspekte der Programmierung (z.B. Funktionen, Objektorientierung) nennen und erläutern? S:* *Wo sieht man das?*

**[08F-Produktion]** - **Erzählimpuls 4 (Dk8 8.2.2W):** *Lernen die SchülerInnen im Verlauf der Spieleentwicklung mit Unity unterschiedliche Programmiersprachen und Produktionsabläufe kennen? S:* *Wenn ja, welche und womit?*

**[09F-Modell]** - **Erzählimpuls 5 (4.2.5A):** *Erwerben die SchülerInnen die Fertigkeit, Aufgaben mit Mitteln der Informatik zu modellieren (Datenstrukturen, Struktogramm, ERM)? S:* *Können Sie vielleicht ein Beispiel nennen, wodurch das der Fall ist? S:* *Sehen Sie das als eine Voraussetzung für die Spieleentwicklung mit Unity oder als dadurch erworben?*

**[10F-DaKIOb]** - **Erzählimpuls 6 (KsmOOP 2A):** *Können die SchülerInnen nach der Spieleentwicklung mit Unity Daten-, Klassen- und Objektstrukturen in einem Algorithmus verwenden? S:* *Wenn ja, mit welcher Übung erlernen sie das?*

**[11F-SoftwareProj]** - Erzählimpuls 7 (4.2.7A, 4.2.8R): *Sind die SchülerInnen nach der Spieleentwicklung mit Unity fähig, ein Softwareprojekt zu planen und durchzuführen, oder sogar zu reflektieren? S: Wenn ja, warum?*

**[12F-EntwicklungsW]** - Erzählimpuls 8 (4.2.9R): *Lernen die SchülerInnen die Angemessenheit von Entwicklungswerkzeugen grob einzuschätzen? S: Wodurch?*

**[13F-AlgoMuster]** - Erzählimpuls 9 (Dk8 8.1.5R): *Entdecken SchülerInnen Gemeinsamkeiten und Regeln (Muster) in Handlungsanleitungen bzw. Algorithmen? S: Wo ist Ihnen das aufgefallen?*

**[14F-AlgoBeEff]** - Erzählimpuls 10 (Dk8 8.1.6R, 4.2.10R): *Können die SchülerInnen durch die Spieleentwicklung mit Unity die Bedeutung von Algorithmen erkennen? S: Woran? S: Können sie auch die Effizienz von Algorithmen bewerten? S: Können Sie ein Beispiel nennen, womit sie dazu fähig werden?*

## **2. CLUSTER:**

**[15F-Schwierigkeiten]** - Erzählimpuls 11: *Bei SchülerInnen treten bei einem neuem Lernstoff immer wieder mal Schwierigkeiten auf. Erinnern Sie sich an solche Situationen. Was waren die Schwierigkeiten?*

### **Steuerungs- und Aufrechterhaltungsfragen:**

- **[16F-schlimmer]** - **S: paradoxe Frage:** *Was müsste den SchülerInnen fehlen damit die Situation noch schlimmer wird?*
- **[17F-Bewältigung]** - **S:** *Was würde man zur Bewältigung dieser Schwierigkeiten benötigen?*
- **[18F-SchülerResilienz]** - **S: Ausnahmefrage:** *Was hatten die SchülerInnen, die nicht auf diese Schwierigkeiten gestoßen sind, an sich?*
- **[19F-optimalesRüstzeug]** - **S: Wunderfrage:** *Angenommen, morgen beginnt ein neues Schuljahr und ihre SchülerInnen sind bestens*

*gerüstet, um direkt mit der Spieleprogrammierung in Unity zu beginnen.  
Woran erkennen Sie dieses optimale Rüstzeug Ihrer SchülerInnen?*

- **[20F-unendlichZeit] - S: hypothetische Frage:** *Was wäre, wenn Sie die Zeit für Sie und ihre jeweilige Klasse anhalten könnten? Was würden Sie dabei den SchülerInnen vor der Spieleentwicklung mit Unity beibringen wollen?*

**Erzählimpuls 12:** *Benötigen SchülerInnen Ihrer Ansicht nach für die Spieleentwicklung mit Unity Kenntnisse und Fertigkeiten aus dem Bereich:*

- **[21F-Datenschutz]**  
**S:** *Verantwortung, Datenschutz und Datensicherheit (Bsp.: Lizenz)?*
  - **S:** *Wenn ja, welche?*
- **[22F-Technik]**  
**S:** *Technische Grundlagen und Funktionsweisen (Bsp.: Peripherieg.)?*
  - **S:** *Wenn ja, welche?*
- **[23F-Software]**  
**S:** *Betriebssysteme und Software (Bsp.: Dateimanag., Sicherung)?*
  - **S:** *Wenn ja, welche?*
- **[24F-Medien]**  
**S:** *Produktion digitaler Medien (Bsp.: Text, Bild, Ton; Bearbeitung)?*
  - **S:** *Wenn ja, welche?*
- **[25F-Info]**  
**S:** *Suche, Auswahl und Organisation von Information (Internetsuche)?*
  - **S:** *Wenn ja, welche?*

**[26F-Abschluss] - Abschlussfrage:** *Wir sind jetzt am Ende des Interviews angelangt. Gibt es noch etwas, was Sie ergänzen oder noch loswerden möchten?*

*Ich bedanke mich für das Interview.*

---

### 6.3.3 Interview- und Transkriptionsdurchführung

Für das Interview wurden drei Experten herangezogen, die alle das Unterrichtsprojekt „learn to proGrAME“ bereits mehrmals mit ihren Klassen durchgeführt haben. Bei allen dreien handelt es sich um Informatiklehrer aus Gymnasien, deren Namen aufgrund der Anonymisierung jedoch nicht genannt werden.

Für diese Studie sind nur die Inhalte und keine Gestiken und Mimik oder Sonstiges von Bedeutung, da die Notwendigkeit und der Erwerb von Kompetenzbausteinen für die Spieleentwicklung mit Unity anhand von konkreten Beispielen aus dem Alltag herausgehört werden soll. Deshalb wurden auch zwei der drei Interviews per Skype von zu Hause aus durchgeführt. Das dritte Interview fand in einem ruhigen, kleinen Besprechungszimmer eines Gymnasiums statt. Dadurch wurde in allen drei Interviews sichergestellt, dass sie möglichst ungestört und ohne große Hintergrundgeräusche stattfinden konnten. Die Interviews sind, zumindest aus meiner Sicht sehr entspannt und angenehm verlaufen.

Alle drei Interviews wurden per Diktiergerätfunktion eines Smartphones aufgenommen und mit dem Programm „MAXQDA“ von der Firma „Verbi Software“ transkribiert. Es wurde das von Kuckartz, Dresing, Rädiker und Stefer aufgezeigte Transkriptionssystem verwendet, laut dem wie folgt transkribiert wird (vgl. Kuckartz et al. 2008, S. 27):

- Wörtliche Transkription, keine lautsprachliche oder zusammenfassende
- Sprache wird geglättet, also an das Schriftdeutsch angepasst
- Angaben die eine Personenidentifikation zulassen werden anonymisiert
- Längere Pausen werden durch Auslassungspunkte „...“ gekennzeichnet
- Bestätigende Lautäußerungen (Mhm, Aha, ...) werden nicht mittranskribiert, außer sie unterbrechen den Redefluss
- Lautäußerungen, die eine Aussage unterstützen oder verdeutlichen werden in Klammern notiert

- Die interviewende Person wird durch ein „I“ und die befragte Person durch ein „B“ gekennzeichnet
- Ein Sprecherwechsel wird durch eine Leerzeile zwischen den Sprechern kenntlich gemacht, um die Lesbarkeit zu erhöhen

Die Transkriptionen befinden sich im Anhang und wurden innerhalb von Word mit Hilfe der Kommentarfunktion mit den Tags des Interviewleitfadens und den Codes des dpKa-Modells codiert. Anschließend wurden die Kommentare extrahiert und in Excel paraphrasiert.

### 6.3.4 Auswertung

Die Auswertung erfolgte durch die qualitative Inhaltsanalyse laut Mayring (2002). Die codierten/getaggten Textstellen der Interviewantworten wurden in Word per „DocTools“-Plugin „ExtractData“, das zum Extrahieren von Kommentaren dient, extrahiert und in Excel als Tabelle eingefügt. Dabei wurden einige Spalten dazugefügt, um unterschiedlichste Sortierungen vornehmen zu können. Die folgende Abbildung zeigt einen Ausschnitt aus der Excel-Tabelle:

Nr.	Comment scope	Zusammenfassung der Textstellen bzw. Paraphrasierung	Question	Tag	Memo	Voraussetzung
162:43	1 Wenn ich jetzt irgendwie eine Oberfläche haben möchte, die im Web erscheint, dann werde ich wohl kaum in Unity ein Projekt machen und das Ganze dann ins Web exportieren, sondern da reicht ein schön gestiltes HTML5 und das sollte dann den Zweck erfüllen, wenn es einfach nur um die Oberfläche geht, um eine einfache Interaktion abzubilden	Für eine Oberfläche mit Interaktion im Web reicht HTML5, anstatt Unity zu verwenden	12F-Entwicklungsw	6.13	Prog.R.SoftwareWerkzeug	
162:44	1 Oder Angemessenheit von Entwicklungswerkzeugen wäre auch dann gegeben, wenn man sich überlegt, jetzt möchte ich an der Grafik eine Änderung haben, mache ich das jetzt in Unity oder mache ich das in der Bildbearbeitung. Zum Beispiel möchte ich von meinen Spielfiguren Dinge anders darstellen, beim Elefanten der Rüssel, dann schätzen die Schülerinnen und Schüler in den meisten Fällen ein, naja, das in Unity zu transformieren ist vielleicht nicht so angemessen, wie einfach nochmal in die Bildbearbeitung zu gehen ins Gimp und dann das zu verändern und dann wieder in Unity weiterzuarbeiten. Wohingegen, wenn es darum	BEISPIEL: Schüler schätzen in den meisten Fällen ein, dass die Transformation des Rüssels eines Elefanten in Gimp einfacher ist als in Unity, wohingegen für die Änderung der Farbe eines Elefanten innerhalb von Unity der Professor gefragt wurde	12F-Entwicklungsw	6.13	Prog.R.SoftwareWerkzeug	E+++
163:45	1 Wenn zum Beispiel die Objekte herunterfallen, diese Game-Objects, da geht es ja darum, da sind Elefanten und irgendwann lässt der verrückte Dr. Clamberwoods den Müll regnen und da kommen dann solche Aussagen: "Ah, das ist ja eh in Wirklichkeit wieder wie der Elefant, nur halt kann ich das nicht steuern". Das wäre ein Beispiel. Also sie entdeckten Gemeinsamkeiten und Regeln und erkennen durchaus gewisse Muster eigentlich	BEISPIEL: Bei Bouncy Fant erkennen sie Game-Objects, also dass die Elefanten genauso wie der herunterfallende Müll sind, nur dass man den Müll nicht steuern kann (Aussage eines Schülers)	13-13F-AlgoMuster	6.14	Prog.R.AlgorithmusMuster	E+++
163:46	1 Und in der State-Machine habe ich vorher schon erwähnt, dass sich solche algorithmischen Repräsentanzen herausbilden, die eben nicht notwendigerweise genau das was implementiert worden ist abbilden, aber die sie halt dazu einsetzen, um fast so ein bisschen nach dem Schnittstellen-Prinzip oder Blackbox-Prinzip, um abzuschätzen was jetzt passiert, wenn sie das manipulieren oder wenn sie dieses oder jenes wollen	Schüler entdecken die Regeln von Algorithmen auch anhand der State-Machine, wobei sie immer wieder Änderungen machen und diese austesten	13F-13F-AlgoMuster	6.14	Prog.R.AlgorithmusMuster	E+
163:47	1 Die Bedeutung auf jeden Fall. Und es gibt ja Bedeutungen darüber, sozusagen im lockeren Sinn, einfach was mein Algorithmus macht für mein Softwareprojekt oder überhaupt für die Informatik	Schüler erkennen die Bedeutung von Algorithmen	14F-14F-AlgoBeff	6.15	Prog.R.AlgorithmusBedeutung	E++
164:48	1 Also wenn sie ein Stück Code sehen und genau sagen können, das ist die Bedeutung für mich im engeren Sinne - was bedeutet dieser Code, oder was macht dieser Code. Zum Beispiel haben wir versucht das auch abzufragen in so einem IT-Skills-Test, wo einfach Lernende eine Funktion analysieren sollen und dann nur die Ausgabe dieser Funktion als Antwort geben sollen. Wenn man das eben nicht verstanden hat und die Bedeutung nicht entschlüsseln kann, dann ist das ein Schuss ins Blaue und wenn man das aber verstanden hat und die Bedeutung kennt, dann ist das eigentlich recht einfach da die Lösungen anzugeben	BEISPIEL: Die Bedeutung von Algorithmen wurde auch per IT-Skills-Test abgefragt, wobei Lernende eine Funktion analysieren sollen und die Ausgabe der Funktion als Antwort geben sollen.	14F-14F-AlgoBeff	6.15	Prog.R.AlgorithmusBedeutung	E+++
164:49	1 Ja, ja durchaus. Da gibt es diese eine Übung wo man wirklich den Boden mittels einer Schleife, mittels eines einfachen Algorithmus erzeugen und zuflastern soll und da habe ich die Anmerkung erhalten: "Das ist ja gar nicht effizient, Herr Professor, wie wäre es wenn man das einfach sozusagen das Ganze einmal unten hinzieht und dupliziert, das man diese Tiles wiederholt und dann einen gemeinsamen Collider macht." Bei dem Algorithmus wird sozusagen anhand eines Prefabs, anhand einer Vorlage wirklich der Boden wie beim Fließen legen gelegt und diese einzelnen Boden-Tiles, die haben auch die Colliders und da hat ein Schüler	BEISPIEL: Schüler sagt zur Erstellung des Bodens im Bouncy Fant Tutorial: "Das ist überhaupt nicht effizient, wie wäre es wenn man das Ganze einmal unten hinzieht, dupliziert und dann einen gemeinsamen Collider macht."	14F-14F-AlgoBeff	6.16	Prog.R.AlgorithmusEffizienz	E+++

Abbildung 28: Screenshot aus der Excel-Tabelle, in der die Paraphrasierung und Gewichtung stattgefunden hat. (Eigene Darstellung)



Die ersten drei Spalten beinhalten die Seite der Textstelle, eine fortlaufende Nummer für die Textstellen und die Nummer der befragten Person (gelbe Spalte in der Abbildung). Die vierte Spalte „Comment scope“ beinhaltet die wortwörtlich übernommenen Textstellen der Interviewantworten. In der fünften Spalte „Zusammenfassung der Textstellen bzw. ...“ wurden die Textstellen zusammengefasst und paraphrasiert und das so vorsichtig wie möglich, sodass keine Inhalte verändert wurden. Um das sicherzustellen entsprechen manche Zusammenfassungen exakt oder fast exakt der Textstelle, da ich nicht immer eine inhaltsgetreue Paraphrasierung gefunden habe. Die Spalten sechs und sieben sind für die Fragenummer (für die Sortierung; in der Abbildung in orange) und den Tag der jeweiligen Frage. Die Nummer für den jeweiligen dpKa-Modell-Punkt (in der Abbildung violett) und der Code/Tag des Modell-Punktes befinden sich in Spalte acht und neun. Bei der zehnten Spalte handelt es sich um eine Memo-Spalte, wo ich mir Notizen hinzugefügt habe. Die letzte Spalte „Voraussetzung / Erwerb“ beinhaltet eine Zuordnung zu den Forschungsfragen in Form eines „V“ für die Voraussetzungen und/oder eines „E“ für den Erwerb von Kompetenzbausteinen. Außerdem wurde dieser Zuordnung eine Gewichtung in Form von einem bis drei Plus oder einem bis drei Minus hinzugefügt. Wie diese Gewichtung zu verstehen ist, zeigt folgendes Beispiel:

[Erwerb/Voraussetzung] [Gewichtung]	Bedeutung:
E +	Hinweis der für einen Erwerb spricht
E ++	Klare Aussage die für einen Erwerb spricht
E +++	Ein Beispiel, wo man auch gesehen hat, dass der Baustein erworben wurde
V +	Hinweis der für eine Voraussetzung spricht
E -	Hinweis der gegen einen Erwerb spricht
... (für alle restlichen Kombinationen)	(V/E mit Hinweis/Aussage/Beispiel pro/contra)

Nun wird auf die für diese Studie interessanten Textstellen eingegangen. Die Befragten Personen werden mit den Bezeichnungen B1, B2 und B3 gekennzeichnet. Für die Darstellung ihrer Interviewantworten werden mehrheitlich nicht die Textstellen ihrer Interviewantworten selbst, sondern die paraphrasierten Zusammenfassungen der Textstellen verwendet. Die Punkte des dpKa-Modells werden nun der Reihe nach behandelt, sodass man sieht, welche Punkte davon eine Voraussetzung darstellen oder auch nicht und welche in der Spieleentwicklung mit Unity erworben werden oder auch nicht. Sollten zusätzliche Punkte auftauchen, die im dpKa-Modell nicht enthalten sind, werden diese nach den dpKa-Modell-Punkten besprochen und eventuell in Form einer induktiven Kategorienbildung in das Modell aufgenommen. Es folgt jetzt also die Beschreibung der Ergebnisse je Kategorie:

### *6.3.4.1 dpKa-Modell Bereich „Verantwortung, Datenschutz und Datensicherheit“*

Dieser Bereich wurde von den Befragten sehr allgemein angesprochen und als eher nebensächlich behandelt, sodass hierin nicht auf einzelne dpKa-Modell-Punkte eingegangen wird. B1 sagte, dass Kenntnisse in diesem Bereich nicht zwingend vorher notwendig sind, aber dass es, wenn vorher nicht vorhanden, spätestens innerhalb der Spielentwicklung mit Unity thematisiert wird, weil man wissen muss, was das Recht am Bild, das Urheberrecht und die Lizenzen bedeuten. Es wurden von B1 dazu auch Beispiele genannt, wo man sieht, dass sie es bei B1 benötigt haben, und zwar folgende. Die SchülerInnen haben die Lehrperson gefragt, welche Bilder sie verwenden dürfen, ob man sich gegenseitig fotografieren und diese Fotos in das Spiel einbauen darf und ob sie Unity-Analytics verwenden sollen. Dadurch wurde der Datenschutz zum Thema, nämlich unter welchen Lizenzen die Grafiken stehen die sie sich „checken“ und auch ob und wie sie Projektdateien im Web oder Netzwerk veröffentlichen können. Auch laut B2 benötigen die SchülerInnen keine Datenschutzvorkenntnisse, allerdings werden sie sich im Verlauf des Projektes der Meinung von B2 nach nicht zwangsweise damit auseinandersetzen. Dazu

wurde von B2 ein Beispiel genannt, wo ein Projektteam einfach aus dem Web Grafiken verwendet hat, sich darüber aber keine Gedanken gemacht hat. B3 meinte nur, dass es nicht schlecht ist, wenn man vor der Spielentwicklung mit Unity bereits über Datenschutz Bescheid weiß.

Das heißt, dass dieser Bereich keine Voraussetzungen für die Spielentwicklung mit Unity stellt. Ob Kompetenzbausteine innerhalb der Spielentwicklung mit Unity erworben werden oder nicht, lässt sich nicht sagen, da das offensichtlich abhängig von der Unterrichtsgestaltung ist, also ob man darauf eingeht oder nicht. Außerdem haben wir im Kapitel Unterrichtskonzept bereits gesehen, dass dieser Bereich nicht explizit im Unterrichtskonzept vorhanden ist. Daher muss man davon ausgehen, dass dieser Punkt nicht erworben wird. Es sei hier trotzdem angemerkt, dass dieser Bereich als positiver Nebeneffekt erworben werden kann, zum Beispiel bei der Aktivität des Einfügens von Spielegrafiken.

**„Verantwortung, Datenschutz und Datensicherheit“: Voraussetzung = nein, Erwerb = nein**

---

#### **6.3.4.2 dpKa-Modell Bereich „Technische Grundlagen und Funktionsweisen“**

Zuerst kommen wir zum dpKa-Modell-Punkt „Technik.W.Computer-2.1.1“. SchülerInnen müssen wissen, dass Spiele ressourcenintensiv sind, sagte B1, bei dem ein Schüler bei einem Spieletest erkannte, dass das Spiel mit den in Schulrechnern eingebauten Onboard-Grafikchips nicht ordentlich funktionieren kann. Durch den Kauf und Einbau von Nvidia-Grafikkarten hat man später auch gesehen, dass es besser geht als mit den Onboard-Chips. B1 sagte weiter, dass man Grundlagen in Bezug auf Funktionsweisen benötigt und dass das auch spezifischer im Bereich der Peripherie und Hardware vorgekommen ist. B1 hält diesen Punkt also für eine Voraussetzung für die Spielentwicklung mit Unity. So sah das auch B2, der sagte, dass es enorm hilft, wenn man versteht, wie Windowssachen funktionieren, man es aber nicht im Detail wissen muss. Von B3 kam hierzu keine für diese Studie verwertbare Aussage.

Dieser Punkt wird als Voraussetzung angesehen und es gibt mit dem Beispiel der Onboard-Grafikchips ein klares Signal, dass dieser Punkt im Verlauf der Spielentwicklung mit Unity erworben wird.

### **Technik.W.Computer-2.1.1: Voraussetzung = ja, Erwerb = ja**

Beim Punkt „Technik.A.Computer-2.1.3“ kamen von allen drei Befragten Antworten zur Bedeutung dieses Punktes und dass bei etwaigen Mängeln darin stets Schwierigkeiten auftraten. Wobei der Fokus auf dem Aspekt des Dateimanagements lag. B1 nannte eine sattelfeste Handhabung des PCs, das Dateimanagement und Kenntnisse und Fertigkeiten bei der Sicherung als Voraussetzung für Unity, um dabei den Prozess reibungs- und verlustfrei zu gestalten. B1 geht mit den SchülerInnen bereits am Anfang des Schuljahres auf Dateimanagement und Sicherung ein. Um das ganze Projekt dann zu realisieren benötigt man an irgendeinem Punkt alle diese Fähigkeiten, so B1. Das spricht für einen Erwerb. B2 nennt das Fehlen von Grundlagen, wie dem Umgang mit dem PC, mit Dateien und Programmen als unpraktikabel. Das Dateimanagement wird noch einmal zusätzlich hervorgehoben, indem es von B2 als Muss genannt wird, ohne dem die Spieleprogrammierung in Unity keinen Sinn macht. Das darf laut B2 nicht im Weg stehen, weil man sich nachher zwangsweise damit auseinandersetzen muss, für die Assets, Grafiken, Sounds und Skripte in Unity, die man speichern, finden und hineinladen muss. B3 beklagt sogar, dass viele beim Programmieren in Unity ausgestiegen sind, die Schwierigkeiten mit dem Dateimanagement hatten, weil dadurch dann zu viele Voraussetzungen gefehlt haben. Die Funktion des Dateimanagements bleibt im Unterricht lange ein Thema. Dadurch lernt man innerhalb von Unity aber auch den Umgang mit dem Dateimanagement. Genauso sind Grundlagen wie ein Betriebssystem funktioniert wichtig, um sich auf Unity einzulassen, sagt B3.

### **Technik.A.Computer-2.1.3: Voraussetzung = ja, Erwerb = ja**

Bei „Technik.R.Computer-2.1.5“ hat man bereits am Beispiel von B1 mit dem Onboard-Grafikchip gesehen, wo der Schüler das digitale Endgerät in Bezug auf seine technischen Eigenschaften und seine Leistungsfähigkeit einschätzen

und bewerten konnte und die restlichen SchülerInnen diese Erfahrung durch den Umbau der Grafikkarte machen konnten. Ob der Schüler, der das gemerkt hat, das schon vorher konnte, oder im Laufe des Unterrichts erworben hat, weiß man nicht, aber die restlichen SchülerInnen haben es erworben. Da dieser Grafikkartenumbau sich jedoch aus einer geschilderten zufälligen Situation ergeben hat und somit anzunehmen ist, dass das nicht jedes Mal umgebaut wird, können wir hier trotz allem nicht auf einen Erwerb schließen. Es gab zu einem Erwerb bezüglich diesem Kompetenzpunkt keine konkrete Frage, weshalb wir den Erwerb nicht beantworten können. Als Voraussetzung wurde der Punkt von niemandem genannt.

**Technik.R.Computer-2.1.5: Voraussetzung = nein, Erwerb = unbeantwortet**

Der dpKa-Modell-Punkt „Technik.R.Fehler-2.1.6“ stellt den Befragten zu Folge keine Voraussetzung dar, wird aber innerhalb der Spieleprogrammierung mit Unity erworben, was sich in Form eines konkreten Beispiels von B2 herauskristallisiert hat. Dabei wird darüber berichtet, dass die SchülerInnen zu Hause mit einer neueren Version von Unity weitergearbeitet haben und als sie wieder in die Schule gekommen sind, funktionierte plötzlich nichts mehr, da hier eine ältere Version von Unity lief. Die SchülerInnen haben dieses Problem auf unterschiedliche Art und Weise behoben. Ein Projektteam hat den eigenen Laptop mitgenommen, andere haben Skripte geschrieben, ohne sie live zu testen, die dann erst in das Programm eingefügt wurden und den anderen Teams wurde von der Lehrperson geholfen. Die SchülerInnen wissen laut B2 immer wieder einmal nicht, wie etwas funktioniert und wie man ein Problem lösen kann, obwohl sie schon eine halbe Idee im Kopf haben. Es wird ihnen dann zum Ausprobieren dieser Idee geraten, wodurch sie die Chance haben diesen Kompetenzpunkt zu erwerben.

**Technik.R.Fehler-2.1.6: Voraussetzung = nein, Erwerb = ja**

### 6.3.4.3 dpKa-Modell Bereich „Betriebssysteme und Software“

Im Punkt „Software.W.Arten-2.2.2“ hob B3 den Teilbereich der Dateiendungen als eine Voraussetzung um mit Unity starten zu können hervor. Für diesen Punkt soll man Kategorien von Software nennen, deren Anwendung beschreiben können und über die Dateiendungen Bescheid wissen. Hier sticht sofort der enge Zusammenhang zweier Kompetenzpunkte heraus, in denen einer (Technik.A.Computer-2.1.3) vom Dateimanagement und dieser von den Dateiendungen handelt. Um Dateien managen zu können, sollte man auch wissen, um welche Dateien es sich handelt, weshalb die Dateiendungen dafür sehr bedeutend sind. Dateimanagement wurde von allen als absolute Muss-Voraussetzung hingestellt und ist gemeinsam mit diesem Kompetenzpunkt und dem der Datensicherung (Software.A.Sicherung-Dk8 7.3.3) komplettiert. Auf die Frage des Erwerbs wurde nicht eingegangen, wobei sich diese von selbst beantwortet, in dem man beim Umgang mit Unity eine Softwarekategorie, nämlich die der Entwicklungswerkzeuge und deren Anwendung erlernt.

#### **Software.W.Arten-2.2.2: Voraussetzung = ja, Erwerb = ja**

Mit dem Punkt „Software.A.Arten-2.2.4“ ist wieder etwas beschrieben, wo sich die Frage des Erwerbs mehr oder weniger von selbst beantwortet. Durch den Umgang mit Unity lernt man sich in die Bedienung für einen selbst neuer Software einzuarbeiten. Auch wenn einem bei den Tutorials noch alles vorgekaut wird, ist man spätestens bei den eigenen Projekten auf sich allein gestellt und befasst sich automatisch mit dem Programm bzw. mit der Software. Dies unterstreicht sowohl die Aussage von B2, laut der man es lernt, sich in eine komplexe Entwicklungsumgebung einzuarbeiten, als auch die von B3, laut der man sich mit einer hochkomplexen Benutzerumgebung mit vielen Einstellungen beschäftigt, bei der das Setzen eines Häkchens gleich etwas bedeutet und auch etwas schrotten kann. B2 sieht den Umgang mit Programmen, also das Öffnen verschiedener Programme, das Wechseln zwischen ihnen und das Arbeiten mit ihnen als eine dieser Grundlagen, ohne der die Spieleentwicklung mit Unity unpraktikabel wäre. B1 erkannte ebenso

Schwierigkeiten bei SchülerInnen, wenn ihnen bei der Bedienung von Tools des PCs etwas fehlte und sagte, dass SchülerInnen besser mit Anwendungssoftware umgehen können sollten. Damit gilt dieser Punkt sowohl als Voraussetzung als auch als damit erworben.

**Software.A.Arten-2.2.4: Voraussetzung = ja, Erwerb = ja**

„Software.A.Text-Dk8 3.2.1“ und „Software.A.Korrektur-Dk8 3.2.3“ sind Punkte die gemeinsam behandelt werden. Beim ersten Punkt geht es um die zügige Texteingabe und beim Zweiten um das Durchführen von Textkorrekturen. B1 brachte es hierbei sehr gut auf den Punkt, indem B1 davon spricht, dass es kaum vorkommt, dass man mit der Textverarbeitung nicht in Berührung gekommen ist und in die Spieleprogrammierung geht. Trotzdem merkte B1 an, dass es wichtig wäre, wenn SchülerInnen Sachanleitungen richtig eingeben können. Es wird jedoch von niemandem als konkrete Voraussetzung genannt. In jedem Fall erwirbt man diese Punkte im Verlauf des Projektes, also bei der Programmierung mit Unity, da bei den Tutorials und möglicherweise auch bei den eigenen Spielen danach Skripte in C# geschrieben werden.

**Software.A.Text-Dk8 3.2.1: Voraussetzung = nein, Erwerb = ja**

**Software.A.Korrektur-Dk8 3.2.3: Voraussetzung = nein, Erwerb = ja**

Der Punkt „Software.A.Hilfe-Dk8 7.3.2“ will wissen, ob SchülerInnen Hilfesysteme verwenden. B1 äußerte sich dazu bei der Frage nach einem optimalen Rüstzeug der SchülerInnen bevor sie mit Unity beginnen. B1 sieht im optimalen Rüstzeug, dass sie unter anderem strukturiert Nachschauen, was ein Programm kann und was es darin alles gibt. Es ist jedoch keine Rede von einer Voraussetzung oder einem Erwerb dieses Punktes. Daher stellt er keine Voraussetzung dar und da er bezüglich des Erwerbs nicht abgefragt wurde, bleibt er dazu unbeantwortet.

**Software.A.Hilfe-Dk8 7.3.2: Voraussetzung = nein, Erwerb = unbeantwortet**

Der dpKa-Modell-Punkt „Software.A.Sicherung-Dk8 7.3.3“ wurde laut B3 und B1 bereits beim Punkt rund um das Dateimanagement beantwortet. B3 sagte, dass B3 mit Dateimanagement nämlich auch Datensicherung und

Datenwiederherstellung meint und B1 spricht diese Dinge überhaupt alle schon bei den Fragen rund um das Dateimanagement konkret an. Das heißt, dass dieser Punkt eine Voraussetzung darstellt und auch erworben wird.

**Software.A.Sicherung-Dk8 7.3.3: Voraussetzung = ja, Erwerb = ja**

---

### *6.3.4.4 dpKa-Modell Bereich „Produktion digitaler Medien“*

Mit einem erworbenen Kompetenzbaustein „Medien.W.Formate-3.1.1“ ist man fähig, gängige Medienformate und ihre Eigenschaften zu beschreiben. Bei der Spieleentwicklung mit Unity werden Grafiken schon im Bouncy Fant Tutorial verwendet und möglicherweise auch noch beim eigenen Projekt danach. Außerdem erstellt man mit dem Tutorial und dem eigenen Projekt ein Spiel, was ebenfalls ein Medienformat darstellt. Das heißt, man lernt Medienformate und deren Eigenschaften kennen und erwirbt somit diesen Kompetenzbaustein. Zur Frage ob es eine Voraussetzung ist, sagen alle drei, dass es im Bereich der Medien keine Voraussetzung gibt. B1 meint, dass alles so konzipiert ist, dass man im Bereich Medien von Null anfangen kann.

**Medien.W.Formate-3.1.1: Voraussetzung = nein, Erwerb = ja**

Den Punkt „Medien.A.Formate-3.1.3“ würde B3, wenn B3 unendlich Zeit mit den SchülerInnen hätte, öfter üben, damit sie lernen wie diese Layering-Geschichte funktioniert, die auch in Unity benötigt wird. Dazu würde B3 viel mit Grafiken und GIMP arbeiten. Aber B3 stellt auch klar, dass es keine Voraussetzung ist, da man keine selbst erstellten Grafiken nehmen muss, sondern irgendwelche Grafiken nehmen kann. B1 erklärt es auch als Vorteil, wenn man mit GIMP umgehen kann, sieht hier aber ebenso keine Voraussetzung. Der Erwerb dieses Kompetenzbausteines geht schon aus dem Tutorial hervor, wo man lernt, die Spiele und somit digitale Medien zu fertig ausführbaren Programmen zu machen. Dadurch lernt man digitale Medien sachgerecht zu bearbeiten und produzieren.

**Medien.A.Formate-3.1.3: Voraussetzung = nein, Erwerb = ja**



Der Erwerb der Kompetenzbausteine „Medien.R.Formate-3.1.4“ und „Medien.R.Selbstwirksamkeit-Dk8 4.2.1“ wurde mit einem Beispiel von B2 untermauert, wo die SchülerInnen ein Problem hatten und sich eine Lehrperson etwas dazu überlegt und ihnen auch präsentiert hat. Doch die SchülerInnen haben sich gegen diesen Lösungsvorschlag entschieden, da er ihrer Meinung nach zu kompliziert war. Damit haben sie bereits eine Einschätzung und Bewertung eines digitalen Produktes vorgenommen (Medien.R.Formate-3.1.4). Anschließend haben sie sich dann selbstständig über Tutorials, Videos und Foren informiert und schließlich ihre eigene Lösung implementiert, was für den Baustein spricht, laut dem sie sich selbstwirksam erleben, indem sie digitale Technologien kreativ und vielfältig nutzen. B1 sagte sogar ganz klar, dass die Lernenden in ihrer Selbstwirksamkeit gestärkt hervorgehen. Sie trauen sich mehr zu und schätzen sich in Bereichen wie Algorithmen, Datenstrukturen oder Modellierung zum Beispiel kompetenter ein. Somit wurden beide Punkte erworben und wie schon bei den vorigen Punkten angesprochen, wird dieser ganze Bereich nicht als Voraussetzung angesehen. Zum Punkt „Medien.R.Techniken-Dk8 4.2.5“ ließ sich nichts Verwertbares in den Interviewantworten finden, außer dass er in den selben Bereich fällt und somit keine Voraussetzung ist.

**Medien.R.Formate-3.1.4: Voraussetzung = nein, Erwerb = ja**

**Medien.R.Selbstwirksamkeit-Dk8 4.2.1: Voraussetzung = nein, Erwerb = ja**

**Medien.R.Techniken-Dk8 4.2.5: Voraussetz. = nein, Erwerb = unbeantwortet**

---

#### ***6.3.4.5 dpKa-Modell Bereich „Suche, Auswahl und Organisation von Information***

Die Punkte „Info.W.SucheQuellen-3.3.1“ und „Info.W.SucheMethoden-3.3.2“ stellen im Grunde eine Aufspaltung von einem sonst möglich gewesenem, übergeordneten Punkt „Info.W.Suche“ dar. Denn auf der Wissensebene dieses Bereiches wären dann die Ideen, dass man wichtige Informationsquellen im Internet anführen kann und dass man in Netzwerken Methoden der Informationsgewinnung benennen kann, in einem Punkt vereint. Jedenfalls

erklärt B1 wie gut es wäre, wenn man Recherche-Kompetenzen hat, weil man dann Assets und Dinge, die man implementieren will, im Internet suchen und holen kann und dann auch das Richtige findet. Damit ist zudem auch noch der dazugehörige Punkt auf der Fertigkeitsebene „Info.A.Suche-3.3.4“ angesprochen, laut dem man unter Verwendung passender Dienste und Angebote Informationen und Medien gezielt suchen und auswählen kann. B1 fügt hinzu, dass es keine Voraussetzung darstellt, da im Projektpaket die Assets zum Beispiel mit eingeschnürt sind und man nur im Bereich der Tutorials stöbern müsste. Trotz allem zählt B1 diese drei Punkte in Form der Recherche-Kompetenz beim optimalen Rüstzeug vor der Spielentwicklung mit Unity auf. B2 spezifiziert das Ganze, indem B2 sagt, dass die gezielte Recherche für das detaillierte Tutorial-Projekt nicht, für das eigenständige Projekt aber sehr relevant ist. Laut B2 ist die Spielentwicklung gut geeignet um Kompetenzen im Bereich der Suche, Auswahl und Organisation von Informationen zu erwerben, da hier die Motivation sehr hoch ist. So auch die Meinung von B3, indem gesagt wird, dass man diese Dinge „on-the-way“ lernt, es aber nicht schlecht ist, wenn man vorher schon weiß, wie man Finder oder Explorer einsetzen kann. Man wird es aber notgedrungen lernen, wenn man Unity verwendet, so B3. Zusammenfassend gesagt, stellt es keine Voraussetzung dar, auch wenn Vorkenntnisse und Fertigkeiten hier wünschenswert sind.

Zum Erwerb dieser Kompetenzbausteine findet man ein Beispiel von B2, dass eben diesen aufzeigt. Dabei handelt es sich um SchülerInnen, die ein 2D-Spiel mit top-down-Perspektive und der Bewegung einer Spielfigur wie auf einem Schachbrett implementiert haben. Da diese Dinge in Unity standardmäßig nicht als Vorlage enthalten sind, ist die Position der Spielfigur zum Beispiel in Fließkommazahlen anzugeben. Für diese Implementierung haben sie laut B2 gelernt, sich durch Foren und Tutorials durchzukämpfen und haben zum Beispiel von Stack-Overflow Lösungen kopiert und für ihr Projekt adaptiert. B1 sprach darüber, dass zumindest die Musterschüler von B1 verstehen, dass Hilfe und Recherche viel Zeit sparen kann und andere Perspektiven für mögliche Problemlösungen eröffnen. Als ein Aspekt der Informationssuche und -auswahl

beschreibt der Punkt „Info.A.SucheBedarf-Dk8 2.1.1“ das Formulieren der Bedürfnissen der SchülerInnen für die Informationssuche, worüber das Beispiel von B2 spricht, wo SchülerInnen ohne Enumerationen zu kennen, diese im Internet gefunden haben und dann auch in ihrem Projekt als passende Lösung umgesetzt haben. Man erwirbt also alle diese Punkte.

**Info.W.SucheQuellen-3.3.1: Voraussetzung = nein, Erwerb = ja**

**Info.W.SucheMethoden-3.3.2: Voraussetzung = nein, Erwerb = ja**

**Info.A.Suche-3.3.4: Voraussetzung = nein, Erwerb = ja**

**Info.A.SucheBedarf-Dk8 2.1.1: Voraussetzung = nein, Erwerb = ja**

Somit bleiben noch die Punkte „Info.W.Wissensmanagement-3.3.3“ und „Info.A.Wissensmanagement-3.3.5“ aus diesem Bereich, die nicht als Voraussetzung genannt wurden und deren Erwerb nicht abgefragt und auch nicht herausgelesen werden konnte.

**Info.W.Wissensmanagement-3.3.3: Vor. = nein, Erwerb = unbeantwortet**

**Info.A.Wissensmanagement-3.3.5: Vor. = nein, Erwerb = unbeantwortet**

---

#### **6.3.4.6 dpKa-Modell Bereich „Algorithmen, Datenstrukturen und Programmierung**

In diesem Bereich liegt das Hauptaugenmerk dieser Arbeit auf dem Erwerb von Kompetenzbausteinen. Dennoch kann es vorkommen, dass auch hier Voraussetzungen zu finden sind, da Unity, wie wir gehört haben, eine sehr komplexe Entwicklungsumgebung ist. Genauso haben wir in den bisherigen Bereichen Kompetenzbausteine gefunden, die durch Unity erworben werden, obwohl die Untersuchung des Erwerbs der bisherigen Kompetenzbausteine ursprünglich nicht geplant war. Die Punkte in diesem Bereich wurden sehr konkret befragt, weshalb teilweise eher kurze, aber klare Aussagen als Antworten gekommen sind, außer die befragten Personen fanden ein Beispiel aus dem Alltag dazu.

Beginnen wir mit dem Punkt „Prog.W.Datenstruktur-[KsmOOP 1]“. Variablen sollten in einer isolierten Umgebung kennengelernt werden, da in Unity dann

viel zu viel Drumherum ist, war die Aussage von B2 dazu. B1 unterstützt dieses Argument, da B1 genauso die Datenstrukturen bereits im Konsole-Tutorial durchmacht. Für den Erwerb kam von B2 ein aussagekräftiges Beispiel zur Sprache, wo SchülerInnen für die Zustände eines Spielercharakters mit einer Enumeration modelliert haben und diese Enumeration dann in eigenen Worten B2 erklärt haben. Sie wurde so beschrieben, dass man eine Variable mit festgesetzten Wertmöglichkeiten hat, so wie Boolean, nur mit mehr als zwei. Die SchülerInnen haben damit eine Datenstruktur beschrieben, was in dem Punkt genau so verlangt wird. B3 und B1 unterstreichen das mit ihren Aussagen. Der Kompetenzbaustein ist somit Voraussetzung, um effektiv in Unity Spiele zu entwickeln, könnte darin aber auch noch erworben bzw. verbessert werden.

### **Prog.W.Datenstruktur-[KsmOOP 1]: Voraussetzung = ja, Erwerb = ja**

SchülerInnen können den Algorithmus-Begriff erklären, besagt der Punkt „Prog.W.Algorithmus-4.2.1“. B3 sagt, dass B3 schon in der zweiten Unterrichtsstunde bzw. beim Tutorial über Algorithmen spricht, also dass es um Algorithmen geht und was ein Algorithmus ist. Außerdem stellt B3 klar, dass man, sobald man selbstständig zu arbeiten beginnt und etwas entwickelt, nicht an Algorithmen vorbeikommt. Auch B2 und B1 erklären den SchülerInnen den Algorithmus-Begriff im Unterricht. Laut B1 wird beim Bouncy-Fant-Tutorial zwischendurch auf den Algorithmus-Begriff eingegangen.

### **Prog.W.Algorithmus-4.2.1: Voraussetzung = nein, Erwerb = ja**

Der Kompetenzbaustein „Prog.W.Modellierung-4.2.2“ wird vor allem beim eigenen Projekt in Unity erworben, wo SchülerInnen sich vorher einmal etwas überlegen müssen, bevor sie es ausarbeiten, so B3. B1 spricht dabei zusätzlich von einem Beispiel mit einer State-Machine, die zur Modellierung dient. Dort müssen SchülerInnen Statusübergänge mit Bedingungen realisieren und da bilden sich Repräsentanzen von Algorithmen in den Köpfen der Lernenden. Es kann aber durchaus auch eine Abstraktion von einem tatsächlichen Algorithmus sein, der schon in Unity implementiert ist und mit dem die Lernenden dann

arbeiten, wobei sie sagen, dass wenn ich dieses und jenes mache, dann passiert dieses und jenes mit dem Objekt, bei diesem Übergang. Dann haben sie sozusagen eine eigene algorithmische Repräsentanz von diesem Vorgang. Der Kompetenzbaustein wird somit erworben, wird aber nicht vorausgesetzt.

**Prog.W.Modellierung-4.2.2: Voraussetzung = nein, Erwerb = ja**

Der Punkt „Prog.W.Software-4.2.3“, der nun an der Reihe wäre, wird erst später beim Punkt „Prog.A.Software-4.2.7“ mitbesprochen, da es dazu kaum etwas Verwertbares gibt, der Punkt aber sehr stark mit dem späteren zusammenhängt.

Wesentliche Aspekte der Prozeduralen, Funktionalen und Objektorientierten Programmierung nennen und an Beispielen erläutern zu können, ist die Forderung vom Punkt „Prog.W.Aspekte-4.2.4“. Es geht also einfach ausgedrückt um wesentliche Aspekte der Programmierung. Laut B1 erklären die Lernenden die Aspekte, wenn man danach fragt und dabei bekommt man sogar die Fachbegriffe zu hören, was B1 ganz besonders gefreut hat. B3 sagt, dass diese Aspekte durchaus explizit erwähnt und behandelt werden. B2 wiederum hat klar gesagt, dass die SchülerInnen diese Aspekte nennen können. Dieser Punkt ist also wieder keine Voraussetzung, wird aber erworben.

**Prog.W.Aspekte-4.2.4: Voraussetzung = nein, Erwerb = ja**

Auf die Frage, ob SchülerInnen Programmiersprachen und Produktionsabläufe kennen lernen, antworteten alle mit ja, und zwar bei der Programmiersprache lernen sie alle C#. Gemeint ist hier der Punkt „Prog.W.Sprache-Dk8 8.2.2“. Die gelernten Produktionsabläufe variieren sehr stark, sind aber vorhanden, so B2. B3 weist bei den Produktionsabläufen auf die Animationsabläufe hin, die in Unity erstellt werden. Der Erwerb ist hier gegeben. Hier wird auch gleich auf den nächsten Punkt „Prog.A.Sprache-Dk8 8.1.2“ eingegangen, da diese beiden vom gleichen Aspekt, nämlich der Sprache handeln. Der Kompetenzbaustein wird von B2 bei den Schwierigkeiten bei der Programmierung angesprochen. Die Probleme findet man bei der Klammersetzung und der Syntax von C#, wo am Befehlsende ein Strichpunkt verlangt wird beispielsweise. B3 hält Unity nicht

für die erste Wahl um diese ganzen Programmiergeschichten und wie Programmieren funktioniert zu lernen. Stattdessen sollte man das in einer anderen Programmierumgebung machen. So war auch die Meinung von B2, dass die Grundlagen vom Coden Dinge sind, die den Arbeitsfluss wirklich behindern können und daher vorher behandelt werden sollten. B2 zeigt in diesem Bereich ganz genau auf, was vorher gemacht werden sollte, sodass man die Lerneffekte in Unity dann dort hat, wo man sie haben will, nämlich ein algorithmisches Denken beibringen und den Umgang mit textbasiertem Code üben. B2 verwendet für Letzteres html, das zwar eine Auszeichnungssprache und keine wirkliche Programmiersprache ist, aber sehr wohl den Umgang mit textbasiertem Code schult. Diese beiden Punkte stellen also eine Voraussetzung dar und werden auch erworben, sofern man darüber noch nicht verfügt oder sonst einfach verbessert.

**Prog.W.Sprache-Dk8 8.2.2: Voraussetzung = ja, Erwerb = ja**

**Prog.A.Sprache-Dk8 8.1.2: Voraussetzung = ja, Erwerb = ja**

Als nächstes kommen wir zum Punkt „Prog.A.Modellierung-4.2.5“. Laut B1 passiert die Formalisierung und Modellierung von Problemen auf nicht-universitärem Niveau bereits in der Schule. B2 sagt ebenso, dass sich die SchülerInnen darüber Gedanken gemacht haben, wie sie bestimmte Dinge modellieren, aber dass es keine Voraussetzung für eine erfolgreiche Spieleentwicklung mit Unity ist. Für B3 ist es wichtig, dass sich die SchülerInnen darauf einlassen. Und diejenigen die das getan und ein eigenes Spiel gemacht haben, gelernt haben, Aufgaben mit Mitteln der Informatik zu modellieren. B3 erzählt dabei von einem Projektteam, das 3D-Pong als Spiel entwickelt hat, das in keiner Form in den Tutorials vorkommt. Das Team hat sich das Spiel selbst überlegt und implementiert. Dabei wurde eine 3D-Umgebung verwendet, die im Tutorial ebenso nie verwendet wurde, sondern nur die 2D-Umgebung. Zudem wurden die für Pong benötigten 3D-Objekte gestaltet und deren Bewegung ermöglicht. B3 zählt die Modellierung nicht zu den Voraussetzungen, sondern ist sogar sehr gut im Unity-Editor erlernbar, indem man für ein Spiel die Spielszenen modelliert zum Beispiel. Dieser Punkt

ist also keine Voraussetzung, wird aber in der Spielentwicklung mit Unity erworben.

**Prog.A.Modellierung-4.2.5: Voraussetzung = nein, Erwerb = ja**

Der Baustein „Prog.A.Algorithmus-4.2.6“ gibt vor, dass man Algorithmen entwerfen, formal darstellen, implementieren und testen kann. B1 sagt, dass eine Konfrontation mit Algorithmen auf verschiedene Arten und Weisen zustande kommen, indem die SchülerInnen im Unterricht zum Beispiel einen Algorithmus für die Bewegung eines Spielobjektes schreiben oder den Boden vom Bouncy Fant Tutorial mit einer Schleife zeichnen. Auch B3 sagt, dass man sich bei den eigenen Spielen sehr stark mit Algorithmen beschäftigt, um die Spiellogik zu programmieren. Dabei wird an bestimmte Probleme per „trial-and-error“ und mit Fragen an Lehrpersonen herangegangen. B2 berichtete ebenfalls, dass Dinge wie Verzweigungen (if-else) und Funktionen in all den selbst erstellten Spielen vorgekommen sind. Leider kamen Schleifen allerdings selten vor, weil die Spielschleife im Hintergrund läuft und sehr viel über die Updatemethode periodisch aufgerufen wird. B2 konnte diesen Erwerb zum Beispiel bei seinen SchülerInnen die das 2D-Spiel mit „top-down-Perspektive“ entwickelten beobachten, da sie Lösungen von „Stack-Overflow“ kopiert haben und für ihr Spiel adaptieren mussten. Für B2 und B3 stellt dieser Punkt eine Voraussetzung dar, da es ohne dem algorithmischen Denken zu Schwierigkeiten kommt und der Arbeitsfluss zu sehr ins Stocken gerät, sodass die Lerneffekte nicht effektiv gesetzt werden können. B3 schlägt dazu vor, mehr Zeit in Scratch zu verbringen. B2 nutzt für die Schulung des algorithmischen Denkens ebenso Scratch. Also lässt sich dieser Punkt, so wie die Punkte rund um die Programmiersprache selbst, unter der Kategorie Voraussetzung einordnen. Und auch er kann mit Unity erworben bzw. verbessert werden.

**Prog.A.Algorithmus-4.2.6: Voraussetzung = ja, Erwerb = ja**

Beim dpKa-Modell-Punkt „Prog.A.Datenstruktur-[KsmOOP 2]“ geht es um die Verwendung von Daten-, Klassen- und Objektstrukturen in einem Algorithmus. Bei B1 werden die Datenstrukturen zum Beispiel beim Textadventure in Unity

behandelt, aber nicht nur dort, da bei Implementierungen in Unity fast täglich Daten-, Klassen- und Objektstrukturen in Algorithmen verwendet werden. B1 untermauert diesen Erwerb mit dem Beispiel, dass der Elefant im Bouncy Fant Tutorial als Gameobject instanziiert wird. Danach wird von außerhalb darauf verwiesen und im Quellcode werden die Eigenschaften und das Verhalten schließlich manipuliert. Die Manipulation der Bewegung erfolgt wiederum über den Inputcontroller, also einem anderen Objekt, so B1. B2 bestätigt das, dass die SchülerInnen bei der Spielentwicklung mit Unity Variablen, Funktionen und Klassen verwendet werden und sie daher dazu fähig sind. B2 führt dazu ein Beispiel an, bei dem SchülerInnen für zwei Spielfiguren eine Oberklasse oder Parentklasse gemacht haben, damit sie das Verhalten der Spielfiguren nur einmal schreiben müssen und vererben können. In einem SchülerInneninterview konnten sie das danach auch erklären und beschreiben, warum sie sich dazu entschieden haben und warum das Sinn macht, sagt B2. Dem Baustein wird keine Voraussetzung zugeschrieben.

### **Prog.A.Datenstruktur-[KsmOOP 2]: Voraussetzung = nein, Erwerb = ja**

Nun wird nicht nur auf den Punkt „Prog.A.Software-4.2.7“, sondern auch auf den zuvor ausgelassenen Punkt „Prog.W.Software-4.2.3“ eingegangen, denn wenn man ein Softwareprojekt planen und durchführen kann, dann kann man auch wesentliche Aspekte und Methoden der Softwareentwicklung und des Softwareprojektmanagements erklären. Gleich vorweg lässt sich sagen, dass beide Punkte von den Befragten nicht als Voraussetzung genannt wurden. Zum Erwerb sagt B1, dass die SchülerInnen das Tutorial auch als Projekt wahrnehmen, sie dann aber bei den eigenen Spielen auch lernen, ein Softwareprojekt zu planen und durchzuführen. Dabei bemerken sie auch wo Schwierigkeiten auftreten können und was man machen und vermeiden sollte. B2 und B3 schließen sich dieser Meinung an indem B3 sagt, dass die SchülerInnen in seinen Klassen eindeutig Projekte gemacht haben und diese wirklich verantwortungsvoll selbst ausgearbeitet haben. Laut B2 sind sie nach der Spielentwicklung mit Unity fähig selbstständig Softwareprojekte durchzuführen. Man erlernt mit Unity Programmlogiken, die nicht C#-spezifisch



sind, nämlich dass man nachher weiß was Funktionen sind und wie Operatoren und Variablen funktionieren, sodass man auch in einer anderen Programmiersprache ein kleines Programm schreiben oder abändern kann, so B2. Weiter ist die Spielentwicklung sehr gut für Projektarbeiten geeignet, da sie sehr viele Aspekte der Informatik vernetzt, wie zum Beispiel das Erstellen und Bearbeiten von Bildern, 3D-Modellen, Animationen und Soundeffekten. Dadurch werden im Unterricht getrennt behandelte Sachen in einem Spielprojekt verknüpft. Zudem erfolgt laut B2 eine Individualisierung durch die Spielentwicklung, da sich jeder auf seine Stärken fokussieren kann, indem der der gerne zeichnet die vorwiegend die Grafiken zeichnen kann und ein anderer sich wiederum auf die Soundeffekte und die Musik stürzt. Beide Punkte werden innerhalb der Spielentwicklung mit Unity erworben.

**Prog.W.Software-4.2.3: Voraussetzung = nein, Erwerb = ja**

**Prog.A.Software-4.2.7: Voraussetzung = nein, Erwerb = ja**

Der Punkt „Prog.R.Software-4.2.8“ wird, genauso wie die beiden Punkte zuvor, erworben, was sich an den Aussagen aller Befragten verdeutlicht. Beschrieben wird mit dem Punkt, dass man die Schritte der Softwareentwicklung reflektieren kann. Die SchülerInnen bei B1 haben zum Beispiel gelernt, dass sie den Blick für das Wesentliche behalten müssen, da sie eigentlich auch noch dieses und jenes machen wollen. Doch für den Projektfortschritt muss man den einen oder anderen Nebenquest sein lassen, um schließlich den Hauptpreis zu holen. B1 fügt hinzu, dass die Reflexion der eigenen Spieleprojekte zu großem Teil auch durch „Interviewbased Code-Reviews“ passiert. B3 erzählt von einem Bericht, den die SchülerInnen abgeben, indem festgehalten wird, was sie gemacht haben und dass dieser Bericht in irgendeiner Form eine Reflexion ist. Zuvor, beim Durcharbeiten der Tutorials erfolgt laut B3 jedoch noch keine Reflexion, da diese sehr fachlich tool- und programmierlernorientiert sind. B2 ist der Überzeugung, dass die SchülerInnen nach dem Projekt über Softwareprojekte reflektieren können und geht darauf ein, dass Spielentwicklung gut für Projektarbeiten genutzt werden kann, da dabei freies Arbeiten, das Verfolgen eigener Ziele und die Teamarbeit ermöglicht werden können. Und gerade die

Teamarbeit regt nach der Aussage von B2 dazu an, sich mit der Lesbarkeit und Wartbarkeit von Code auseinanderzusetzen, womit eine Reflexion erfolgt. Der Punkt kann also erworben werden.

### **Prog.R.Software-4.2.8: Voraussetzung = nein, Erwerb = ja**

Ob SchülerInnen die Angemessenheit der Entwicklungswerkzeuge grob einschätzen können, ist die Frage hinter dem Kompetenzbaustein „Prog.R.SoftwareWerkzeug-4.2.9“. B3 zu Folge können sie das bei B3 nicht, da sie dazu mehrere Entwicklungswerkzeuge kennen und verstehen müssten worum es dabei geht. Laut B1 lernen sie durch die vorbereitenden Tätigkeiten – bzw. Arbeiten mit GIMP und Scratch –, ganz gut einzuschätzen, welches Entwicklungswerkzeug angemessen ist. Aber auch innerhalb von Unity lernen sie das, sagt B1 und hängt das Beispiel an, dass die SchülerInnen in den meisten Fällen selbst einschätzen, dass die Transformation des Rüssels eines Elefanten in GIMP einfacher ist als in Unity, wohingegen für die Änderung der Farbe eines Elefanten innerhalb von Unity nicht auf GIMP zurückgegriffen wurde. Dieses Beispiel hat leider einen kleinen Schönheitsfehler, nämlich dass ein Vergleich zwischen Unity und GIMP stattfindet, die Studie aber den Erwerb innerhalb von Unity untersucht. B2 sagt dazu, dass SchülerInnen teilweise sicher im Stande sind, die Angemessenheit der Entwicklungsumgebungen grob einzuschätzen. Dieser Punkt lässt sich bezüglich eines Erwerbs nicht eindeutig beantworten, da wir von B3 ein starkes Argument gegen den Erwerb, mit B2 und B1 jeweils eine Aussage für den Erwerb haben, wobei das Beispiel von B1 zeigt, dass die dazugehörige Aussage in Bezug auf unsere Studie nicht für einen Erwerb spricht, da hier das Projekt als Ganzes und nicht nur die reine Spielentwicklung mit Unity herangezogen wurde. Persönlich tendiere ich aufgrund der Argumentation von B3 zu keinem Erwerb, womit ich allerdings die Aussage von B2 untergraben würde. Man müsste diesen Punkt noch einmal untersuchen, um eine fehlerlose Antwort geben zu können. Daher bleibt der Erwerb hier leider unbeantwortet.

### **Prog.R.SoftwareWerkzeug-4.2.9: Vorauss. = nein, Erwerb = unbeantwortet**

Der Baustein „Prog.R.AlgorithmusMuster-Dk8 8.1.5“ findet beim Erwerb klare Zustimmung. B3 sagt, dass die SchülerInnen, die sich damit auseinandersetzen, diese Regeln und Muster in den Handlungsanleitungen auch entdecken. B1 gibt dazu das Beispiel an, dass die SchülerInnen beim Bouncy Fant Tutorial die Game-Objects erkennen, was B1 erlebt hat, indem ein Schüler gesagt hat, dass die Elefanten eigentlich gleich sind, wie der herunterfallende Müll, nur dass man den Müll nicht steuern kann. B2 erklärt sich das so, dass einige SchülerInnen bei ihren eigenen Projekten versucht haben Dinge aus dem Tutorial-Spiel nachzubauen und abzuändern. Außerdem kennen die SchülerInnen laut B2 Programmierkonzepte, sodass sie wissen wie Schleifen und Funktionen funktionieren und sie, wenn sie eine andere Programmiersprache benötigen, sich diese dadurch einfach aneignen können. Somit wurden eindeutige Nachweise für den Erwerb gefunden.

**Prog.R.AlgorithmusMuster-Dk8 8.1.5: Voraussetzung = nein, Erwerb = ja**

Der Punkt „Prog.R.AlgorithmusBedeutung-Dk8 8.1.6“ handelt vom Erkennen der Bedeutung von Algorithmen. B1 sagt, dass das per IT-Skills-Test abgefragt wird, wobei Lernende eine Funktion analysieren und die Ausgabe der Funktion als Antwort geben sollen. Und ja, laut B1 erkennen die SchülerInnen die Bedeutung von Algorithmen. B3 sagt ebenso, dass die SchülerInnen die Bedeutung von Algorithmen erkennen lernen, da Algorithmen im Unterricht ausgearbeitet werden. B3 bringt auch ein Beispiel wo man das sieht, nämlich an zwei Gruppen, von denen eine Gruppe einen eigenen Dating-Algorithmus selbst entwickelt und die andere Gruppe sich den Dijkstra-Algorithmus angeschaut, ausgearbeitet und vorgestellt hat. Die Bedeutung eines Algorithmus zu erfassen ist laut B2 im Bereich der Spieleentwicklung viel einfacher, da man den Algorithmus durchspielen kann, und anhand des Spieles wirklich die Auswirkungen vom Algorithmus sieht. Bestimmte Algorithmen lassen sich in Unity sehr gut anschaulich darstellen und man hat sofort Feedback, da Änderungen im Code bei der Spieldurchführung sofort ersichtlich werden. Dafür ist ein Konsolenprogramm ungünstig, da es wahnsinnig schwer

ist, anhand des Codes zu erfassen was ein Algorithmus tatsächlich macht, so B2. Der Erwerb ist anhand dieser beiden Beispiele gut ersichtlich.

### **Prog.R.AlgorithmusBedeutung-Dk8 8.1.6: Voraussetzung = nein, Erwerb = ja**

Können die SchülerInnen auch die Effizienz von Algorithmen bewerten, wird im Punkt „Prog.R.AlgorithmusEffizienz-4.2.10“ gefragt. B3 ist nicht der Meinung, dass sie das können, solange man den Unterricht nicht entsprechend ausrichtet. Außerdem ist Unity, laut B3 für einen Unterricht rund um die Effizienz eines Algorithmus ungeeignet. B2 sagt, dass wenn man die Effizienz von Algorithmen im Unterricht nicht extra behandelt, es den SchülerInnen auch nicht zwangsweise klar ist, dass man bei Performance-Problemen den Algorithmus effizienter gestalten kann. Sie gehen dann nur davon aus, dass die Hardware zu langsam ist. Und es ist in der Praxis auch nicht vorgekommen, dass sie das Problem auf den Algorithmus bezogen haben. Ein Erwerb ist hier nicht gegeben.

### **Prog.R.AlgorithmusEffizienz-4.2.10: Voraussetzung = nein, Erwerb = nein**

Der letzte Punkt des dpKa-Modells „Prog.R.Sprache-4.2.11“ besagt, dass man gezielt nach Programmfehlern suchen und diese korrigieren kann. B2 sagt, dass SchülerInnen verstehen was einzelne Statements machen, aber sich bei Problemen trotz halber Ideen nicht sicher sind, wie sie es lösen. Zuerst wird ihnen dann eben zum Ausprobieren ihrer Ideen geraten und sollte das nicht das Problem lösen, wird ihnen durch die Lehrperson geholfen. Laut B1 und B3 gehen die SchülerInnen bei ihren eigenen Projekten hunderte „trial-and-error“-Phasen durch, bis einmal alles funktioniert. Dadurch lernt man wo Programmierfehler auftauchen können und wie man sie behebt. Also wird auch dieser Punkt im Laufe der Spielentwicklung mit Unity erworben.

### **Prog.R.Sprache-4.2.11: Voraussetzung = nein, Erwerb = ja**

#### **6.3.4.7 eventuelle zusätzliche Punkte**

Es wird hier auf digitale Bausteine eingegangen, die zusätzlich zu den abgefragten Kategorien bzw. dpKa-Modell-Punkten zur Sprache gekommen sind und einen Grund liefern, dass eine ergänzende induktive Kategorienbildung stattfinden sollte. Im Laufe der Interviews wurden tatsächlich einige weitere Dinge angesprochen, die aber alle keine digitalen Kompetenzbausteine sind, sondern Softskills oder Kompetenzbausteine außerhalb des digitalen Bereiches. Daher kommt es auch zu keiner induktiven Kategorienbildung. Trotzdem möchte ich hier kurz auf eine Fertigkeit eingehen, die einen sehr großen Einfluss auf die Programmierung haben kann, nämlich das Lesen. B3 hat die Bedeutung dieser Fertigkeit wie folgt hervorgehoben (zusammengefasstes Zitat): „Lesefähigkeit ist etwas, dass sehr nahe bei den Grundkompetenzen, was Lernen und Leben angeht ist und ist gerade beim Programmieren sehr wichtig. Wir haben es mit Programmier-"Sprachen" zu tun, was eigentlich immer etwas mit Sprachen-lernen zu tun hat. Wenn den SchülerInnen da etwas fehlt, also beim Lesen, Sprachverständnis und damit zusammenhängend dem logischen Denken, dann tun sie sich noch schwerer. In der Unterstufe sollte man den Fokus darauflegen, dass die Leute Lesen, Schreiben und Rechnen können. Wichtig ist vor allem eben auch Rechnen und Angaben in der Mathematik zu verstehen, also dass man Textbeispiele versteht. Wenn da eine gute Grundlage geschaffen wird, sodass Leute auch mehrere Sätze sinnerfassend lesen können, möglicherweise sogar auf Englisch, dann ist das Programmieren ein guter nächster Schritt.“ Erinnern wir uns auf ein Zitat von Hauke Fehr innerhalb dieser Arbeit. Er weist dort ebenfalls auf die Bedeutung des Lesens, ganz besonders in Englisch hin. Die Fertigkeit sinnerfassend lesen zu können, scheint für die Programmierung einen enorm hohen Stellenwert einzunehmen und könnte daher auch für die Programmierung mit Unity eine sehr gute Basis bzw. einen Grundstein von höherer Bedeutung darstellen. Dieser Arbeit beschränkt sich jedoch auf die digitalen Kompetenzbausteine, weshalb hier leider nicht weiter darauf eingegangen werden kann.

### 6.4 Ergebnisdarstellung

In diesem Kapitel erfolgt eine interpretative Einordnung der Ergebnisse in den größeren theoretischen Kontext. Dazu wurde eine Vergleichsmatrix mit den Punkten des dpKa-Modelles und den interpretierten Ergebnissen der Forschung erstellt. Die Interpretationen wurden mit dem Material abgesichert und argumentativ begründet, um in sich schlüssig zu sein. Bei der Auswertung und der damit verbundenen Interpretationen der Ergebnisse bezüglich der dpKa-Modell-Punkte im vorigen Kapitel, wurde darauf geachtet, dass ein vielschichtiges Bild, vor allem durch Beispiele der Befragten gezeichnet wurde und keine Quantifizierung der Aussagen stattgefunden hat.

Die im Folgenden dargestellte, sogenannte Vergleichsmatrix besteht im Grunde aus den Punkten des dpKa-Modells, wobei die einzelnen Punkte jeweils in einer Tabelle dargestellt sind, in der eben zum Vergleich auch eingetragen ist, ob durch die Betrachtung des Unterrichtskonzeptes der Erwerb des Punktes erwartet wurde, ob die Fragebogenantworten der SchülerInnen auf einen Erwerb gedeutet haben, ob die interviewbasierte Forschung einen Erwerb innerhalb der Spielentwicklung mit Unity nachweist und ob der jeweilige Punkt laut interviewbasierter Forschung eine Voraussetzung für die erfolgreiche Spieleentwicklung mit Unity darstellt. Das Unterrichtskonzept wurde zusätzlich noch aufgeteilt in „Vorbereitung“ und „Unity“, wobei „Vorbereitung“ alle vorbereitenden Tätigkeiten meint, die im derzeitigen Unterrichtskonzept durchgeführt werden, wie zum Beispiel die Dateimanagementübungen, die Spielfigurenerstellung in GIMP und die Programmierübungen mit Scratch und mit Microsoft Visual Studio Express 2010. Der Bereich „Unity“ hingegen betrifft jene Punkte die wirklich aus den Arbeiten mit Unity herausgelesen wurden.

Bei einer entsprechenden Zustimmung wurde in die darunterliegende Zelle „Ja“ geschrieben und sie wurde ausgefüllt, bei einer Ablehnung wurde „Nein“ hineingeschrieben und wenn der Punkt diesbezüglich unbeantwortet ist, wurde die Zelle leer gelassen. Sehen wir uns das Ganze an einem fiktiven Beispiel an:

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja	Ja		Ja	Nein
<b>Prog.W.Objektorientierung</b> - Ich kann die Objektorientierung erklären.				

Hier geht es um den erfundenen Punkt „Prog.W.Objektorientierung“, laut dem man die Objektorientierung erklären kann.

Laut dieser Tabelle hat sich nach der Durchsicht des Unterrichtskonzeptes herausgestellt, dass man diesen Kompetenzbaustein voraussichtlich bei den vorbereitenden Tätigkeiten (Vorbereitung) erwerben wird, aber auch, wenn man nur die Aufgaben innerhalb von Unity durchmacht (Unity). Vorsicht, es handelt sich bei den Einträgen „Vorbereitung“ und „Unity“ nur um Hinweise und man darf diese nicht in Kombination mit dem Eintrag „Erwerb“ sehen, den dieser stellt nur den nachgewiesenen Erwerb innerhalb von Unity dar und hat nichts mit den Einträgen des Unterrichtskonzeptes zu tun.

Im Bereich „Fragebogen“ wurde nichts eingetragen, was bedeutet, dass dieser Punkt vom Fragebogen nicht behandelt wurde oder keine Aussagekraft hat, sodass dieser Punkt in Bezug auf den Fragebogen unbeantwortet bleibt.

Der Eintrag „Erwerb“ im Bereich „Interview“ stellt also tatsächlich den durch die Interviews nachgewiesenen Erwerb des Punktes „Prog.W.Objektorientierung“ innerhalb von Unity dar.

Bei „Voraussetzung“ wurde „Nein“ eingetragen, da sich anhand des Interviewmaterials herausgestellt hat, dass der Punkt „Prog.W.Objektorientierung“ keine Voraussetzung für eine erfolgreiche Spieleentwicklung mit Unity darstellt. Es folgen nun die tatsächlichen Punkte.

6.4.1 Verantwortung, Datenschutz und Datensicherheit

**Dk12 – 1.2 Verantwortung, Datenschutz und Datensicherheit**

**Wissen und Verstehen (Wissen):**

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
			Nein	Nein
<p><b>Recht.W.Daten - 1.2.1</b> Ich kann meine Rechte und Pflichten in der Nutzung von Informatiksystemen beschreiben und wesentliche Aspekte des Datenschutzes und der Datensicherheit erklären. <b>[DC 2.5, 3.3, 4.1, 4.2], [Dk8 2.4.2]</b>                      SchülerInnen kennen die Grundzüge des Urheberrechts sowie des Datenschutzes (insb. das Recht am eigenen Bild) ...</p>				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
			Nein	Nein
<p><b>Recht.W.Lizenz - Dk8 2.4.3</b> SchülerInnen kennen Lizenzmodelle, insb. Offene (Creative Commons, Open Educational Resources). <b>[DC 3.3]</b></p>				

**Anwenden und Gestalten (Fertigkeiten):**

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
			Nein	Nein
<p><b>Recht.A.Daten - 1.2.2</b> Ich kann beim Einsatz von Informatiksystemen mein Wissen um Pflichten und Rechte in Bezug auf meine Person und meine Arbeitsumgebung, auf persönliche und fremde Daten verantwortungsbewusst anwenden. <b>[DC 2.5, 4.1, 4.2] [Dk8 2.4.2]</b> ... und wenden diese Bestimmungen [bzgl. Urheberrecht und Datenschutz] an.</p>				



## 6.4.2 Technische Grundlagen und Funktionsweisen

## Dk12 – 2.1 Technische Grundlagen und Funktionsweisen

## Wissen und Verstehen (Wissen):

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
			Ja	Ja
<p><b>Technik.W.Computer - 2.1.1</b> Ich kann Komponenten von Informatiksystemen beschreiben und ihre Funktionsweise und ihr Zusammenwirken erklären. [Dk8 7.1.1] SchülerInnen kennen die Bestandteile und Funktionsweise eines Computers und eines Netzwerks</p>				

## Anwenden und Gestalten (Fertigkeiten):

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja			Ja	Ja
<p><b>Technik.A.Computer - 2.1.3</b> Ich kann ein Computersystem samt Peripheriegeräten sachgerecht nutzen. [DC 0.2, 5.2] [Dk8 3.1.1] SchülerInnen nutzen die zum Normalbetrieb notwendigen Funktionen eines Betriebssystems einschließlich des <u>Dateimanagements</u> sowie der Druckfunktion. [Dk8 7.2.3] SchülerInnen nutzen unterschiedliche digitale Geräte entsprechend ihrer Einsatzmöglichkeiten</p>				

## Reflektieren und Bewerten (Handlungsbereitschaft):

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
				Nein
<p><b>Technik.R.Computer - 2.1.5</b> Ich kann unterschiedliche digitale Endgeräte bzw. Informatiksysteme in Bezug auf ihre technischen Eigenschaften und ihre Leistungsfähigkeit bewerten. [DC 5.2]</p>				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja	Ja		Ja	Nein
<b>Technik.R.Fehler - 2.1.6</b> Ich kann einfache Fehler diagnostizieren und beheben. [DC 5.1] [Dk8 7.3.1] SchülerInnen erkennen technische Probleme in der Nutzung von digitalen Geräten und melden eine konkrete Beschreibung des Fehlers an die richtigen Stellen				

### 6.4.3 Betriebssysteme und Software

#### Dk12 – 2.2 Betriebssysteme und Software

##### Wissen und Verstehen (Wissen):

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja	Ja		Ja	Ja
<b>Software.W.Arten - 2.2.2</b> Ich kann Kategorien von Software nennen und deren Anwendung beschreiben. [DC 5.2] [Dk8 7.1.2] SchülerInnen kennen gängige proprietäre und offene Anwendungsprogr. und zugehörige Dateitypen.				

##### Anwenden und Gestalten (Fertigkeiten):

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
			Ja	Ja
<b>Software.A.Arten - 2.2.4</b> Ich kann mich in die Bedienung für mich neuer Software selbstständig einarbeiten. [DC 0.2]				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja			Ja	Nein
<b>Software.A.Text - Dk8 3.2.1</b> SchülerInnen geben Texte zügig ein, [DC 0.2]				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja			Ja	Nein
<b>Software.A.Korrektur - Dk8 3.2.3</b> SchülerInnen führen Textkorrekturen durch (inkl. Zuhilfenahme von Überarbeitungsfunktionen, Rechtschreibprüf.). <b>[DC 0.2]</b>				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
				Nein
<b>Software.A.Hilfe - Dk8 7.3.2</b> SchülerInnen nutzen Hilfesysteme bei der Problemlösung				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
			Ja	Ja
<b>Software.A.Sicherung - Dk8 7.3.3</b> SchülerInnen führen Datensicherungen und -Wiederherstellungen aus.				

#### 6.4.4 Produktion digitaler Medien

### Dk12 – 3.1 Produktion digitaler Medien

#### Wissen und Verstehen (Wissen):

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja			Ja	Nein
<b>Medien.W.Formate - 3.1.1</b> Ich kann gängige Medienformate und ihre Eigenschaften beschreiben. <b>[DC 3.1]</b>				

**Anwenden und Gestalten (Fertigkeiten):**

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja	Ja		Ja	Nein
<p><b>Medien.A.Formate - 3.1.3</b> Ich kann digitale Medien in Form von Text, Ton, Bildern und Filmen sachgerecht bearbeiten, produzieren und publizieren. <b>[DC 3.1], [Dk8 4.2.2]</b> SchülerInnen gestalten digitale Medien mittels aktueller Technologien, ggf. unter Einbeziehung anderer Medien: Texte, Präsentationen, Audio-, Video-beiträge sowie multimediale Lernmaterialien</p>				

**Reflektieren und Bewerten (Handlungsbereitschaft):**

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
			Ja	Nein
<p><b>Medien.R.Formate - 3.1.4</b> Ich kann digitale Produkte (Artefakte) in Bezug auf inhaltliche Relevanz, Wirkung und Design einschätzen und bewerten.</p>				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja	Ja		Ja	Nein
<p><b>Medien.R.Selbstwirksamkeit - Dk8 4.2.1</b> SchülerInnen erleben sich selbstwirksam, indem sie digitale Technologien kreativ und vielfältig nutzen</p>				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
				Nein
<p><b>Medien.R.Techniken - Dk8 4.2.5</b> SchülerInnen setzen Wissen über Techniken und Ästhetiken populärer Medienkulturen eigenverantwortlich um, <b>[DC 3.1]</b></p>				

## 6.4.5 Suche, Auswahl und Organisation von Information

## Dk12 – 3.3 Suche, Auswahl und Organisation von Information

## Wissen und Verstehen (Wissen):

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja	Ja		Ja	Nein
<b>Info.W.SucheQuellen - 3.3.1</b> Ich kann wichtige Informationsquellen im Internet anführen, die für meine schulischen und privaten Informationsbedürfnisse nützlich und notwendig sind. [DC 5.2, 5.3]				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
	Ja		Ja	Nein
<b>Info.W.SucheMethoden - 3.3.2</b> Ich kann lokal und in Netzwerken Methoden der Informationsgewinnung und -organisation benennen. [DC 1.1, 5.3]				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
				Nein
<b>Info.W.Wissensmanagement - 3.3.3</b> Ich kann Möglichkeiten grundlegenden digitalen Wissensmanagements beschreiben.				

## Anwenden und Gestalten (Fertigkeiten):

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
			Ja	Nein
<b>Info.A.SucheBedarf - Dk8 2.1.1</b> SchülerInnen formulieren ihre Bedürfnisse für die Informationssuche, [DC 1.1]				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
	Ja		Ja	Nein
<p><b>Info.A.Suche - 3.3.4</b> Ich kann unter Verwendung passender Dienste und Angebote und Wahl geeigneter Suchmethoden Informationen und Medien gezielt suchen und auswählen. [DC 1.1, 5.3], [Dk8 2.1.2] SchülerInnen planen zielgerichtet und selbstständig die Suche nach Informationen, Daten und digitalen Inhalten mit Hilfe geeigneter Strategien und Methoden (z. B. Suchbegriffe), passender Werkzeuge bzw. nützlicher Quellen.</p>				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
				Nein
<p><b>Info.A.Wissensmanagement - 3.3.5</b> Ich kann im Rahmen persönlichen Lernmanagements Informationen und Medien strukturiert speichern und verfügbar halten., [DC 1.3], [Dk8 2.3.1] SchülerInnen speichern Informationen, Daten und digitale Inhalte sowohl im passenden Format als auch in einer sinnvollen Struktur, in der diese gefunden und verarbeitet werden können.</p>				

---

### 6.4.6 Algorithmen, Datenstrukturen und Programmierung

#### Dk12 – 4.2 Algorithmen, Datenstrukturen und Programmierung

##### Wissen und Verstehen (Wissen):

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja	Ja	Ja	Ja	Ja
<p><b>Prog.W.Datenstruktur - [KsmOOP 1]</b> Ich kann unterschiedliche Datenstrukturen beschreiben <b>5.1 data structure (graph, tree, array)</b></p>				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
		Ja	Ja	Nein
<b>Prog.W.Algorithmus - 4.2.1</b> Ich kann den Algorithmus-begriff erklären.				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja	Ja	Ja	Ja	Nein
<b>Prog.W.Modellierung - 4.2.2</b> Ich kann Aufgaben und Problemstellungen algorithmisch und formalsprachlich in geeigneten Datenstrukturen beschreiben. [DC 3.4], [Dk8 8.1.4] SchülerInnen formulieren eindeutige Handlungsanleitungen (Algorithmen) verbal und schriftlich [siehe auch 4.2.6]				
<b>5.3 algorithmic structure (loops, conditional statement)</b>				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
	Ja	Ja	Ja	Nein
<b>Prog.W.Software - 4.2.3</b> Ich kann wesentliche Aspekte und Methoden der Softwareentwicklung und des Softwareprojektmanagements erklären. <b>5.4 notional machine (data, working memory, processor, statement, program, automation)</b>				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja	Ja		Ja	Nein
<b>Prog.W.Aspekte - 4.2.4</b> Ich kann wesentliche Aspekte der <i>Prozeduralen, Funktionalen und Objektorientierten</i> [sic!] Programmierung nennen und an Beispielen erläutern. <b>5.2 class &amp; object structure (object, attribute, association)</b>				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja	Ja		Ja	Ja

**Prog.W.Sprache - Dk8 8.2.2** SchülerInnen kennen unterschiedliche Programmiersprachen und Produktionsabläufe. **6 Mastering representation (language, syntax, semantics)**

**Anwenden und Gestalten (Fertigkeiten):**

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja	Ja	Ja	Ja	Ja

**Prog.A.Sprache - Dk8 8.1.2** SchülerInnen verwenden, erstellen [und reflektieren] Codierungen (z. B. Geheimschrift, QR-Code), **[DC 0.1] 6 Mastering representation (language, syntax, semantics)**

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
			Ja	Nein

**Prog.A.Modellierung - 4.2.5** Ich kann Aufgaben mit Mitteln der Informatik modellieren.

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja	Ja	Ja	Ja	Ja

**Prog.A.Algorithmus - 4.2.6** Ich kann Algorithmen entwerfen, diese formal darstellen, implementieren und testen. **[DC 3.4], [Dk8 8.1.4]** SchülerInnen formulieren eindeutige Handlungsanleitungen (Algorithmen) verbal und schriftlich [siehe auch 4.2.2] **[Dk8 8.2.3]** SchülerInnen beherrschen grundlegende Programmierstrukturen (Verzweigung, Schleifen, Prozeduren). **5.3 algorithmic structure (loops, conditional statement)**



Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
	Ja		Ja	Nein
<b>Prog.A.Datenstruktur - [KsmOOP 2]</b> Ich kann Datenstrukturen, sowie Klassen- und Objektstrukturen in einem Algorithmus verwenden <b>5.1 data structure (graph, tree, array), 5.2 class &amp; object structure (object, attribute, association)</b>				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
	Ja	Ja	Ja	Nein
<b>Prog.A.Software - 4.2.7</b> Ich kann ein Softwareprojekt planen und durchführen. <b>[Dk8 8.2.1]</b> SchülerInnen erstellen einfache Programme oder Webanwendungen mit geeigneten Tools, um ein bestimmtes Problem zu lösen oder eine bestimmte Aufgabe zu erfüllen, <b>[DC 5.3] 5.4 notional machine (data, working memory, processor, statement, program, automation)</b>				

**Reflektieren und Bewerten (Handlungsbereitschaft):**

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
			Ja	Nein
<b>Prog.R.Software - 4.2.8</b> Ich kann die Schritte der Softwareentwicklung reflektieren.				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
Ja				Nein
<b>Prog.R.SoftwareWerkzeug - 4.2.9</b> Ich kann die Angemessenheit der Entwicklungswerkzeuge grob einschätzen.				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
			Ja	Nein
<b>Prog.R.AlgorithmusMuster - Dk8 8.1.5</b> SchülerInnen entdecken Gemeinsamkeiten und Regeln (Muster) in Handlungsanleitungen, [DC 0.1]				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
		Ja	Ja	Nein
<b>Prog.R.AlgorithmusBedeutung - Dk8 8.1.6</b> SchülerInnen erkennen die Bedeutung von Algorithmen in automatisierten digitalen Prozessen (z. B. automatisiertes Vorschlagen von potenziell interessanten Informationen).				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
			Nein	Nein
<b>Prog.R.AlgorithmusEffizienz - 4.2.10</b> Ich kann die Effizienz von Algorithmen bewerten.				

Unterrichtskonzept		Fragebogen	Interview	
Vorbereitung	Unity		Erwerb	Voraussetzung
	Ja	Ja	Ja	Nein
<b>Prog.R.Sprache - 4.2.11</b> Ich kann gezielt nach Programmfehlern suchen und diese korrigieren. [DC 5.1]				

## 6.5 Deutung und Diskussion der Ergebnisse

In diesem Kapitel finden eine Deutung und zusammenfassende Diskussion der Ergebnisse, sowie die Beantwortung der Forschungsfragen statt.

---

### 6.5.1 Forschungsfrage 1: Voraussetzungen für Unity

Zuerst gehen wir auf die Forschungsfrage mit den Voraussetzungen ein, die wie folgt lautet:

- Welche digitalen Kompetenzen und Fertigkeiten und welches Vorwissen benötigen jugendliche Schülerinnen und Schüler, um im Unterricht erfolgreich Computerspiele mit Unity zu entwickeln?

Dazu könnte man die Frage stellen, warum man den überhaupt nach Voraussetzungen sucht, immerhin kann man doch auch direkt mit Unity beginnen und darin alles Benötigte lernen. Doch die Unterrichtszeit, die man mit den SchülerInnen zu Verfügung hat ist begrenzt und daher ist es naheliegend, dass man diese Zeit bestmöglich nutzt. Dementsprechend macht es Sinn nach Voraussetzungen zu suchen, um diese dann mit einfacheren bzw. geeigneteren Tools im Unterricht zu behandeln.

Unity stellt den drei Befragten Personen zufolge eine für SchülerInnen komplexe und herausfordernde Entwicklungsumgebung dar. Daher ist es laut B2 in Unity schwieriger, den Fokus auf bestimmte Aspekte zu setzen, wenn den SchülerInnen bei den Grundlagen zu viel fehlt. Das ganze Drumherum, wie zum Beispiel die Gameschleife und andere Unity-spezifischen Sachen machen es da schwieriger bestimmte Dinge herauszupicken, wohingegen man mit der Konsolenprogrammierung auf eine sehr fokussierte Art Operatoren beispielsweise lernen kann, indem man vielleicht einen Taschenrechner programmiert, so B2. Die Spieleentwicklung in Unity ist aber auch ohne C# bzw. Code zu schreiben möglich. Man könnte jederzeit mit Unity beginnen, indem man sie mit der Entwicklungsumgebung vertraut macht und ein bestehendes Spiel modifiziert, wobei man zum Beispiel Parameter und dadurch den Schwierigkeitsgrad des Spieles verändert, oder ein neues, kleines Feature

einbaut. Dazu müssen sie nur lesen können. Man kann ihnen auch nur erklären, wie man sich ein Spiel zusammenklickt oder man lässt sie Bilder machen und fügt diese ins Spiel ein, oder zeigt ihnen nur wie 2D-Animationen funktionieren. Dann kann man Unity hernehmen, ohne dass sie dafür eine einzige Zeile Code schreiben müssen. B2 sagt weiter, dass man also jederzeit mit Unity anfangen könnte und sich nur die Frage stellen muss, wie man es aufziehen will und wieviel Zeit man bereit ist zu investieren.

Wie man sieht, kann man also auch ohne Vorkenntnisse und Fertigkeiten mit Unity starten. Derselben Meinung ist auch B1. Auf die Frage hin, ob man dann noch effektiv bzw. erfolgreich mit Unity Spiele programmieren kann, oder ob sich dann das Projekt verzögert, antwortet B1 damit, dass sich das Projekt dadurch erheblich verzögert und es dadurch länger dauern kann, als wenn man vorbereitende Maßnahmen setzt, weil dann das Lernen in so einem Spannungsgeflecht ist, dass es nicht so leicht stattfinden kann. B1 erklärt das an einem Beispiel, nämlich dass man das bei neuen Sportarten oder beim Bewegungslernen sieht. Wenn man da methodisch heran geht, ist das oft hilfreicher, um etwas technisch richtig zu lernen, als wenn man völlig chaotisch herangeht, also alles gleichzeitig irgendwie lernt und dann schaut was man braucht, denn dann kann es sein, dass sich das gegenseitig hemmt. Genau aus diesem Grund soll untersucht werden, welche Voraussetzungen man benötigt, um erfolgreich Spiele mit Unity zu programmieren. Dementsprechend lautete die Forschungsfrage auch so:

- Welche digitalen Kompetenzen und Fertigkeiten und welches Vorwissen benötigen jugendliche Schülerinnen und Schüler, um im Unterricht erfolgreich Computerspiele mit Unity zu entwickeln?

Die Antwort darauf wird nun durch Auflistung der entsprechenden dpKa-Modell-Punkte dargestellt, wobei sie in der Reihenfolge des digi.komp12 Kompetenzmodelles aufgelistet werden und die abgedeckten Punkte des digi.komp8 und DigComp2.2AT ebenfalls zu sehen sind.

## Dk12 – 2.1 Technische Grundlagen und Funktionsweisen

### Wissen und Verstehen (Wissen):

- **Dk12 2.1.1** Ich kann Komponenten von Informatiksystemen beschreiben und ihre Funktionsweise und ihr Zusammenwirken erklären. **[Dk8 7.1.1]** SchülerInnen kennen die Bestandteile und Funktionsweise eines Computers und eines Netzwerks

### Anwenden und Gestalten (Fertigkeiten):

- **Dk12 2.1.3** Ich kann ein Computersystem samt Peripheriegeräten sachgerecht nutzen. **[DC 0.2, 5.2] [Dk8 3.1.1]** SchülerInnen nutzen die zum Normalbetrieb notwendigen Funktionen eines Betriebssystems einschließlich des Dateimanagements sowie der Druckfunktion. **[Dk8 7.2.3]** SchülerInnen nutzen unterschiedliche digitale Geräte entsprechend ihrer Einsatzmöglichkeiten

## Dk12 – 2.2 Betriebssysteme und Software

### Wissen und Verstehen (Wissen):

- **Dk12 2.2.2** Ich kann Kategorien von Software nennen und deren Anwendung beschreiben. **[DC 5.2] [Dk8 7.1.2]** SchülerInnen kennen gängige proprietäre und offene Anwendungsprogramme und zugehörige Dateitypen.

### Anwenden und Gestalten (Fertigkeiten):

- **Dk12 2.2.4** Ich kann mich in die Bedienung für mich neuer Software selbstständig einarbeiten. **[DC 0.2]**
- **Dk8 7.3.3** SchülerInnen führen Datensicherungen und -Wiederherstellungen aus.

## Dk12 – 4.2 Algorithmen, Datenstrukturen und Programmierung

### Wissen und Verstehen (Wissen):

- **[KsmOOP 1]** Ich kann unterschiedliche Datenstrukturen beschreiben **5.1 data structure (graph, tree, array)**
- **Dk8 8.2.2** SchülerInnen kennen unterschiedliche Programmiersprachen und Produktionsabläufe. **6 Mastering representation (language, syntax, semantics)**

### Anwenden und Gestalten (Fertigkeiten):

- **Dk8 8.1.2** SchülerInnen verwenden, erstellen [und reflektieren] Codierungen (z. B. Geheimschrift, QR-Code), **[DC 0.1] 6 Mastering representation (language, syntax, semantics)**
- **Dk12 4.2.6** Ich kann Algorithmen entwerfen, diese formal darstellen, implementieren und testen. **[DC 3.4], [Dk8 8.1.4]** SchülerInnen formulieren eindeutige Handlungsanleitungen (Algorithmen) verbal und schriftlich [siehe auch 4.2.2] **[Dk8 8.2.3]** SchülerInnen beherrschen grundlegende Programmierstrukturen (Verzweigung, Schleifen, Prozeduren). **5.3 algorithmic structure (loops, conditional statement)**

Wie man an obiger Auflistung und somit den Ergebnissen der Studie sieht, kommen nur Kompetenzbausteine aus diesen drei Bereichen vor, wobei keiner der Kompetenzbausteine aus einer entsprechenden dritten Ebene „Reflektieren und Bewerten (Handlungsbereitschaft)“ kommt. Daher benötigt man als Voraussetzung für die erfolgreiche Spieleentwicklung mit Unity nur Wissen und Fertigkeiten, aber keine vollständige Kompetenz, da dazu ein entsprechender Kompetenzbaustein im Bereich des Reflektierens und Bewertens nötig wäre.

Dieses Ergebnis stellt keine wirkliche Überraschung dar, da zu erwarten war, dass man mit dem Computer, dem Betriebssystem und Anwendungsprogrammen umgehen können soll. Interessant ist vor allem, dass im Bereich der Programmierung Vorwissen und Fertigkeiten gefragt sind. Diese Voraussetzungen sollen daher auch mit den vorbereitenden Maßnahmen erworben werden, die genau dafür im Unterrichtskonzept eingesetzt werden. Ob diese Voraussetzungen in den vorbereitenden Tätigkeiten auch erworben werden, wurde allerdings nicht untersucht und wäre daher mit einer entsprechenden Forschungsfrage zu untersuchen.

---

### 6.5.2 Forschungsfrage 2: Kompetenzerwerb in Unity

Die zweite Forschungsfrage lautet:

- Welche digitalen Kompetenzen, Fertigkeiten und welches Wissen werden in der Computerspieleentwicklung mit Unity erworben?

Auch diese Frage wird mit der Auflistung der entsprechenden dpKa-Modell-Punkte beantwortet, wobei man wieder die Punkte der Kompetenzmodelle digi.komp12, digi.komp8 und DigComp2.2AT sieht. Dabei sind Punkte aufgelistet, die auch unter Voraussetzungen fallen. Das bedeutet, dass diese Punkte für eine erfolgreiche Spieleentwicklung mit Unity erforderlich sind und durch Unity verbessert werden, oder wenn sie eben noch nicht vorhanden sind, durch Unity angeeignet werden können. Jene Punkte, die nicht nur erworben werden können, sondern gleichzeitig auch eine Voraussetzung darstellen und somit, wenn bereits vorhanden lediglich verbessert werden können, wurden mit

dem Tag „**Voraussetzung**“ gekennzeichnet. Nun zu den Kompetenzbausteinen:

## Dk12 – 2.1 Technische Grundlagen und Funktionsweisen

### Wissen und Verstehen (Wissen):

- **Voraussetzung** Dk12 2.1.1 Ich kann Komponenten von Informatiksystemen beschreiben und ihre Funktionsweise und ihr Zusammenwirken erklären. [Dk8 7.1.1] SchülerInnen kennen die Bestandteile und Funktionsweise eines Computers und eines Netzwerks

### Anwenden und Gestalten (Fertigkeiten):

- **Voraussetzung** Dk12 2.1.3 Ich kann ein Computersystem samt Peripheriegeräten sachgerecht nutzen. [DC 0.2, 5.2] [Dk8 3.1.1] SchülerInnen nutzen die zum Normalbetrieb notwendigen Funktionen eines Betriebssystems einschließlich des Dateimanagements sowie der Druckfunktion. [Dk8 7.2.3] SchülerInnen nutzen unterschiedliche digitale Geräte entsprechend ihrer Einsatzmöglichkeiten

### Reflektieren und Bewerten (Handlungsbereitschaft):

- Dk12 2.1.6 Ich kann einfache Fehler diagnostizieren und beheben. [DC 5.1] [Dk8 7.3.1] SchülerInnen erkennen technische Probleme in der Nutzung von digitalen Geräten und melden eine konkrete Beschreibung des Fehlers an die richtigen Stellen.

## Dk12 – 2.2 Betriebssysteme und Software

### Wissen und Verstehen (Wissen):

- **Voraussetzung** Dk12 2.2.2 Ich kann Kategorien von Software nennen und deren Anwendung beschreiben. [DC 5.2] [Dk8 7.1.2] SchülerInnen kennen gängige proprietäre und offene Anwendungsprogramme und zugehörige Dateitypen.

### Anwenden und Gestalten (Fertigkeiten):

- **Voraussetzung** Dk12 2.2.4 Ich kann mich in die Bedienung für mich neuer Software selbstständig einarbeiten. [DC 0.2]
- Dk8 3.2.1 SchülerInnen geben Texte zügig ein, [DC 0.2]
- Dk8 3.2.3 SchülerInnen führen Textkorrekturen durch (inkl. Zuhilfenahme von Überarbeitungsfunktionen, Rechtschreibprüfung). [DC 0.2]
- **Voraussetzung** Dk8 7.3.3 SchülerInnen führen Datensicherungen und -Wiederherstellungen aus.

## Dk12 – 3.1 Produktion digitaler Medien

### Wissen und Verstehen (Wissen):

- Dk12 3.1.1 Ich kann gängige Medienformate und ihre Eigenschaften beschreiben. [DC 3.1]

### Anwenden und Gestalten (Fertigkeiten):

- **Dk12 3.1.3** Ich kann digitale Medien in Form von Text, Ton, Bildern und Filmen sachgerecht bearbeiten, produzieren und publizieren. **[DC 3.1]**, **[Dk8 4.2.2]** SchülerInnen gestalten digitale Medien mittels aktueller Technologien, ggf. unter Einbeziehung anderer Medien: Texte, Präsentationen, Audio-, Video-beiträge sowie multimediale Lernmaterial.

### Reflektieren und Bewerten (Handlungsbereitschaft):

- **Dk12 3.1.4** Ich kann digitale Produkte (Artefakte) in Bezug auf inhaltliche Relevanz, Wirkung und Design einschätzen und bewerten.
- **Dk8 4.2.1** SchülerInnen erleben sich selbstwirksam, indem sie digitale Technologien kreativ und vielfältig nutzen,

## Dk12 – 3.3 Suche, Auswahl und Organisation von Information

### Wissen und Verstehen (Wissen):

- **Dk12 3.3.1** Ich kann wichtige Informationsquellen im Internet anführen, die für meine schulischen und privaten Informationsbedürfnisse nützlich und notwendig sind. **[DC 5.2, 5.3]**
- **Dk12 3.3.2** Ich kann lokal und in Netzwerken Methoden der Informationsgewinnung und -organisation benennen. **[DC 1.1, 5.3]**

### Anwenden und Gestalten (Fertigkeiten):

- **Dk8 2.1.1** SchülerInnen formulieren ihre Bedürfnisse für die Informationssuche, **[DC 1.1]**
- **Dk12 3.3.4** Ich kann unter Verwendung passender Dienste und Angebote und Wahl geeigneter Suchmethoden Informationen und Medien gezielt suchen und auswählen. **[DC 1.1, 5.3]**, **[Dk8 2.1.2]** SchülerInnen planen zielgerichtet und selbstständig die Suche nach Informationen, Daten und digitalen Inhalten mit Hilfe geeigneter Strategien und Methoden (z. B. Suchbegriffe), passender Werkzeuge bzw. nützlicher Quellen.

## Dk12 – 4.2 Algorithmen, Datenstrukturen und Programmierung

### Wissen und Verstehen (Wissen):

- **[Voraussetzung] [KsmOOP 1]** Ich kann unterschiedliche Datenstrukturen beschreiben **5.1 data structure (graph, tree, array)**
- **Dk12 4.2.1** Ich kann den Algorithmus-begriff erklären.
- **Dk12 4.2.2** Ich kann Aufgaben und Problemstellungen algorithmisch und formalsprachlich in geeigneten Datenstrukturen beschreiben. **[DC 3.4]**, **[Dk8 8.1.4]** SchülerInnen formulieren eindeutige Handlungsanleitungen (Algorithmen) verbal und schriftlich [siehe auch 4.2.6] **5.3 algorithmic structure (loops, conditional statement)**
- **Dk12 4.2.3** Ich kann wesentliche Aspekte und Methoden der Softwareentwicklung und des Softwareprojektmanagements erklären. **5.4 notional machine (data, working memory, processor, statement, program, automation)**



- **Dk12 4.2.4** Ich kann wesentliche Aspekte der *Prozeduralen, Funktionalen und Objektorientierten* [sic!] Programmierung nennen und an Beispielen erläutern. **5.2 class & object structure (object, attribute, association)**
- **[Voraussetzung] Dk8 8.2.2** SchülerInnen kennen unterschiedliche Programmiersprachen und Produktionsabläufe. **6 Mastering representation (language, syntax, semantics)**

**Anwenden und Gestalten (Fertigkeiten):**

- **[Voraussetzung] Dk8 8.1.2** SchülerInnen verwenden, erstellen [und reflektieren] Codierungen (z. B. Geheimschrift, QR-Code), **[DC 0.1] 6 Mastering representation (language, syntax, semantics)**
- **Dk12 4.2.5** Ich kann Aufgaben mit Mitteln der Informatik modellieren.
- **[Voraussetzung] Dk12 4.2.6** Ich kann Algorithmen entwerfen, diese formal darstellen, implementieren und testen. **[DC 3.4], [Dk8 8.1.4]** SchülerInnen formulieren eindeutige Handlungsanleitungen (Algorithmen) verbal und schriftlich [siehe auch 4.2.2] **[Dk8 8.2.3]** SchülerInnen beherrschen grundlegende Programmierstrukturen (Verzweigung, Schleifen, Prozeduren). **5.3 algorithmic structure (loops, conditional statement)**
- **[KsmOOP 2]** Ich kann Datenstrukturen, sowie Klassen- und Objektstrukturen in einem Algorithmus verwenden **5.1 data structure (graph, tree, array), 5.2 class & object structure (object, attribute, association)**
- **Dk12 4.2.7** Ich kann ein Softwareprojekt planen und durchführen. **[Dk8 8.2.1]** SchülerInnen erstellen einfache Programme oder Webanwendungen mit geeigneten Tools, um ein bestimmtes Problem zu lösen oder eine bestimmte Aufgabe zu erfüllen, **[DC 5.3] 5.4 notional machine (data, working memory, processor, statement, program, automation)**

**Reflektieren und Bewerten (Handlungsbereitschaft):**

- **Dk12 4.2.8** Ich kann die Schritte der Softwareentwicklung reflektieren.
- **Dk8 8.1.5** SchülerInnen entdecken Gemeinsamkeiten und Regeln (Muster) in Handlungsanleitungen, **[DC 0.1]**
- **Dk8 8.1.6** SchülerInnen erkennen die Bedeutung von Algorithmen in automatisierten digitalen Prozessen (z. B. automatisiertes Vorschlagen von potenziell interessanten Informationen).
- **Dk12 4.2.11** Ich kann gezielt nach Programmfehlern suchen und diese korrigieren. **[DC 5.1]**

Der Schwerpunkt der Betrachtung von zu erwerbenden Kompetenzbausteinen in Unity lag im Bereich „Algorithmen, Datenstrukturen und Programmierung“. Aber wie sich herausgestellt hat, können auch viele Kompetenzbausteine aus anderen Bereichen, nämlich aus „Technische Grundlagen und

Funktionsweisen“, „Betriebssysteme und Software“, „Produktion digitaler Medien“ und „Suche, Auswahl und Organisation von Information“ erworben werden. Die Betonung liegt hier auf „können“ in Bezug auf den Erwerb, denn es gibt immer wieder SchülerInnen in den Klassen oder Gruppen, die sich für ein Fach bzw. Projekt wenig bis gar nicht interessieren und daher auch nicht viel erwerben werden. Aber die Masse der SchülerInnen sollte durch die Spieleentwicklung mit Unity diese Kompetenzbausteine mehr oder weniger stark ausgeprägt erwerben oder wenn etwas schon vorhanden war verbessern.

Besonders spannend ist, dass die SchülerInnen in einigen Bereichen sogar tatsächlich eine Kompetenz erwerben, da sie sich Kompetenzbausteine aus allen drei Kompetenzstufen aneignen und somit über das Wissen und Fertigkeiten in einem Bereich hinausgehen, indem sie zusätzlich die Handlungsbereitschaft erwerben. Das ist zum Beispiel im Bereich „Technische Grundlagen und Funktionsweisen“ durch den Punkt Dk12 2.1.6 der Fall, den sie durch die Spielentwicklung mit Unity erwerben. Im Bereich „Produktion digitaler Medien“ erwerben sie auch durch Unity eine ganze Kompetenz. Bei „Suche, Auswahl und Organisation von Information“ hingegen schafft man es nur bis zur Kompetenzstufe der Fertigkeiten. Der letzte große Bereich „Algorithmen, Datenstrukturen und Programmierung“ ist wiederum mit dem Erwerb von Kompetenzen gekennzeichnet. Der Plural „Kompetenzen“ wird hier deshalb verwendet, weil es darin viele Teilbereiche mit zusammenhängenden Kompetenzbausteinen aus unterschiedlichen Kompetenzstufen gibt, die jeweils gemeinsam, also wenn man in dem Teilbereich alle drei Kompetenzstufen erreicht hat, eine Kompetenz darstellen. Ein so ein Teilbereich ist zum Beispiel „Softwareprojekte“, der sich aus den Punkten „Dk12 4.2.3“ der Kompetenzstufe „Wissen und Verstehen (Wissen)“, „Dk12 4.2.7“ der Kompetenzstufe „Anwenden und Gestalten (Fertigkeiten)“ und „Dk12 4.2.8“ der Kompetenzstufe „Reflektieren und Bewerten (Handlungsbereitschaft)“ zusammensetzt.

Der Erwerb von Kompetenzbausteinen aus den ersten zwei Bereichen, also „Technische Grundlagen und Funktionsweisen“ und „Betriebssysteme und

Software“ ist irgendwo am naheliegendsten, da man sich durch die Spieleentwicklung mit Unity mit dem Computer und der Software beschäftigt. Spannend sind jedoch die Kompetenzbausteinerwerbe aus den Bereichen „Produktion digitaler Medien“ und „Suche, Auswahl und Organisation von Information“, denn diese hängen möglicherweise sehr stark mit dem Softwareprojektcharakter zusammen, den die Spieleentwicklung mit Unity, insbesondere durch die selbst zu erstellenden Spiele mit sich bringt. Das drückt sich meiner Meinung nach auch im Bereich „Algorithmen, Datenstrukturen und Programmierung“ aus, indem eine vollständige Kompetenz in Bezug auf die Softwareprojektentwicklung erworben werden kann. Aber ob das tatsächlich so ist müsste man wiederum untersuchen.

---

### 6.5.3 Ausblick und Reflexion der Empirie

Sieht man sich nun die Punkte des digi.komp12 im Bereich „4.2 Algorithmen, Datenstrukturen und Programmierung“ noch einmal näher an und vergleicht diese mit den in der Spieleentwicklung mit Unity erworbenen Kompetenzbausteinen, so fällt auf, dass erwiesen wurde, dass alle bis auf zwei Punkte erworben werden. Nicht erworben werden lediglich die Punkte 4.2.9 und 4.2.10, die im folgenden Ausschnitt des digi.komp12 unterstrichen wurden.

## **4 Praktische Informatik (21 Punkte)**

### **4.1 Konzepte der Informationsverarbeitung (3 Punkte)**

- ...

### **4.2 Algorithmen, Datenstrukturen und Programmierung (11 Punkte)**

#### **Wissen und Verstehen:**

- **4.2.1** Ich kann den Algorithmus-begriff erklären.
- **4.2.2** Ich kann Aufgaben und Problemstellungen algorithmisch und formalsprachlich in geeigneten Datenstrukturen beschreiben.
- **4.2.3** Ich kann wesentliche Aspekte und Methoden der Softwareentwicklung und des Softwareprojektmanagements erklären.
- **4.2.4** Ich kann wesentliche Aspekte der Prozeduralen, Funktionalen und Objektorientierten Programmierung nennen und an Beispielen erläutern.

### Anwenden und Gestalten:

- **4.2.5** Ich kann Aufgaben mit Mitteln der Informatik modellieren.
- **4.2.6** Ich kann Algorithmen entwerfen, diese formal darstellen, implementieren und testen.
- **4.2.7** Ich kann ein Softwareprojekt planen und durchführen.

### Reflektieren und Bewerten:

- **4.2.8** Ich kann die Schritte der Softwareentwicklung reflektieren.
- **4.2.9** Ich kann die Angemessenheit der Entwicklungswerkzeuge grob einschätzen.
- **4.2.10** Ich kann die Effizienz von Algorithmen bewerten.
- **4.2.11** Ich kann gezielt nach Programmfehlern suchen und diese korrigieren.

### 4.3 Datenmodelle und Datenbanksysteme (4 Punkte)

- ...

### 4.4 Intelligente Systeme (3 Punkte)

- ...

Diese beiden unterstrichenen Punkte werden möglicherweise jedoch durch die vorbereitenden Tätigkeiten, also durch die Programmierung in Scratch, die Spielfigurenerstellung mit GIMP und/oder die Programmierung in Visual Studio Express 2010 erworben. Wenn das nicht bereits passiert, wäre das zu empfehlen.

Nun noch ein paar Worte bezüglich einer Reflexion der Empirie. In einigen Bereichen habe ich das Unterrichtsprojekt zu allgemein behandelt und dadurch den Rahmen für meine Untersuchung zu wenig eng gesetzt. Dies fällt vor allem bei den Interviewfragen auf. Auf der einen Seite soll man Ice-breaker-Fragen und Fragen die nach Antworten aus dem Alltag verlangen im Interviewleitfaden haben, auf der anderen Seite habe ich das aber scheinbar etwas zu gut in Bezug auf meine Untersuchung gemeint. Die zwei Fragen aus dem ersten Fragencluster, nämlich wodurch sich ein guter Programmierer auszeichnet und wo man das in Bezug auf digitale Kompetenzen sieht, brachte durch die Fragestellungen selbst kein verwertbares Material. Erst dadurch, dass die befragten Personen glücklicherweise auch hier schon einen Vergleich mit ihren

eigenen SchülerInnen gezogen haben, ließ sich ein wenig Material von den entsprechenden Antworten verwerten.

Da ich bisher noch nie ein Interview geführt habe, war natürlich auch während der Interviews eine gewissen Anspannung da. Das war vor allem dem geschuldet, dass ich mir immer wieder Sorgen gemacht habe, ob ich wirklich für die Studie verwertbares Material bekommen werde. Dadurch war ich immer hin- und hergerissen von dem Gedanken, ob ich nun eine Steuerungsfrage einwerfen sollte, oder doch weiter frei erzählen lasse, damit ich den Gedankenfluss der jeweils befragten Person nicht unterbreche. Ich denke, dass man da einfach mehr Übung benötigt, damit man ein besseres Gefühl dafür bekommt, wann man steuert und wann man den Gesprächsfluss weiterlaufen lässt. Bei einigen Fragen habe ich auch unerwartet lange Erzählungen erhalten, die dann teilweise auch stark abgedriftet sind. Ich habe den Erzählfluss dann aber öfter weiterlaufen lassen, da ich auf einen eventuell erklärten Zusammenhang zur Frage gewartet habe, der dann aber immer wieder Mal nicht gekommen ist. Dort wäre mehr Steuerung oder wären eventuell sogar bessere Fragen nötig gewesen.

### 7 Zusammenfassung und Schlussbetrachtung

Anhand dieser Studie wurden Voraussetzungen in Form von digitalen Kompetenzbausteinen für eine erfolgreiche Spieleentwicklung mit Unity untersucht und gefunden. Weiter wurden die Kompetenzbausteine von drei Kompetenzmodellen, nämlich dem des digi.komp12, des digi.komp8 und des DigComp2.2AT, im Bereich der Programmierung auf einen Erwerb innerhalb der Spielentwicklung mit Unity untersucht.

Da man nun die Voraussetzungen kennt, kann man das Unterrichtskonzept dahingehend adaptieren oder verbessern, dass man sich gezielt auf die Voraussetzungen konzentriert und damit den Unterricht in Bezug auf die mangelnde Zeit effektiver gestaltet. Was den Erwerb von Kompetenzen angeht, bietet sich ebenfalls eine Anpassung oder Überprüfung an, indem man die beiden Punkte des digi.komp12, deren Erwerb durch die Spieleentwicklung mit Unity nicht abgedeckt ist, nachprüft ob sie über die vorbereitenden Maßnahmen des Unterrichtskonzeptes abgedeckt sind und wenn nicht, diese dort einbaut.

Anhand der Interviewantworten konnten keine neuen Kompetenzbausteine gefunden werden, wodurch auch keine Erweiterung des dpKa-Modells durch eine induktive Kategorienbildung nötig wurde bzw. stattgefunden hat. Dementsprechend gab es auch keine Hinweise auf eventuelle Unvollständigkeiten, in Form von fehlenden Kompetenzbausteinen innerhalb des dpKa-Modells und der vier, in dieser Arbeit vorgestellten Kompetenzmodelle (digi.komp12, digi.komp8, digi.komp4 und DigComp2.2AT). Allerdings sollten meiner Meinung nach diese vier Kompetenzmodelle bis auf den digi.komp12 in der Wahlpflichtfach-Version noch einmal überarbeitet werden, sodass alle über die Kompetenzstufen Wissen, Fertigkeit und Handlungsbereitschaft strukturiert sind. Dadurch wären nicht nur die einzelnen Kompetenzbausteine einfacher zu definieren bzw. zu beschreiben, sondern man würde auch im Sinne von Blooms erweiterter Taxonomie sehen, auf welcher Stufe man sich befindet und ob man bereits von einer Kompetenz in einem Bereich sprechen kann.

Es sei noch gesagt, dass ich nicht vorhatte, so eine umfangreiche Diplomarbeit zu schreiben, ganz besonders deshalb, weil das Diplomstudium am Auslaufen war. Die Deadline hat sich aber gegen Ende der Arbeit aufgrund des Coronavirus nach hinten verschoben. Leider habe ich mich jedoch, rückblickend viel zu lange und zu genau mit den Definitionen der Begriffe beschäftigt. Obwohl ich schon der Meinung bin, dass es gut ist, wenn man sich mit den Begriffen, mit denen man hantiert gut auskennt und für ein gemeinsames Verständnis definiert, so habe ich das in dieser Arbeit trotzdem etwas übertrieben. Auch das Zustandekommen der Kompetenzmodelle war zu ausführlich. Zudem konnte ich nicht einschätzen wie umfangreich und aufwändig ein Experteninterview, angefangen von der Leitfadenerstellung, über die Transkription, bis hin zur Auswertung sein würde. Das liegt vor allem daran, dass ich zuvor noch nie ein Interview geführt habe.

Abschließend möchte ich noch anmerken, dass die Diplomarbeit ein großartiges und herausforderndes Projekt war und mich vor allem das Interesse am Thema der Diplomarbeit und dem damit verbundenen Projekt „learn to proGrAME“ bis zum Schluss motiviert hat.

## 8 Abbildungsverzeichnis

Abbildung 1: DIKW-Pyramide (Matthew.viel) Quelle: <a href="https://commons.wikimedia.org/wiki/File%3AKM_Pyramid_Adaptation.png">https://commons.wikimedia.org/wiki/File%3AKM_Pyramid_Adaptation.png</a> ....	23
Abbildung 2: Schlüsselqualifikationen (Reibold und Regier 2009, S. 91–92) ...	33
Abbildung 3: Konzept-Dummie (Eigene Darstellung) .....	35
Abbildung 4: Wissen, Qualifikation, Kompetenz. Screenshot aus dem pdf-Dokument im Internet. Quelle: <a href="http://www.ciando.com/img/books/extract/3799263675_lp.pdf">http://www.ciando.com/img/books/extract/3799263675_lp.pdf</a> (Heyse und Erpenbeck 2010, XI) .....	37
Abbildung 5: Fünf Kompetenzbereiche der "digitalen Kompetenz". Screenshot aus dem pdf-Dokument im Internet. Quelle: <a href="https://publications.jrc.ec.europa.eu/repository/bitstream/JRC110624/dc_guide_may18.pdf">https://publications.jrc.ec.europa.eu/repository/bitstream/JRC110624/dc_guide_may18.pdf</a> (12.04.2020) .....	42
Abbildung 6: Kompetenzbereiche inkl. Kompetenzen. Screenshot aus dem pdf-Dokument im Internet. Quelle: <a href="https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz_Arbeitsmarkt.html">https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz_Arbeitsmarkt.html</a> (12.04.2020) .....	45
Abbildung 7: dreidimensionales Würfel-Modell des DigComp 2.2 AT. Screenshot aus dem pdf-Dokument im Internet. Quelle: <a href="https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz_Arbeitsmarkt.html">https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz_Arbeitsmarkt.html</a> (12.04.2020) .....	46
Abbildung 8: Nationaler Qualifikationsrahmen Österreich. Screenshot von der Website. Quelle: <a href="https://www.qualifikationsregister.at/">https://www.qualifikationsregister.at/</a> (12.04.2020) .....	47
Abbildung 9: Bloom's Taxonomy.. Screenshot von der Website. Quelle: <a href="https://tophat.com/blog/blooms-revised-taxonomy-pyramid/">https://tophat.com/blog/blooms-revised-taxonomy-pyramid/</a> (Vanderbilt University Center for Teaching) (12.04.2020) .....	48
Abbildung 10: DigComp 2.2 AT - Kompetenzstufen. Screenshot aus dem pdf-Dokument im Internet. Quelle: <a href="https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz_Arbeitsmarkt.html">https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz_Arbeitsmarkt.html</a> (12.04.2020) .....	49



Abbildung 11: DigComp 2.2 AT - Kompetenzstufen. Screenshot aus dem pdf-Dokument im Internet. Quelle: <a href="https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz_Arbeitsmarkt.html">https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz_Arbeitsmarkt.html</a> (12.04.2020) .....	66
Abbildung 12: Dateimanagement. Screenshot vom Moodle-Kurs des GRG16 Maroltingergasse (Comber, O. et. al., 2019) .....	78
Abbildung 13: GIMP. Screenshot vom Moodle-Kurs des GRG16 Maroltingergasse (Comber, O. et. al., 2019) .....	78
Abbildung 14: Scratch. Screenshot vom Moodle-Kurs des GRG16 Maroltingergasse (Comber, O. et. al., 2019) .....	79
Abbildung 15: Microsoft Visual Studio Express 2010. Screenshot vom Moodle-Kurs des GRG16 Maroltingergasse (Comber, O. et. al., 2019).....	79
Abbildung 16: Unity. Screenshot vom Moodle-Kurs des GRG16 Maroltingergasse (Comber, O. et. al., 2019) .....	80
Abbildung 17: Unity-Tutorial (Comber, O. et. al., 2019) Screenshot von der Website. Quelle: <a href="https://learn2progame.github.io/learn2proGrAME-Tutorial/">https://learn2progame.github.io/learn2proGrAME-Tutorial/</a> (14.04.2020).....	81
Abbildung 18: Verständnis der Lernenden des Ablaufs von Programmen (N=124) (Eigene Darstellung).....	89
Abbildung 19: Erwerb der Fertigkeit Programmieren (N=124) (Eigene Darstellung).....	90
Abbildung 20: Verständnis des Algorithmusbegriffes (N=124) (Eigene Darstellung).....	91
Abbildung 21: Verständnis der Lernenden von Softwareprojekten (N=124) (Eigene Darstellung) .....	92
Abbildung 22: Verständnis der Lernenden von Problemen bei der Entwicklung (N=124) (Eigene Darstellung).....	93
Abbildung 23: evtl. Verbesserung von Projektmanagementfähigkeiten (N=124) (Eigene Darstellung) .....	95
Abbildung 24: evtl. Erwerb bzw. Verbesserung der Programmierfertigkeit (N=124) (Eigene Darstellung).....	96

## KAPITEL: Abbildungsverzeichnis

---

Abbildung 25: evtl. Verständnis von Datenstrukturen (N=124) (Eigene Darstellung).....	97
Abbildung 26: evtl. Verständnis von Algorithmen (N=124) (Eigene Darstellung).....	98
Abbildung 27: evtl. Einblick in die Logik von Programmen (N=124) (Eigene Darstellung).....	99
Abbildung 28: Screenshot aus der Excel-Tabelle, in der die Paraphrasierung und Gewichtung stattgefunden hat. (Eigene Darstellung).....	112

## 9 Literaturverzeichnis

- Anderson, L. W., Krathwohl, D. R. & Airasian, P. W. (Hrsg.). (2001). *A taxonomy for learning, teaching, and assessing. A revision of Bloom's taxonomy of educational objectives* (Abridged ed.). New York: Longman/Langenscheidt ELT.
- Aronson, E., Wilson, T. D. & Akert, R. M. (2008). *Sozialpsychologie* (Pearson Studium - Psychologie, 6., aktualisierte Aufl.). München: Pearson Studium.
- Asendorpf, J. & Neyer, F. J. (2012). *Psychologie der Persönlichkeit. Mit 110 Tabellen* (Springer-Lehrbuch, 5., vollst. überarb. Aufl.). Berlin: Springer.
- Baacke, D. (1997). *Medienpädagogik* (Grundlagen der Medienkommunikation, Bd. 1). Tübingen: Niemeyer.
- Bechmann, S. (2016). *Sprachwandel - Bedeutungswandel. Eine Einführung* (UTB Sprachwissenschaft, Bd. 4536). Tübingen: A. Francke Verlag.
- Beer, R. (Hrsg.). (2011). *Kompetenzorientierter Unterricht in Theorie und Praxis. Information für Lehrer/innen* (Standards). Graz: Leykam.
- Bildungskommission NRW. (1995). *Zukunft der Bildung - Schule der Zukunft. Denkschrift der Kommission "Zukunft der Bildung - Schule der Zukunft" beim Ministerpräsidenten des Landes Nordrhein-Westfalen*. Neuwied: Luchterhand.
- Griesenauer, G. & Kaschnig-Löbel, H. (Mitarbeiter) (Bundesinstitut BIFIE, Hrsg.). (2019). *Bildungsstandards und Kompetenzorientierung im Unterricht*, BIFIE – Bundesinstitut für Bildungsforschung, Innovation & Entwicklung des österreichischen Schulwesens. <https://www.bifie.at/>. Zugegriffen: 20. Januar 2020.

Bundesministerium für Bildung, Wissenschaft und Forschung, Österreichische Austauschdienst GmbH (OeAD) (Mitarbeiter) (Mayrhofer, M. & Siegele, P., Hrsg.). Forschungsprogramm: Sparkling Science, Bundesministerium für Bildung, Wissenschaft und Forschung. <https://www.sparkling-science.at/>. Zugegriffen: 29. November 2019.

(Bundesministerium für Bildung, Wissenschaft und Forschung (BMBWF), Hrsg.). (2020). *digi.komp. Digitale Kompetenzen Informatische Bildung*. <https://www.digikomp.at/>. Zugegriffen: 29. November 2019.

(Bundesministerium für Digitalisierung und Wirtschaftsstandort (BMDW), Hrsg.). (2018). *Digitale Kompetenz und Arbeitsmarkt. DigComp 2.2 AT*. [https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz\\_Arbeitsmarkt.html](https://www.bmdw.gv.at/Themen/Digitalisierung/Gesellschaft/Digitale-Kompetenz_Arbeitsmarkt.html). Zugegriffen: 20. Januar 2020.

(Bundesministerium für Digitalisierung und Wirtschaftsstandort (BMDW), Hrsg.). (2020). *RIS - Rechtsinformationssystem des Bundes*. <https://www.ris.bka.gv.at/>. Zugegriffen: 22. Januar 2020.

Mag. Radnitzky, E. & Schmied, C. (Mitarbeiter) (Bundesministerium für Unterricht, Kunst und Kultur (BMUKK), Hrsg.). (2007). Initiative „25+“: Individualisierung des Unterrichts. Persönlichkeit und Lernvoraussetzungen der einzelnen Schülerinnen und Schüler in den Mittelpunkt stellen. Rundschreiben Nr. 9/2007. [https://www.bmbwf.gv.at/Themen/schule/schulrecht/rs/1997-2017/2007\\_09.html](https://www.bmbwf.gv.at/Themen/schule/schulrecht/rs/1997-2017/2007_09.html). Zugegriffen: 15. Dezember 2019.

Cambridge Dictionary (Cambridge Free English Dictionary and Thesaurus, Hrsg.). (2019). Titel bzw. Eintrag: siehe Angabe bei Zitierstelle, Cambridge University Press. <https://dictionary.cambridge.org/dictionary/>. Zugegriffen: 4. Januar 2020.

Carretero, S., Vuorikari, R. & Punie, Y. (2017). *DigComp 2.1. The digital competence framework for citizens with eight proficiency levels and*

*examples of use* (EUR, Scientific and technical research series, Bd. 28558).  
Luxembourg: Publications Office.

Comber, O. (2018). *Zwischenbericht - 24.8.2018 - SPA 06/082 - Learn to proGrAME*. Wien: Universität Wien, Fakultät für Informatik.

Comber, O. (2020, 27. Februar). Kurs: MC learn to proGrAME, GRG16 Maroltingergasse. <https://www4.lernplattform.schule.at/g16slsz/course/view.php?id=183>. Zugegriffen: 27. Februar 2020.

COMM/DG/UNIT (Europäische Kommission, Hrsg.). (2019). Council Recommendation on Key Competences for Lifelong Learning - Allgemeine und berufliche Bildung - European Commission. [https://ec.europa.eu/education/education-in-the-eu/council-recommendation-on-key-competences-for-lifelong-learning\\_de](https://ec.europa.eu/education/education-in-the-eu/council-recommendation-on-key-competences-for-lifelong-learning_de). Zugegriffen: 20. Januar 2020.

Duden Wörterbuch, Dudenredaktion (Mitarbeiter) (Duden online, Hrsg.). (2019). Titel bzw. Eintrag: siehe Angabe bei Zitierstelle, Bibliographisches Institut GmbH. <https://www.duden.de/>. Zugegriffen: 4. Januar 2020.

DWDS, Dr. Alexander Geyken, Dr. Lothar Lemnitzer (Mitarbeiter) (DWDS – Digitales Wörterbuch der deutschen Sprache, Hrsg.). (2019). Titel bzw. Eintrag: siehe Angabe bei Zitierstelle, Berlin-Brandenburgische Akademie der Wissenschaften. <https://www.dwds.de/>. Zugegriffen: 4. Januar 2020.

(Europäische Kommission, Hrsg.). (2019). Europäische Kommission, Europäische Union. [https://ec.europa.eu/info/index\\_de](https://ec.europa.eu/info/index_de).

(European Centre for the Development of Vocational Training (CEDEFOP), Hrsg.). (2020). Europass, Europäische Kommission. <https://europass.cedefop.europa.eu/>. Zugegriffen: 20. Januar 2020.

Fehr, H. (2017). *Eigene Apps programmieren. Ohne Vorkenntnisse einsteigen* (Rheinwerk Computing, 1. Auflage, 1. Nachdruck). Bonn: Rheinwerk Verlag.

- Oswald, P. & Domany-Funtan, U. F. (Mitarbeiter) (fit4internet, Hrsg.). (2019). Kompetenzmodell - fit4internet, Bundesministerium für Digitalisierung und Wirtschaftsstandort. <https://www.fit4internet.at/view/verstehen-das-modell>. Zugegriffen: 22. Januar 2020.
- Gregory, J. (2014). *Game Engine Architecture, Second Edition* (2nd ed.). Natick: CRC Press.
- Heyse, V. & Erpenbeck, J. (2010). *Kompetenztraining. Informations- und Trainingsprogramme* (2., überarb. u. erw. Aufl.). Stuttgart: Schäffer-Poeschel.
- Jank, W. & Meyer, H. (2009). *Didaktische Modelle* (9. Aufl.). Berlin: Cornelsen Scriptor.
- John, O. P., Naumann, L. P. & Soto, C. J. (2008). Paradigm Shift to the Integrative Big Five Trait Taxonomy. History, Measurement, and Conceptual Issues. In O. P. John, R. W. Robins & L. A. Pervin (Hrsg.), *Handbook of personality. Theory and research* (3. ed., ). New York, NY: Guilford Press.
- Kanning, U. P. (2009). *Diagnostik sozialer Kompetenzen* (Kompendien psychologische Diagnostik, Bd. 4, 2., aktualisierte Auflage). Göttingen: Hogrefe.
- Kramer, M., Hubwieser, P. & Brinda, T. (2016). *2016 International Conference on Learning and Teaching in Computing and Engineering. LaTiCE 2016 : Mumbai, India, 31 March-3 April 2016 : proceedings*. Piscataway, NJ: IEEE.
- Krisper-Ullyett, L. (2013). *Kein Kind ohne digitale Kompetenzen. kreativinnovativ2020 Gesprächsreihe*. ki2020 im Gespräch mit den Projektverantwortlichen von digi.komp (Bundesministerium für Unterricht, Kunst und Kultur (BMUKK), Hrsg.). Wien: Bundesministerium für Unterricht, Kunst und Kultur (BMUKK). <https://www.informatische-grundbildung.com/app/download/6450497564/digikomp-interview.pdf?t=1483020240>.

Kuckartz, U., Dresing, T., Rädiker, S. & Stefer, C. (2008). *Qualitative Evaluation. Der Einstieg in die Praxis* (2., aktualisierte Auflage). Wiesbaden: VS Verlag für Sozialwissenschaften / GWV Fachverlage GmbH Wiesbaden.

Kühberger, C. (2015). *Kompetenzorientiertes historisches und politisches Lernen. Methodische und didaktische Annäherungen für Geschichte, Sozialkunde und Politische Bildung* (Österreichische Beiträge zur Geschichtsdidaktik, Band 2, 3., überarbeitete und erweiterte Auflage). Innsbruck: StudienVerlag.

Kultusministerkonferenz (Sekretariat der Kultusministerkonferenz, Hrsg.). (2018). Handreichung für die Erarbeitung von Rahmenlehrplänen der KMK für den berufsbezogenen Unterricht und ihre Abstimmung mit Ausbildungsordnungen des Bundes für anerkannte Ausbildungsberufe. [https://www.kmk.org/fileadmin/Dateien/veroeffentlichungen\\_beschluesse/2011/2011\\_09\\_23-GEP-Handreichung.pdf](https://www.kmk.org/fileadmin/Dateien/veroeffentlichungen_beschluesse/2011/2011_09_23-GEP-Handreichung.pdf).

Lexico Dictionaries. (2019). Titel bzw. Eintrag: siehe Angabe bei Zitierstelle, Lexico Dictionaries. <https://www.lexico.com/>. Zugegriffen: 4. Januar 2020.

Mayer, H. (2018). *Die Probleme von Schülerinnen und Schüler beim Programmierenlernen mit Unity im Rahmen des ProGrAME Projekts*. Bachelorarbeit, Universität Wien. Wien.

Merriam-Webster Dictionary (Merriam-Webster, Incorporated, Hrsg.). (2019). Titel bzw. Eintrag: siehe Angabe bei Zitierstelle, Merriam-Webster, Incorporated. <https://www.merriam-webster.com/>. Zugegriffen: 4. Januar 2020.

Meyer, H. (2011). *Was ist guter Unterricht?* (8. Aufl.). Berlin: Cornelsen-Scriptor.

(Micheuz, P., Hrsg.). (2011). Informatische Grundbildung. Kompetenzmodell, Bundesministerium für Unterricht, Kunst und Kultur. <https://www.informatische-grundbildung.com/digi-komp-bm-bf-alt/das-kompetenzmodell/>. Zugegriffen: 3. Februar 2020.

- Miljković, N. & Merten, R. (Hrsg.). (2017). *Erfolg in Studium und Karriere - Fit durch Selbstcoaching* (UTB Schlüsselkompetenzen, Bd. 4770). Opladen: Verlag Barbara Budrich.
- Motschnig, R., Hlavacs, H., Schritteser, I., Swertz, C. & Comber, O. (2016). *Antrag - Sparkling Science - Online Antrag inkl. ausführlicher Projektbeschreibung. Learn to proGrAME - Programmieren lernen durch Computerspielentwicklung. : CSLEARN - Educational Technologies* (Fakultät für Informatik), Universität Wien. <https://progame.at/projekt>.
- Nárosy, T., Röthler, T. & Svencik, D. (2018). *DigComp 2.2 AT* (Bundesministerium für Digitalisierung und Wirtschaftsstandort (BMDW), Hrsg.). Wien: Bundesministerium für Digitalisierung und Wirtschaftsstandort (BMDW). [https://www.bmdw.gv.at/dam/jcr:54bbe103-7164-494e-bb30-cd152d9e9b33/DigComp2.2\\_V33-barrierefrei.pdf](https://www.bmdw.gv.at/dam/jcr:54bbe103-7164-494e-bb30-cd152d9e9b33/DigComp2.2_V33-barrierefrei.pdf).
- Calice, J. (Mitarbeiter) (Nationale Koordinierungsstelle für den NQR in Österreich (NKS), Hrsg.). (2019). *Qualifikationsregister. Der Nationale Qualifikationsrahmen (NQR), Österreichische Austauschdienst GmbH*. <https://www.qualifikationsregister.at/>. Zugegriffen: 23. Januar 2020.
- Olechowski, R. & Bábosik, I. (Hrsg.). (2003). *Lehren - Lernen - Prüfen* (Schule - Wissenschaft - Politik, Bd. 16). Frankfurt am Main: Lang.
- Österreichische Austauschdienst GmbH. OeAD-GmbH. <https://oead.at/>. Zugegriffen: 29. November 2019.
- PONS Wörterbuch, Gabriele Schmidt (Mitarbeiter). (2019). Titel bzw. Eintrag: siehe Angabe bei Zitierstelle. <https://de.pons.com/>. Zugegriffen: 4. Januar 2020.
- Radnitzky, E. & Westfall-Greiter, T. (2009) Comenius wäre begeistert! In C. Schrack & T. Nárosy (Hrsg.), *Individualisieren mit eLearning. Neues Lernen in heterogenen Lerngemeinschaften* (S. 14–17). Wien. Bundesministerium für Unterricht, Kunst und Kultur (BMUKK).



(Rat der Europäischen Union, Hrsg.). (2018). Empfehlung des Rates vom 22. Mai 2018 zu Schlüsselkompetenzen für lebenslanges Lernen, Rat der Europäischen Union. [https://eur-lex.europa.eu/legal-content/DE/TXT/HTML/?uri=CELEX:32018H0604\(01\)&qid=1577807163576&from=DE](https://eur-lex.europa.eu/legal-content/DE/TXT/HTML/?uri=CELEX:32018H0604(01)&qid=1577807163576&from=DE). Zugegriffen: 31. Dezember 2019.

(Rat und Parlament der Europäischen Union, Hrsg.). (2006). Empfehlung des Europäischen Parlaments und des Rates vom 18. Dezember 2006 zu Schlüsselkompetenzen für lebensbegleitendes Lernen. <https://eur-lex.europa.eu/legal-content/DE/TXT/HTML/?uri=CELEX:32006H0962&from=DE>. Zugegriffen: 31. Dezember 2019.

Reibold, D. K. & Regier, S. (2009). *Training der Schlüsselqualifikationen in Schule und Beruf. Handreichung für Lehrkräfte der Sekundarstufe I sowie für betriebliche Ausbilder in Wirtschaft und Verwaltung* (Expert-Taschenbuch, Nr. 108). Renningen: expert-Verl.

Roth, H. (1971). *Pädagogische Anthropologie. Bd. 2. Entwicklung und Erziehung. Grundlagen einer Entwicklungspädagogik* (Bd. 2, 1. Auflage). Hannover: Hermann Schroedel Verlag KG.

Schell, J. & Feilen, M. (2012). *Die Kunst des Game Designs. Bessere Games konzipieren und entwickeln* (1. Aufl.). Heidelberg: mitp.

Swertz, C. (2019). DigComp 2.2 AT. Hintergründe und Kontexte. *Medienimpulse* 57 (1). <https://journals.univie.ac.at/index.php/mp/article/view/mi1337>. Zugegriffen: 22. Januar 2020.

Üçoluk, G. & Kalkan, S. (2012). *Introduction to Programming Concepts with Case Studies in Python*. Vienna: Springer.

Vogt, S. & Werner, M. (2014, 05. August). *Forschen mit Leitfadeninterviews und qualitativer Inhaltsanalyse* (Fachhochschule Köln, Hrsg.). Köln: Fakultät für angewandte Sozialwissenschaften. <https://www.th-koeln.de/mam/bilder/>

hochschule/fakultaeten/f01/skript\_interviewsqualinhaltsanalyse-fertig-05-08-2014.pdf.

Weinert, F. E. (Hrsg.). (2002). *Leistungsmessungen in Schulen* (Beltz Pädagogik, 2., unveränd. Aufl., Dr. nach Typoskript). Weinheim: Beltz.

Wikipedia, Wikipedia-Autoren (Mitarbeiter) (Wikipedia – Die freie Enzyklopädie, Hrsg.). Titel bzw. Eintrag: siehe Angabe bei Zitierstelle. <https://de.wikipedia.org/wiki/Wikipedia:Hauptseite>. Zugegriffen: 2020.

Wilhelm, O. & Nickolaus, R. (2013). Was grenzt das Kompetenzkonzept von etablierten Kategorien wie Fähigkeit, Fertigkeit oder Intelligenz ab? *Zeitschrift für Erziehungswissenschaft* 16 (S1), 23–26. doi:10.1007/s11618-013-0380-6

Wimmer, B. (futurezone - Technology News, Hrsg.). (2019). Österreicher sollen mit 2,5 Millionen Euro internetfit gemacht werden, Futurezone GmbH. <https://futurezone.at/netzpolitik/oesterreicher-sollen-mit-25-millionen-euro-internetfit-gemacht-werden/400390610>. Zugegriffen: 7. Februar 2020.

Wimmer, J. (2013). *Massenphänomen Computerspiele. Soziale, kulturelle und wirtschaftliche Aspekte*. München: UVK Verl.-Ges.

---

## 10 Anhang

Die einzelnen Punkte des Anhangs befinden sich als Links auch im Inhaltsverzeichnis, sodass man schneller den gesuchten Punkt findet.

### **Hinweis zu den zueinander gemappten Kompetenzmodellen:**

Die zur Zeit der Diplomarbeit gültigen und verwendeten drei Kompetenzmodelle lauten DigComp2.2AT (**DC**), digi.komp12 (**Dk12**), und digi.komp8 (**Dk8**) wobei in eckiger Klammer bei den Punkten die jeweils zugeordneten Punkte aus den anderen Kompetenzmodellen angegeben sind. Beispielsweise bedeutet **[Dk12 2.1.2W, 4.1.1W]**, dass die Punkte **2.1.2** und **4.1.1** aus dem „**Dk12**“ also **digi.komp12** dem aktuellen Punkt dieses Kompetenzmodelles entsprechen, das heißt übereinstimmend, oder zumindest überwiegend deckungsgleich sind, oder evtl. einer für den anderen eine Grundvoraussetzung ist. Das beim Punkt angehängte **W**, **A** oder **R** deutet auf die **Handlungsdimension** des Punktes im digi.komp12 an. **W** beschreibt die Handlungsdimension „**Wissen und Verstehen**“, **A** steht für „**Anwenden und Gestalten**“ und **R** für „**Reflektieren und Bewerten**“. Beim DigComp2.2AT sind jene Punkte in **Lila**, um die der österreichische DigComp2.2AT den europäischen DigComp2.1 erweitert. Zuletzt ist auch noch der digi.komp4, zur Überprüfung von Hinweisen darauf angehängt.

---

### **10.1 DigComp2.2AT – ergänzt die europ. Version 2.1 (25 Punkte)**

#### **0 Grundlagen und Zugang (3 Punkte)**

##### **0.1 Konzepte der Digitalisierung verstehen**

Verständnis für den technischen Unterschied zwischen „analog“ und „digital“ und für die grundlegenden Strukturen, Eigenschaften und Begrifflichkeiten des Internet. **[Dk8 3.1.2, 8.1.2, 8.1.5]**, **[Dk12 2.1.2W, 4.1.1W]**

##### **0.2 Digitale Geräte bedienen**

Sich unterschiedliche Bedienkonzepte von digitalen Geräten aneignen und verwenden können. [Dk8 3.1.1, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 3.3.4, 3.3.5, 3.3.6, 3.4.1, 3.4.2, 3.4.3, 3.4.4, 3.4.5, 3.4.6], [Dk12 2.1.3A, 2.2.4A]

### **0.3 Inklusive Formen des Zugangs zu digitalen Inhalten nutzen und bereitstellen**

Technische Umsetzungsformen digitaler Barrierefreiheit (wie z.B. automatische Übersetzung, Vorleseoption) verstehen und anwenden bzw. bereitstellen können. Bewusstsein hinsichtlich Gender, Diversität, kulturellem Kontext, Menschen mit besonderen Bedürfnissen wie z.B. geistige und körperliche Behinderung. [Dk8 1.3.3], [Dk12 2.4.1W, 2.4.2A, 2.4.3A]

## **1 Umgang mit Informationen und Daten (3 Punkte)**

### **1.1 Daten, Informationen und digitale Inhalte recherchieren, suchen und filtern**

Informationsbedarf artikulieren; in digitalen Umgebungen nach Daten, Informationen und Inhalten suchen, auf sie zugreifen und zwischen ihnen navigieren; persönliche Suchstrategien erstellen und aktualisieren. [Dk8 2.1.1], [Dk12 3.3.2W, 3.3.4A]

### **1.2 Daten, Informationen und digitale Inhalte **kritisch** bewerten und interpretieren**

Glaubwürdigkeit und Zuverlässigkeit von Datenquellen, Informationen und digitale Inhalte analysieren, vergleichen und kritisch bewerten; Daten, Informationen und digitale Inhalte analysieren, interpretieren und kritisch bewerten. [Dk8 2.2.1, 2.2.2, 2.2.5], [Dk12 1.2.6R, 3.3.6R]

### **1.3 Daten, Informationen und digitale Inhalte verwalten**

Daten, Informationen und Inhalte in digitalen Umgebungen organisieren, speichern und abrufen; Daten, Informationen und Inhalte in einer strukturierten Umgebung organisieren und verarbeiten. [Dk8 2.3.1], [Dk12 3.3.5A]

## **2 Kommunikation und Zusammenarbeit (6 Punkte)**

### **2.1 Mithilfe digitaler Technologien kommunizieren**

Durch eine Vielzahl von digitalen Technologien interagieren und geeignete digitale Kommunikationsmittel für einen bestimmten Kontext (auch hinsichtlich kultureller, sozialer, genderspezifischer etc. Unterschiede) verstehen. [Dk8 4.2.1, 5.1.6], [Dk12 3.4.1W, 3.4.2A, 3.4.3R]

## **2.2 Mithilfe digitaler Technologien Daten und Informationen teilen und zusammenarbeiten**

Daten, Informationen und digitale Inhalte mit anderen mithilfe geeigneter digitaler Technologien austauschen; als Vermittler fungieren; über Verweis- und Zuordnungspraktiken Bescheid wissen. Digitale Tools und Technologien für kooperative Prozesse sowie für die gemeinsame Erstellung und Erarbeitung von Ressourcen und Wissen nutzen. [Dk8 2.4.1], [Dk12 3.4.1W, 3.4.2A, 3.4.3R]

## **2.3 Digitale Technologien für die gesellschaftliche Teilhabe verwenden**

KEINE BESCHREIBUNG VORHANDEN [Dk8 2.4.1], [Dk12 3.4.2A, 3.4.4R]

## **2.4 Ein- und Verkäufe durchführen**

Kommerzielle (Ein- und Verkauf, Ver- und Ersteigerung) sowie nicht-kommerzielle (Tausch, Geschenk) Transaktionen von Waren und Dienstleistungen aller Art durchführen. [Dk8 so nicht vorhanden, evtl. 6.2.3]

## **2.5 Angemessene Ausdrucksformen verwenden („Netiquette“)**

Sich der Verhaltensweisen und -normen bei der Nutzung digitaler Technologien und der Interaktion in digitalen Umgebungen bewusst sein; Kommunikationsstrategien an die jeweils Beteiligten anpassen und sich der Vielfalt der Kulturen und Generationen in digitalen Umgebungen bewusst sein. Erkennen, welcher Kommunikationskanal für welche adäquat ist bzw. ob formelle oder informelle Kommunikation vorzuziehen ist. Angemessenes Verhalten bei Online-Diskussionen. [Dk8 5.1.7], [Dk12 1.2.1W, 1.2.2A, 1.2.4R]

## **2.6 Die digitale Identität gestalten**

Eine oder mehrere digitale Identitäten erstellen und verwalten; das eigene Ansehen bewahren; sich um die Daten kümmern, die man durch verschiedene digitale Tools, Umgebungen und Dienste erzeugt. [Dk8 5.3.1]

### **3 Kreation digitaler Inhalte (4 Punkte)**

#### **3.1 Digitale Inhalte entwickeln**

Digitale Inhalte in verschiedenen Formaten erstellen, bearbeiten und situationsadäquat anwenden können; sich mit digitalen Mitteln ausdrücken.

[Dk8 4.2.5], [Dk12 3.1.1W, 3.1.3A]

#### **3.2 Digitale Inhalte integrieren und neu erarbeiten**

Informationen und Inhalte in einen bestehenden Wissensfundus integrieren, modifizieren, verfeinern und verbessern; neue, originelle und relevante Inhalte und Kenntnisse erschaffen. [Dk8 4.2.2]

#### **3.3 Werknutzungsrecht und Lizenzen**

Verstehen, wie Werknutzungsrechte und Lizenzen für Daten, Informationen und digitale Inhalte gelten. [Dk8 2.4.3], [Dk12 1.2.2W]

#### **3.4 Programmieren und Abläufe automatisieren**

Eine Abfolge verständlicher Anweisungen für ein Computersystem zur Lösung eines bestimmten Problems oder zum Ausführen einer bestimmten Aufgabe planen und entwickeln. [Dk8 8.1.3, 8.1.4], [Dk12 4.2.2W, 4.2.6A]

### **4 Sicherheit (5 Punkte)**

#### **4.1 Geräte schützen**

Geräte und digitale Inhalte schützen und Risiken und Bedrohungen in digitalen Umgebungen verstehen; Sicherheits- und Sicherungsmaßnahmen kennen sowie Zuverlässigkeit und Privatsphäre gebührend berücksichtigen. [Dk8

6.1.3], [Dk12 1.2.1W, 1.2.2A, 1.2.3A, 1.2.5R]

#### **4.2 Personenbezogene Daten und Privatsphäre schützen**

Persönliche Daten und die Privatsphäre in digitalen Umgebungen schützen; verstehen, wie man persönlich identifizierbare Informationen verwendet und teilt, gleichzeitig sich und andere vor Schäden schützen; verstehen, dass digitale Dienste eine „Datenschutzrichtlinie“ verwenden, um über die Verwendung personenbezogener Daten zu informieren. [Dk8 6.2.2], [Dk12

1.2.1W, 1.2.2A]

#### **4.3 Gesundheit und Wohlbefinden schützen**

Gesundheitsrisiken und Bedrohungen für das körperliche und seelische Wohlbefinden beim Einsatz digitaler Technologien vermeiden können; sich selbst und andere vor möglichen Gefahren in digitalen Umgebungen schützen können (z. B. Cybermobbing); sich der digitalen Technologien für soziales Wohlergehen und soziale Inklusion bewusst sein. **[Dk8 1.3.2]**

#### **4.4 Sich vor Betrug und Konsumentenrechtsmissbrauch schützen**

Erkennen von unseriösen Online-Shops; Kenntnis der wichtigsten rechtlichen Bestimmungen; Maßnahmen des Käuferschutzes anwenden sowie Preisvergleiche anstellen können. **[Dk8 6.2.3]**, **[Dk12 1.2.6R]**

#### **4.5 Umwelt schützen**

Sich der Umweltauswirkungen digitaler Technologien und ihrer Nutzung bewusst sein. **[Dk8 1.1.5]**

### **5 Problemlösen und Weiterlernen (4 Punkte)**

#### **5.1 Technische Probleme lösen**

Technische Probleme beim Betrieb von Geräten und beim Einsatz digitaler Umgebungen identifizieren und lösen (von der Fehlersuche bis zur Lösung komplexerer Probleme). **[Dk8 7.3.1]**, **[Dk12 2.1.6R, 4.2.11R]**

#### **5.2 Bedürfnisse und technologische Antworten darauf erkennen**

Bedürfnisse erkennen und identifizieren sowie digitale Werkzeuge und mögliche technologische Antworten zu deren Lösung bewerten, auswählen und verwenden; digitale Umgebungen an persönliche Bedürfnisse anpassen (z.B. Zugänglichkeit). **[Dk8 7.2.3]**, **[Dk12 1.1.1W, 1.1.2A, 2.1.2W, 2.1.3A, 2.1.5R, 2.2.2W, 2.2.5R, 2.4.1W, 2.4.3A, 3.3.1W]**

#### **5.3 Kreativ mit digitalen Technologien umgehen**

Digitale Werkzeuge und Technologien zur Schaffung von Wissen und zur Innovation von Prozessen und Produkten nutzen; sich individuell und gemeinsam mit anderen in Denkprozessen auseinandersetzen, um konzeptionelle Probleme und Problemsituationen in digitalen Umgebungen zu verstehen und zu lösen. **[Dk8 8.2.1]**, **[Dk12 3.3.1W, 3.3.2W, 3.3.4A, 3.4.1W, 3.4.2A]**

#### **5.4 Digitale Kompetenzlücken erkennen**

Verstehen, wo die eigene digitale Kompetenz verbessert oder aktualisiert werden muss; andere bei ihrer digitalen Kompetenzentwicklung unterstützen; nach Gelegenheiten zur Selbstentwicklung suchen und mit der digitalen Evolution Schritt halten.



---

## 10.2 Digi.komp12 – Wahlpflichtfach Informatik (81 Punkte)

### 1 Informationstechnologie, Mensch und Gesellschaft (15 Punkte)

#### 1.1 Bedeutung von IT in der Gesellschaft (3 Punkte)

##### Wissen und Verstehen:

- 1.1.1 Ich kann Beispiele für den Einsatz von Informatiksystemen und ihre gesellschaftlichen und wirtschaftlichen Auswirkungen beschreiben. [Dk8 1.2.1], [DC 5.2]

##### Anwenden und Gestalten:

- 1.1.2 Ich kann Wissen über Informatiksysteme im digitalen privaten und schulischen Umfeld zielgerichtet anwenden und nutzen. [Dk8 1.1.1], [DC 5.2]

##### Reflektieren und Bewerten:

- 1.1.3 Ich kann den Einfluss von Informatiksystemen auf meinen Alltag, auf die Gesellschaft und Wirtschaft einschätzen und an konkreten Beispielen Vor- und Nachteile abwägen. [Dk8 1.1.3]

#### 1.2 Verantwortung, Datenschutz und Datensicherheit (6 Punkte)

##### Wissen und Verstehen:

- 1.2.1 Ich kann meine Rechte und Pflichten in der Nutzung von Informatiksystemen beschreiben und wesentliche Aspekte des Datenschutzes und der Datensicherheit erklären. [Dk8 2.4.2, 6.2.1], [DC 2.5, 3.3, 4.1, 4.2]

##### Anwenden und Gestalten:

- 1.2.2 Ich kann beim Einsatz von Informatiksystemen mein Wissen um Pflichten und Rechte in Bezug auf meine Person und meine Arbeitsumgebung, auf persönliche und fremde Daten verantwortungsbewusst anwenden. [Dk8 2.4.2, 5.1.7, 6.2.2], [DC 2.5, 4.1, 4.2]
- 1.2.3 Ich kann für den Schutz und die Sicherheit von Informatiksystemen, mit denen ich arbeite, sorgen. [Dk8 6.1.2, 6.1.3], [DC 4.1]

##### Reflektieren und Bewerten:

- 1.2.4 Ich kann meine Verantwortung beim Einsatz von Informatiksystemen sowohl in der Quantität als auch in der Qualität reflektieren. [DC 2.5]
- 1.2.5 Ich kann verschiedene Schutzmaßnahmen für Daten und IT-Systeme beurteilen und empfehlen. [DC 4.1]
- 1.2.6 Ich kann die Rechtskonformität einer Website in Grundzügen bewerten. [Dk8 2.2.1], [DC 1.2, 4.4]

#### 1.3 Geschichte der Informatik (3 Punkte)

##### Wissen und Verstehen:

- **1.3.1** Ich kann Meilensteine in der Entwicklung der Computertechnik beschreiben und maßgebliche dahinterstehende Persönlichkeiten nennen. **[Dk8 1.4.1]**

### **Anwenden und Gestalten:**

- **1.3.2** Ich kann mein geschichtliches Wissen in Beziehung zur aktuellen Situation setzen und daraus gegebenenfalls mögliche Zukunftsszenarien ableiten.

### **Reflektieren und Bewerten:**

- **1.3.3** Ich kann anhand der Entwicklung der IT zwischen kurzlebigen Hard- und Softwareprodukten und langlebigen Prinzipien unterscheiden.

## **1.4 Berufliche Perspektiven (3 Punkte)**

### **Wissen und Verstehen:**

- **1.4.1** Ich kann Berufsfelder benennen, in denen die Anwendung der IT eine bedeutende Rolle spielt, und die Vielfalt an IT-Berufen kategorisieren. **[Dk8 1.2.1]**

### **Anwenden und Gestalten:**

- **1.4.2** Ich kann mein Wissen und meine schulischen Erfahrungen im Zusammenhang mit IT für meine künftige Erwerbsbiografie nutzen.

### **Reflektieren und Bewerten:**

- **1.4.3** Ich kann die wirtschaftliche Bedeutung der IT in den diversen Berufsfeldern einordnen und die Chancen von IT-Berufen abschätzen.

## **2 Informatiksysteme (22 Punkte)**

### **2.1 Technische Grundlagen und Funktionsweisen (6 Punkte)**

#### **Wissen und Verstehen:**

- **2.1.1** Ich kann Komponenten von Informatiksystemen beschreiben und ihre Funktionsweise und ihr Zusammenwirken erklären. **[Dk8 7.1.1]**
- **2.1.2** Ich kann grundlegende technische Konzepte von Informatiksystemen erklären. **[DC 0.1, 5.2]**

#### **Anwenden und Gestalten:**

- **2.1.3** Ich kann ein Computersystem samt Peripheriegeräten sachgerecht nutzen. **[Dk8 3.1.1, 7.2.3], [DC 0.2, 5.2]**
- **2.1.4** Ich kann ein Computersystem zusammenstellen und zusammenschließen. **[Dk8 7.2.1]**

#### **Reflektieren und Bewerten:**

- **2.1.5** Ich kann unterschiedliche digitale Endgeräte bzw. Informatiksysteme in Bezug auf ihre technischen Eigenschaften und ihre Leistungsfähigkeit bewerten. **[DC 5.2]**
- **2.1.6** Ich kann einfache Fehler diagnostizieren und beheben. **[Dk8 7.3.1], [DC 5.1]**

## 2.2 Betriebssysteme und Software (5 Punkte)

### Wissen und Verstehen:

- 2.2.1 Ich kann die Kernaufgaben und Arbeitsweisen von Betriebssystemen beschreiben und erklären. [Dk8 3.1.2]
- 2.2.2 Ich kann Kategorien von Software nennen und deren Anwendung beschreiben. [Dk8 7.1.2], [DC 5.2]

### Anwenden und Gestalten:

- 2.2.3 Ich kann ein Betriebssystem installieren, Systemkonfigurationen vornehmen und seine wichtigsten Funktionen nutzen.
- 2.2.4 Ich kann mich in die Bedienung für mich neuer Software selbstständig einarbeiten. [DC 0.2]

### Reflektieren und Bewerten:

- 2.2.5 Ich kann Software (inklusive Betriebssysteme) zur Bewältigung von Aufgaben bewerten und die Wahl für meinen Lösungsweg begründen. [Dk8 7.1.4], [DC 5.2]

## 2.3 Netzwerke (7 Punkte)

### Wissen und Verstehen:

- 2.3.1 Ich kann Netzwerke und Protokolle beschreiben und ihre Funktions- und Wirkungsweise erklären.
- 2.3.2 Ich kann verschiedene Internetdienste nennen und ihre Einsatzmöglichkeiten und Funktionsweisen beschreiben und erklären.

### Anwenden und Gestalten:

- 2.3.3 Ich kann ein einfaches Computernetzwerk konzipieren, aufbauen, verwalten und nutzen.
- 2.3.4 Ich kann Maßnahmen zur Netzwerksicherheit umsetzen.
- 2.3.5 Ich kann verschiedene Internetdienste nutzen.

### Reflektieren und Bewerten:

- 2.3.6 Ich kann technische Aspekte von Netzwerken hinsichtlich der Verfügbarkeit und Qualität einschätzen.
- 2.3.7 Ich kann die Einsatzmöglichkeiten verschiedener Internetdienste bewerten.

## 2.4 Mensch-Maschine-Schnittstelle (4 Punkte)

### Wissen und Verstehen:

- 2.4.1 Ich kann verschiedene Arten der Mensch-Maschine Schnittstelle beschreiben und die Bedeutung der Barrierefreiheit für Menschen mit besonderen Bedürfnissen erklären. [DC 0.3, 5.2]

### Anwenden und Gestalten:

- 2.4.2 Ich kann verschiedene Mensch-Maschine Schnittstellen sicher und zügig bedienen. [DC 0.3]

- **2.4.3** Ich kann meine digitale Umgebung lokal und im Netz für mich passend gestalten. **[DC 0.3, 5.2]**

### Reflektieren und Bewerten:

- **2.4.4** Ich kann die Benutzerfreundlichkeit von Mensch-Maschine Schnittstellen einschätzen und die Bedeutung für die Anwender bewerten.

## **3 Angewandte Informatik (23 Punkte)**

### **3.1 Produktion digitaler Medien (4 Punkte)**

#### **Wissen und Verstehen:**

- **3.1.1** Ich kann gängige Medienformate und ihre Eigenschaften beschreiben. **[DC 3.1]**
- **3.1.2** Ich kann grundlegende Richtlinien, die bei der Produktion digitaler Medien von Bedeutung sind, erläutern. **[Dk8 4.2.3]**

#### **Anwenden und Gestalten:**

- **3.1.3** Ich kann digitale Medien in Form von Text, Ton, Bildern und Filmen sachgerecht bearbeiten, produzieren und publizieren. **[Dk8 4.2.2, 4.2.4], [DC 3.1]**

#### **Reflektieren und Bewerten:**

- **3.1.4** Ich kann digitale Produkte (Artefakte) in Bezug auf inhaltliche Relevanz, Wirkung und Design einschätzen und bewerten.

### **3.2 Kalkulationsmodelle und Visualisierung (8 Punkte)**

#### **Wissen und Verstehen:**

- **3.2.1** Ich kann Grundbegriffe strukturierter und tabellarisch erfasster Daten und Operationen benennen.
- **3.2.2** Ich kann den (informatischen) Funktionsbegriff erklären.
- **3.2.3** Ich kann digitale Visualisierungsmöglichkeiten beschreiben.

#### **Anwenden und Gestalten:**

- **3.2.4** Ich kann Kalkulationsmodelle zur Lösung von Problemen gestalten und implementieren. **[Dk8 3.4.3]**
- **3.2.5** Ich kann Datenbestände mit entsprechender Software auswerten.
- **3.2.6** Ich kann den Anforderungen und Daten entsprechend visualisieren. **[Dk8 3.4.4]**

#### **Reflektieren und Bewerten:**

- **3.2.7** Ich kann die Korrektheit von Kalkulationsmodellen und Berechnungsmethoden reflektieren und Alternativen prüfen.
- **3.2.8** Ich kann Varianten von Visualisierungen bewerten.

### **3.3 Suche, Auswahl und Organisation von Information (7 Punkte)**

#### **Wissen und Verstehen:**

- **3.3.1** Ich kann wichtige Informationsquellen im Internet anführen, die für meine schulischen und privaten Informationsbedürfnisse nützlich und notwendig sind. [DC 5.2, 5.3]
- **3.3.2** Ich kann lokal und in Netzwerken Methoden der Informationsgewinnung und -organisation benennen. [DC 1.1, 5.3]
- **3.3.3** Ich kann Möglichkeiten grundlegenden digitalen Wissensmanagements beschreiben.

**Anwenden und Gestalten:**

- **3.3.4** Ich kann unter Verwendung passender Dienste und Angebote und Wahl geeigneter Suchmethoden Informationen und Medien gezielt suchen und auswählen. [Dk8 2.1.2], [DC 1.1, 5.3]
- **3.3.5** Ich kann im Rahmen persönlichen Lernmanagements Informationen und Medien strukturiert speichern und verfügbar halten. [Dk8 2.3.1], [DC 1.3]

**Reflektieren und Bewerten:**

- **3.3.6** Ich kann Informationen hinsichtlich ihrer Relevanz und Qualität einschätzen und bewerten. [Dk8 2.2.5], [DC 1.2]
- **3.3.7** Ich kann adäquate Werkzeuge und Methoden der Daten- und Informationsorganisation beurteilen.

**3.4 Kommunikation und Kooperation (4 Punkte)**

**Wissen und Verstehen:**

- **3.4.1** Ich kann wichtige Webanwendungen für den Informationsaustausch und die Zusammenarbeit benennen und ihre Grundlagen erklären. [Dk8 5.1.1, 5.4.1], [DC 2.1, 2.2, 5.3]

**Anwenden und Gestalten:**

- **3.4.2** Ich kann Netzwerke mit geeigneten Webanwendungen zum Informationsaustausch, zur Diskussion und zur Zusammenarbeit sinnvoll und verantwortungsbewusst nutzen. [Dk8 5.1.3, 5.2.2, 5.4.2], [DC 2.1, 2.2, 2.3, 5.3]

**Reflektieren und Bewerten:**

- **3.4.3** Ich kann den situationsgerechten Einsatz von Kommunikations- und Kooperationssystemen bewerten. [Dk8 5.4.4], [DC 2.1, 2.2]
- **3.4.4** Ich kann ihre Bedeutung für mich und die Gesellschaft reflektieren. [DC 2.3]

**4 Praktische Informatik (21 Punkte)**

**4.1 Konzepte der Informationsverarbeitung (3 Punkte)**

**Wissen und Verstehen:**

- **4.1.1** Ich kann wesentliche informatische Konzepte und fundamentale Ideen der Informatik benennen und an Hand von Beispielen erklären. [DC 0.1]

### Anwenden und Gestalten:

- **4.1.2** Ich kann bei der Lösung konkreter Aufgaben Heuristiken, Grundprinzipien und Konzepte der Informatik anwenden und informatische Modelle gestalten.

### Reflektieren und Bewerten:

- **4.1.3** Ich kann unterschiedliche Lösungsansätze in Bezug auf zugrunde liegende Konzepte reflektieren und in konkreten Handlungssituationen bewerten.

## 4.2 Algorithmen, Datenstrukturen und Programmierung (11 Punkte)

### Wissen und Verstehen:

- **4.2.1** Ich kann den Algorithmus-begriff erklären.
- **4.2.2** Ich kann Aufgaben und Problemstellungen algorithmisch und formalsprachlich in geeigneten Datenstrukturen beschreiben. **[Dk8 8.1.4], [DC 3.4]**
- **4.2.3** Ich kann wesentliche Aspekte und Methoden der Softwareentwicklung und des Softwareprojektmanagements erklären.
- **4.2.4** Ich kann wesentliche Aspekte der Prozeduralen, Funktionalen und Objektorientierten Programmierung nennen und an Beispielen erläutern.

### Anwenden und Gestalten:

- **4.2.5** Ich kann Aufgaben mit Mitteln der Informatik modellieren.
- **4.2.6** Ich kann Algorithmen entwerfen, diese formal darstellen, implementieren und testen. **[Dk8 8.1.4, 8.2.3], [DC 3.4]**
- **4.2.7** Ich kann ein Softwareprojekt planen und durchführen. **[Dk8 8.2.1]**

### Reflektieren und Bewerten:

- **4.2.8** Ich kann die Schritte der Softwareentwicklung reflektieren.
- **4.2.9** Ich kann die Angemessenheit der Entwicklungswerkzeuge grob einschätzen.
- **4.2.10** Ich kann die Effizienz von Algorithmen bewerten.
- **4.2.11** Ich kann gezielt nach Programmfehlern suchen und diese korrigieren. **[DC 5.1]**

## 4.3 Datenmodelle und Datenbanksysteme (4 Punkte)

### Wissen und Verstehen:

- **4.3.1** Ich kann den Begriff Datenbanken und wichtige Fachbegriffe beschreiben und an Beispielen erklären.
- **4.3.2** Ich kann Datenbankmodelle, Tabellen und ihre Beziehungsmuster sowie weitere Datenbankobjekte erklären.

### Anwenden und Gestalten:

- **4.3.3** Ich kann Daten strukturiert (in Tabellen) erfassen, abfragen, auswerten sowie Datenbanken modellieren und einfache automatisierte Datenbanklösungen entwickeln.

### Reflektieren und Bewerten:

- **4.3.4** Ich kann Datenmodelle hinsichtlich der Datentypen, Redundanz, Integrität und Relevanz bewerten.

#### **4.4 Intelligente Systeme (3 Punkte)**

##### **Wissen und Verstehen:**

- **4.4.1** Ich kann Bereiche beschreiben, in denen sich Informatiksysteme bzw. Computer intelligent verhalten. Ich kann den Unterschied zwischen menschlicher und maschineller Intelligenz erklären.

##### **Anwenden und Gestalten:**

- **4.4.2** Ich kann intelligente Informatiksysteme anwenden.

##### **Reflektieren und Bewerten:**

- **4.4.3** Ich kann Merkmale menschlicher Intelligenz und künstlicher Intelligenz vergleichen und einschätzen.

## 10.3 Digi.komp8 – inkl. schulauton. Vertiefung (105 Punkte)

### **1 Gesellsch. Aspekte von Medienwandel & Digitalisierung (15 Pkte)**

#### **1.1 Digitalisierung im Alltag (5 Punkte):**

Schülerinnen und Schüler

- **1.1.1** können die Nutzung digitaler Geräte in ihrem persönlichen Alltag gestalten, [Dk12 1.1.2A]
- **1.1.2** reflektieren die eigene Medienbiografie sowie Medienerfahrungen im persönlichen Umfeld,
- **1.1.3** beschreiben mögliche Folgen der zunehmenden Digitalisierung im persönlichen Alltag. [Dk12 1.1.3R]
- **1.1.4 VERTIEFUNG\_2:** kennen die Dynamik und Bedeutung von Werten, Normen und unterschiedlichen Interessen im Hinblick auf die Nutzung von digitalen Medien (ökonomisch, religiös, politisch, kulturell),
- **1.1.5 VERTIEFUNG\_2:** wissen, inwieweit die Nutzung digitaler Technologien der Umwelt schadet oder zum Umweltschutz beiträgt. [DC 4.5]

#### **1.2 Chancen und Grenzen der Digitalisierung (6 Punkte):**

Schülerinnen und Schüler

- **1.2.1** kennen wichtige Anwendungsgebiete der Informationstechnologie und informationstechnologische Berufe, [Dk12 1.1.1W, 1.4.1W]
- **1.2.2** sind sich gesellschaftlicher und ethischer Fragen im Zusammenhang mit technischen Innovationen bewusst,
- **1.2.3** können die gesellschaftliche Entwicklung durch die Teilnahme am öffentlichen Diskurs mitgestalten. [DC 2.3]
- **1.2.4 VERTIEFUNG:** erkennen die Wechselwirkungen zwischen Natur, Technik und Gesellschaft
- **1.2.5 VERTIEFUNG:** erkennen Chancen und Risiken der Mediennutzung und geschlechtsspezifische Aspekte.
- **1.2.6 VERTIEFUNG\_2:** erkennen Entwicklungen, die eine Gefahr für Chancengleichheit bei der Nutzung von Informationstechnologien darstellen, und nennen Handlungsoptionen.

#### **1.3 Gesundheit und Wohlbefinden (3 Punkte):**

Schülerinnen und Schüler

- **1.3.1** reflektieren, welche gesundheitlichen Probleme die übermäßige Nutzung von digitalen Medien nach sich ziehen kann,
- **1.3.2** vermeiden Gesundheitsrisiken und Bedrohungen für das körperliche und seelische Wohlbefinden in Bezug auf digitale Technologien. [DC 4.3]
- **1.3.3 VERTIEFUNG:** erkennen, wie digitale Technologien soziales Wohlbefinden und Inklusion fördern. [DC 0.3]



#### 1.4 Geschichtliche Entwicklung (1 Punkt):

- **1.4.1 VERTIEFUNG\_2:** kennen die geschichtliche Entwicklung der Informations- und Kommunikationstechnologie und Informatik insb. Von Social Media unter Berücksichtigung menschenrechtlicher und ethischer Fragestellungen. [Dk12 1.3.1W]

### 2 Informations-, Daten- und Medienkompetenz (12 Punkte)

#### 2.1 Suchen und Finden (2 Punkte):

Schülerinnen und Schüler

- **2.1.1** formulieren ihre Bedürfnisse für die Informationssuche, [DC 1.1]
- **2.1.2** planen zielgerichtet und selbstständig die Suche nach Informationen, Daten und digitalen Inhalten mit Hilfe geeigneter Strategien und Methoden (z. B. Suchbegriffe), passender Werkzeuge bzw. nützlicher Quellen. [Dk12 3.3.4A]

#### 2.2 Vergleichen und bewerten (6 Punkte):

Schülerinnen und Schüler

- **2.2.1** wenden Kriterien an, um die Glaubwürdigkeit und Verlässlichkeit von Quellen zu bewerten (Quellenkritik, Belegbarkeit von Wissen), [DC 1.2], [Dk12 1.2.6R]
- **2.2.2** erkennen und reflektieren klischeehafte Darstellungen und Zuschreibungen in der medialen Vermittlung, [DC 1.2]
- **2.2.3** können mit automatisiert aufbereiteten Informationsangeboten eigenverantwortlich umgehen.
- **2.2.4 VERTIEFUNG:** erkennen unterschiedliche, auch widersprüchliche Wahrheitsansprüche,
- **2.2.5 VERTIEFUNG:** vergleichen, analysieren und bewerten Informationen und digitale Inhalte kritisch (manipulative und monoperspektivische Darstellungen). [DC 1.2], [Dk12 3.3.6R]
- **2.2.6 VERTIEFUNG\_2:** entwickeln ein Verständnis für die Konstruktion von Medienwirklichkeit durch die Erhebung und Analyse von Informationen und Daten bzw. die Mechanismen der Bild- und Datenmanipulation.

#### 2.3 Organisieren (1 Punkt):

Schülerinnen und Schüler

- **2.3.1** speichern Informationen, Daten und digitale Inhalte sowohl im passenden Format als auch in einer sinnvollen Struktur, in der diese gefunden und verarbeitet werden können. [DC 1.3], [Dk12 3.3.5A]

#### 2.4 Teilen (3 Punkte):

Schülerinnen und Schüler

- **2.4.1** teilen Informationen, Daten und digitale Inhalte mit anderen durch geeignete digitale Technologien, **[DC 2.2]**
- **2.4.2** kennen die Grundzüge des Urheberrechts sowie des Datenschutzes (insb. das Recht am eigenen Bild) und wenden diese Bestimmungen an. **[Dk12 1.2.1W, 1.2.2A]**
- **2.4.3 VERTIEFUNG:** kennen Lizenzmodelle, insb. Offene (Creative Commons, Open Educational Resources). **[DC 3.3]**

### **3 Betriebssysteme und Standard-Anwendungen (17 Punkte)**

#### **3.1 Grundlagen des Betriebssystems (2 Punkte):**

Schülerinnen und Schüler

- **3.1.1** nutzen die zum Normalbetrieb notwendigen Funktionen eines Betriebssystems einschließlich des Dateimanagements sowie der Druckfunktion. **[DC 0.2], [Dk12 2.1.3A]**
- **3.1.2 VERTIEFUNG\_2:** kennen die wichtigsten Aufgaben eines Betriebssystems und die wichtigsten Betriebssysteme. **[DC 0.1], [Dk12 2.2.1W]**

#### **3.2 Textverarbeitung (3 Punkte):**

Schülerinnen und Schüler

- **3.2.1** geben Texte zügig ein, **[DC 0.2]**
- **3.2.2** strukturieren und formatieren Texte unter Einbeziehung von Bildern, Grafiken und anderen Objekten, **[DC 0.2]**
- **3.2.3** führen Textkorrekturen durch (ggf. unter Zuhilfenahme von Überarbeitungsfunktionen, Rechtschreibprüfung oder Wörterbuch). **[DC 0.2]**

#### **3.3 Präsentationssoftware (6 Punkte):**

Schülerinnen und Schüler

- **3.3.1** gestalten Präsentationen unter Einbeziehung von Bildern, Grafiken und anderen Objekten, **[DC 0.2]**
- **3.3.2** beachten Grundregeln der Präsentation (z. B. aussagekräftige Bilder, kurze Texte). **[DC 0.2]**
- **3.3.3 VERTIEFUNG:** kennen unterschiedliche Präsentationsansichten und wissen, wann man diese einsetzt,
- **3.3.4 VERTIEFUNG:** nutzen verschiedene Folienlayouts und Foliendesigns. **[DC 0.2]**
- **3.3.5 VERTIEFUNG\_2:** erstellen und formatieren Diagramme, **[DC 0.2]**
- **3.3.6 VERTIEFUNG\_2:** fügen Effekte wie Animation und Übergang zu Präsentationen hinzu. **[DC 0.2]**

#### **3.4 Tabellenkalkulation (6 Punkte):**

Schülerinnen und Schüler

- **3.4.1** beschreiben den grundlegenden Aufbau einer Tabelle, [DC 0.2]
- **3.4.2** legen Tabellen an, ändern und formatieren diese, [DC 0.2]
- **3.4.3** führen mit einer Tabellenkalkulation einfache Berechnungen durch und lösen altersgemäße Aufgaben, [DC 0.2], [Dk12 3.2.4A]
- **3.4.4** stellen Zahlenreihen in geeigneten Diagrammen dar [DC 0.2], [Dk12 3.2.6A]
- **3.4.5 VERTIEFUNG:** erfassen Daten; speichern, ändern und sortieren diese, [DC 0.2]
- **3.4.6 VERTIEFUNG:** suchen gezielt nach Daten und selektieren diese. [DC 0.2]

## **4 Mediengestaltung (13 Punkte)**

### **4.1 Digitale Medien rezipieren (5 Punkte):**

Schülerinnen und Schüler

- **4.1.1** kennen mediale Gestaltungselemente und können medien spezifische Formen unterscheiden,
- **4.1.2** erkennen Medien als Wirtschaftsfaktor (z. B. Finanzierung, Werbung),
- **4.1.3** nehmen die Gestaltung digitaler Medien und damit verbundenes kommunikatives Handeln reflektiert wahr: den Zusammenhang von Inhalt und Gestaltung (z. B. Manipulation), problematische Inhalte (z. B. sexualisierte, gewaltverherrlichende) sowie stereotype Darstellungen in Medien.
- **4.1.4 VERTIEFUNG:** analysieren Interessen und Bedingungen der Medienproduktion und Medienverbreitung.
- **4.1.5 VERTIEFUNG\_2:** erkennen und benennen Medieneinflüsse und Wertvorstellungen

### **4.2 Digitale Medien produzieren (6 Punkte):**

Schülerinnen und Schüler

- **4.2.1** erleben sich selbstwirksam, indem sie digitale Technologien kreativ und vielfältig nutzen,
- **4.2.2** gestalten digitale Medien mittels aktueller Technologien, ggf. unter Einbeziehung anderer Medien: Texte, Präsentationen, Audiobeiträge, Videobeiträge sowie multimediale Lernmaterialien, [DC 3.2], [Dk12 3.1.3A]
- **4.2.3** beachten Grundregeln der Mediengestaltung, [Dk12 3.1.2W]
- **4.2.4** veröffentlichen Medienprodukte in geeigneten Ausgabeformaten auf digitalen Plattformen (z. B. Blog). [Dk12 3.1.3A]
- **4.2.5 VERTIEFUNG\_2:** setzen Wissen über Techniken und Ästhetiken populärer Medienkulturen eigenverantwortlich um, [DC 3.1]
- **4.2.6 VERTIEFUNG\_2:** planen die Produktion von Medien hinsichtlich Inhalt, Format und Zielgruppe.

### 4.3 Inhalte weiterentwickeln (2 Punkte):

Schülerinnen und Schüler

- **4.3.1** können Informationen und Inhalte aktualisieren, verbessern sowie zielgruppen-, medienformat- und anwendungsgerecht aufarbeiten. [DC 2.1]
- **4.3.2 VERTIEFUNG\_2:** binden Informationen inhaltlich, organisatorisch und sprachlich in bestehende Wissensorganisationsformate ein.

## 5 Digitale Kommunikation und Social Media (17 Punkte)

### 5.1 Interagieren und kommunizieren (7 Punkte):

Schülerinnen und Schüler

- **5.1.1** kennen verschiedene digitale Kommunikationswerkzeuge, [Dk12 3.4.1W]
- **5.1.2** beschreiben Kommunikationsbedürfnisse und entsprechende Anforderungen an digitale Kommunikationswerkzeuge,
- **5.1.3** schätzen die Auswirkungen des eigenen Verhaltens in virtuellen Welten ab und verhalten sich entsprechend, [Dk12 3.4.2A]
- **5.1.4** erkennen problematische Mitteilungen und nutzen Strategien, damit umzugehen (z. B. Cybermobbing, Hasspostings).
- **5.1.5 VERTIEFUNG:** wählen zielgerichtet geeignete digitale Technologien für konkrete Kommunikationsszenarien aus und bedenken bei der Auswahl die Interessen der Anbieter von Social Media, den Einfluss von Social Media auf ihre Wahrnehmung der Welt und Art und Umfang der Daten, die durch die Nutzung entstehen.
- **5.1.6 VERTIEFUNG\_2:** adaptieren Kommunikationsstrategien für spezifische Zielgruppen, [DC 2.1]
- **5.1.7 VERTIEFUNG\_2:** wenden Verhaltensregeln für die Nutzung digitaler Technologien und zur Interaktion in digitalen Umgebungen an („Netiquette“). [DC 2.5], [Dk12 1.2.2A]

### 5.2 An der Gesellschaft teilhaben (2 Punkte):

Schülerinnen und Schüler

- **5.2.1** begreifen das Internet als öffentlichen Raum und erkennen damit verbundenen Nutzen und Risiken.
- **5.2.2 VERTIEFUNG:** nutzen die demokratische Kommunikationskultur durch öffentliche Äußerungen unter Verwendung digitaler Technologien. [Dk12 4.3.2A]

### 5.3 Digitale Identitäten gestalten (4 Punkte):

Schülerinnen und Schüler

- **5.3.1** gestalten und schützen eigene digitale Identitäten reflektiert, [DC 2.6]

- **5.3.2** erkennen Manipulationsmöglichkeiten durch digitale Identitäten (z. B. Grooming),
- **5.3.3** verfolgen den Ruf eigener digitaler Identitäten und schützen diesen.
- **5.3.4 VERTIEFUNG\_2:** entwickeln ein Bewusstsein für die Pluralität von Onlineidentitäten und die Differenz zur eigenen Persönlichkeit.

#### **5.4 Zusammenarbeiten (4 Punkte):**

Schülerinnen und Schüler

- **5.4.1** wissen, wie cloudbasierte Systeme grundsätzlich funktionieren und achten auf kritische Faktoren (z. B. Standort des Servers, Datensicherung), [\[Dk12 3.4.1W\]](#)
- **5.4.2** nutzen verantwortungsvoll passende Werkzeuge und Technologien (etwa Wiki, cloudbasierte Werkzeuge, Lernplattform, ePortfolio). [\[Dk12 3.4.2A\]](#)
- **5.4.3 VERTIEFUNG\_2:** formulieren Bedürfnisse für die gemeinsame Erarbeitung von Inhalten und Wissen mit Hilfe digitaler Technologien,
- **5.4.4 VERTIEFUNG\_2:** wählen zielgerichtet geeignete Werkzeuge und Technologien für Prozesse der Zusammenarbeit aus. [\[Dk12 3.4.3R\]](#)

### **6 Sicherheit (8 Punkte)**

#### **6.1 Geräte und Inhalte schützen (4 Punkte):**

Schülerinnen und Schüler

- **6.1.1** sind sich Risiken und Bedrohungen in digitalen Umgebungen bewusst
- **6.1.2** überprüfen den Schutz ihrer digitalen Geräte und wenden sich im Bedarfsfall an die richtigen Stellen, [\[Dk12 1.2.3A\]](#)
- **6.1.3** treffen entsprechende Vorkehrungen, um ihre Geräte und Inhalte vor Viren bzw. Schadsoftware/Malware zu schützen. [\[DC 4.1\]](#), [\[Dk12 1.2.3A\]](#)
- **6.1.4 VERTIEFUNG\_2:** verwenden Software zur Verschlüsselung von Daten.

#### **6.2 Persönliche Daten und Privatsphäre schützen (4 Punkte):**

Schülerinnen und Schüler

- **6.2.1** verstehen, wie persönlich nachvollziehbare Informationen verwendet und geteilt werden können, [\[Dk12 1.2.1W\]](#)
- **6.2.2** treffen Vorkehrungen, um ihre persönlichen Daten zu schützen, [\[DC 4.2\]](#), [\[Dk12 1.2.2A\]](#)
- **6.2.3** kennen Risiken, die mit Geschäften verbunden sind, die im Internet abgeschlossen werden. [\[DC 4.4\]](#)

- **6.2.4 VERTIEFUNG\_2:** verstehen, wie Anbieter digitaler Services darüber informieren, auf welche Art und Weise persönliche Daten verwendet werden.

## **7 Technische Problemlösung (12 Punkte)**

### **7.1 Technische Bedürfnisse und entsprechende Möglichkeiten**

#### **identifizieren (5 Punkte):**

Schülerinnen und Schüler

- **7.1.1** kennen die Bestandteile und Funktionsweise eines Computers und eines Netzwerks, [\[Dk12 2.1.1W\]](#)
- **7.1.2** kennen gängige proprietäre und offene Anwendungsprogramme und zugehörige Dateitypen. [\[Dk12 2.2.2W\]](#)
- **7.1.3 VERTIEFUNG:** formulieren Bedürfnisse für den Einsatz digitaler Geräte,
- **7.1.4 VERTIEFUNG:** bewerten mögliche technologische Lösungen und wählen eine passende aus, auch unter Berücksichtigung proprietärer und freier Software. [\[Dk12 2.2.5R\]](#)
- **7.1.5 VERTIEFUNG\_2:** passen digitale Umgebungen an die eigenen Bedürfnisse an und treffen persönliche Einstellungen (z. B. barrierefreie Einstellungen im Betriebssystem).

#### **7.2 Digitale Geräte nutzen (4 Punkte):**

Schülerinnen und Schüler

- **7.2.1** schließen die wichtigsten Komponenten eines Computers richtig zusammen und identifizieren Verbindungsfehler, [\[Dk12 2.1.4A\]](#)
- **7.2.2** verbinden digitale Geräte mit einem Netzwerk und tauschen Daten zwischen verschiedenen elektronischen Geräten aus.
- **7.2.3 VERTIEFUNG:** nutzen unterschiedliche digitale Geräte entsprechend ihrer Einsatzmöglichkeiten, [\[DC 5.2\]](#), [\[Dk12 2.1.3A\]](#)
- **7.2.4 VERTIEFUNG:** nutzen verschiedene Arten von Speichermedien und Speichersystemen.

#### **7.3 Technische Probleme lösen (3 Punkte):**

Schülerinnen und Schüler

- **7.3.1** erkennen technische Probleme in der Nutzung von digitalen Geräten und melden eine konkrete Beschreibung des Fehlers an die richtigen Stellen. [\[DC 5.1\]](#), [\[Dk12 2.1.6R\]](#)
- **7.3.2 VERTIEFUNG:** nutzen Hilfesysteme bei der Problemlösung,
- **7.3.3 VERTIEFUNG:** führen Datensicherungen und -Wiederherstellungen aus.

## **8 Computational Thinking (11 Punkte)**

### 8.1 Mit Algorithmen arbeiten (7 Punkte):

Schülerinnen und Schüler

- **8.1.1** nennen und beschreiben Abläufe aus dem Alltag,
- **8.1.2** verwenden, erstellen und reflektieren Codierungen (z. B. Geheimschrift, QR-Code), **[DC 0.1]**
- **8.1.3** vollziehen eindeutige Handlungsanleitungen (Algorithmen) nach und führen diese aus, **[DC 3.4]**
- **8.1.4** formulieren eindeutige Handlungsanleitungen (Algorithmen) verbal und schriftlich. **[DC 3.4], [Dk12 4.2.2W, 4.2.6A]**
- **8.1.5 VERTIEFUNG:** entdecken Gemeinsamkeiten und Regeln (Muster) in Handlungsanleitungen, **[DC 0.1]**
- **8.1.6 VERTIEFUNG:** erkennen die Bedeutung von Algorithmen in automatisierten digitalen Prozessen (z. B. automatisiertes Vorschlagen von potenziell interessanten Informationen).
- **8.1.7 VERTIEFUNG\_2:** können intuitiv nutzbare Benutzeroberflächen und dahinterstehende technische Abläufe einschätzen.

### 8.2 Kreative Nutzung von Programmiersprachen (4 Punkte):

Schülerinnen und Schüler

- **8.2.1** erstellen einfache Programme oder Webanwendungen mit geeigneten Tools, um ein bestimmtes Problem zu lösen oder eine bestimmte Aufgabe zu erfüllen, **[DC 5.3], [Dk12 4.2.7A]**
- **8.2.2** kennen unterschiedliche Programmiersprachen und Produktionsabläufe.
- **8.2.3 VERTIEFUNG:** beherrschen grundlegende Programmierstrukturen (Verzweigung, Schleifen, Prozeduren). **[Dk12 4.2.6A]**
- **8.2.4 VERTIEFUNG\_2:** reflektieren die Grenzen und Möglichkeiten von Simulationen.

## 10.4 Digi.komp4 – Kompetenzmodell (49 Punkte)

### **1 Informationstechnologie, Mensch und Gesellschaft (16 Punkte)**

#### **1.1 Bedeutung von IT in der Lebenswelt der Kinder (3 Punkte)**

- Ich kann wichtige Anwendungsgebiete der Informationstechnologie aus der Lebensumwelt anführen.
- Ich kann Bereiche nennen, in denen Computer Menschen nicht ersetzen können.
- Ich denke über meine Nutzung digitaler Medien nach und kann darüber mit meinen Eltern und Lehrpersonen sprechen

#### **1.2 Verantwortung bei der Nutzung von IT (8 Punkte)**

- Ich kann reale und virtuelle Welten unterscheiden.
- Ich kann mein digitales Ich im Web gestalten.
- Ich weiß, dass ich im Internet Spuren hinterlasse und identifizierbar bin. Daher verhalte ich mich entsprechend.
- Ich kenne grundlegende Rechte und Pflichten im Umgang mit eigenen und fremden Daten.
- Ich beachte das Urheberrecht (Musik, Film, Bilder, Texte, Software) und das Recht auf den Schutz persönlicher Daten, insbesondere das Recht am eigenen Bild.
- Ich kenne Risiken bei der Nutzung von Informationstechnologien und weiß, wie ich mich im gegebenen Fall verhalten soll.
- Ich kenne mögliche Gefahren im Umgang mit Personen, die ich nur aus dem Internet kenne und kann mir Hilfe holen.
- Ich weiß, dass auch im Internet Geschäfte abgeschlossen werden und damit Risiken verbunden sind.

#### **1.3 Datenschutz und Datensicherheit (3 Punkte)**

- Ich weiß, dass es Bedrohungen wie Schadprogramme gibt, besonders wenn ich Daten austausche oder das Internet nutze.
- Ich kenne Möglichkeiten zum Schutz meines Computers und weiß, an wen ich mich im Bedarfsfall wenden kann.
- Ich weiß, dass es Daten gibt, zu denen ich mir keinen Zugriff verschaffen darf und dass missbräuchlicher Zugriff in manchen Fällen strafbar ist.

#### **1.4 Entwicklung und berufliche Perspektiven (2 Punkte)**

- Ich kann einige Berufe nennen, in denen Computersysteme wichtig sind.
- Ich kenne die geschichtliche Entwicklung der Kommunikationstechnologie in groben Umrissen.

### **2 Informatiksysteme (13 Punkte)**

#### **2.1 Technische Bestandteile und deren Einsatz (2 Punkte)**



- Ich kann digitale Geräte des täglichen Lebens benennen und verantwortungsvoll verwenden.
- Ich kann Speichermedien nennen und nutzen.

## **2.2 Gestaltung und Nutzung persönlicher Informatiksysteme (7 Punkte)**

- Ich verwende digitale Geräte und Internet beim Lernen.
- Ich kann einen Computer starten und herunterfahren.
- Ich kann mich ordnungsgemäß an- und abmelden.
- Ich kann Programme starten und darin arbeiten.
- Ich kann Dateien in einem Ordnungssystem speichern, wiederfinden und öffnen.
- Ich kann Dateien einfügen, verschieben, kopieren und löschen.
- Ich kann Plattformen, die für mich geeignet sind, nutzen.

## **2.3 Datenaustausch in Netzwerken (3 Punkte)**

- Ich kann Netzwerke zum Suchen und Darstellen von Informationen nutzen.
- Ich kann Netzwerke zum Kommunizieren nutzen.
- Ich kann Netzwerke zur Zusammenarbeit nutzen.

## **2.4 Mensch-Maschine-Schnittstelle (1 Punkt)**

- Ich weiß, dass digitale Geräte unterschiedlich zu bedienen sind und kann sie im täglichen Leben nutzen.

## **3 Anwendungen (15 Punkte)**

### **3.1 Dokumentation, Publikation und Präsentation (6 Punkte)**

- Ich kann Texte eingeben und diese formatieren.
- Ich kann Elemente kopieren, einfügen, verschieben und löschen.
- Ich kann Texte korrigieren und, wenn nötig, Rechtschreibhilfen verwenden.
- Ich kann meine Arbeiten mit Bildern und Grafiken gestalten und medial präsentieren.
- Ich kann digitale Zeichnungen und Bilder erstellen und gestalten.
- Ich kann digitale Audio- und Videodateien nutzen.

### **3.2 Berechnung und Visualisierung (4 Punkte)**

- Ich verstehe den Aufbau einer Tabelle.
- Ich kann eine Tabelle erstellen und gestalten.
- Ich kann altersgemäße Berechnungen durchführen.
- Ich kann ein Diagramm erstellen.

### **3.3 Suche, Auswahl und Organisation von Information (2 Punkte)**

- Ich kenne Suchmaschinen für Kinder und kann sie verwenden.
- Ich kann Informationen aus dem Internet in meinen Arbeiten nutzen.

### **3.4 Kommunikation und Kooperation (3 Punkte)**

- Ich kann digitale Nachrichten schreiben, versenden und empfangen.

- Ich beachte Umgangsformen im Internet.
- Ich nutze digitale Werkzeuge zur Zusammenarbeit.

#### **4 Konzepte (5 Punkte)**

##### **4.1 Darstellung von Information (1 Punkt)**

- Ich kann einige Informationen aus dem Alltag verschlüsseln und entschlüsseln.

##### **4.2 Strukturieren von Daten (1 Punkt)**

- Ich kann Daten erfassen, speichern und ändern.

##### **4.3 Automatisierung von Handlungsanweisungen (2 Punkte)**

- Ich kann einfache Anleitungen verstehen und ausführen.
- Ich kann einfache Anleitungen erstellen.

##### **4.4 Koordination und Steuerung von Abläufen (1 Punkt)**

- Ich weiß, dass ein Computerprogramm entsteht, indem Anweisungen aneinandergereiht werden.

## 10.5 Fragebogen

Dieser Fragebogen wird von den SchülerInnen jeweils am Ende des Unterrichtsprojektes bzw. am Ende des Schuljahres ausgefüllt:

**Fragebogen - Informatikunterricht**

**Statistische Daten**

Geschlecht				weiblich <input type="radio"/>		männlich <input type="radio"/>	
Alter	jünger <input type="radio"/>	14 <input type="radio"/>	15 <input type="radio"/>	16 <input type="radio"/>	17 <input type="radio"/>	18 <input type="radio"/>	älter <input type="radio"/>
Ich habe zu Hause oder am Smartphone einen Internetzugang				Ja <input type="radio"/>		Nein <input type="radio"/>	

**Computernutzung**

Meine Computerkenntnisse bewegen sich auf folgendem Niveau:	Anfänger <input type="radio"/>	Anwender <input type="radio"/>	Fortgeschrittener <input type="radio"/>	Power-user <input type="radio"/>	Experte <input type="radio"/>
Wie viele Stunden verbringst du pro Woche in deiner Freizeit vor dem Computer?	bis zu 5h <input type="radio"/>	5+ bis 10h <input type="radio"/>	10+ bis 15h <input type="radio"/>	15+ bis 20h <input type="radio"/>	20h+ <input type="radio"/>
Wieviele Stunden pro Woche verbringst du mit ...					
Social Networks (Facebook, Netlog,...), Chatten (WhatsApp), Forumdiskussionen, e-mail	gar keine <input type="radio"/>	wenig bis 2h <input type="radio"/>	2 bis 6h <input type="radio"/>	6 bis 10h <input type="radio"/>	mehr als 10h <input type="radio"/>
Spielen (PC, Smartphone, Konsole), inklusive Flashgames, Facebookspiele	gar keine <input type="radio"/>	wenig bis 2h <input type="radio"/>	2 bis 6h <input type="radio"/>	6 bis 10h <input type="radio"/>	mehr als 10h <input type="radio"/>
Tätigkeiten für die Schule	gar keine <input type="radio"/>	wenig bis 2h <input type="radio"/>	2 bis 6h <input type="radio"/>	6 bis 10h <input type="radio"/>	mehr als 10h <input type="radio"/>
Programmieren aus eigenem Interesse (also nicht für die Schule)	gar keine <input type="radio"/>	wenig bis 2h <input type="radio"/>	2 bis 6h <input type="radio"/>	6 bis 10h <input type="radio"/>	mehr als 10h <input type="radio"/>
Internetsurfen: Lesen von Online-Zeitschriften, Nachrichten	gar keine <input type="radio"/>	wenig bis 2h <input type="radio"/>	2 bis 6h <input type="radio"/>	6 bis 10h <input type="radio"/>	mehr als 10h <input type="radio"/>
Sonstige Tätigkeiten am Computer	gar keine <input type="radio"/>	wenig bis 2h <input type="radio"/>	2 bis 6h <input type="radio"/>	6 bis 10h <input type="radio"/>	mehr als 10h <input type="radio"/>

## KAPITEL: Anhang

### Spielerprogrammierung

Ein Computerspiel im Unterricht zu entwickeln finde ich interessant!	Trifft völlig zu <input type="radio"/>	Trifft eher zu <input type="radio"/>	Weder noch <input type="radio"/>	Trifft wenig zu <input type="radio"/>	Trifft nicht zu <input type="radio"/>
<b>Durch die Spielerprogrammierung ...</b>					
... verstehe ich besser wie Programme ablaufen!	Trifft völlig zu <input type="radio"/>	Trifft eher zu <input type="radio"/>	Weder noch <input type="radio"/>	Trifft wenig zu <input type="radio"/>	Trifft nicht zu <input type="radio"/>
... habe ich gelernt besser zu programmieren!	Trifft völlig zu <input type="radio"/>	Trifft eher zu <input type="radio"/>	Weder noch <input type="radio"/>	Trifft wenig zu <input type="radio"/>	Trifft nicht zu <input type="radio"/>
... verstehe ich besser was ein Algorithmus ist!	Trifft völlig zu <input type="radio"/>	Trifft eher zu <input type="radio"/>	Weder noch <input type="radio"/>	Trifft wenig zu <input type="radio"/>	Trifft nicht zu <input type="radio"/>
... verstehe ich besser wie Softwareprojekte ablaufen!	Trifft völlig zu <input type="radio"/>	Trifft eher zu <input type="radio"/>	Weder noch <input type="radio"/>	Trifft wenig zu <input type="radio"/>	Trifft nicht zu <input type="radio"/>
... welche Probleme beim Entwickeln von Software auftreten!	Trifft völlig zu <input type="radio"/>	Trifft eher zu <input type="radio"/>	Weder noch <input type="radio"/>	Trifft wenig zu <input type="radio"/>	Trifft nicht zu <input type="radio"/>
Die Aussicht ein Spiel programmieren zu können hat mich motiviert!	Trifft völlig zu <input type="radio"/>	Trifft eher zu <input type="radio"/>	Weder noch <input type="radio"/>	Trifft wenig zu <input type="radio"/>	Trifft nicht zu <input type="radio"/>
Die tatsächliche Spielerprogrammierung hat mich motiviert!	Trifft völlig zu <input type="radio"/>	Trifft eher zu <input type="radio"/>	Weder noch <input type="radio"/>	Trifft wenig zu <input type="radio"/>	Trifft nicht zu <input type="radio"/>

### Vergleich zu herkömmlichen Informatikunterricht

Spielerprogrammierung hat im Vergleich zum herkömmlichen Informatikunterricht mir in folgenden Bereichen "mehr" oder "weniger" gebracht:					
Teamwork	viel weniger <input type="radio"/>	weniger <input type="radio"/>	gleich viel <input type="radio"/>	mehr <input type="radio"/>	viel mehr <input type="radio"/>
Projektmanagementfähigkeiten	viel weniger <input type="radio"/>	weniger <input type="radio"/>	gleich viel <input type="radio"/>	mehr <input type="radio"/>	viel mehr <input type="radio"/>
Verwenden von Anwendungsprogrammen	viel weniger <input type="radio"/>	weniger <input type="radio"/>	gleich viel <input type="radio"/>	mehr <input type="radio"/>	viel mehr <input type="radio"/>
Programmieren	viel weniger <input type="radio"/>	weniger <input type="radio"/>	gleich viel <input type="radio"/>	mehr <input type="radio"/>	viel mehr <input type="radio"/>
Verstehen von Datenstrukturen	viel weniger <input type="radio"/>	weniger <input type="radio"/>	gleich viel <input type="radio"/>	mehr <input type="radio"/>	viel mehr <input type="radio"/>
Verständnis von Algorithmen	viel weniger <input type="radio"/>	weniger <input type="radio"/>	gleich viel <input type="radio"/>	mehr <input type="radio"/>	viel mehr <input type="radio"/>
Einblick in die Logik von Programmen	viel weniger <input type="radio"/>	weniger <input type="radio"/>	gleich viel <input type="radio"/>	mehr <input type="radio"/>	viel mehr <input type="radio"/>
Spaß	viel weniger <input type="radio"/>	weniger <input type="radio"/>	gleich viel <input type="radio"/>	mehr <input type="radio"/>	viel mehr <input type="radio"/>
Motivation	viel weniger <input type="radio"/>	weniger <input type="radio"/>	gleich viel <input type="radio"/>	mehr <input type="radio"/>	viel mehr <input type="radio"/>

Wie hast du die Spieleprogrammierung im Informatikunterricht erlebt?	<input type="text"/>
Kann deiner Meinung Spieleprogrammierung die Motivation steigern?	<input type="text"/>
Ist das für die Spieleprogrammierung der Fall gewesen? (Warum?/Warum nicht?)	<input type="text"/>
Sollte die Lehrperson eine noch genauere Struktur für den Projektablauf vorgeben?	<input type="text"/>
Was ich sonst noch mitteilen möchte:	<input type="text"/>
<b>Fragebogen abschicken!</b>	
<input type="button" value="Abschicken!"/>	

---

## 10.6 Interviewvereinbarung

Name des/der Interviewten:

Anschrift:

Tel.:

Email:

Ort des Interviews:

Datum des Interviews:

Ich, der/die Interviewte, erkläre

Ich habe an dem oben genannten Interview teilgenommen und war mit der Aufzeichnung des Interviews auf Audioband per Smartphone einverstanden. Ich überlasse Herrn Mario Winkler für die Erarbeitung seiner Diplomarbeiten-These als Schenkung alle Nutzungsrechte an den im Rahmen des Interviews entstandenen Dokumenten und stimme einer Verwendung für ausschließlich wissenschaftliche Zwecke in anonymisierter Form zu.

Datum:

---

Unterschrift InterviewpartnerIn

---

Unterschrift Interviewer

---

## 10.7 Interviews

---

### 10.7.1 Interview 01

---

1 [0:00:00] **I:** Beschreiben Sie bitte Ihre Erfahrungen mit dem Unterrichtsprojekt  
2 "learn to proGrAME".  
3  
4  
5 [0:00:05] **B:** Also grundlegend, wenn man als Informatik Lehrer oder Lehrerinnen  
6 in eine Schulklasse geht, um zu unterrichten. Da fragen die Lernenden sehr  
7 schnell einmal nach drei speziellen Bereichen der Informatik: Wann lernen wir  
8 hacken, programmieren wir auch Viren und können wir unsere eigenen  
9 Computerspiele entwickeln. Während die ersten und zweiten Fragen tatsächlich  
10 in einzelnen Unterrichtsstunden mehr theoretisch abgehandelt werden, also in  
11 Form von Sicherheitsbewusstsein, Security und wie man Netzwerke schützt  
12 bzw. bei den Viren in ähnlichen Themenbereichen über Sicherheit-Verbreitung  
13 und auch Programmieren geredet wird, ist das Game Programming ganz  
14 besonders reizvoll, weil da kann man wirklich mit den Lernenden in medias res  
15 gehen und mit ihnen Computerspiele entwickeln. Was so interessant ist am  
16 Projekt "learn to proGrAME" ist, dass man eine professionelle  
17 Spielentwicklungsumgebung verwendet. Bisher ist sehr erfolgreich die  
18 Einstiegs-Programmierungsumgebung Scratch verwendet worden und professionelle  
19 Kits wie Unity eher im universitären Bereich zum Einsatz gekommen und auch  
20 die Erforschung des Programmierens mit Unity war ganz, ganz dünn gesät zum  
21 Projektbeginn. Was aber wesentlich bei den Lernenden auszumachen war, es  
22 ist irgendwie so ein Zug zu diesen professionellen Tools und das was man im  
23 Englischen als getting the real deal beschreiben wird, ganz stark zu spüren und  
24 wenn man da noch interessante Physikeffekte einbauen kann mit wenig  
25 Aufwand, dann fesselt das die Jugendlichen sowohl Mädchen als auch  
26 Burschen und man kann also unserer Meinung nach den Unterricht sehr  
27 motivierend und engagierend, also auch für die Schülerinnen sehr  
28 Schülerinnen-orientiert vermitteln und durchführen. Gut und falls ich noch ein  
29 wenig zu "learn to proGrAME" weitererzählen kann, also wir haben dann mit

30 mehreren Schulklassen, das waren 14- bis 15-jährige Schülerinnen und  
31 Schülern tatsächlich Tutorials zur Computerspielentwicklung gemacht. Sie  
32 haben am Anfang dieses „Bouncy Fant“ Tutorial durchlaufen und das Projekt ist  
33 gewachsen. Es sind noch mehr Tutorials dazugekommen. Also Textadventures,  
34 Ellis großes Abenteuer oder das Asteroids-Tutorial mit Partikelsystemen und  
35 die Lernenden haben wirklich in acht bis 16 Doppelstunden mit dem  
36 professionellen Toolkit ihre eigenen Computerspiele entwickelt. Später auch  
37 dann modifiziert. Gibt es noch irgendwas im Detail, was sie interessiert? Zum  
38 Projekt selbst?

39

40 [0:02:49] **I:** Nein da jetzt nicht, so allgemein jetzt nicht. Aber wollen Sie sonst  
41 noch etwas weitererzählen?

42

43 [0:03:03] **B:** Jetzt habe ich schon durch das viele erzählen ein bisschen den  
44 Faden verloren. Was war das Kernelement?

45

46 [0:03:16] **I:** Nur die Erfahrungen damit und die waren eigentlich enthalten.

47

48 [0:03:24] **B:** Zu den Erfahrungen kann ich sagen: Also die Erfahrungen, die wir  
49 gemacht haben, sind, das Spieleentwicklung mit Unity eine hochkomplexe  
50 herausfordernde Angelegenheit ist. Mit einer ganz herausfordernden Lernkurve,  
51 also nicht so eine flache Lernkurve wie bei Scratch. Aber das die  
52 Programmieraktivitäten sehr gut von den Lernenden angenommen worden sind.  
53 Das Feedback war aber schon auch und die Erfahrung durchwegs, dass es  
54 tatsächlich herausfordernd ist und das gegenseitiger Support und diese  
55 Hilfestellungen eine wichtige Rolle spielen, um etwas zu erreichen. Und dass  
56 man ohne detaillierte Lernmaterialien das weder allein zu Hause machen  
57 könnte noch gut in der Schule bewältigen kann. Für die Schule ist wichtig, ein  
58 guter Mix zwischen Lehrperson und gegenseitigem Support und ja eigentlich  
59 Experten-Support gemischt mit diesen Materialien, den Lernmaterialien also  
60 Tutorials. Für Zuhause ist nochmal eine Spur wichtiger, dass es eben diese auf



61 die Bedürfnisse der Lernenden zugeschnittenen Tutorials gibt, wo wir halt auch  
62 im Projekt Rückmeldungen eingesammelt haben. Weil an und für sich könnte  
63 man sagen, Unity ist weit verbreitet. Liebe Lernende, schauts einfach im  
64 Internet was ihr findets und arbeitet damit. Das hat aber kaum zu irgendwelchen  
65 Aktivitäten der Schülerinnen geführt, sondern erst tatsächlich, dass man das im  
66 Unterricht thematisiert hat und Mal den Lernenden das grundlegend gezeigt hat  
67 und dann halt auch diese speziellen Lernmaterialien bereitgestellt hat. Und  
68 dann haben aber schon wiederum die Lernenden auch zuhause weiter  
69 gemacht, also nach dieser Starthilfe. Man könnte fast sagen, das ist wie ein  
70 Katalysator, also im chemischen Sinne etwas das eine Reaktion beschleunigt  
71 oder überhaupt erst ermöglicht.

72

73 [0:05:21] **I:** Schlussendlich ist ja das Ziel aus den Schülerinnen und Schülern  
74 gute Programmierer zu machen. Und was zeichnet Ihrer Meinung nach einen  
75 guten Programmierer aus?

76

77 [0:05:36] **B:** Also gute Programmiererinnen und Programmierer zeichnet sicher  
78 aus, die Fähigkeit logisch und strukturiert zu denken. Problemstellungen zu  
79 erfassen, zu formulieren und zu exportieren. Und dann auch wiederum  
80 strukturiert vorzugehen. Was ganz wesentlich ist, glaube ich auch einerseits  
81 eben abstrakte Inhalte zu erfassen, zur erkennen und zu formulieren und  
82 artikulieren, gleichzeitig aber auch für die Implementierung ins Detail zu gehen,  
83 also ins Detail springen zu können. So ist dann wirklich so ein Abwechseln  
84 zwischen Detail und abstrakten Arbeiten. Dann Kreativität ist wichtig, auch  
85 wenn es für Nicht-ProgrammiererInnen so aussieht als wäre das sozusagen nur  
86 ein aus Regeln vorgegebenes Code-Getippsel. Das immer wieder Kreativität  
87 bei der Lösung von Problemen im Unterricht auftaucht, auch bei der Lösung  
88 von selbstgesteckten Problemen, die die Schülerinnen und Schüler haben ein  
89 ganz positiver wesentlicher Faktor ist, um zum Ziel zu gelangen. Und dann  
90 diese ganzen Sachen wie formalisieren und Probleme modellieren, das macht  
91 man vielleicht nicht auf universitären Niveau in der Schule aber all diese Muster

92 tauchen auch schon bei programmierenden Schülerinnen und Schülern auf.  
93 Was jetzt sozusagen abseits dieser technischen Skills interessant ist: es ist  
94 sicherlich auch wichtig eine gute Recherche, eine wichtige Kommunikation, eine  
95 gute Kommunikationsfähigkeit und eine gute Teamfähigkeit, weil auch wenn  
96 man irgendwo hängt und sich austauschen will bei Probleme, dann bringt halt  
97 einfach eine gute Kommunikationsfähigkeit die Probleme zu artikulieren oder  
98 gute Teamfähigkeit, das zusammen zu lösen sehr viel. Also, in Wirklichkeit  
99 zeichnet gute Programmierer auch ein gewisses Set an Softskills aus, die diese  
100 mitbringen und nicht nur das reine Coden.

101

102 [0:07:37] **I:** Das war jetzt sehr stark auch in Richtung Soft Skills gehend. Wie  
103 sieht das ganz speziell jetzt im Bezug zu digitalen Kompetenzen aus, was einen  
104 guten Programmierer ausmacht?

105

106 [0:07:56] **B:** Das ist irgendwie fast schon eine notwendige Voraussetzung das  
107 Programmierende in diesen digitalen Grundkompetenzen fit sind. Und wir  
108 haben auch im Projekt gesehen, immer dann, wenn es an digitalen  
109 Grundkompetenzen gemangelt hat, zum Beispiel beim Filemanagement, dass  
110 man sich vernünftige Ordnerstrukturen anlegt und das man auch in denen sich  
111 zurechtfindet, immer dann ist es besonders schwierig geworden oder die  
112 Lernenden sind in Sackgassen geraten. Es war einmal beim Filemanagement  
113 konkret, es war auch aber bei einer Recherche-Kompetenz, dass man einfach  
114 Sachen nachschauen kann, die dann wirklich binnen weniger Sekunden  
115 beantwortet sind. Das würde ich einmal verallgemeinern, auch allgemeine  
116 digitale Kompetenzen. Also es ist fast eine notwendige Voraussetzung, dass  
117 man da ein gewisses Level schon hat, damit man eben mit so einer komplexen  
118 Spielentwicklungsumgebung wie Unity die Spiele programmieren kann.

119

120 [0:08:47] **I:** Mhm. Fallen Ihnen da noch weitere Beispiele ein vielleicht? Wie  
121 Dateimanagement haben Sie jetzt gesagt, oder das Finden von Informationen.

122

123 [0:09:10] **B:** Das Finden von Informationen war ein Beispiel eben im Unterricht.  
124 Ich überlege jetzt nochmal grad, welche Sachen da noch aufgetreten sind, die  
125 ganz, ganz wesentlich waren. Mehr Anwendungssoftware-Kompetenzen, wie  
126 die Lernenden mit der Bildbearbeitung zu tun gehabt haben. Das zum Beispiel  
127 ein Thema, also wirklich mit der Anwendungssoftware umgehen zu können. Es  
128 waren aber eher mehr so konkrete Kompetenzen. Was fehlt mir noch ein? Ich  
129 muss das wieder aus der geistigen Lade hervorkramen.

130

131 [0:09:42] **I:** Ja, klar.

132

133 [0:09:44] **B:** Naja was dann, es waren auch teilweise aber so Dinge dabei wie  
134 z.B. welche Bilder können Sie nehmen. Das waren eher so Kompetenzen in  
135 rechtlichen Belangen. Was kann man machen, was kann man nicht machen,  
136 kann man sich gegenseitig fotografieren und die gegenseitigen Gesichter ins  
137 Spiel einbauen oder unter welchen Lizenzen stehen die Grafiken die ich mir aus  
138 dem Internet "checke", wie die Schülerinnen und Schüler das gesagt haben. Es  
139 hat aber auch natürlich teilweise mit Kompetenzen zu tun gehabt, die sich auf  
140 das Schulnetzwerk bezogen haben, also warum man eben online oder offline  
141 arbeiten kann in Unity und wie das mit den Netzwerk-Ordnern ist im Bereich der  
142 Zusammenarbeit, wenn man jetzt zu mehr ein Projekt macht, also dass man  
143 eben das Wissen und die Kompetenzen im Bereich des Schulnetzwerks  
144 reflektiert umgeht. Was fällt mir da noch ein? Hm. Eben ja, diese  
145 Bildbearbeitung haben wir schon gehabt. Informationssuche und Recherche  
146 haben wir auch schon gehabt. Von Kompetenzen, naja es wäre eigentlich eine  
147 altmodische Grundtugend, diese Genauigkeit, also die Fähigkeit genauer zu  
148 arbeiten und Sachanleitungen richtig einzugeben. Das wäre vielleicht noch was,  
149 was aufgefallen ist im Projekt. Naja, ein bisschen vielleicht doch hinter die  
150 Kulissen blicken, so in die Tiefe, wie an und für sich technische Artefakte wie  
151 der Computer funktioniert. Da kommt man halt zuerst einmal das man versucht  
152 zu analysieren, wie funktioniert überhaupt das Spiel, wie funktioniert die  
153 Entwicklungsumgebung dieses Spiels und warum funktionieren manche

154 Sachen so und so nicht und was ist da abgebildet. Und auch gewisse Hardware  
155 Sachen. Wenn man ganz aufwendige Szenen hat und die sind halt dann in der  
156 Vorschau, wenn man auf Unity „Play“ gedrückt hat, langsam gelaufen. Und  
157 warum das langsam läuft und was in den Rechnern für Grafikkarten drinnen  
158 sind, zur Vorstellung welche Prozessoren drinnen sind, also eigentlich hat dann  
159 ein Schüler der sich gut mit Hardware ausgekannt hat, sagt der der diese  
160 Hardware, diese technische Kompetenz gehabt hat: Ja, das kann in der Schule  
161 nicht gehen, mit diesen Onboard-Grafikchips. Und solche Dinge sind da auch  
162 im Bereich eigentlich eingeflossen. Es war wirklich ein ganz buntes  
163 Sammelsurium an Kompetenzen, die dann einerseits zutage getreten sind und  
164 andererseits aber auch notwendig waren oder von Interesse waren, einfach  
165 nach dem gefragt worden ist.

166

167 [0:12:32] **I:** Und war dann innerhalb des Unterrichtsprojektes sozusagen die Zeit  
168 dazu, dass man auf die Punkte näher eingegangen ist, oder wurde das eher  
169 kurz quasi angesprochen.

170

171 [0:12:47] **B:** Also, das ist ehrlich gesagt kurz angesprochen worden, wirklich in  
172 so Nebenplots, also eben, dass mal der Rechner zu langsam war, oder das  
173 Dateimanagement nicht gepasst hat, dann ist es eben schnell mit der  
174 betreffenden Schülerin besprochen worden. Und nur vereinzelt in Bezug auf  
175 das Dateimanagement, eben wie so ein Projektordner ausschaut, das ist dann  
176 halt in der nächsten Stunde in Form einer Wiederholung aufgegriffen worden.  
177 Was wichtig ist. Größtenteils in Form von Anmerkungen, also Annotations an  
178 die Schülerinnen und Schüler, abgehandelt worden. Aber bei wichtigen Sachen  
179 die einfach so ein Knock-Out-Kriterium waren, so einen Stop in die Entwicklung  
180 eingebracht hätten, da ist es dann noch einmal für alle erklärt worden. Da sind  
181 auch die Lernenden in die Mitte geholt worden und es wurde besprochen. Aber  
182 sonst war es tatsächlich eine Zeitfrage, dass man nicht diese unzähligen  
183 Fragen dann alle nochmal aufrollt. Das ist wirklich eine Zeitgeschichte.

184

185 [0:13:48] **I:** Und wo, also wenn wir jetzt wieder den Bogen zurückspannen zu  
186 "wodurch sich ein guter Programmierer auszeichnet", wo kann da die  
187 Spieleentwicklung mit Unity ihren Beitrag leisten?

188

189 [0:14:08] **B:** Naja, die Spieleentwicklung mit Unity, die stellt tatsächlich die  
190 Lernenden vor eine ganz komplexe, herausfordernde Entwicklungsumgebung.  
191 Und was da allgemein beobachtet worden ist, eben die Scheu sich damit  
192 auseinanderzusetzen hat irrsinnig stark abgenommen. Wo man früher vielleicht  
193 Schülerinnen und Schüler gehabt haben, die gesagt haben, nein machen gar  
194 nichts was über diese Komplexitätsebene N hinausgeht, die haben nachher  
195 dann ganz beeindruckend auch mit Besucherinnen von der Universität mit  
196 Quellcode hantiert und mit dieser ID gearbeitet. Das war mal die eine Ebene  
197 und ansonsten bei der Spieleentwicklung selbst, da sind irrsinnig viele Prozesse  
198 und Themen, die angesprochen werden, die im Bereich digitaler Kompetenzen  
199 liegen, die tatsächlich dann auch abgedeckt werden. Also wenn ich jetzt zum  
200 Beispiel Objektorientierung hab einfach. Die objektorientierte Programmierung  
201 ist in Unity sehr stark verankert. Da hat man eben diese Objekte und teilweise  
202 ist es sogar so extrem stark verankert, dass es durch eine intensive Nutzung  
203 vorgegebener Objekte gekennzeichnet ist, dass die Schülerinnen und Schüler  
204 irgendwann einmal in diese Welt eintauchen und erfahren, oh, da gibt es ja  
205 wirklich so eine Welt, so eine objektorientierte Welt und da macht man sich die  
206 einzelnen Objekte, also macht man sich eigentlich die Entitäten dieser Objekte  
207 zu Nutze, um dann Dinge zu bewerkstelligen, die schon jemand anderer für  
208 einen vorbereitet oder vorprogrammiert oder geschafft hat, wenn man  
209 Funktionen einsetzt von einem bestimmten Objekt oder wenn man eben  
210 Instanzen setzt, oder Referenzen. All das lässt dann ein bisschen die  
211 Lernenden, wenn das Ganze erfolgreich ist, sowas wie die Zahlen im Film The  
212 Matrix sehen, wenn dann die Hauptperson, der Neo Anderson, dann  
213 irgendwann einmal am Ende des ersten Teils dann die Matrix sieht und die  
214 Zahlen umdreht. Also so ist mir das ein wenig vorgekommen.

215

216 [0:16:18] **I:** Wenn Sie jetzt an einen Musterschüler von ihnen denken. Was  
217 meinen Sie, nahm derjenige oder diejenige mit für den Job als Programmierer?

218

219 [0:16:30] **B:** Also bei den Musterschülern würde ich sagen, die haben  
220 mitgenommen, dass es wichtig ist, den Blick für das Wesentliche zu bewahren  
221 und auch das Gefühl für das Wesentliche zu bewahren und sie nicht irgendwo  
222 zu versteigen. Gerade bei den besonders guten Schülerinnen und Schülern ist  
223 es so, dass sie dann dieses und jenes probieren und das machen wollen und  
224 das und das und das. Wenn es aber darum geht, dass man im Projekt etwas  
225 weiterbringt, da muss man schon den ein oder anderen Nebenquest mal sein  
226 lassen und verlassen, um dann den Hauptpreis zu verfolgen. Dieses Gefühl fürs  
227 Wesentliche glaube ich schon, dass die die besonders involviert waren, die  
228 sogenannten Musterschülerinnen mitgenommen haben und dass eben auch  
229 gewisse Situationen auftreten können wo eine Hilfe oder eine Recherche enorm  
230 viel Zeit sparen kann und einem eine andere Perspektive einem aus gewissen  
231 Sackgassen herausführen kann. Das wäre dann wieder so eine Art  
232 Selbstkompetenz/Recherchekompetenz, aber eben auch dieser Blick fürs  
233 Wesentliche. Und was vielleicht für den Schulunterricht noch gut zu fördern ist,  
234 man braucht als Programmiererin so eine gewisse Resilienz, um doch nicht  
235 gleich nach dem ersten Fehlversuch aufgeben, sonst wird das nichts. Ich kenne  
236 keinen einzigen Programmierer, keine einzige Programmiererin, die Code  
237 schreibt und dieser Code funktioniert dann beim allerersten Mal, wenn es eine  
238 neuartige Aufgabe ist. Natürlich wenn man Lehrpersonen an der Universität ein  
239 Szenario vorprogrammiert, schön auf dem Screen und es wird mit dem  
240 Videoprojektor projiziert, man hat das schon hundertmal gemacht, dann kommt  
241 es schon vor. Aber wenn man als Programmiererin neuen Code für eine neue  
242 Aufgabe schreibt, die man neu erlebt hat, da wage ich zu behaupten, dass es  
243 solche Leute entweder gar nicht gibt oder in so geringer Anzahl gibt, dass es  
244 eben nicht auftritt, dass man einfach Code schreibt, der funktioniert. Es ist in  
245 der Praxis eher so, man muss hundertmal probieren, hundertmal kompilieren  
246 oder öfter bis man mal das ganze Ding am Laufen hat. Da ist eben diese

247 Resilienz, das nicht aufgeben, wichtig, weil da kann man, wenn man sich mit  
248 den Schülerinnen und Schülern zusammen hinsetzt und das durchdenkt und  
249 das vorlebt, sozusagen das Probieren und das Durchdenken und das auch  
250 strukturierte Vorgehen, dann kann man diese Resilienz ein bisschen sozusagen  
251 näherbringen oder übertragen, würde ich fast sagen.

252

253 [0:19:02] **I:** Das ist jetzt auch ein bisschen in Richtung Softskills würde ich fast  
254 sagen mit der Resilienz. Versuchen wir bei den digitalen Kompetenzen mehr zu  
255 bleiben. Wo sehen Sie da den größten Nutzen von der Spielentwicklung mit  
256 Unity?

257

258 [0:19:21] **B:** Auf jeden Fall, dass man irgendwie objektorientiert denkt, dann das  
259 das algorithmische Denken geschult wird. Dass das Strukturieren gefördert  
260 wird. Dass man die Grundbegriffe eigentlich vom Programmieren an und für  
261 sich verinnerlicht. Dass man gewisse Variablen einsetzen kann,  
262 Datenstrukturen beschreiben kann. Diese Geschichten. Dass man sich mit if-  
263 Anweisungen auseinandersetzt und auskennt, also dass auf jeden Fall, dass  
264 man die verschiedenen Schleifentypen kennt, wann man die Schleifen einsetzt  
265 und darüber hinaus noch immer Alternativen zu Schleifen kennt, vielleicht  
266 rekursive Programmierung oder Alternativen zu conditional statements zum if-  
267 statement, die case-Verzweigung fällt mir da ein, case-switch. Und das wäre  
268 eher sozusagen bei den Hard Skill, würde ich sagen.

269

270 [0:20:17] **I:** Hat die Spielentwicklung mit Unity Einfluss auf das Verständnis der  
271 Schülerinnen eines Algorithmus?

272

273 [0:20:30] **B:** Das würde ich sagen, auf jeden Fall. Wenn man auf verschiedene  
274 Arten und Weisen mit Algorithmen konfrontiert ist. Also man nimmt einmal  
275 irgendein Spieleobjekt und dann schreibt man zum Beispiel einen Algorithmus  
276 für die Bewegung. Oder es ist auch teilweise so ein bisschen verdeckt. Man hat  
277 diese State-Machine die eigentlich zuerst der Modellierung dient, in erster Linie,

278 also die bildet sehr gut die Modellierung ab. Wie diese unterschiedlichen states  
279 stattfinden, aber dann hat man auf einmal diese Übergänge, von einem Status  
280 zum anderen. Es sind halt diese Bedingungen drinnen. Aber dann ist die Frage,  
281 wie wird dieser Übergang eigentlich mit Bedingungen realisiert und da bilden  
282 sich so Repräsentanzen eigentlich von Algorithmen, die die Lernenden dann im  
283 Kopf haben. Das kann durchaus eine Abstraktion von einem tatsächlichen  
284 Algorithmus sein der schon in Unity, in der Spiele-Entwicklungsumgebung  
285 implementiert ist, mit denen die Lernenden dann arbeiten. Dann sagen sie,  
286 dieses und jenes mache ich und dann passiert dieses und jenes mit dem  
287 Objekt, bei diesem Übergang, bei diesen Werten und dann haben sie  
288 sozusagen eine eigene algorithmische Repräsentanz von diesem Vorgang. Das  
289 ist auch ganz interessant, es ist nicht nur ein Algorithmus in diesem wirklich  
290 strengen Sinne, dass sie einen Algorithmus selbst coden oder als Quellcode  
291 anschauen, sondern auch in der erwähnten Weise.

292

293 [0:22:04] **I:** Wird das dann auch explizit erwähnt, dass da mit Algorithmen  
294 gearbeitet wird?

295

296 [0:22:12] **B:** Ja. z.B. es gibt ein Spiel, das ist nicht so direkt drinnen, aber das ist  
297 dieses "Let's make a Snake", dieses Lemmers, das ich ganz am Anfang des  
298 Projekts geschrieben habe. Da geht es um unseren besonders schlaunen  
299 Algorithmus und da taucht immer wieder das Wort Algorithmus auf. Was kann  
300 man da machen, um das zu verbessern. Es kommt halt auch immer wieder  
301 beim Bouncy Fant zwischendurch durch, weil wir Informatikerinnen sind und  
302 Effizienz lieben. Oder es gibt doch so mini Algorithmen-Geschichten wie, dass  
303 man den Boden zeichnet in einer Schleife. Könnte man auch schon sagen, ja  
304 das ist schon die einfachste Stufe eines Algorithmus, wenn man die Lernenden  
305 dann mit der Aufgabe betraut, den Boden nicht einfach hinzuklatschen, sondern  
306 in Form eines Algorithmus zeichnen zu lassen zur Laufzeit.

307



308 [0:23:19] **I:** Können die Schülerinnen nach der Spielentwicklung mit Unity  
309 wesentliche Aspekte der Programmierung nennen und erläutern wie Funktionen  
310 oder Objektorientierung?

311

312 [0:23:41] **B:** Also ich denke, da kann Ihnen (anonymisierte Person) in den Code  
313 Reviews mehr sagen, (anonymisierte Person) hat nämlich mit den älteren  
314 Schülerinnen und Schülern im Wahlpflichtfach diese Code-Reviews gemacht,  
315 diese Interview-based Code-Reviews. Im Wesentlichen wenn man die  
316 Lernenden danach fragt, dann kommen sehr häufig die Erklärungen von diesen  
317 ganzen Aspekten. Manchmal freut man sich als Lehrperson eben, dass man  
318 wirklich die Fachbegriffe hört.

319

320 [0:24:09] **I:** Lernen die Schülerinnen im Verlauf der Spieleentwicklung mit Unity  
321 unterschiedliche Programmiersprachen und Produktionsabläufe kennen?

322

323 [0:24:27] **B:** Mit Unity und seinen vorbereitenden Tätigkeiten auf jeden Fall. Da  
324 wird ja auch Scratch gemacht. Bei den anderen Tätigkeiten (...). In Unity wird  
325 mit C# programmiert. Das ist sozusagen die Programmiersprache, die wir an  
326 der Schule verwenden. Wenn man dann den Webexport macht, könnte man  
327 sich über die Web-Geschichten unterhalten. html-coden, wenn man html als  
328 eine Programmiersprache bezeichnen mag. Ist halt eine Ansichtssache. Aber  
329 zumindest hat man da eine Auszeichnungssprache, die interpretiert wird.  
330 Ansonsten, diese state-machine hat eine eigene Programmier-Logik. Aber es ist  
331 auch keine eigene Programmiersprache. Im Zuge der gesamten  
332 Projektaktivitäten ist es schon so dass im Unterricht die Lernenden über  
333 Scratch und die console-Programmierung in C# und dann letztendlich mit Unity  
334 diesen Weg nehmen. Aber so direkt im Unity ist das jetzt nicht in unserem  
335 Curriculum drinnen.

336

337 [0:25:44] **I:** Erwerben die Schülerinnen die Fähigkeit, dass sie Aufgaben mit  
338 Mitteln der Informatik modellieren können?

339

340 [0:25:54] **B:** Ich denke ja und das wird in der bereits erwähnten state-machine  
341 zum Beispiel diese Modellierung von Zustandsübergängen so oft gemacht.

342

343 [0:26:06] **I:** Sehen Sie das auch als eine Voraussetzung für die  
344 Spieleentwicklung mit Unity oder erwerben sie diese erst dadurch?

345

346 [0:26:20] **B:** Naja, es ist insofern, in unseren Aktivitäten haben wir schon diese  
347 ganzen Geschichten wie Datenstrukturen im Konsole-Tutorial, also im Bereich  
348 des Konsolen-Tutorial gemacht. Beziehungsweise kann man das auch mit dem  
349 Textadventure den Lernenden näherbringen. Das heißt, die Antwort ist eine  
350 ganz klare teils, teils. Es wird zum Teil vorbereitet und es wird zum Teil aber  
351 auch dann wirklich erst im Zuge der Entwicklung mit Unity erlernt und vertieft.

352

353 [0:26:50] **I:** Aber Sie würden also sagen, es sollte schon ein bisschen ein  
354 Vorwissen, ein bisschen etwas da sein an Fertigkeit.

355

356 [0:27:01] **B:** Ja, deswegen machen wir auch diese Voraktivitäten. Obwohl es  
357 auch die Idee gibt, dass man nur sozusagen mit Unity gleich einsteigt. Das ist  
358 auch ein Ansatz, den einige Kollegen auch verfolgen.

359

360 [0:27:20] **I:** Können die Schülerinnen dann nach der Spielentwicklung mit Unity  
361 Daten-, Klassen- und Objekt-Strukturen in einem Algorithmus verwenden?

362

363 [0:27:38] **B:** Das auf jeden Fall. Das ist ja fast eine tägliche, oder  
364 wiederkehrende Arbeit, wenn man irgendetwas ausimplementiert in Unity  
365 selbst. Da muss ich schon einfach neue Objekte instanziiieren und mit  
366 Datentypen arbeiten und Objekt-Strukturen mir zu eigen machen. Einfach damit  
367 ich ans Ziel gerate. Weil es eine riesengroße Schnittstelle ist, die Unity  
368 bereitstellt, um Dinge tatsächlich aus zu implementieren im eigentlichen Spiel.

369

370 [0:28:23] **I:** Haben Sie da vielleicht ein konkretes Beispiel.

371

372 [0:28:27] **B:** Ja, zum Beispiel wenn sie irgendwie den Elefanten als Gameobject  
373 erstellt haben, dann wird der Elefant sozusagen in Unity, wenn ich jetzt zum  
374 Coden beginne instanziiert und dann wird immer auf diesen Elefanten in  
375 meinem Quellcode verwiesen. Und auch im Quellcode werden dann die  
376 Eigenschaften und das Verhalten des Elefanten manipuliert. Und wenn ich zum  
377 Beispiel die Bewegung manipulierte, dann hole ich das wieder von einer  
378 anderen Seite, dann hole ich das sozusagen vom Inputkontroller und frag das  
379 von anderen Objekten ab. Wie z.B. die Horizontalbewegung ist, oder solche  
380 Dinge. Also, ich hole es von dem einen Objekt und gebe es zum anderen  
381 Objekt weiter.

382

383 [0:29:12] **I:** Das ist ja auch ein Projekt quasi ein Unterrichtsprojekt. Und wenn  
384 man da an Software Projekte denkt, werden die Schülerinnen dann auch fähig  
385 danach ein Softwareprojekt selbstständig zu planen und durchzuführen? Und  
386 das vielleicht sogar noch zu reflektieren?

387

388 [0:29:50] **B:** Also, schon allein durch das einmalige Durchführen der  
389 verschiedenen frei gestaltbaren Projektteile, also zuerst macht man mal das  
390 Tutorial. Und da lernt man schon, das Tutorial ist auch in Wirklichkeit, ... hm, für  
391 die Lernenden wird das aufgefasst wie ein Projekt. Aber danach sollen sie halt  
392 irgendwelche eigenen Spiele implementieren und da lernen sie in der Planung  
393 schon ein Softwareprojekt ja, also da lernen sie im Registrieren des Projektes  
394 zu planen, durchzuführen, wo da die Schwierigkeiten sind und was man eben in  
395 einem Projekt tut und was man besser vermeiden sollte. Und dann haben wir  
396 noch drangehängt, diese Interviewbased Code-Reviews und da wird schon  
397 auch zu einem großen Teil das realisierte Projekt reflektiert. Da wird halt nicht  
398 mit der Lehrperson gemeinsam mit den Schülern das Projekt durchgegangen.

399

400 **I:** [0:30:43] Lernen sie dabei auch die Entwicklungswerkzeuge einzuschätzen?  
401 Innerhalb von Unity?

402

403 **B:** [0:30:55] Innerhalb von Unity und auch gewiss außerhalb von Unity. Wenn  
404 ich zum Beispiel irgendwie ganz einfache Dinge bewerkstelligen möchte, dann  
405 brauch ich nicht mit Kanonen auf Spatzen schießen und Unity verwenden, aber  
406 wenn ich irgendwelch ein Spiel mit Physikengine, oder Physikanbindung haben  
407 will, das gut exportierbar ist und wo die Grafiken halbwegs gut ausschauen und  
408 nicht so zu ruckeln beginnen wie bei Scratch, dann brauch ich eben schon  
409 Unity. Und ich denke durch diese vorbereitenden Tätigkeiten lernen dann die  
410 Schülerinnen und Schüler ganz gut einzuschätzen, welches  
411 Entwicklungswerkzeug angemessen ist. Wenn ich jetzt irgendwie eine  
412 Oberfläche haben möchte, die im Web erscheint, dann werde ich wohl kaum in  
413 Unity ein Projekt machen und das Ganze dann ins Web exportieren, sondern da  
414 reicht ein schön gestiltes html5 und das sollte dann den Zweck erfüllen, wenn  
415 es einfach nur um die Oberfläche geht, um eine einfache Interaktion  
416 abzubilden. Oder Angemessenheit von Entwicklungswerkzeugen wäre auch  
417 dann gegeben, wenn man sich überlegt, jetzt möchte ich an der Grafik eine  
418 Änderung haben, mache ich das jetzt in Unity oder mache ich das in der  
419 Bildbearbeitung. Zum Beispiel möchte ich von meinen Spielfiguren Dinge  
420 anders darstellen, beim Elefanten der Rüssel, dann schätzen die Schülerinnen  
421 und Schüler in den meisten Fällen ein, naja, das in Unity zu transformieren ist  
422 vielleicht nicht so angemessen, wie einfach nochmal in die Bildbearbeitung zu  
423 gehen ins Gimp und dann das zu verändern und dann wieder in Unity  
424 weiterzuarbeiten. Wohingegen, wenn es darum geht, ändere ich jetzt die Farbe  
425 von meinem Elefanten, dann machen wir das gleich meistens in Unity. Da  
426 fragen sie, Herr Professor, wie kann man die Farbe ändern und dann zeige ich  
427 ihnen das. Und das ist schon ein starkes Indiz für mich, dass sie die  
428 Angemessenheit von Entwicklungswerkzeugen richtig einschätzen.

429

430 **I:** [0:33:22] Entdecken die SchülerInnen auch Gemeinsamkeiten und Regeln in  
431 den Handlungsanleitungen bzw. Algorithmen?

432

433 **B:** [0:33:33] Wenn zum Beispiel die Objekte herunterfallen, diese Game-  
434 Objects, da geht es ja darum, da sind Elefanten und irgendwann lässt der  
435 verrückte Dr. Clamberwoods den Müll regnen und da kommen dann solche  
436 Aussagen: "Ah, das ist ja eh in Wirklichkeit wieder wie der Elefant, nur halt kann  
437 ich das nicht steuern". Das wäre ein Beispiel. Also sie entdeckten  
438 Gemeinsamkeiten und Regeln und erkennen durchaus gewisse Muster  
439 eigentlich. Und in der State-Machine habe ich vorher schon erwähnt, dass sich  
440 solche algorithmischen Repräsentanzen herausbilden, die eben nicht  
441 notwendigerweise genau das was implementiert worden ist abbilden, aber die  
442 sie halt dazu einsetzen, um fast so ein bisschen nach dem Schnittstellen-  
443 Prinzip oder Blackbox-Prinzip, um abzuschätzen was jetzt passiert, wenn sie  
444 das manipulieren oder wenn sie dieses oder jenes wollen.

445

446 **I:** [0:34:44] Können die SchülerInnen durch die Spielentwicklung mit Unity die  
447 Bedeutung von den Algorithmen erkennen?

448

449 **B:** [0:34:55] Die Bedeutung auf jeden Fall. Und es gibt ja Bedeutungen darüber,  
450 sozusagen im lockeren Sinn, einfach was mein Algorithmus macht für mein  
451 Softwareprojekt oder überhaupt für die Informatik. Und dann ist aber auch  
452 wesentlich, dass man sich anschaut, ob sie wirklich den Algorithmus  
453 entschlüsseln können, also dekodieren. Also wenn sie ein Stück Code sehen  
454 und genau sagen können, - das ist die Bedeutung für mich im engeren Sinne -  
455 was bedeutet dieser Code, oder was macht dieser Code. Zum Beispiel haben  
456 wir versucht das auch abzufragen in so einem IT-Skills-Test, wo einfach  
457 Lernende eine Funktion analysieren sollen und dann nur die Ausgabe dieser  
458 Funktion als Antwort geben sollen. Wenn man das eben nicht verstanden hat  
459 und die Bedeutung nicht entschlüsseln kann, dann ist das ein Schuss ins Blaue

460 und wenn man das aber verstanden hat und die Bedeutung kennt, dann ist das  
461 eigentlich recht einfach da die Lösungen anzugeben.

462

463 **I:** [0:35:55] Und wenn sie so Codeteile oder Algorithmen dekodieren, können  
464 sie die Effizienz des Algorithmus einschätzen/bewerten?

465

466 **B:** [0:36:17] Ja, ja durchaus. Da gibt es diese eine Übung wo man wirklich den  
467 Boden mittels einer Schleife, mittels eines einfachen Algorithmus erzeugen und  
468 zupflastern soll und da habe ich die Anmerkung erhalten: "Das ist ja gar nicht  
469 effizient, Herr Professor, wie wäre es wenn man da einfach sozusagen das  
470 Ganze einmal unten hinzieht und dupliziert, das man diese Tiles wiederholt und  
471 dann einen gemeinsamen Collider macht." Bei dem Algorithmus wird  
472 sozusagen anhand eines Prefabs, anhand einer Vorlage wirklich der Boden wie  
473 beim Fließen legen gelegt und diese einzelnen Boden-Tiles, die haben auch die  
474 Colliders und da hat ein Schüler dann zurecht gesagt, das ist überhaupt nicht  
475 effizient, das kann man ja wiederholen lassen und einen gemeinsamen Collider  
476 machen.

477

478 **I:** [0:37:09] Wir kommen jetzt zu einem zweiten Teil sozusagen des Interviews.  
479 Der ist etwas kürzer. Bei SchülerInnen treten ja immer wieder Mal  
480 Schwierigkeiten auf. Vor allem bei neuem Lehrstoff. Wenn Sie sich an solche  
481 Situationen erinnern, was waren da die Schwierigkeiten?

482

483 **B:** [0:37:36] Also es gab einerseits eben wirklich die Toolbezogenen  
484 Schwierigkeiten, bezogen auf Unity und andererseits die  
485 Vorbereitungsschwierigkeiten. Zum Beispiel so eine Vorbereitungsschwierigkeit  
486 ist, wenn man die ersten Schritte mit Unity herzeigt und selber Unity startet und  
487 dann loslegt und dann sagt: "Liebe SchülerInnen und Schüler es kommt jetzt  
488 ein Tutorial", aber vergisst zu erwähnen: "Ja ihr müsst euch vorher am besten  
489 noch registrieren.", dann sitzen die beim Registrieren irrsinnig lang dran, weil  
490 das Schulnetzwerk gleich einmal schlecht ist und sie jetzt eben nicht genau

491 wissen, aha, sollen wir uns registrieren oder nicht und dann die Fragen sind  
492 das. Das sind eben die strukturellen Schwierigkeiten, die man durch gute  
493 Vorbereitung, gute Erklärung und Unterrichtsplanung rausnehmen kann. Dann  
494 gibt es eben die technischen Schwierigkeiten, die da auch schon ein bisschen  
495 drinnen waren, schlechte Netzwerkverbindung, dann kommen viele Anfragen  
496 von einer IP-Adresse, dann wird wieder ein anzulegender Account sozusagen  
497 geblockt von irgendeiner Heuristik, weil alles von der gleichen IP-Adresse  
498 kommt. Das sind die technischen Schwierigkeiten. Und dann gibts tatsächlich  
499 die Schwierigkeiten mit dem Stoff, dass der Stoff am Anfang teilweise sehr viel  
500 erscheint für die Lernenden und recht komplex erscheint und das einfach das  
501 mit der kognitiven Erfassung nicht so leicht geht. Ich versuche dann immer  
502 entgegenzusteuern indem ich sage, wir setzen uns vorher einmal entspannt  
503 zusammen hin und machen so ein „Situating Learning“, so nach Lave und  
504 Wenger, wie bei den „Communities of Practice“ und wir machen einmal ein  
505 Outline von dem Problem und besprechen wo wir hinwollen und dann zeige ich  
506 das kurz vor. Das müsst ihr euch nicht alles auswendig merken, sage ich dann  
507 den Lernenden, aber es soll euch irgendwie ein Gefühl vermitteln und euch  
508 vorbereiten für das, was ihr nachher mit Hilfe der Unterlagen lernen könnt. Aus  
509 der Lernpsychologie weiß man, dass sozusagen vorher in so einem  
510 LehrerInnen-SchülerInnen-Setting, das förderlich für das Lernen ist, Und dass  
511 man eben zuerst das durchgeht und dann erst die Lernenden mit den Lernhilfen  
512 und Tutorials arbeiten lässt.

513

514 **I:** [0:39:51] Was müsste den SchülerInnen fehlen damit es noch schlimmer  
515 werden würde?

516

517 **B:** [0:39:59] Noch schlimmer? An der Schule, mit der Infrastruktur? Naja,  
518 vielleicht das das Internet gar nicht mehr geht, dass die Computer noch  
519 langsamer sind und abstürzen, dass die technischen Schwierigkeiten größer  
520 werden, also technische Ressourcen. Ein ruhiger Raum zum Lernen müsste  
521 den SchülerInnen fehlen, dass sie ständig, immer wieder gestört werden im

522 Lernprozess. Was müsste ihnen noch fehlen? Ja, die LehrerInnen Erklärung,  
523 also dass die Lehrperson die Inhalte erklärt und der gegenseitige Support und  
524 der Lehrersupport. Also wenn der ihnen fehlen würde, dann würde die Situation  
525 sehr, sehr unbefriedigend werden. Also ich glaube nicht, dass da noch was  
526 weitergehen würde. Und die Lernmaterialien am besten noch unvollständig  
527 formuliert, oder rudimentärer, gar nicht vorhanden und dann sagen, ja schaut  
528 das nach, googelt das, macht das mit den Tutorials, die ihr im Internet findet.  
529 Weil da findet man so schwer den Einstieg und die Anknüpfungspunkte. Ich  
530 glaube, dann könnte das Ganze ordentlich schief gehen.

531

532 **I:** [0:41:10] Was müsste man zur Bewältigung der Schwierigkeiten tun?

533

534 **B:** [0:41:14] Zur Bewältigung der Schwierigkeiten braucht man eine gute,  
535 motivierte Lehrperson, die schon auch ein ExpertInnenwissen hat. Also es  
536 reicht nicht einfach der Cheering-Coach zu sein und sagen: "Ja, liebe Schüler  
537 ich find euch super, ihr machts das, ihr schaffts das, ich glaube an euch",  
538 sondern da muss man tatsächlich dann für Fragen gewappnet sein und  
539 vorbereitet sein. Also eine gute, vorbereitete Lehrperson, dann das die  
540 technische Infrastruktur auf einem neueren Stand als dem vor zehn Jahren ist,  
541 dass die Internetanbindung mehr als sozusagen eine unterdurchschnittliche  
542 Heiminternetanbindung hergibt. Das heißt die technische Infrastruktur muss  
543 passen und es müsste alles an Tools und Software installiert sein, die man  
544 braucht. Und ein gutes Klassenklima, indem auch die Kultur des gegenseitigen  
545 Helfens funktioniert. Also ich versuche immer im Laufe des Schuljahres im  
546 Klassenraum so eine Kultur des gegenseitigen Helfens zu etablieren, aber  
547 gleichzeitig auch mit so einer Supportpyramide. Das heißt, dass das nicht  
548 Überhand nimmt. Supportpyramide ist für mich, dass zuerst einmal die  
549 SchülerInnen und Schüler ihr Problem nochmal im Geiste formulieren und sich  
550 die Frage stellen, habe ich sozusagen dem Algorithmus, dem  
551 Entwicklungswerkzeug alles richtig eingegeben. Ich sage dann immer ganz  
552 gerne, Programme folgen deinen Befehlen, nicht deinen Absichten. Dann die



553 Frage für sich neu formulieren und dann nochmal im Lernmaterial nachschauen  
554 oder irgendwo anders nachrecherchieren, dass ... die nächste Stufe nach  
555 sozusagen dem Hinterfragen, das Recherchieren, dann an die MitschülerInnen  
556 wenden und nach Problemhilfe fragen. Und jetzt in der letzten Instanz, also auf  
557 der vierten Ebene die Lehrperson bemühen. Denn die Lehrperson kann sich  
558 sowieso nicht zwölfteln, oder vierzehnteln, oder sechzehnteln, da hat man  
559 nachher nur die typische Turnschuh-Didaktik, wo die Lehrperson herumrennt.  
560 Und es soll auch nicht so sein, dass die SchülerInnen und Schüler beim  
561 kleinsten Lernwiderstand immer auf externe Hilfe, also auf MitschülerInnen und  
562 Lehrpersonen zurückgreifen und immer sagen, Herr Professor, Herr Professor  
563 es geht nicht, sondern wirklich Mal Stufe eins und Stufe zwei, also die untersten  
564 Ebenen dieser Pyramide durchmachen.

565

566 **I:** Und was hatten die SchülerInnen dann an sich, die nicht auf die  
567 Schwierigkeiten gestoßen sind?

568

569 **B:** [0:43:40] Die SchülerInnen, die nicht auf diese Schwierigkeiten gestoßen  
570 sind, die hatten schon größtenteils ein solides Vorwissen im Bereich der  
571 Bedienung und Handhabung von PCs und den Tools auf Personal Computern  
572 und in Bezug auf unterschiedliche Software. Dann haben die aber auch  
573 meistens eine überdurchschnittliche Auffassungsgabe gehabt und waren sehr  
574 offen, also haben so eine gewisse Offenheit gegenüber Prozessen gehabt, also  
575 haben nicht sofort versucht ihre eigenen Schablonen auf das Ganze anzulegen,  
576 sondern sich auch auf neue Dinge eingelassen und haben probiert neue Muster  
577 auszuprobieren. Dann haben die, die nicht so lange, also auf Schwierigkeiten  
578 sind fast alle gestoßen, aber die die die Schwierigkeiten selbst bewältigen  
579 konnten, die haben eben die gerade vorhin genannten Kompetenzen, die in den  
580 unteren Ebenen dieser Supportpyramide im Sinn gehabt und haben aber schon  
581 auch eine gewisse Hartnäckigkeit und Resilienz gehabt, bei der Lösung der  
582 Probleme. Ja, das war es eigentlich. Aber das ist ja eh schon eine ganze Latte  
583 an Eigenschaften, die man mitbringen muss.

584

585 **I:** [0:44:46] Wenn wir uns jetzt Mal etwas Fiktives vorstellen. Angenommen  
586 morgen beginnt ein neues Schuljahr und die SchülerInnen sind bestens  
587 gerüstet, um direkt eben mit der Spielentwicklung mit Unity zu beginnen. Woran  
588 würden Sie diese optimale Rüstzeug der SchülerInnen erkennen?

589

590 **B:** [0:45:05] Also, dass die grundlegende Handhabung eines PCs sattelfest ist.  
591 Das heißt, das erwähnte Filemanagement, das Umgehen mit Programmen, wie  
592 ich eben Programme auch vielleicht selbst mir aneigne und erforsche, also so  
593 explorative Muster. Ich habe jetzt ein neues Tool, wie schaue ich strukturiert  
594 nach, was es alles kann und was es gibt. Dann schon auch eine gewisse  
595 soziale Kompetenz, eine Recherche-Kompetenz und dann alles was ich  
596 irgendwie unter ein Vorwissen für Informationstechnologien an sich  
597 subsumieren würde, also dass man sich das Ganze vorstellen kann, wie  
598 Technologien funktionieren und wie Technologien funktionieren können und so  
599 ein technisches Verständnis im Allgemeinen. Das wäre ein gutes Rüstzeug.

600

601 **I:** Wenn Sie die Zeit für Sie und die Klasse anhalten können, was würden Sie  
602 den SchülerInnen - also vor, noch bevor der Spielentwicklung mit Unity -  
603 beibringen, unbedingt beibringen wollen?

604

605 **B:** [0:46:28] Also, wenn ich ausreichend viel Zeit hätte?

606

607 **I:** [0:46:32] Ja.

608

609 **B:** [0:46:33] Dann würde ich auf jeden Fall davor mich schon, das war zwar  
610 nicht Ziel des Projektes, aber davor mich schon mit objektorientierter  
611 Programmierung auf der Konsole zum Beispiel auseinandersetzen, würde ich  
612 den SchülerInnen und Schülern davor diese ganzen Tools, die die Assests  
613 zuliefern, zum Beispiel Bildbearbeitungsprogramm, oder Programm zum  
614 Modellieren, 3D-modellieren, wie zum Beispiel Blender würde ich ihnen dann

615 näher bringen. Audacity würde ich ihnen zum Aufnehmen, zum Audio-  
616 engineering einfach näherbringen, dass sie dann wirklich alles selber machen  
617 könnten. Also wenn die Zeit quasi unbegrenzt wäre, oder beliebig wäre, dann  
618 würde ich auch die ganzen Sachen drumherum den Lernenden näherbringen.  
619 Dann würde ich ihnen noch Game-Design näherbringen, also würde ich die  
620 Prinzipien des Game-Designs explizieren, weil so machen sie halt das was sie  
621 cool finden und was sie implizit mitbekommen haben, weil sie, weil die meisten  
622 ja selber Spiele spielen und eigentlich fast alle Spiele kennen. Und dann würde  
623 ich alle möglichen Sachen, die ich jetzt erzählt habe in ausführlicher  
624 Gründlichkeit und Genauigkeit, je nach den Wünschen und Bedürfnissen der  
625 Lernenden durchmachen und mit den Lernenden gemeinsam erforschen.

626

627 **I:** [0:47:59] Man könnte das fast so sagen, also ein ganzes Softwareprojekt  
628 quasi durchspielen.

629

630 **B:** [0:48:07] Genau, mit allen möglichen Rollen, dies es in so einer  
631 Spieleentwicklung gibt, von der Projektleitung über Audioengineers und  
632 Grafikerinnen und vorwiegend Engine-ProgrammiererInnen und allen möglichen  
633 Bestandteilen. Dass sie in jedem Feld von dem schnuppern könnten, Game-  
634 DesignerInnen.

635

636 **I:** [0:48:27] Jetzt noch bereichsspezifische Fragen, die ich Ihnen stellen würde.  
637 Benötigen SchülerInnen Ihrer Ansicht nach für die Spielentwicklung mit Unity  
638 Kenntnisse und Fertigkeiten aus dem Bereich Verantwortung, Datenschutz und  
639 Datensicherheit?

640

641 **B:** [0:48:47] Durchaus, ja, da werden zum Beispiel gleich am Anfang bei Unity  
642 die Fragen gestellt: "Sollen wir Unity-Analytics verwenden? Was macht das?"  
643 Also, es ist nicht zwingend notwendig, aber es wird thematisiert, also  
644 Verantwortung, Datenschutz und Datensicherheit. Dann die ganze große Frage  
645 der Lizenzen von Bildern und Assets die ich hernehme. Dann auch die

646 erwähnte Angelegenheit mit dem sich gegenseitig zu fotografieren und die  
647 eigenen Gesichter sozusagen herumspringen lassen oder herumpurzeln  
648 lassen. Das heißt, in Wirklichkeit muss man wissen sozusagen was das Recht  
649 an Bild bedeutet und was Urheberrecht bedeutet, was die verschiedenen  
650 Lizenzen bedeuten. Dann eben auch warum man Unity-Analytics verwendet  
651 oder eben nicht und wenn man dann Projektdateien teilt, ob man die gleich im  
652 Web veröffentlicht zum Download, oder wie man es am Netzwerk teilt, das sind  
653 alles Sachen, die im Bereich Verantwortung, Datenschutz und Datensicherheit  
654 für mich fallen. Und die tatsächlich, wenn sie nicht vorher vorhanden sind, nach  
655 der digitalen Grundbildung, dass die dann spätestens dann thematisiert werden.  
656 Aber auf die direkte Frage, ob sie es benötigen, kann ich antworten: "ja".

657

658 **I:** [0:50:03] Wie sieht das Ganze bezüglich Kenntnisse und Fertigkeiten im  
659 Bereich der technischen Grundlagen und Funktionen, Funktionsweisen aus,  
660 sprich Peripheriegeräte.

661

662 **B:** [0:50:15] Na im Speziellen haben wir da immer die Geschichte gehabt mit  
663 der nicht so starken Grafikkarte, also mit der OnBoard-Grafikkarte und mit der  
664 CPU, die das dann nicht gestemmt hat. An und für sich muss man auf jeden  
665 Fall wissen, dass Spiele ressourcenintensiv sind, dass die schon von den  
666 Hardwareressourcen eines Rechners einiges abverlangen. Also das braucht  
667 man auf jeden Fall. Und dann hat es so nette Details gegeben, wie irgendwann  
668 einmal haben wir Nvidia-Grafikkarten auftreiben können und durften die dann,  
669 weil die gerade so billig waren tatsächlich kaufen, von der Direktorin aus und  
670 die haben wir dann eingebaut und dann haben wir gesehen, ja also mit den  
671 Grafikkarten geht es dann besser als mit den Onboard-Chips. Also man braucht  
672 die Grundlagen im Grundlagenbereich und auf die Funktionsweisen bezogen  
673 sowieso und vorgekommen ist es dann auch, sogar spezifischer, im Bereich auf  
674 Peripherie und Hardware.

675

676 I: [0:51:01] Und in Bezug auf Betriebssysteme und Software? Wie eben  
677 Dateimanagement oder Datensicherung und auch die Wiederherstellung  
678 davon?

679

680 B: [0:51:17] Da sind die Kenntnisse höchst wünschenswert. Also, das hat auch  
681 sehr viele Probleme gemacht, weil die SchülerInnen und Schüler ihr  
682 Dateimanagement, das waren zum Glück nur Ausnahmefälle, nicht beherrscht  
683 haben und sich im eigenen Projekt nicht mehr ausgekannt haben.  
684 Dateimanagement ist fast eben Voraussetzung. Das mache ich auch relativ am  
685 Anfang eines Schuljahres, also das man alles organisieren kann in Ordern. Und  
686 bei der Sicherung genauso. Also wenn man ein Projekt so verbockt hat, dass es  
687 einfach Fubar ist, dann muss man einfach auf eine Sicherung oder auf eine  
688 frühere Variante zurückgehen. Zusätzlich haben wir im Projekt noch eingebaut,  
689 dass wir nach jedem Einzelschritt die Zip-Pakete angeboten haben, dass man  
690 sozusagen falls wirklich gar nichts mehr gegangen ist, dass man beginnend  
691 vom Schritt drei zum Beispiel wieder weiterarbeiten hat können. Aber erwünscht  
692 war es und auch erforderlich, um den Prozess reibungs- und verlustfrei zu  
693 gestalten, dass man in beiden Bereichen, also Sicherung und  
694 Dateimanagement Kenntnisse und Fertigkeiten vorweist.

695

696 I: [0:52:42] Gibt es da in diesem Bereich vielleicht weitere Punkte, vielleicht  
697 betriebssystemspezifische, wie bei den Einstellungen oder so, oder bei der  
698 Software mit den Office-Programmen zum Beispiel?

699

700 B: [0:52:54] Ja, Office-Programme (...). Ich überlege gerade. Einstellungen bei  
701 den Betriebssystemen fällt mir ad-hoc nichts ein. Aber vielleicht weiß ich auch  
702 nur nicht was gemeint ist.

703

704 I: [0:53:10] Vielleicht, wenn man mit Audacity etwas aufnehmen will für Ton zum  
705 Beispiel oder so, dass man da Mikrofoneinstellungen ändern muss, oder solche  
706 Dinge zum Beispiel. Oder vielleicht auch um den Dark-Mode einzustellen?

707

708 **B:** [0:53:25] Ja durchaus, aber das ist eher im Bereich (...). Also man stellt das  
709 im Betriebssystem ein, das stimmt und da ist vielleicht der Sound ausgestellt,  
710 aber das hätte ich jetzt eher dem Bereich der Asset-Gestaltung zugeordnet.  
711 Aber stimmt ja, das greift ins Betriebssystem mit ein.

712

713 **I:** [0:53:57] Dann zum Bereich Produktion digitaler Medien, also Text, Bild, Ton  
714 und die Bearbeitung. Benötigen sie da Kenntnisse und Fertigkeiten bereits  
715 vorher?

716

717 **B:** [0:54:18] Also, im Bereich Text, vielleicht wenn man Storys für ein Spiel  
718 schreiben will, dann ja Textverarbeitung. Aber es kommt kaum vor, dass man  
719 ohne mit Text, oder Textverarbeitung oder Texten zu tun gehabt hat, dann  
720 irgendwie in die Spieleprogrammierung geht. Beim Audioengineering im  
721 Bereich Ton braucht man eigentlich nichts. Ich meine, das Tröten des  
722 Elefanten, das war einmal Thema, ob man das selbst aufnimmt oder darstellt.  
723 Und im Bereich Bildbearbeitung, also es wird zwar alles von Anfang an erklärt,  
724 aber es ist schon ganz gut, wenn man mit Gimp umgehen kann. Das gilt  
725 eigentlich für alle Fragen in diesem Bereich, theoretisch ist es so konzipiert,  
726 dass man von Null anfangen kann. Also sowohl was den Bereich des  
727 Datenschutzes und den technischen Grundlagen, ja was diese Bereiche betrifft,  
728 oder Betriebssysteme und Software oder Audio, Video, man könnte theoretisch  
729 eigentlich von Null anfangen und das erklären, nur braucht man dann halt  
730 theoretisch mehr Zeit.

731

732 **I:** [0:55:20] Aber was heißt direkt neu beginnen? Direkt mit der Spielentwicklung  
733 mit Unity oder eben mit dem ganzen Projekt, beginnend mit Gimp, Scratch  
734 usw.?

735

736 **B:** [0:55:34] Für Unity selbst? Für das Coden?

737

738 I: [0:55:41] Ja.

739

740 B: [0:55:42] Braucht man eigentlich, naja von diesen Sachen ad-hoc jetzt  
741 vielleicht ein bisschen Dateimanagement, damit man weiß, wo man die Assets  
742 speichert. Dann kann man schon vorgegebenen Pfaden mit Unterstützung  
743 folgen. Aber um das Projekt zu realisieren, braucht man an irgendeinem Punkt  
744 all diese Fähigkeiten, aber man kann auch, und das hat der Kollege  
745 (anonymisierte Person) gemacht, er hat gesagt, er beginnt jetzt einfach wirklich  
746 mit der Spieleentwicklung, Spieleentwicklungsprojekten und erklärt alles im  
747 Zuge dieser Projekte.

748

749 I: [0:56:20] Und dann noch ein letzter Bereich. Die Suche und die Auswahl und  
750 Organisation von Informationen?

751

752 B: [0:56:32] Also da gilt das Gleiche. Wenn man es kann, ist es super, weil  
753 dann kann man sich gleich die eigenen Assets suchen und schaut sich das an,  
754 welche frei verfügbar sind. Oder man kann sich gleich irgendwelche Sachen,  
755 die man selber implementieren will, auch im Internet anschauen, hervorholen  
756 und dann eben auch die Wahrscheinlichkeit steigern, dass man das Richtige  
757 findet. Oder eben, wenn man es nicht hat, geht das Ganze etwas langsamer  
758 von statten. Aber es ist so, dass im Projektpaket die Assets mit reingeschnürt  
759 sind. Die kann man sich herunterladen. Ich müsste eigentlich nur im Bereich  
760 des Tutorials stöbern und dann dort alle Informationen finden und nutzen.

761

762 I: [0:57:33] Also stellen keine unbedingten Voraussetzungen dar?

763

764 B: [0:57:37] Genau. Also es ist nicht so, dass man nicht anfangen könnte. Und  
765 in der Tat hat mein Kollege gesagt, er will jetzt im Zuge des Projektes einmal so  
766 anfangen mit den Lernenden, ohne das er das explizit vorher den Lernenden  
767 angeeignet hat und dann gesagt hat, so das braucht ihr jetzt, sonst geht es  
768 nicht. Also, man kann immer alles on-the-fly lernen.

769

770 **I:** [0:58:03] Die Frage die sich da aber stellt ist, wie verzögert sich da das  
771 Projekt? Oder ist es dann noch effektiv, dass man da wirklich effektiv mit Unity  
772 Spiele entwickeln kann, oder?

773

774 **B:** [0:58:16] Ja genau. Das ist ein guter Hinweis. Das verzögert sich nämlich  
775 dann deutlich, erheblich.

776

777 **I:** [0:58:21] Sodass es sogar mehr Aufwand ist oder länger dauert, das Ganze  
778 als wenn man eben so vorbereitende Maßnahmen setzt?

779

780 **B:** [0:58:35] Das kann auch noch länger dauern, ja, weil sozusagen dann das  
781 Lernen in so einem Spannungsgeflecht ist und nicht so leicht stattfinden kann.  
782 Das sieht man vielleicht auch bei neuen Sportarten oder beim  
783 Bewegungslernen, wenn man da methodisch heran geht, das ist oft hilfreicher,  
784 das technisch richtig zu lernen, als wie wenn man völlig chaotisch herangeht  
785 und dann schaut wann man was braucht. Wenn man schon einmal was  
786 gefestigt hat, dann kann man auch darauf zurückgreifen und durch Verknüpfung  
787 neues darauf aufbauen. Und wenn man alles gleichzeitig irgendwie lernt, dann  
788 kann sein, dass sich das gegenseitig hemmt.

789

790 **I:** [0:59:25] Ok. Wir wären jetzt am Ende des Interviews. Gibt es vielleicht noch  
791 etwas das Sie los werden wollen oder ergänzen wollen?

792

793 **B:** [0:59:35] Hmm. Was ich los werden will. Naja, das was mich an diesen  
794 Spielentwicklungsprojekten erstaunt hat ist, dass die Lernenden irgendwie in  
795 ihrer Selbstwirksamkeit sehr gestärkt hervorgehen. Also sie trauen sich mehr zu  
796 und sie schätzen sich auch bezüglich ihrer Kompetenzen in verschiedenen  
797 Bereiche, wie Algorithmen, Datenstrukturen oder Modellierung und all diesen  
798 Dingen, kompetenter ein. Und dass das eine gute Grundlage ist, wenn man mit  
799 zukünftigen Bildungswegen/Karrierewegen dann anschauen möchte, wie es da



800 weitergehen kann für einen selbst. Das ist vielleicht auch irgendwie die Affinität  
801 im Bezug zur Informatik zu verbessern. Und ansonsten, dass das Feedback  
802 sehr oft gekommen ist, es war anstrengend und es gab Schwierigkeiten, aber  
803 die Probleme konnten gelöst werden und es war sehr viel Spaß, es war mehr  
804 Spaß als der herkömmliche Unterricht.  
805  
806

---

1 **10.7.2 Interview 02**

2 [0:00:00] **I:** Beschreiben Sie bitte einmal Ihre Erfahrungen mit dem  
3 Unterrichtsprojekt "learn to proGrAME".  
4  
5 **B:** [0:00:10] Also ich habe angefangen zu arbeiten auf der Uni Wien als  
6 wissenschaftlicher Mitarbeiter und habe dann im Zuge dieser Arbeit  
7 Lehrmaterial erstellt und den Unterricht in zwei Klassen begleitet. Das war  
8 einmal eine fünfte Klasse im Realgymnasium und einmal eine siebente Klasse  
9 im Wahlpflichtfach. Und meine Erfahrungen waren, also grundlegende  
10 Eindrücke waren, dass die Schülerinnen und Schüler sehr, sehr motiviert  
11 waren, vor allem am Anfang und das die Erwartungen nicht unbedingt erfüllt  
12 worden sind, also, dass die Erwartungen vielleicht zu hoch waren, für ja wir  
13 programmieren jetzt ein eigenes Spiel. Und am Schluss, am Ende vom  
14 Semester haben wir auf einmal GTA6 programmiert. Ein bisschen überspitzt  
15 formuliert. Es ist dann durchaus irgendwann so der Effekt gekommen, ok, es ist  
16 doch nicht so einfach eigentlich. Aber auch danach war die Motivation noch  
17 recht hoch, also, nach meinem Eindruck noch immer höher als in anderen  
18 Themen. Das zum einen und zum anderen habe ich insbesondere ein paar  
19 Projektarbeiten begleitet, die ich auch im Rahmen von meiner Diplomarbeit  
20 analysiert habe und da ist mir vor allem aufgefallen, dass die Schülerinnen und  
21 Schüler sich sehr unterschiedlich auf verschiedene Bereiche der  
22 Spielentwicklung fokussieren und dementsprechend natürlich auch sehr  
23 unterschiedliche Lernerfolge dann haben schlussendlich. Das heißt, Game-

24 Development ist ein wahnsinnig breites Gebiet, wo man sehr, sehr viele  
25 Kompetenzen in verschiedensten Bereichen haben müsste oder haben muss,  
26 wenn man ein vollständiges Spiel entwickeln mag. Und meine Erfahrung war,  
27 dass wenn man sie dann in Projekten arbeiten lasst, wo sie selbstständig  
28 arbeiten, dass sie dann mitunter gewisse Teile vielleicht ein bisschen  
29 untergehen lassen oder hintan rücken, andere Teile aber sich sehr intensiv  
30 damit beschäftigen.

31

32 **I:** [0:02:21] Also, wenn ich das richtig verstehe, eine starke Spezialisierung fast  
33 stattfindet.

34

35 **B:** [0:02:28] Ja, das konnte ich auf jeden Fall beobachten, ja.

36

37 **I:** [0:02:37] Wollen Sie noch weitere Erfahrungen oder so, oder?

38

39 **B:** [0:02:42] Hm. Was gibt es da noch? In welche Richtung? Also, einfach nur  
40 Erfahrungen ist irgendwie sehr, sehr breit. Mir fällt jetzt im Moment nichts ein.

41

42 **I:** [0:02:52] Oder was vielleicht sehr auffällig oder herausgestochen ist für Sie.

43

44 **B:** [0:02:59] Es ist auch sehr herausgestochen, dass für mich das einzelne  
45 Schülerinnen und Schüler sehr, sehr motiviert waren und andere aber weniger.

46

47 **I:** [0:03:17] Also doch eine sehr große Diskrepanz, manche stark interessiert,  
48 andere ...

49

50 **B:** [0:03:23] Ja schon, ja.

51

52 **I:** [0:03:28] Wodurch zeichnet sich Ihrer Meinung nach ein guter Programmierer  
53 aus?

54

55 **B:** [0:03:34] Durchhaltevermögen und Neugierde. Das sind so zwei wirklich  
56 wichtige Dinge, weil ähm (...). Also Durchhaltevermögen, damit meine ich, ich  
57 muss etwas schaffen wollen, also ich brauche die Neugierde dafür, ich muss,  
58 ich will, oder ich habe keine Angst vor Problemen, oder ich habe ein Ziel und  
59 ich will dieses Ziel erreichen und ich lasse mich nicht davon verunsichern, dass  
60 ich vielleicht keine Ahnung habe, wie das überhaupt funktioniert und wie das  
61 geht. Dass wenn ich einmal etwas ausprobieren, dass ich mich nicht davon  
62 unterkriegen lasse, dass es nicht funktioniert, sondern dass ich dann versuche  
63 den Fehler zu finden. Also auch ich bin jetzt am Ende von meinem Studium in  
64 Informatik und auch ich mache nach wie vor, wenn ich etwas programmiere,  
65 dann ist das meistens, ok, ich kenne die Sprache vielleicht nicht, ich schaue mir  
66 einmal irgendein Programm an und ich würde gerne irgendwo hinkommen und  
67 dann schaue ich mir an, wie man das machen könnte und dann probiere aus  
68 und dann suche ich mir selbstständig Tutorials, Inhalte dazu, recherchiere  
69 selber und das ist glaube ich ein Skill den man immer braucht wenn man  
70 programmiert, weil es kommt immer eine neue Sprache, es kommen immer  
71 neue Features und man muss sich diese selbst beibringen können und diese  
72 Neugierde bewahren.

73

74 **I:** [0:04:59] Dahingehend, dass sie sich auch sehr schnell verändern kann.

75

76 **B:** [0:05:03] Ja. Ich glaube ein ganz, ganz wesentlicher Faktor ist eben der,  
77 dass ich wirklich was erreichen will und da durchbeiße. Weil wenn ich eben, ich  
78 will ein Spiel programmieren und ich will, dass die Figur springt, wenn ich die  
79 Leertaste drücke und ich will, dass sie an der Wand dann abprallt, dann muss  
80 ich das irgendwie schaffen, dass ich das programmiere, beziehungsweise in  
81 Unity die Engine dafür verwenden und das funktioniert auf Anhieb  
82 wahrscheinlich nicht. Und das wirkt, wenn man sich gar nicht auskennt, dann  
83 wirkt das oft sehr einschüchternd und das schreckt glaube ich viele Leute ab.  
84 Und deswegen ist das glaube ich so eine der zentralen Aspekte, die man  
85 braucht. Man muss es wirklich wollen. Es ist schwer finde ich, Schülerinnen und

86 Schüler dazu zu bringen, im speziellen in Programmieren, generell auch  
87 allgemein, aber im Programmieren kommt es sehr stark heraus, wenn sie es  
88 nicht lernen wollen. Wenn sie einen Arbeitsauftrag haben, programmiere mir ein  
89 Programm, das die Fläche eines Rechtecks berechnet, dann gibt es ein paar,  
90 die machen das, dann gibt es ein paar die versuchen es, aber scheitern dann  
91 daran und dann gibt es ein paar die sowieso sagen, sie haben keine Ahnung.  
92 Aber wenn es vielleicht ein anderes Programm ist, das sie persönlich  
93 interessanter finden, zum Beispiel eben, programmiere, wenn du die Leertaste  
94 drückst, dann soll die Figur springen, dann ist da viel mehr Eigenmotivation da  
95 und das ist dann das, was es eigentlich ausmacht, was den Leuten zum Erfolg  
96 verhilft. Denn es ist nicht einfach, man muss es wirklich lernen, es ist schwierig.  
97 Und ohne Eigenmotivation bleiben die Lernerfolge meistens, nach meiner  
98 Beobachtung, aus. Ich glaube, das ist eben ein Punkt, wo die  
99 Spieleprogrammierung die Jugendlichen sehr gut abholen kann, weil ist von  
100 vornherein sehr viel Motivation da, im Gegensatz zu Konsolenprogrammen, die  
101 irgendetwas berechnen.

102

103 **I:** [0:07:06] Ich schmunzle deshalb so, weil habe es ja selbst auch ein bisschen  
104 so, bin ich es durchgegangen und das war bei mir das Gleiche, dass ich zuerst  
105 einmal gesagt habe, wow, das ist sehr komplex, das braucht ein bisschen zum  
106 Einarbeiten, da muss man schon genau schauen und so. Ok, das war jetzt ein  
107 wenig abseits vom Interview quasi, aber. Wo sehen Sie da einen guten  
108 Programmierer in Bezug auf, wirklich speziell auf digitale Kompetenzen jetzt,  
109 weil da waren ja auch Softskills jetzt dabei, vor allem mit dem Durchhalten und  
110 der Neugierde. Aber wo ist es vielleicht spezieller mit den digitalen  
111 Kompetenzen, in diese Richtung?

112

113 **B:** [0:07:50] Ja also wirklich in Bezug auf die digitalen Kompetenzen ist es,  
114 glaube ich kann man sagen, einen guten Programmierer, eine gute  
115 Programmiererin unterscheidet sich insofern von einem schlechteren oder  
116 schlechten, dass vielleicht nicht nur der Fokus darauf liegt, dass das Programm

117 funktioniert, sondern auch dass es wartbar ist und dass es dokumentiert ist und  
118 dass andere Leute damit arbeiten können. Das ist ein sehr, sehr weit  
119 verbreitetes Problem in der Informatik, dass jemand ein System entwickelt und  
120 dann in Pension geht, oder an einem anderen Projekt weiter arbeitet und dann  
121 stirbt das Projekt einfach nur weil jemand anderes nichts damit anfangen kann,  
122 weil ich komme hin und schaue mir den Code an und denke, was macht das,  
123 wie. Da programmiere ich es lieber von Grund auf neu. Und ich glaube das ist  
124 sehr weit verbreitet und das bekomme ich auch von Freunden und Kollegen zu  
125 hören, die als Programmierer arbeiten, dass das sehr weit verbreitet ist, oft das  
126 ein Programmierer, auch wenn es fachlich gut funktioniert zum Beispiel das  
127 Programm, ein Programm macht, das dann nicht zukunftstauglich ist, weil es  
128 einfach schlecht dokumentiert ist zum Beispiel, oder bestimmte coding-  
129 Konventionen nicht eingehalten werden, oder Programmierparadigmen. Das ist  
130 so auf der Programmiererebene ein Skill glaube ich, wo sich so gut und schlecht  
131 unterscheidet. Und ansonsten die Anpassungsfähigkeit und dass man wirklich  
132 Systeme versteht und wirklich versteht warum, wie eine Sprache funktioniert,  
133 was die Sprache genau macht, was das Statement genau macht.

134

135 **I:** [0:09:43] Sozusagen den Algorithmus versteht.

136

137 **B:** [0:09:47] Ja. Dass ich analytisch denken kann, dass ich mir überlege, was  
138 soll das Programm können, oder was soll diese Funktion, oder diese  
139 Komponente vom Programm können und mir dann überlege, ok wie kann ich  
140 jetzt die, mit den Limitierungen, die vielleicht die Sprache hat, oder das System  
141 hat, mit dem ich arbeite, wie kann ich dieses Problem lösen. Auf der einen Seite  
142 eben nur dieses auf die Funktionalität hin, aber dann gleichzeitig auch auf die  
143 Lesbarkeit und Wartbarkeit von Programmen, das ist wichtig.

144

145 **I:** [0:10:21] Ok. Und wo kann dabei die Spieleentwicklung mit Unity einen  
146 Beitrag leisten?

147

148 **B:** [0:10:28] Einerseits wie vorher schon angesprochen, ein wesentlicher Faktor  
149 ist die Eigenmotivation. Sei es jetzt, wenn ich professioneller Programmierer  
150 bin, dass es mein Brot-Job ist, ich verdiene damit Geld, deswegen muss ich das  
151 programmieren und deswegen lerne ich es dann auch und deswegen kann ich  
152 es und was in der Schule halt nicht gegeben ist. Und deswegen muss man  
153 glaube ich schauen, dass man sich interessante Aufgabenstellungen überlegt,  
154 oder interessante Anwendungen überlegt, wo Schülerinnen und Schüler dann  
155 wirklich hergehen und sagen, ja das ist cool, das mag ich machen und schau  
156 was ich mach und schau was ich programmiert habe, schau was ich machen  
157 kann, das ist cool. Und dieser Aspekt ist bei der Spieleentwicklung von  
158 vorneherein normal irgendwie mehr da als bei vielen anderen Sachen.  
159 Gleichzeitig geht damit einher, dass es komplizierter ist, wirklich Software zu  
160 schreiben, die dann wirklich angewendet wird, sei es jetzt eben irgendwelche  
161 Programme oder wirklich auch Spiele. Weil wenn ich es dann wirklich spielen  
162 mag, dann muss ich natürlich noch ein ganz anderes Level an Perfektion  
163 mithinein bringen. Und dazu kommt noch, dass Spieleprogrammierung an sich  
164 schon ein recht breites Thema ist und es schwerer ist, in meinen Augen den  
165 Fokus auf bestimmte Aspekte zu setzen. Zum Beispiel wenn ich jetzt Konzepte  
166 wie Variablen oder Funktionen vermitteln will und das anhand von  
167 Spielentwicklung machen will, dann ist es einfach schwierig, denn ich habe so  
168 viel drumherum, das muss ich dann irgendwie gut verpacken oder verstecken  
169 oder schon zur Verfügung stellen. Und ich kann den Fokus, von dem was an  
170 Kompetenzen vermittelt werden soll, ist bei Spieleprogrammierung für mich als  
171 Lehrperson ein bisschen schwieriger, diesen Fokus zu setzen, weil immer viel  
172 mehr drumherum ist.

173

174 **I:** [0:12:38] Weil viel mitschwingt sozusagen.

175

176 **B:** [0:12:39] Weil viel mitschwingt. Ich habe die Gameschleife, ich habe die  
177 Unity-spezifischen Sachen, ich kann weniger mir die bestimmten Aspekte  
178 herauspicken und mit denen was machen, als wenn ich jetzt zum Beispiel sage,

179 wir machen ein Konsolenprogramm und heute lernen wir über Operatoren und  
180 ihr sollt ein Programm schreiben, das ist ein Taschenrechner, der soll  
181 automatisch flächen berechnen von Dreiecken, Rechtecken, was weiß ich was  
182 und dann schreiben wir dann eine Funktion jeweils, eine Funktion für Fläche  
183 Rechteck, eine Funktion für Fläche Dreieck. Da kann ich mir sehr viel  
184 fokussierter bestimmte Aspekte herausholen und die vermitteln. Bei der  
185 Spieleentwicklung kann man das natürlich auch machen, aber es ist didaktisch  
186 schwieriger für mich. Und ansonsten wo es eben auch sehr gut, wofür sich  
187 Spieleentwicklung sehr gut eignet ist, meiner Meinung nach für Projektarbeiten,  
188 wo Schülerinnen und Schüler dann sehr frei arbeiten können und ihre eigenen  
189 Ziele verfolgen können und auch im Team arbeiten und die vorher von mir  
190 angesprochene Lesbarkeit und Wartbarkeit von Code zum Beispiel, die kommt  
191 wenn ich selbst in einem Team ein Programm schreibe, wird das nötig. Dann  
192 habe ich nämlich entweder den Effekt, dass nur einer programmiert und der  
193 Rest damit nichts zu tun hat, weil den Code verstehe ich sowieso nicht von dem  
194 anderen, oder, das wäre der bessere Fall, sie arbeiten gemeinsam und der eine  
195 sagt, ich verstehe nicht was du da machst, kannst du mir erklären, was dein  
196 Code macht und dann kommt man vielleicht darauf, ok, schreib das ein  
197 bisschen anders, oder füge da noch einen Kommentar ein, oder sonst  
198 irgendwas. Das heißt, dieser Aspekt wird dann automatisch fokussiert, wenn sie  
199 im Team arbeiten. Und was auch noch ein ganz relevanter Punkt ist, bei der  
200 Spieleprogrammierung werden sehr, sehr viele Aspekte der Informatik vernetzt.  
201 Und dieses, man könnte da fast vielleicht von einem transdisziplinären Rahmen  
202 sprechen, weil es hat zwar noch immer alles mit Computern zu tun, aber es ist  
203 trotzdem, ich kann, ich muss meine Bilder zeichnen, ich muss meine 3D-  
204 Modelle machen, die Animationen, Soundeffekte. Dann muss ich das alles  
205 irgendwie verknüpfen, ich muss ein Programm schreiben, ich muss den Code  
206 schreiben für das Verhalten, für die Programmlogik, für die Animationen, ich  
207 muss die Animationen steuern. Da kommt einfach wahnsinnig viel zusammen  
208 und diese Sachen, die sonst vielleicht im Unterricht getrennt behandelt werden  
209 oder wurden, können dann in so einem Spielprojekt verknüpft werden. Und das

210 kann dann wahnsinnig gut funktionieren. Das hat teilweise im Projekt sehr, sehr  
211 gut funktioniert. Teilweise, also das war mein, was ich auf jeden Fall  
212 mitgenommen habe ist, dass man sich die Projekte sehr gut strukturieren muss.  
213 Wenn ich einfach nur sage, macht ein Spiel, jetzt habt ihr ja gelernt wie die  
214 Grundlagen funktionieren und jetzt macht ihr euer eigenes Projekt, geht in  
215 zweier/dreier Teams zusammen und macht euer eigenes Spiel, dann  
216 funktioniert das manchmal, wenn die Leute eine sehr hohe Eigenmotivation  
217 haben und vielleicht auch gute Ideen haben, aber andere sind dann überfordert,  
218 oder wenig motiviert. Das heißt, man muss schon irgendwie den Rahmen  
219 festlegen und sie unterstützen dabei. Und dadurch dass ebenso ein breites  
220 Spektrum an Kompetenzen in verschiedenen Bereichen nötig sind, bietet die  
221 Spieleentwicklung in meinen Augen auch wiederum großartige Möglichkeiten  
222 zur Individualisierung oder zur Stärkung von individuellen Begabungen, oder  
223 Talenten, oder Interessen, weil zum Beispiel wir haben ein Projektteam, wir  
224 machen ein Spiel zu viert, ich zeichne gerne, ich mache jetzt einmal echt coole  
225 Grafiken für das Spiel und die andere Person macht die Soundeffekte oder die  
226 Musik dazu und so kann jeder seine eigenen Stärken fokussieren und diese  
227 einbringen und dafür eignet es sich auch sehr stark. Dieser Aspekt ist auch  
228 einer der glaube ich sehr gut von selbst passiert und gleichzeitig ist es dann  
229 auch immer schwierig, ok wo will ich eigentlich hin. Wenn ich jetzt wirklich das  
230 vernetzende Denken oder die vernetzende Arbeit fokussieren mag mit so einem  
231 Spielprojekt aus der didaktischen Sicht, dann muss ich das wirklich auch aktiv  
232 einleiten und den Rahmen dafür setzten.

233

234 **I:** [0:17:17] Inwiefern, das verstehe ich jetzt nicht ganz.

235

236 **B:** [0:17:20] Also Beispiel, wir machen im Unterricht Unity durch, die  
237 Grundlagen werden alle erklärt, wir machen Beispiele durch und dann zum  
238 Schluss machen wir ein großes Projekt, wo die Schülerinnen und Schüler in  
239 Viererteams ein eigenes Spiel entwickeln oder ein vorhandenes Spiel  
240 modifizieren. Und wenn ich sie einfach so arbeiten lasse, dann habe ich eben



241 eventuell den Effekt, dass eine Person kümmert sich um Grafiken, die andere  
242 um die Sounds, die anderen beiden programmieren alles und fügen alles  
243 zusammen. Und dann habe ich einen Lerneffekt, bei der einen Person in Bezug  
244 auf die Erstellung von Grafiken, bei der anderen Person in Bezug auf Musik und  
245 Sound und bei den anderen beiden Programmierung und vielleicht Unity-Engine  
246 usw. Wenn das das Ziel ist von dem Projekt, dann ist es natürlich super  
247 geeignet dafür, wenn ich allerdings eben will, dass alle programmieren oder alle  
248 Grafiken erstellen oder sich alle zumindest auskennen damit, dann muss ich als  
249 Lehrperson da aktiv einen Rahmen setzen, denke ich.

250

251 **I:** [0:18:29] Ok, so war das gemeint, verstehe schon.

252

253 **B:** [0:18:31] In diesen Projekten, die ich betreut habe, hat man das dann auch  
254 gesehen, dass teilweise in Projekten die Arbeiten einfach aufgeteilt wurden, aus  
255 Interessen oder aus anderen praktischen Gründen, teilweise auch nicht. Also  
256 das ist (...)

257

258 **I:** [0:18:52] Gruppenabhängig.

259

260 **B:** [0:18:55] Ja.

261

262 **I:** [0:18:59] Wenn Sie jetzt an eine Musterschülerin oder einen Musterschüler im  
263 Projekt denken, was denken Sie nahm der- oder diejenige für einen etwaigen  
264 Job als Programmierer mit?

265

266 **B:** [0:19:13] Wie. In Bezug auf die Spielentwicklung, oder?

267

268 **I:** [0:19:20] Ja genau.

269

270 **B:** [0:19:20] Also, wenn die Musterschüler in Bezug auf Spielentwicklung, was  
271 die dann tatsächlich für das Arbeitsleben lernen?

272

273 **I:** [0:19:25] Ja genau. Denken Sie da wirklich an eine konkrete Musterschülerin  
274 oder einen konkreten Musterschüler. Und was hat der- oder diejenige eben  
275 mitgenommen?

276

277 **B:** [0:19:36] Ja, also ich habe da zwei im Kopf und die haben eben ein super  
278 Spiel gemacht. Und was die auf jeden Fall wirklich gelernt haben ist, sie haben  
279 sich selbstständig mit der Thematik beschäftigt und haben, ähm sie wollten zum  
280 Beispiel ein 2D-Spiel machen, ein Puzzle-Game, so top-down-Perspektive,  
281 kann man sich das vorstellen wie diese alten Pokemon Spiele, wo man eine  
282 Figur hat und die kann man rechts, links, oben steuern und sie wollten die  
283 Person, also die Figur auf einem Grid bewegt. Das heißt, ich drücke einmal kurz  
284 nach rechts, dann geht sie einen Schritt nach rechts und sie bewegt sich auf so  
285 einem Raster, das heißt, die Position muss immer in einem von diesen Raster,  
286 wie auf einem Schachbrett sein quasi. Das ist etwas was in Unity von Haus aus  
287 nicht drinnen ist. Die Position sind Fließkommazahlen, die ja. Und sie wollten  
288 das unbedingt so machen und haben sich das in den Kopf gesetzt und haben  
289 das dann auch geschafft umzusetzen. Und es war ein schwieriger Weg für sie.  
290 Was sie dabei auf jeden Fall gelernt haben ist, sie haben sich durch Foren  
291 gekämpft, durch Tutorials gekämpft, haben verschieden Ansätze versucht,  
292 haben von Stack-Overflow irgendwelche Lösungen für Probleme "copy-paste"  
293 in das Programm eingefügt, hat dann nicht auf Anhieb funktioniert, dann haben  
294 sie versucht das zu adaptieren usw. Also die typische Arbeit, die Programmierer  
295 und Programmiererinnen beim Lernen von Programmiersprachen oder wie  
296 auch immer, eigentlich auch machen. Und das haben sie da wirklich gemacht  
297 und dann, sind damit zu einem Ziel gekommen. Und ich glaube das ist eine  
298 wahnsinnig wertvolle Kompetenz, wenn ich irgendwann tatsächlich als  
299 Programmierer arbeiten mag, weil ich eben, auch wenn ich Informatik studiere,  
300 ich habe vielleicht Java gelernt, ich habe vielleicht C gelernt und dann fange ich  
301 bei einem Job an und auf einmal haben die ihr ganzes System in Rust und  
302 dann ok, jetzt muss ich Rust lernen. Ok, aber ich weiß, ich kenne die

303 Programmierkonzepte, ich weiß wie eine Schleife funktioniert, ich weiß wie  
304 Funktionen funktionieren, ich weiß vielleicht wie Generalität funktioniert in Java.  
305 Wie schaut das in Rust aus, das kann ich mir dann aneignen. Und diese  
306 Kompetenz haben sie auf jeden Fall gelernt. Sie haben auch zum Beispiel  
307 Zustände vom Spielercharakter dann mit einer Enumeration modelliert, was wir  
308 auch nie im Unterricht behandelt haben. Das war auch was, wo sie sich zuerst  
309 irgendwie überlegt haben, ok sie codieren das mit Zahlen, ob die Figur steht, ob  
310 sie geht, oder ob sie irgendwas anderes gerade macht und dann sind sie  
311 irgendwo im Internet anscheinend auf die Enumeration gestoßen, also diesen  
312 Datentyp und haben dann erkannt, ok, das ist eigentlich viel, viel besser und  
313 viel cooler und das verwenden wir jetzt um den Zustand von unserem Charakter  
314 zu beschreiben. Da war auch ganz interessant, wie sie dann, ich habe dann  
315 noch nach dem Projekt eben ein Interview auch geführt mit ihnen. Wie sie dann  
316 in ihren eigenen Worten erklärt haben, was eine Enumeration ist und wie die  
317 funktioniert und wie sie das mit vorhandenem Wissen, das wir im Unterricht  
318 behandelt haben, dann verknüpft haben und dann haben sie eben irgendwie so  
319 beschrieben von wegen, der eine hat gemeint die Enumeration ist wie ein Array  
320 mit verschiedenen Daten, aber alle vom gleichen Typ irgendwie, also da habe  
321 ich nicht genau gewusst worauf er jetzt genau hinaus will. Und dann hat der  
322 andere, ist der andere Schüler dahergekommen und hat eingehakt, ja er findet,  
323 es ist wie ein „Boolean“ mit mehr als einem Datentyp, also mehr als true/false,  
324 also quasi du hast eine Variable die festgesetzte Wertmöglichkeiten hat, so wie  
325 ein Boolean mit true und false, nur mit mehr als zwei. Und das finde ich, ist  
326 eigentlich eine sehr, sehr treffende Beschreibung von einer Enumeration, die  
327 ich aber so noch nie gehört habe. Ja, also was die beiden auf jeden Fall  
328 mitgenommen haben ist, sich selbst diese Dinge beizubringen und dazu kommt  
329 natürlich, sie haben gelernt mit Unity umzugehen, sie haben C#-spezifische  
330 Sachen gelernt.

331

332 I: [0:23:54] Und was zeichnet da die Spielentwicklung mit Unity besonders aus?  
333 Wo ist der größte Nutzen?

334

335 **B:** [0:24:04] Warum Unity finde ich eine gute Wahl ist, ist Unity ist einerseits  
336 eine professionelle Spielentwicklungsumgebung, also es werden viele  
337 professionelle Spiele damit gemacht, ich bin mir jetzt nicht ganz sicher, aber ich  
338 glaube "Ori and the blind forest", ein bekanntes Spiel von einer österreichischen  
339 Firma. Jetzt ist gerade der zweite Teil herausgekommen. Da war zumindest der  
340 erste Teil glaube ich mit Unity erstellt und ich glaube der zweite Teil auch. Das  
341 heißt, es ist einerseits, hat es schon den Appeal von wegen, das ist keine  
342 Lernumgebung, sondern es ist wirklich ein professionelles Tool und ich lerne da  
343 nicht jetzt irgendwie mit einem Programm umzugehen, das ich danach eh nie  
344 mehr brauche, angenommen, ich will wirklich Spiele programmieren und das  
345 professionell machen, dann habe ich da schon etwas gelernt, was ich dann  
346 auch verwenden kann. Das ist einmal der erste Punkt und der zweite Punkt ist,  
347 Unity ist gratis, ich kann es gratis verwenden, solange ich weniger als \$100.000  
348 im Jahr verdiene, oder irgendwie so. Und als dritten Punkt und das ist gerade in  
349 der Didaktik finde ich auch ein ganz wichtiger Punkt, Unity hat extrem viele  
350 Tutorials. Es gibt extrem viele Youtuber, die zum Beispiel Inhalte zu Unity  
351 machen und Unity selbst stellt auch wahnsinnig viele Tutorials zur Verfügung.  
352 Das ist zum Beispiel bei anderen Engines, wie der Unreal Engine weniger der  
353 Fall. Es gibt auch viele Tutorials dazu, aber bei Unity ist es einfach wirklich,  
354 wirklich viel.

355

356 **I:** [0:25:42] Ok. Und wo ist der größte Nutzen auch in Bezug auf was sie  
357 mitnehmen, was Kompetenzen sind, in die Richtung, oder auch Wissen und  
358 Fertigkeiten?

359

360 **B:** [0:25:54] Ein Aspekt ist auch der das, sie lernen mit komplexen  
361 Entwicklungsumgebungen umzugehen. Unity, wenn ich es zum ersten Mal  
362 öffne hat viele verschiedene Fenster und Komponenten und das ist ungefähr  
363 so, wie wenn ich von Paint auf Photoshop umsteige, da bin ich am Anfang auch  
364 überfordert, aber ich lerne damit umzugehen. Und das ist ein Skill, der einem

365 auch immer wieder unterkommt, wenn ich irgendetwas mit dem Computer  
366 mache, ganz egal was. Ich muss mich immer wieder in neue Systeme  
367 einarbeiten und das kommt bei Unity, glaube ich, sehr gut rüber, einerseits.  
368 Andererseits ist Unity, noch immer finde ich eine, also es ist eine komplexe  
369 Entwicklungsumgebung, verglichen mit Scratch zum Beispiel, oder  
370 GameMaker, aber es ist noch immer finde ich relativ einfach eigentlich und  
371 leicht zu erlernen. Und was sie dann konkret wirklich lernen dabei, das hängt  
372 sehr vom Setting ab, vom Unterricht und auch vom Fokus der einzelnen  
373 Schülerinnen und Schüler. Also, ich kann in Unity Spiele machen, ohne wirklich  
374 zu programmieren, ohne C#, also ohne wirklich Code zu schreiben und mir  
375 vielleicht auch einfach nur Code mit Copy-Paste von irgendwo anders herholen.  
376 Aber natürlich kann ich auch viel programmieren und ob ich dann das  
377 Programmieren lerne, hängt davon ab, ob ich es auch mache im Unterricht.

378

379 **I:** [0:27:26] Und hat die Spielentwicklung mit Unity einen Einfluss auf das  
380 Verständnis der SchülerInnen von einem Algorithmus?

381

382 **B:** [0:27:35] Ja. Also man kann bestimmte Algorithmen sehr gut einfach  
383 anschaulich darstellen und man hat sofort das Feedback. Also wenn ich jetzt  
384 ein Spiel programmiere und ich ändere was im Algorithmus für, also ein "Jump  
385 'n' Run" zum Beispiel für den Sprung, dann sehe ok, ich sehe sofort, was macht  
386 das mit dem Programm, was macht das mit dem Spiel und dem Spielerlebnis  
387 wenn ich da jetzt irgendwas verändere, oder was anders programmiere. Oder  
388 ich mache irgendein Programm, also schreibe ein Script für die Bewegung von  
389 irgendwelchen Hindernissen oder so im Spiel und dann funktioniert es nicht so  
390 wie ich will, dann muss ich analysieren, ok, warum funktioniert es nicht so wie  
391 ich es will. Und dann schaue ich mir den Algorithmus noch einmal an und  
392 schaue mir genau durch was es macht und habe sofort irgendwie den Connect  
393 zwischen dem Algorithmus wie er im Code steht und was dann tatsächlich im  
394 Spiel passiert. Und das ist bei Spielen auch sehr einfach möglich. Eben um  
395 wieder auf das Beispiel von so Konsolenprogrammen zurückzukommen, wenn

396 ich nur einen Input habe und einen Output und der Algorithmus einfach im  
397 Hintergrund sein Ding macht und dann etwas ausspuckt, dann habe ich diesen  
398 Effekt nicht, oder weniger. Und Code statisch zu verstehen ist wahnsinnig  
399 schwierig. Anhand des Codes, der im Script steht, genau zu erfassen was ein  
400 Algorithmus tatsächlich macht, ist wahnsinnig schwer und es geht viel  
401 einfacher, wenn ich den Algorithmus durchspiele, oder eben einfach anhand  
402 eines Spieles wirklich die Auswirkung vom Algorithmus sehe.

403

404 **I:** [0:29:31] Und wird da dann innerhalb des Unterrichtes auch bewusst darauf  
405 eingegangen oder so, ok ihr arbeitet jetzt mit Algorithmen, oder so, oder mit  
406 Handlungsanleitungen, so in die Richtung?

407

408 **B:** [0:29:45] Ja schon, schon. Es ist ja so, wenn ich jetzt ein Spiel habe und ich  
409 will, dass die eine Spielfigur, ich will, dass die etwas macht, dann muss man  
410 überlegen, ok, was genau soll sie machen. Und dann muss ich mir überlegen,  
411 ok wie soll sie das, also wie bringe ich das Programm dazu, wie bringe ich das  
412 Spiel dazu, dass das passiert. Und da muss man sich zwangsweise dann mit  
413 dem Algorithmus beschäftigen. Natürlich, dass man das Ganze Algorithmus  
414 nennt, muss natürlich im Unterricht von der Lehrperson erfolgen, aber ja, das  
415 passiert eigentlich ganz von selbst, dass man sich mit dem algorithmischen  
416 Denken dann auseinandersetzt.

417

418 **I:** [0:30:35] Können die SchülerInnen nach der Spielentwicklung mit Unity  
419 wesentliche Aspekte der Programmierung, also, wie zum Beispiel Funktionen,  
420 oder Objektorientierung nennen und erläutern?

421

422 **B:** [0:30:50] Teilweise ja, teilweise nein. Also nicht zwangsweise sage ich  
423 einmal. Das hängt mit dem, einerseits dann an dem unterschiedlichen Fokus  
424 von Schülerinnen und Schülern, oder eben dem Unterricht selbst. Wobei eben,  
425 in meiner Arbeit mir auf jeden Fall als junge Lehrkraft sehr bewusst worden ist,  
426 dass einfach der Rahmen, den ich im Unterricht setze und vorgebe, da eine

427 wesentliche Auswirkung hat natürlich. Und dadurch, dass ich in der  
428 Spieleentwicklung ebenso viele verschiedene Sachen machen kann, sind die  
429 Outcomes auch sehr unterschiedlich. Und es gibt, also ich habe im Rahmen  
430 meiner Diplomarbeit fünf Projekte analysiert und ist ein zentraler Aspekt, der  
431 eben herausgekommen ist, ist das es wahnsinnig unterschiedlich ist, was die  
432 Leute gelernt haben und teilweise, zum Beispiel in einem Projekt ist Vererbung  
433 vorgekommen, in allen anderen Projekten nicht. Das heißt eben, das Konzept  
434 der Vererbung in der objektorientierten Programmierung kann vorkommen,  
435 muss nicht. Was war noch so ein Beispiel. Ja ich habe es jetzt nicht mehr  
436 genau im Kopf, aber es sind verschiedene Aspekte der Programmierung sind  
437 teilweise eben wirklich nur in ein zwei Projekten vorgekommen, in anderen gar  
438 nicht. Dann hat es wieder andere Aspekte gegeben, die sehr viel  
439 vorgekommen. Was zum Beispiel sehr interessant war, Schleifen sind fast nie  
440 vorgekommen. Was meines Erachtens durchaus daran liegt, das Schleifen jetzt  
441 nicht, also bei einfachen Algorithmen in der Spieleprogrammierung nicht  
442 wirklich eine, natürlich nicht oft vorkommen und zwar deshalb, weil die  
443 Spielschleife ja an sich schon im Hintergrund läuft und sehr viel über diese  
444 Updatemethode in Unity, die einfach periodisch immer aufgerufen wird, jeden  
445 Frame abgehandelt wird. Das heißt wenn ich etwas öfter machen will, dann  
446 schreibe ich es einfach in die Update-Methode und dann wird es eh schon  
447 sechzig Mal pro Sekunde ausgeführt, sozusagen. Und dementsprechend muss  
448 man für einfache Aufgaben oft, braucht man Schleifen einfach nicht. Und ich  
449 glaube es war eben auch wirklich so, dass von diesen fünf Projekten haben nur  
450 zwei oder ein Projekt aktiv eine Schleife in ihren Programmen verwendet.  
451 Wohingegen Verzweigungen mit „if-else“ zum Beispiel, das ist überall  
452 vorgekommen und Funktionen auch. Das liegt auch daran, wie Unity strukturiert  
453 ist und wie Unity arbeitet. Man muss generell wissen, wo man seinen Code  
454 hineinschreibt, also in welche Funktionen, welche Funktionen von der Game-  
455 Engine aufgerufen werden, also zum Beispiel im Update oder Start oder  
456 FixedUpdate und wo dann der entsprechende Code hineinmuss. Und dass sie  
457 dann auch ihren Code selbst faktorisiert haben und eigene Funktionen

458 geschrieben haben, das ist auch vorgekommen, also ziemlich durchgehend  
459 glaube ich.

460

461 **I:** [0:34:23] Und jetzt nochmal konkret quasi auf die Frage, also denken Sie,  
462 dass die SchülerInnen eben solche Aspekte dann nennen und erläutern  
463 können?

464

465 **B:** [0:34:34] Ja, können sie. Also, wenn sie sich wirklich damit  
466 auseinandergesetzt haben und den Code nicht einfach irgendwo her kopiert  
467 haben, dann können sie schon erklären wie das funktioniert, grob zumindest.  
468 Vielleicht nicht unbedingt einhundertprozentig korrekt, sondern, wenn sie es  
469 jetzt wirklich nur angewendet haben, dann in dem Horizont in dem sich ihre  
470 Anwendung befunden hat und mit dem Eindruck den sie davon haben und das  
471 Konzept, das sie so erarbeitet haben. Dabei kann es natürlich auch zu  
472 Fehlkonzepten kommen. Es ist nicht so, dass man einfach sagt, ok,  
473 programmiert ein Spiel und ihnen vielleicht erklärt oder mit ihnen gemeinsam  
474 ein Spiel programmiert und dann können sie die Sachen, sondern ich glaube  
475 man soll trotzdem den Fokus auf die einzelnen Konzepte legen. Kann man  
476 durchaus im Rahmen der Spieleprogrammierung machen, aber man muss  
477 diesen Fokus setzen. Also es passiert nicht einfach so von selbst, vor allem  
478 wenn ich will, dass wirklich alle diese Konzepte kennen lernen, weil eben nicht  
479 alles einfach so vorkommt in einem einfachen Spiel.

480

481 **I:** [0:35:43] Lernen die SchülerInnen im Verlauf der Spieleentwicklung mit Unity  
482 unterschiedliche Programmiersprachen und Produktionsabläufe kennen?

483

484 **B:** [0:35:53] Nein. Also Programmiersprachen, ich meine man kann in Unity  
485 theoretisch mit JavaScript programmieren, es gibt da sogar noch eine dritte  
486 Sprache glaube ich, aber generell würde ich, also JavaScript mag ich jetzt  
487 persönlich einfach nicht. Kann man sicher auch machen, es gibt sicher auch  
488 Gründe, die dafür sprechen, aber erstens kenne ich mich damit nicht so gut aus



489 und zweitens finde ich ist es didaktisch weniger sinnvoll als Sprachen wie C#.  
490 Dementsprechend haben bei uns alle mit C# programmiert, weil wir es alle ...  
491 [unverständlich]. Also als Sprache ist bei Unity eigentlich C# gängig. Auch alle  
492 Tutorials die man eigentlich findet, verwenden im Grunde C#, es gibt glaube ich  
493 sehr wenige die JavaScript verwenden. Was war der andere Aspekt noch  
494 einmal?

495

496 **I:** [0:36:53] Produktionsabläufe.

497

498 **B:** [0:36:55] Produktionsabläufe. Ja, sie lernen verschiedene Sachen, wenn  
499 man sie ihnen beibringt und, diese Produktionsabläufe. Wenn man es ihnen  
500 nicht beibringt, dann kann es sein, dass sie es lernen, aber es kann auch sein,  
501 dass sie es nicht lernen. Es gibt so wahnsinnig viele verschiedene Sachen die  
502 man machen kann mit Unity und je nachdem was man machen mag/kann,  
503 braucht man verschiedene Dinge und dadurch das die Zeit an der Schule sehr  
504 begrenzt ist, kommen in verschiedenen Arten von Spielen oder verschiedenen  
505 Arten von Projekten auch nur bestimmte Produktionsabläufe zum Tragen. Zum  
506 Beispiel in dem einen Spiel, das die beiden Schüler, von denen ich vorher als  
507 Musterschüler geredet habe, die haben zum Beispiel keine Animationen in  
508 ihrem Spiel gehabt. Sie haben geplant gehabt, dass sie das irgendwann später  
509 einbauen, aber sie wollten sich im Moment nicht damit auseinandersetzen und  
510 haben genug damit zu tun gehabt und auch in ihrer Freizeit programmiert, damit  
511 dass sie sich um die anderen Konzepte kümmern und haben dementsprechend  
512 den Animator in Unity überhaupt nicht verwendet und sich damit auch nicht  
513 wirklich befasst. Dementsprechend weiß ich auch nicht, ob sie da irgendwelche  
514 Kompetenzen haben. Ich bin dann darauf nicht eingegangen. Oder eben, wenn  
515 ich fertige Grafiken verwende, dann muss ich mir die Grafik nicht selber  
516 zeichnen klarerweise. Anderer Aspekt, je nachdem welche Arten von Grafiken  
517 ich verwende und ob ich vielleicht ein 2D- oder 3D-Spiel mache, gibt dann  
518 schon vor, womit ich mich auseinandersetze und wo ich Kompetenzen erlernen  
519 werde. Wenn ich ein 2D-Spiel mache, dann werde ich keine 3D-Modelle

520 brauchen und mich auch mit der dreidimensionalen Physik-Engine von Unity  
521 nicht auseinandersetzen. Also zusammenfassend, die Produktionsabläufe, die  
522 gelernt werden, die variieren stark. Beziehungsweise wenn man nicht will das  
523 sie variieren, dann muss man als Lehrkraft den Fokus entsprechend setzen und  
524 den Rahmen entsprechend setzen.

525

526 **I:** [0:39:14] Erwerben die SchülerInnen die Fertigkeit, Aufgaben mit Mitteln der  
527 Informatik zu modellieren? Also Datenstrukturen, Struktogramme, ER-Modelle  
528 jetzt als Beispiele genannt.

529

530 **B:** [0:39:30] Ja, ja, generell ja. Ist auf jeden Fall auch passiert in den Projekten.  
531 Wieder ist es nicht durchgängig passiert, aber die Schülerinnen und Schüler  
532 haben sich Gedanken darüber gemacht wie sie bestimmte Dinge modellieren  
533 können, damit sie eben. Wenn ich als Ziel habe, ich will ein Spiel haben wo  
534 man von Plattform zu Plattform springen kann und irgendwie ein Affe mit  
535 Bananen auf mich wirft, dann muss ich das auf die entsprechenden  
536 Kernelemente reduzieren und dann modellieren und das wurde teilweise  
537 gemacht.

538

539 **I:** [0:40:19] Und sehen Sie das als Voraussetzung für die Spieleentwicklung mit  
540 Unity, oder erlernt man es damit?

541

542 **B:** [0:40:27] Ich sehe es nicht als Voraussetzung. Ich glaube, dass es durchaus  
543 damit erlernt werden kann. Und ich glaube solche Dinge lernt man auch am  
544 besten, wenn man sie anwendet und wenn man eine konkrete Anwendung  
545 dafür hat. Und im Idealfall ist es eine Anwendung wo ich enthusiastisch bin und  
546 wo ich motiviert bin, dass auch tatsächlich zu machen und das ist eben bei der  
547 Spielentwicklung sehr oft gegeben.

548

549 **I:** [0:40:56] Ok. Können die SchülerInnen nach der Spielentwicklung mit Unity  
550 Daten-, Klassen- oder Objektstrukturen in einem Algorithmus verwenden?

551

552 **B:** [0:41:04] Ja, also grundsätzlich ja. Das ist natürlich immer unterschiedlich  
553 und das hat man auch generell. Wenn ich jetzt sage, im Mathematikunterricht  
554 können sie, nachdem dieses Thema durchgenommen ist, das auch tatsächlich  
555 machen, dann gibt es immer Leute, die das weniger gut können und manche  
556 die das dann sehr gut können. Und dementsprechend würde ich dann aber  
557 auch sagen, ja in der Spielentwicklung mit Unity, sie verwenden Variablen, sie  
558 verwenden Funktionen, sie verwenden Klassen, zwar in diesem Unity-Setting  
559 und ich glaube da muss man dann auch eben aufpassen, dass man das  
560 entsprechend im Unterricht dann auch behandelt. Aber vor allem wenn man  
561 tiefer in die Materie gehen mag, muss ich mich zwangsweise auch wirklich mit  
562 Unity beschäftigen und vielleicht sogar in die Dokumentation schauen, oder mir  
563 Tutorials anschauen, wie die ein Problem lösen und welche Klassen die  
564 verwenden und dann, damit ich dann mein Problem lösen kann. Und zum  
565 Beispiel in einem Team haben sie zwei verschiedene Spielfiguren gehabt, zwei  
566 Fischer, Nemo und Dori und sie haben dann für beide eine Oberklasse, eine  
567 Parent-Klasse gemacht, wo, damit eben sie das Verhalten der Fischer, das  
568 beide gemeinsam haben, sie den Code nur einmal schreiben müssen. Und sie  
569 haben das im Interview dann auch beschreiben können, wieso sie sich dazu  
570 entschieden haben das so zu machen und warum das so sein soll und warum  
571 das Sinn macht. Also diese Lerneffekte sind auf jeden Fall da.

572

573 **I:** [0:42:58] Sind die SchülerInnen nach der Spielentwicklung mit Unity fähig ein  
574 Softwareprojekt zu planen und durchzuführen, oder vielleicht sogar zu  
575 reflektieren?

576

577 **B:** [0:43:10] Das hängt ganz davon ab wieviel Zeit man im Unterricht dafür  
578 aufwendet und wieviel Motivation die Schülerinnen und Schüler mitbringen.  
579 Also ich glaube nicht, dass man davon sprechen kann, wenn ich jetzt ein Monat  
580 lang in der Schule, jede Woche eine Doppelstunde lang Unity mache, dass sie  
581 danach im Stande sind selbstständig ein Spiel zu entwickeln, es sei denn, sie

582 sind wirklich hochmotiviert und sind bereit, sich durch Tutorials zu arbeiten und  
583 selbstständig noch weiter zu lernen. Also es ist ein wahnsinnig großes Feld und  
584 komplexes Gebiet, dass ja also für wirkliche, also dass die Kompetenzen  
585 wirklich so vermittelt werden, dass man sagen kann, die Schülerinnen und  
586 Schüler können, nachdem dieses Thema durchgenommen worden ist sich zu  
587 Hause hinsetzen und einen Prototypen von einem Spiel selbstständig  
588 programmieren, würde ich sagen, grundsätzlich im Rahmen des  
589 Schulunterrichts wahrscheinlich nicht, weil, es sei denn sie bringen sehr, sehr  
590 viel Eigenmotivation mit, dann natürlich. Aber ich kann, wenn ich diesen Effekt  
591 haben will, dann müsste ich wahrscheinlich zwei Jahre lang nur  
592 Spieleentwicklung unterrichten.

593

594 **I:** [0:44:32] Ok. Und gehen wir jetzt Mal ein bisschen den Schritt weg von der  
595 Spielentwicklung, vielleicht allgemein in Richtung Softwareprojekte, auch  
596 vielleicht kleinere Softwareprojekte. Weil, klar, ein Spiel ist oft was doch sehr  
597 Komplexes im Gegensatz zu vielleicht Anforderungen für eine  
598 Rechtsanwaltskanzlei an ein Programm oder so, oder ein kleineres Programm.  
599 Selbstständig?

600

601 **B:** [0:45:00] Ja, können sie schon. Wobei eben da auch zu sagen ist, wenn ich  
602 jetzt Spielentwicklung mit Unity gemacht habe, dann kenne ich mich mit C# aus  
603 und weiß wie C#-Programme in Unity mit der Unity-Game-Engine interagieren.  
604 Wie ich jetzt ein eigenständiges C#-Programm schreibe ist vielleicht wieder ein  
605 anderes Thema, sollte auch möglich sein, aber wenn ich dann in die, in dem  
606 Beispiel jetzt die Rechtsanwaltskanzlei komme und ich soll dort ein PHP-Script  
607 schreiben, das auf der Webseite, also am Webserver läuft, muss ich natürlich  
608 dann wissen wie PHP funktioniert. Das kann ich damit automatisch nach dem  
609 Unity-Grundkurs im Unterricht nicht. Aber gleichzeitig eben, im Idealfall wurden  
610 dabei Kompetenzen erlangt, die einfach Programmlogik und die nicht spezifisch  
611 auf C# sind, sondern ich weiß was Funktionen sind, ich weiß wie Operatoren  
612 funktionieren, Variablen und kann dieses Wissen dann anwenden in einer

613 anderen Programmiersprache und dort dann ein kleines Programm schreiben  
614 oder vielleicht ein Programm abändern oder einen Bug finden, oder sonst  
615 irgendwas. Das ist grundsätzlich schon möglich ja.

616

617 **I:** [0:46:16] Ok, und darüber zu reflektieren?

618

619 **B:** [0:46:18] Darüber zu reflektieren, ja, ich denke auch. Also zum Beispiel  
620 gerade bei Unity auch an den Schul-PCs, die eben nicht die allerschnellsten  
621 sind, merkt man vielleicht, ok wenn ich den Algorithmus irgendwie auf eine  
622 bestimmte Art mache, oder bestimmte Sachen programmiere, dann ist das zum  
623 Beispiel sehr wenig performant und das Spiel laggt dann und dann setze ich  
624 mich auch zwangsweise damit auseinander, was der Code eigentlich macht  
625 und wie lange der vielleicht braucht um ein bestimmtes Problem zu lösen. Und  
626 man kann auch darüber reflektieren. Aber das ist auch glaube ich wieder etwas,  
627 das sehr stark von der Eigenmotivation abhängt, von den Schülerinnen und  
628 Schülern.

629

630 **I:** [0:47:06] Lernen die SchülerInnen die Angemessenheit von  
631 Entwicklungswerkzeugen grob einzuschätzen?

632

633 **B:** [0:47:16] Puhh, das kann ich so nicht beantworten, glaube ich. Also ja, das  
634 wüsste ich jetzt nicht.

635

636 **I:** [0:47:28] Ok, also im Bezug zum Beispiel, eben auch teilweise vielleicht  
637 leistungsbezogen oder so, weil Sie das vorher erwähnt haben.

638

639 **B:** [0:47:39] Nein ich, also teilweise ja, können sie das sicher machen nachher  
640 und wissen vielleicht ok, ich habe dieses Problem, dafür verwende ich jetzt  
641 nicht Unity, sondern vielleicht was anderes, was ich einmal gelernt habe. Ich  
642 glaube aber durchaus auch, dass der Effekt eintreten könnte, dass, ich kenne  
643 mich jetzt mit Unity aus und will ein Programm schreiben das vielleicht kein

644 Spiel ist, sondern irgendwas anderes und dadurch dass ich mich mit Unity  
645 auskenne, versuche ich das mit Unity zu lösen, obwohl vielleicht ein anderes  
646 Werkzeug besser gewesen wäre. Aber das ist nur eine Vermutung von mir. Das  
647 kann ich jetzt so nicht aus eigener Erfahrung bestätigen.

648

649 **I:** [0:48:28] Entdecken die SchülerInnen Gemeinsamkeiten und Regeln, also  
650 Muster in Handlungsanleitungen oder Algorithmen?

651

652 **B:** [0:48:37] Ja, definitiv.

653

654 **I:** [0:48:40] Sieht man das wo? Finden Sie da irgendein Beispiel?

655

656 **B:** [0:48:44] Als Beispiel eben die Projekte die wir gemacht haben, also es war  
657 grundsätzlich so, sie haben zu Beginn haben sie einmal C#-Grundlagen extra  
658 gelernt, mit Konsolenprogrammen größtenteils, danach haben wir die  
659 Grundlagen von Unity gemacht anhand eines Tutorials und dann haben sie ihr  
660 eigenes Spiel erstellen müssen. Und da haben sie zum Beispiel, sind einige  
661 Leute hergegangen und haben eben quasi überlegt ok, ich mag dieses Spiel so  
662 machen und ich weiß wie es in dem Tutorialspiel funktioniert hat und das  
663 versuche ich jetzt so nachzubauen, dass es dann für mein Spiel auch passt und  
664 vielleicht ein bisschen abzuändern und so diese Parallelen zu ziehen. Also es  
665 wurde dann vor allem auf schon vorhandenes Wissen zurückgegriffen und  
666 dieses dann aber modifiziert und ja.

667

668 **I:** [0:49:41] Und können die SchülerInnen durch die Spielentwicklung mit Unity  
669 die Bedeutung von Algorithmen erkennen?

670

671 **B:** [0:49:49] Ja.

672

673 **I:** [0:49:50] Ja?

674

675 **B:** [0:49:51] Also eben wie schon vorher angesprochen die, sie erkennen, ich  
676 schreibe meinen Code, ich gebe die Anweisungen Schritt für Schritt und ich  
677 sehe dann sofort im Spiel, was passiert damit und wie verändert sich der  
678 Spielverlauf, wie verändert sich das Verhalten der Spielfigur, wenn ich eben in  
679 dem Code was ändere und wie sich der Algorithmus direkt auf den Code  
680 auswirkt, also wie sich der Code und der Algorithmus der dahinter steht direkt  
681 auf das Spielgeschehen auswirkt, so.

682

683 **I:** [0:50:22] Und können sie dann auch die Effizienz von einem Algorithmus  
684 bewerten?

685

686 **B:** [0:50:28] Nein, nicht zwangsweise. Es kann sein eben wenn sie an die  
687 Grenzen der Engine oder der Hardware stoßen, dass das Spiel dann ins  
688 Stocken kommt, dass sie dann zwangsweise erkennen ok, da läuft jetzt  
689 irgendetwas nicht rund und so erkennen sie dann vielleicht ok, das Problem  
690 liegt im Code und dann kann man das auch behandeln, aber grundsätzlich, es  
691 ist jetzt so glaube ich nicht vorgekommen, dass das tatsächlich passiert ist. Und  
692 die Effizienz von Algorithmen ist da eher im Hintergrund auf jeden Fall. Ich  
693 glaube auch, das ist etwas, wenn man es nicht irgendwie im Unterricht extra  
694 behandelt, dann ist das auch nicht klar. Dann ist da eher das Konzept da, die  
695 Hardware ist zu langsam und die Computer sind schlecht, oder sonst  
696 irgendetwas, als dass man sich vielleicht überlegt, ok ich könnte den Algorithmus  
697 eigentlich abändern, so dass er effizienter läuft und weniger Ressourcen  
698 braucht.

699

700 **I:** [0:51:44] Jetzt kommen wir zum zweiten Fragecluster sozusagen, der ist jetzt  
701 ein bisschen kürzer. Bei SchülerInnen treten bei einem neuen Lehrstoff immer  
702 wieder Schwierigkeiten auf. Erinnern Sie sich an solche Situationen. Was waren  
703 die Schwierigkeiten da konkret?

704

705 **B:** [0:52:04] Schwierigkeiten. Grundsätzlich programmieren mit C# ist immer  
706 wieder eine Schwierigkeit, die ich beobachte, dass sie zum Beispiel Probleme  
707 haben mit der Klammern Setzung und generell mit der Syntax von C#, die sehr  
708 an C angelehnt ist und da recht pedantisch ist. Also wenn ein Strichpunkt fehlt,  
709 dann gibt es einen Compilererror und die Klammern sind wichtig. Anders als  
710 eben bei JavaScript, wo, ich kann Strichpunkte machen, ich kann keine  
711 machen, ich kann Variablen deklarieren und ich kann auch einfach nur einen  
712 Variablennamen hinschreiben und das Programm macht trotzdem irgendetwas.  
713 Also, das ist natürlich ein Problem und kann sehr, sehr demotivierend wirken,  
714 dass ich, ich schreibe Code, ich habe eine Idee wir das machen und dann  
715 schreibe ich den Code und dann kommt auf einmal eine Welle von  
716 Fehlermeldungen nur weil irgendwo ein Strichpunkt gefehlt hat. Das ist immer  
717 wieder einmal was, also Probleme, die ich beobachte. Andere Probleme. Ja  
718 generell noch ein Problem beim Programmieren ist oft, dass man nicht weiß wo  
719 man anfangen muss. Das heißt ich sitze da und denke mir ok, ich habe keine  
720 Ahnung was ich tun soll, wo fange ich an, wo muss ich meinen Code überhaupt  
721 hinschreiben und wo im Programm finde ich überhaupt meine Scripte und wie  
722 funktioniert das ganze System. Das heißt, ich muss mich da einmal einarbeiten  
723 und das ist auf jeden Fall eine Hürde. Und dass man eben diese Abstraktion  
724 schafft, dass ich vom Problem zu einem Algorithmus und dann zur  
725 Implementierung komme. Das ich mir überlege, ok ich will dass da ein Affe im  
726 Baum sitzt, der Bananen wirft, aber was heißt das eigentlich. Wie kann man so  
727 etwas programmieren. Wie erstelle ich den Affen, wie erstelle ich die Bananen  
728 und wie mache ich dann das der in bestimmten Abständen in die Richtung vom  
729 Spieler diese Bananen wirft. Das ist keine einfache Aufgabe und man muss  
730 eben da diese Abstraktion schaffen und dieses sehr vage definierte Problem  
731 dann abstrahieren und konkret in eine Programmiersprache im Grunde  
732 übersetzen. Das ist auch schwierig. Dann, was mir auch aufgefallen ist mit der  
733 Arbeit mit Tutorials, dass es teilweise eben sprachliche Probleme gibt, also  
734 wenn ein Tutorial mit sehr viel Text da ist, dann wirkt das einfach sehr  
735 einschüchternd und man hat vielleicht nicht unbedingt Lust oder vielleicht auch



736 gar nicht die Kompetenz so lange Texte dann sinngemäß zu verstehen. Da war  
737 auch eine sehr interessante Erfahrung, wie da war ein Schüler da, der noch  
738 nicht so lange in Österreich war und seine Deutschkenntnisse waren nicht so  
739 gut und er hat dementsprechend Probleme gehabt mit dem Text-Tutorial, das  
740 wir ihnen zur Verfügung gestellt haben, das Deutsch war. Aber wir haben eben  
741 auch Video-Tutorials gemacht und er hat dann hauptsächlich mit diesen Video-  
742 Tutorials gearbeitet, die zwar auf Deutsch waren, aber wo man die Handlungen  
743 am PC auch beobachten hat können und dann gesehen hat, ok man muss dort  
744 hin klicken, dorthin über dieses Menü und dann erstellt man dort diese Datei  
745 usw. Das heißt, je nachdem eben, es gibt verschiedene Barrieren beim Lernen  
746 und diese sind individuell unterschiedlich zwischen SchülerInnen und Schülern  
747 und das kann man mit unterschiedlichen Lernmaterialien zum Beispiel lösen,  
748 dieses Problem. Sonst noch Schwierigkeiten, ähm. Eine Schwierigkeit, die auch  
749 immer wieder genannt wurde, war, wir haben für unsere Tutorials Unity-Version  
750 5.6 oder irgendwas verwendet, die mittlerweile sehr veraltet ist. Wenn dann  
751 Schülerinnen und Schüler hergegangen sind und sich zu Hause, zu Hause am  
752 Projekt weiterarbeiten wollten und sich zu Hause Unity installiert haben, haben  
753 sie natürlich die aktuelle Version installiert. Das aufwärtskompatible ist dann  
754 vielleicht noch gegangen. Das heißt, sie haben ihr Projekt zu Hause importiert,  
755 weitergearbeitet. Dann nehmen sie es mit in die Schule. Auf den Schulrechnern  
756 ist die alte Version und auf einmal geht gar nichts mehr und das war ein  
757 Problem, dass bei einigen Leuten aufgetaucht ist und dass es natürlich sehr,  
758 sehr demotivierend ist und dann auch wahnsinnig mühsam zu beheben ist. Und  
759 da muss man eben auch irgendwie von vorneherein klar machen, dass wenn  
760 sie zu Hause arbeiten, dass sie die richtige Version verwenden zum Beispiel  
761 oder eben schauen dass man die Version in der Schule aktuell hält bzw. dann  
762 die Tutorials aktuell hält, was dann insbesondere bei Video-Tutorials sehr  
763 aufwändig wird, wenn ich die von Version zu Version aktuell halten will.

764

765 I: [0:58:05] Konnten die SchülerInnen das dann teilweise selbstständig lösen  
766 oder schon immer mit Hilfe?

767

768 **B:** [0:58:14] Ich überlege gerade. Das eine Projektteam, sie haben dann ihren  
769 eigenen Laptop mitgebracht und haben dann auf dem gearbeitet, einerseits und  
770 andererseits haben sie dann teilweise Scripte geschrieben, die sie dann erst in  
771 das Programm eingefügt haben, ohne sie live zu testen, was sehr aufwändig  
772 und schwierig ist eigentlich. Das hat aber auch funktioniert dann. Und  
773 ansonsten ja also mit Hilfe haben wir die Probleme dann irgendwie lösen  
774 können und dann auch irgendwie Scripte importieren.

775

776 **I:** [0:59:05] Was müsste den SchülerInnen fehlen damit die Situation noch  
777 schlimmer wird, bezüglich der Schwierigkeiten?

778

779 **B:** [0:59:11] Noch schwieriger? Interessante Frage. Unstrukturiert arbeiten. Sich  
780 nicht trauen Dinge zu probieren auf der einen Seite. Also ein Problem, welches  
781 ich immer wieder einmal sehe ist, dass einzelne Leute nicht wissen wie etwas  
782 funktioniert, wie kann ich ein Problem lösen und dann werde ich als Lehrperson  
783 gefragt und sie haben eigentlich schon eine halbe Idee im Kopf, wie es  
784 funktionieren könnte und ich muss dann immer sagen, ja probiere es doch  
785 einfach einmal aus und schaue ob das funktioniert und wenn es nicht  
786 funktioniert, überlege dir vielleicht wieso es nicht funktioniert und wo vielleicht  
787 ein Denkfehler liegt. Und da irgendwie haben sie teilweise wenig Vertrauen in  
788 sich bzw. auch vielleicht auch Bequemlichkeit, von wegen, ich habe einen  
789 Lehrer da, der mir die Frage beantworten kann, warum soll ich selber irgendwas  
790 überlegen, diesen Eindruck habe ich manchmal. Und ich werde sicher keine  
791 Probleme lösen, wenn ich nicht eigenständig versuche Dinge auszuprobieren.

792

793 **I:** [1:00:40] Und was würde man zur Bewältigung dieser Schwierigkeiten  
794 benötigen?

795

796 **B:** [1:00:44] Was sehr, sehr viel hilft, sind einfach gute, gut ausgearbeitete  
797 Unterrichtsmaterialien, mit denen die Schülerinnen und Schüler selbst arbeiten

798 können. Weil wenn ich jetzt eine Klasse habe von sechzehn Schülern und ich  
799 muss jedem die ganze Zeit Fragen beantworten oder Dinge erklären, dann  
800 komme ich nicht sehr weit wahrscheinlich und die Probleme treten meistens  
801 erst beim Arbeiten auf. Ich kann Konzepte, ich kann verschiedene  
802 Arbeitsschritte erklären und sie schauen zu und sie verstehen es vom  
803 Zuschauen, oder glauben es zu verstehen, aber wenn sie es dann selbst  
804 reproduzieren müssen oder eigenständig Dinge kreieren müssen anhand  
805 dieses Wissens, dann ist es natürlich schwierig und dann treten neue Probleme  
806 auf. Und wenn sie dann Anhaltspunkte in Form von Unterrichtsmaterialien,  
807 Tutorials, usw. haben, dann hilft das enorm. Und ansonsten von den  
808 SchülerInnen selbst, ja Eigenmotivation, wirklich den Versuch so, ich will das  
809 lösen, dieses Durchhaltevermögen und ich bekomme eine Fehlermeldung und  
810 dann google ich einfach Mal die Fehlermeldung, vielleicht finde ich was dazu.  
811 Das sind so Sachen, die helfen.

812

813 **I:** [1:02:24] Was hatten die SchülerInnen, die nicht auf diese Schwierigkeiten  
814 gestoßen sind, an sich?

815

816 **B:** [1:02:31] Puh. Also ich sage einmal, es gibt drei Arten von Schüler und  
817 Schülerinnen, wo ich als Lehrperson nicht mitbekomme, wenn sie  
818 Schwierigkeiten haben oder dass sie überhaupt Schwierigkeiten haben. Das  
819 sind einerseits die, die sich dann einfach ruhig verhalten und wenig arbeiten  
820 und unmotiviert sind und andererseits die, die selbstständig arbeiten und dann  
821 auch selbstständig recherchieren und eigenständig ihre Probleme lösen. Eben  
822 die beiden, diese beiden Musterschüler sozusagen, die haben das auf jeden  
823 Fall gehabt, dass sie wirklich, nein wir wollen das machen und mit ihrem Grid-  
824 basierten Bewegungsalgorithmus, haben mich gefragt, wie sie das machen  
825 könnten. Ich habe dazu gesagt, ja ich muss mir das auch überlegen. Dann  
826 haben sie (anonymisierte Person) gefragt, der hat sich selbst etwas überlegt,  
827 hat ihnen das auch präsentiert. Das war ihnen zu kompliziert und das haben sie  
828 nicht gleich verstanden, dann sind sie weitergegangen und haben gesagt, nein,

829 nein, wir machen das selbst, wir recherchieren selber. Und das Fazit am  
830 Schluss war irgendwie so, ja, es war erstaunlich wie kompliziert das Ganze war  
831 und sie haben Tutorials gelesen, sie haben Videos angeschaut, sie haben  
832 Foren durchstöbert, sie haben uns Lehrkräfte gefragt. Schlussendlich haben sie  
833 ihre eigene Lösung implementiert, basierend auf dem ganzen Wissen, das sie  
834 von überall her sich zusammengeholt haben. Also, wenn Probleme nicht  
835 auftreten, dann liegt es glaube ich eher daran, dass den Problemen  
836 ausgewichen wird. Relevanter ist mehr, wie ich mit den Problemen umgehe und  
837 wie ich sie dann lösen kann. Sonst fällt mir da jetzt gerade nichts dazu ein,  
838 glaube ich.

839

840 **I:** [1:04:56] Und angenommen morgen beginnt ein neues Schuljahr und die  
841 SchülerInnen sind bestens gerüstet, um direkt mit der Spielentwicklung mit  
842 Unity zu beginnen. Woran erkennen Sie dieses optimale Rüstzeug?

843

844 **B:** [1:05:16] Also woran ich erkenne, dass meine Schülerinnen und Schüler  
845 bereit sind, dass ich jetzt mit ihnen die Spieleentwicklung anfangen?

846

847 **I:** [1:05:23] Mhm.

848

849 **B:** [1:05:23] Ok. Also erstens einmal, Voraussetzung dafür ist, es muss  
850 Interesse da sein. Das ist meistens kein Problem, vor allem wenn sie noch  
851 keine Ahnung von der Spieleentwicklung haben. Andererseits wenn ich, also  
852 was sie mitbringen können/sollen ist, sie sollen einmal ein grundsätzliches, ein  
853 bisschen ein Verständnis von Code-schreiben haben und wie man Code  
854 strukturiert und wie das gemacht wird. Und die Art wie ich das jetzt im Unterricht  
855 mache, ist, ich fange relativ früh an mit Scratch, wo sie algorithmisches Denken  
856 lernen und auch schon Spiele machen, eigene. Andererseits mache ich schon  
857 im Realgymnasium in der Unterstufe schon html. Weniger aus dem Grund, dass  
858 sie tatsächlich lernen Webseiten zu schreiben, mehr aus dem Grund das sie  
859 lernen Code zu schreiben und Code zu strukturieren und einfach wirklich mit

860 textbasiertem Programmcode umzugehen und eben lernen, dass es relevant  
861 ist, ob da eine spitze Klammer zu viel ist, oder ob da eben ein Strichpunkt fehlt,  
862 oder sonst irgendetwas. Das sind alles Voraussetzungen, die dann, wenn ich  
863 das einmal habe, wenn ich eben gewohnt bin ok, wenn ich meinen Code  
864 schreibe, dann muss ich bestimmte Regeln beachten und wenn ich eine  
865 Funktion aufrufe, dann sind die Parameter zwischen runden Klammern und  
866 nach jedem Statement kommt ein Strichpunkt. Wenn ich diese ganzen Sachen  
867 intus habe und sie kein Problem mehr darstellen, dann kann ich mich sehr viel  
868 besser auf die anderen Sachen konzentrieren. Wenn diese ganzen  
869 Voraussetzungen nicht da sind, dann wird es schwierig, dann ist so viel auf  
870 einmal da, dass wahrscheinlich die Motivation sehr schnell flöten geht und dass  
871 dann auch Lerneffekte nicht unbedingt dort sind, wo ich sie vielleicht gerne  
872 gehabt hätte. Aber grundsätzlich, man kann die Spielentwicklung allgemein  
873 aber auch mit Unity auf so viele verschiedene Arten aufziehen, dass  
874 grundsätzlich meiner Meinung nach, könnte man jederzeit damit anfangen. Die  
875 Frage ist, wie man es aufziehen will und wieviel Zeit man bereit ist zu  
876 investieren, beziehungsweise was man für Vorbereitungen hat. Ein Spiel von  
877 Grund auf zu erstellen, dafür ist sehr viel nötig, meiner Meinung nach. Dafür ist  
878 vor allem auch sehr viel Zeit nötig. Was ich zum Beispiel jederzeit machen kann  
879 ist, dass ich sie mit der Entwicklungsumgebung von Unity vertraut mache und  
880 ein bestehendes Spiel modifiziere. Das kann sein, ich füge vielleicht einfach  
881 meine eigene Grafik ein und ersetze damit die vorhandene Grafik, oder ich  
882 ändere einfach nur das Verhalten durch bestimmte Parameter im Spiel, oder  
883 den Schwierigkeitsgrad dadurch, oder ich modifiziere das, indem ich einfach ein  
884 kleines, neues Feature dazu einbaue. Das kann man glaube ich im Grunde  
885 jederzeit machen, sobald sie lesen können. Aber dazu kommt natürlich, ich  
886 muss, wenn ich so ein vorhandenes Projekt modifizieren will, dann brauche ich  
887 das Projekt und das vorzubereiten und didaktisch sinnvoll aufzubereiten ist viel  
888 Arbeit.  
889

890 **I:** [1:09:09] Also zusammenfassend kann man das so sagen, das wichtigste  
891 Rüstzeug, wenn sie das mitnehmen sollten, wäre das eben Code zu schreiben  
892 oder zu verstehen?

893

894 **B:** [1:09:23] Ja, wenn ich wirklich Code schreiben mag mit ihnen. Wenn ich  
895 ihnen einfach nur erklären mag, wie man sich ein Spiel zusammenklicken kann  
896 und vielleicht einfach, oder angenommen zum Beispiel ich mache  
897 Bildbearbeitung und sie sollen ihre eigenen Bilder zeichnen mit Gimp oder mit  
898 Photoshop oder was auch immer und ich will einfach nur als Motivation  
899 nachher, dass sie diese Bilder in ein Spiel einfügen, das man auch spielen kann  
900 und dass sie lernen wie 2D-Animationen funktionieren mit Frames und so  
901 weiter, dann kann ich Unity genauso dafür hernehmen. Dafür müssen sie keine  
902 einzige Zeile Code schreiben, wenn ich den Code ihnen zur Verfügung stelle  
903 und sie einfach nur in das fertige Spiel ihre Grafiken einfügen und dann noch  
904 vielleicht ihre Animationen irgendwie verlinken. Dann haben sie auch mit Unity  
905 etwas gemacht und interessierte Leute haben sicher dann auch ein bisschen  
906 einen Einblick bekommen, wie so ein Spiel da funktioniert, aber dafür muss ich  
907 keinen Code schreiben. Das kann ich auch so einbauen. Man hat einfach  
908 wahnsinnig viele Möglichkeiten wie man diese Dinge aufziehen kann und wie  
909 man eine Gameengine einsetzen kann und je nachdem was man erreichen will,  
910 glaube ich, gibt es meistens sinnvolle Ansätze.

911

912 **I:** [1:10:54] Und wenn wir jetzt in Richtung dem Job als Programmierer sage ich  
913 einmal, hingeh, wie ist es da dann? Also wenn das die Spieleentwicklung mit  
914 Unity wirklich darauf ausgerichtet ist, welches Rüstzeug würde man dann  
915 benötigen? Gibt es da noch mehr vielleicht?

916

917 **B:** [1:11:19] Definitiv. Also als Rüstzeug bevor man mit Unity beginnt?

918

919 **I:** [1:11:25] Ja genau. Bevor man anfängt.

920

921 **B:** [1:11:29] Ich glaube es wäre sinnvoll Grundlagen in der jeweiligen  
922 Programmiersprache, also in dem Fall, bei Unity C# schon zu beherrschen. Ich  
923 glaube es ist wenig sinnvoll Grundlagen wie Variablen, einfache Operatoren,  
924 Funktionsaufrufe mit der Game-Engine Unity im Hintergrund zu behandeln, weil  
925 einfach viel zu viel Drumherum ist, das vom eigentlichen Kernthema ablenkt. Es  
926 ist glaube ich sehr viel einfacher dann eben so Grundlagen wie Variablen,  
927 Verzweigung, Schleifen in das Spiel sinnvoll einzubauen, wenn ich sie, ich sage  
928 einmal in einer unter Anführungszeichen isolierten Umgebung kennengelernt  
929 habe zumindest und grob verstanden habe.

930

931 **I:** [1:12:24] Und was wäre, wenn Sie die Zeit für Sie und ihre jeweilige Klasse  
932 anhalten könnten, was würden Sie den SchülerInnen vor der Spieleentwicklung  
933 mit Unity unbedingt beibringen wollen?

934

935 **B:** [1:12:43] Also quasi so, ich habe davor beliebig viel Zeit und kann beliebig  
936 viel machen?

937

938 **I:** [1:12:47] Ja.

939

940 **B:** [1:12:50] Ja, ich würde ihnen die Grundlagen in der jeweiligen Sprache  
941 beibringen, aber ich glaube ich würde mich auf die Grundlagen beschränken  
942 und ich würde dann mit gezielt zugeschnittenen Unterrichtsmaterialien und  
943 vorgefertigten Projekten einsteigen und ihnen so die Programmierung näher  
944 bringen.

945

946 **I:** [1:13:18] Bereits mit Unity?

947

948 **B:** [1:13:20] Bereits mit Unity, ja.

949

950 **I:** [1:13:27] Und jetzt kommt noch so ein kleiner Bereich. Benötigen  
951 SchülerInnen Ihrer Ansicht nach für die Spieleentwicklung mit Unity Kenntnisse

952 und Fertigkeiten aus dem Bereich Verantwortung, Datenschutz und  
953 Datensicherheit?

954

955 **B:** [1:13:40] Nein. Sie benötigen sie nicht und wenn man sie nicht behandelt,  
956 dann glaube ich auch nicht, dass sie sich zwangsweise damit  
957 auseinandersetzen. Das ist auch in den Analysen der Projekte bei mir in der  
958 Diplomarbeit herausgekommen, dass zum Beispiel ein Team, die haben  
959 Grafiken aus dem Internet verwendet, aber haben sich keine Gedanken darüber  
960 gemacht, ob sie diese überhaupt verwenden dürfen. Meiner Meinung nach  
961 bietet die Spieleentwicklung da auch viel Potenzial, das sinnvoll im Unterricht  
962 einzubauen und auch gleich praktisch einzubauen, weil wenn ich ein eigenes  
963 Spiel mache und ich zeichne alle Grafiken selbst, dann wird sich das im  
964 Unterricht nicht ausgehen. Dementsprechend kann man sehr gut auf, also  
965 bietet es sich sehr an, das man auf fertige Grafiken zum Beispiel, oder  
966 Soundeffekte genauso zurückgreift und es gibt genügend Plattformen, wo man  
967 sich diese herunterladen kann, zum Beispiel opengamearts.com oder so, wo  
968 teilweise auch verschiedene Lizenzierungen bei den Grafiken sind und je  
969 nachdem welche Lizenzierung es ist, muss man bestimmte Dinge beachten. Ob  
970 das jetzt CreativeCommons mit "by" ist, also dass ich den Autor angeben muss  
971 oder kommerziell oder sonst irgendetwas. Das heißt ich kann diese Sachen auf  
972 jeden Fall behandeln und ich glaube, wenn man sie behandelt, dann ist das  
973 auch eine sehr, sehr gute Umgebung, um das zu behandeln, weil sie gleich  
974 praktische Relevanz hat, aber es passiert nicht automatisch. Wenn ich einfach  
975 nur sage, macht ein Spiel und ihnen zeige wie man ein Spiel macht, dann  
976 werden sie wahrscheinlich, oder zumindest teilweise einfach auf Google gehen  
977 und sich Bilder herunterladen, die von irgendwo her sind und es ist ja eh nur für  
978 den Unterricht, da brauche ich mir keine weiteren Gedanken machen, so auf  
979 diese Art beziehungsweise vielleicht nicht einmal das.

980

981 **I:** [1:15:44] Und benötigen sie Kenntnisse und Fertigkeiten aus dem Bereich  
982 technische Grundlagen und Funktionsweisen, vor der Spieleentwicklung?



983  
984 **B:** [1:15:55] In gewisser Maßen ja, sicher, aber ich glaube man kann es auch im  
985 Rahmen dessen beibringen. Ich meine natürlich, bestimmte Grundlagen, wenn  
986 die nicht vorhanden sind, dann wird es irgendwann unpraktikabel. Also zum  
987 Beispiel einfach grundsätzlich der Umgang mit dem Computer, mit Dateien, wie  
988 speichere ich Dateien, wo finde ich meine Dateien, wie organisiere ich meine  
989 Dateien, wie öffne ich verschiedenen Programme, wie, wenn ich mein Unity  
990 offen habe, wie komme ich dann in den Webbrowser um dort etwas zu  
991 recherchieren, wie öffne ich mein Grafikbearbeitungsprogramm um dort etwas  
992 zu machen und wie kann ich grundsätzlich mit Unity arbeiten. Das sind  
993 Kompetenzen, die nicht selbstverständlich sind. Ich unterrichte sehr viel in der  
994 Unterstufe in der digitalen Grundbildung. Das sind Kompetenzbereiche wo die  
995 vorhandenen Kompetenzen bei den Schülerinnen und Schülern extrem weit  
996 auseinanderklaffen. Da ist es teilweise so, ich gebe Arbeitsaufträge und es gibt  
997 Leute, die sind in zwei Minuten fertig, es gibt Leute, die brauchen eine Stunde  
998 dafür. Weil eben, es ist nicht selbstverständlich, dass Kinder und Jugendliche  
999 einen PC bedienen zu Hause zum Beispiel in ihrer Freizeit. Sachen wie  
1000 Dateimanagement muss man ihnen beibringen. Wenn solche Sachen nicht  
1001 vorhanden sind, dann macht es glaube ich gar keinen Sinn. Technische Sachen  
1002 im Sinne von wie funktioniert jetzt der Computer an sich, was macht der  
1003 Prozessor, oder was macht vielleicht die Grafikkarte oder wie funktioniert die  
1004 Game-Loop im Hintergrund, das muss davor nicht bekannt sein. Das kann man  
1005 dann, wenn es relevant wird aufgreifen und macht auch glaube ich so am  
1006 meisten Sinn, aber ich glaube man kann das sehr gut abstrakt eben auf die  
1007 Spiele führend dann betrachten. Das Drumherum, das kommt dann teilweise  
1008 von selbst. Also ich glaube wichtig ist es, man sollte Dinge die den Arbeitsfluss  
1009 wirklich behindern, also wenn die Kompetenzen nicht vorhanden sind, eben wie  
1010 Dateimanagement, wie vielleicht eben die ganzen Grundlagen beim Code-  
1011 schreiben, von wegen es ist relevant wo ich ein Zeichen hin mache, es ist  
1012 relevant wo ein Befehl steht, in welcher Reihenfolge solche Sachen gemacht  
1013 werden. Also wenn ein Fehlen von Kompetenzen da wirklich ein Hindernis ist

1014 und dann vom eigentlichen Thema ablenkt, dann sollte man es vorher machen.  
1015 Wenn es einfach nur Hintergrundwissen ist, dann kann man das glaube ich auf  
1016 jeden Fall außen vorlassen und wenn dann Interesse da ist, oder wenn die  
1017 Notwendigkeit aufkommt das zu behandeln, dann kann man es behandeln.  
1018 Wobei es natürlich auf im Rahmen vom Unterricht, je nach Gruppengröße dann  
1019 eventuell nicht praktikabel ist. Aber man muss nicht alles, man muss nicht die  
1020 Welt verstehen, um einmal ein kleines Projekt auf die Beine zu stellen.

1021

1022 **I:** [1:19:18] Also, den Bereich haben Sie dann teilweise schon mitbeantwortet,  
1023 also die Betriebssysteme und Software sozusagen.

1024

1025 **B:** [1:19:27] Ja also, es hilft natürlich enorm, wenn ich die Sachen verstehe,  
1026 aber ich muss jetzt nicht im Detail wissen, wie bestimmte Windowssachen  
1027 funktionieren zum Beispiel. Vor allem Unity kann ich sogar unter MacOS oder  
1028 sogar unter Linux verwenden und es ist im Grunde das Betriebssystem dahinter  
1029 relativ egal. Das muss ich nicht verstehen dafür.

1030

1031 **I:** [1:19:51] Und jetzt konkret angesprochen auf Dateimanagement und  
1032 Datensicherung und -Wiederherstellung?

1033

1034 **B:** [1:19:57] Muss ich so verstehen, dass ich es verwenden kann, also ich muss  
1035 es verwenden können. Unity-Projekte bestehen aus vielen, vielen, vielen  
1036 einzelnen Dateien, die ich teilweise managen muss, also zumindest die Dateien  
1037 für die Assets im Spiel, also die Grafiken, Soundeffekte, Skripte und  
1038 Spielobjekte, beziehungsweise vor allem wenn ich dann Grafiken von außen in  
1039 das Spiel einbauen mag, dann muss ich mich zwangsweise damit  
1040 auseinandersetzen. Ich muss nicht im Detail verstehen, also verstehen wie es  
1041 funktioniert. Ich muss es bedienen können. Es darf quasi nicht im Weg stehen.  
1042 Also das finde ich ist das Relevante, auf das soll man achten. Dazu zählen so  
1043 Dinge wie zum Beispiel eben ja, Dateien „copy-paste“ irgendwo hin  
1044 transferieren. Das sollte man draufhaben.

1045

1046 **I:** [1:20:59] Schon als Vorbedingung? Bevor man damit beginnt?

1047

1048 **B:** [1:21:00] Ja. Also ich sage einmal, diese Kompetenzen im Rahmen der  
1049 Spieleentwicklung zu erlernen erachte ich nicht als zielführend. Weil das lenkt  
1050 dann vom eigentlichen Thema nur ab. Und man hat nur eine gewisse  
1051 Aufmerksamkeitsspanne und ich kann nicht alles gleichzeitig lernen.

1052

1053 **I:** [1:21:31] Was meinen Sie bezüglich benötigter Kenntnisse und Fertigkeiten  
1054 im Bereich der Produktion digitaler Medien? Also Text, Bild, Ton, die  
1055 Bearbeitung und Produktion davon?

1056

1057 **B:** [1:21:44] Variiert sehr stark, je nachdem wie man Projekte aufzieht und oder  
1058 wie Schülerinnen und Schüler das machen wollen. Wenn da jetzt jemand ist,  
1059 der unbedingt seine eigene Grafik machen will, dann gibt es natürlich sehr hohe  
1060 Lerneffekte, wenn sie das dann tatsächlich machen. Gleichzeitig macht man im  
1061 Unterricht keine vollständigen Spiele, die, also da kann man auch einmal sagen  
1062 ok, Soundeffekte lasse ich einmal einfach weg und ich kümmerge mich um  
1063 andere wesentliche Aspekte und dann habe ich zum Beispiel gar keine  
1064 Berührungspunkte mit Sound. Ist natürlich nicht optimal, aber kann man so  
1065 machen. Das ist nicht zwangsweise der Fall, aber es bietet auf jeden Fall enorm  
1066 viel Potenzial, dass man diese Produktion von digitalen Medien praktisch  
1067 anwendet und zwar in einem Setting das für viele Schülerinnen und Schüler  
1068 motivierend wirkt, von wegen ich zeichne jetzt ein Bild das dann nicht einfach  
1069 nur irgendwo am PC dahingammelt, sondern ich verwende das dann für ein  
1070 Spiel und dann hüpf die Spielfigur tatsächlich herum und ist nicht einfach nur  
1071 ein Foto, das gibt es irgendwo und starr anzuschauen ist, sondern es macht  
1072 etwas.

1073

1074 **I:** [1:23:13] Und noch zum letzten Bereich, nämlich der Suche, Auswahl und  
1075 Organisation von Informationen. Sind da Vorkenntnisse, Fertigkeiten gut oder  
1076 benötigt?

1077

1078 **B:** [1:23:25] Ich glaube das ist auf jeden Fall gut. Hängt davon ab. Wenn ich  
1079 ihnen zum Beispiel ein detailliertes Tutorial zur Verfügung stelle, anhand  
1080 dessen sie ein Spiel erstellen oder anhand dessen sie dann arbeiten, dann ist  
1081 das nicht so relevant. Wenn sie jetzt ein eigenständiges Projekt machen sollen,  
1082 dann ist es sehr relevant. Gleichzeitig aber ist es auch wieder ein  
1083 Kompetenzbereich wo man glaube ich sagen kann, es eignet sich schon gut,  
1084 also Spielentwicklung eignet sich dafür Kompetenzen in diesem Bereich auch  
1085 zu erwerben. Eben gerade, wenn ein hoher Motivationsfaktor da ist, von wegen,  
1086 ich will ein Spiel machen, dass so ein „grid based movement“ so wie diese alten  
1087 Pokemon-Spiele hat und ich will, dass das funktioniert und ich will das nicht  
1088 anders und deswegen recherchiere ich jetzt und versuche das selbstständig zu  
1089 lösen und dabei erwerbe ich Kompetenzen in diesem Bereich. Also wohingegen  
1090 einfach nur wenn ich sage ok, wir lernen jetzt recherchieren, ich gebe euch  
1091 einen Rechercheauftrag und macht das und da ist kein, also die Motivation  
1092 besteht nur darin, das ist ein Arbeitsauftrag, den ich zu erfüllen habe, dann ist  
1093 der Lerneffekt sicher weniger da, als wenn eine hohe Eigenmotivation da ist.

1094

1095 **I:** [1:25:03] Und wenn man jetzt sagt, so wie das Unterrichtsprojekt momentan  
1096 existiert? Oder gibt es da mehrere Versionen davon? Unterschiedliche?

1097

1098 **B:** [1:25:13] Also, so wie wir es aufgezogen haben, haben wir es grundsätzlich  
1099 stringent so gemacht, dass eben zuerst Grundlagen gemacht werden in C# und  
1100 dann fängt man mit der Spieleentwicklung an und am Schluss erstellen sie  
1101 dann ein eigenes Projekt. Das war so die Idee, wobei die eigenen Projekte  
1102 dann am Schluss teilweise ein bisschen untergegangen sind, wegen  
1103 Zeitmangel. Aber es sind Lernmaterialien entstanden im Zuge dieses Projektes,  
1104 die frei zur Verfügung stehen, kann jeder und jede verwenden im Unterricht

1105 oder auch privat oder wie auch immer und die werden noch verwendet und  
1106 hoffentlich auch noch aktuell gehalten und weiterentwickelt. Und wie man diese  
1107 Unterrichtsmaterialien dann einsetzt, das hängt natürlich wieder davon ab, wie  
1108 man das machen mag. Aber es gibt grundsätzlich die Idee vom Grundlagen-  
1109 Tutorial, dann gibt es ein Tutorial dafür, wie man ein Spiel von Grund auf  
1110 erstellt, anhand eines 2D-Plattformers und dann gibt es noch ein anderes Spiel,  
1111 das quasi schon grob fertig ist, dass man dann nur modifizieren muss, oder  
1112 vervollständigen muss und dann gibt es verschiedene Möglichkeiten das  
1113 weiterzuentwickeln.

1114

1115 **I:** [1:26:47] Haben Sie das dann in unterschiedlichen Varianten schon  
1116 verwendet?

1117

1118 **B:** [1:26:51] Ja, also vor allem das eine Tutorial, dass ein Spiel von Grund auf  
1119 erstellt, das Bouncy Fant, das haben wir sowohl in den fünften Klassen als auch  
1120 in den siebenten Klassen gemacht. Und es war nicht immer gleich vom Setting  
1121 her. Aber ich glaube sie haben immer vorher schon C#-Vorkenntnisse gehabt.

1122

1123 **I:** [1:27:34] Gut, dann wären wir am Ende des Interviews angelangt. Gibt es  
1124 irgendetwas, das sie vielleicht noch ergänzen oder loswerden möchten?

1125

1126 **B:** [1:27:48] Ja also, zusammenfassend ist Spieleentwicklung im Unterricht, es  
1127 bietet enorm viel Potenzial weil von Seiten der Schülerinnen und Schüler viel  
1128 Eigenmotivation von Haus aus einmal da ist, meistens und man Arbeiten in dem  
1129 Bereich sehr, sehr unterschiedlich gestalten kann, sodass man etwas in  
1130 verschiedensten Arten irgendwie einbauen kann im Unterricht. Und ja, es bietet  
1131 enorm viel Potenzial. Gleichzeitig ist es nicht immer sinnvoll glaube ich und man  
1132 muss abwägen und es ist auch sehr aufwändig dafür finde ich,  
1133 Unterrichtsmaterialien zu erstellen und solche Sachen zu machen. Ja und je  
1134 nachdem wie man eben den Unterricht mit der Spielentwicklung gestaltet, kann

1135 man ganz, ganz unterschiedliche Lernerfolge und Kompetenzbereiche fördern  
1136 und es hängt sehr davon ab, wie man es aufzieht, sozusagen.

1137

---

### 1 10.7.3 Interview 03

2 [0:00:00] **I:** Beschreiben Sie bitte Ihre Erfahrungen mit dem Unterrichtsprojekt  
3 "learn to proGrAME".

4

5 **B:** [0:00:12] Wie jetzt Erfahrungen? Wie für mich der Unterricht war, oder wie  
6 die Lernumgebung war, oder? Können Sie das genauer sagen, was Sie da  
7 hören wollen?

8

9 **I:** [0:00:22] Ganz allgemein, wie es für Sie jetzt war und vielleicht auch wie sie  
10 dazu gekommen sind.

11

12 **B:** [0:00:29] Ok, also meine Motivation. Wie ich dazu gekommen bin, über  
13 dieses Projekt und es hat mich interessiert, weil ich natürlich als  
14 Informatiklehrer interessiert daran bin, dass sich Jugendliche mit Informatik,  
15 insbesondere mit Programmieren auseinandersetzen. Und dafür war dieses  
16 Projekt recht gut geeignet und ich war interessiert auch an Unity, das  
17 kennenzulernen. Das war für mich was Neues eigentlich auch. Also ich hatte  
18 vorher noch nicht mit Unity gearbeitet und ich wollte einfach auch diese  
19 Spieleumgebung, diese Spieleprogrammierungsumgebung kennen lernen. Also das  
20 war die Motivation. Ich bin auch überzeugt davon, dass für Kinder und  
21 Jugendliche Spiele programmieren durchaus etwas sein kann, das motivierend  
22 sein kann. Und deshalb bin ich da zum Projekt gekommen. Und wie es nachher  
23 war, war sehr unterschiedlich. Also da gibt es ganz vielfältige Eindrücke. Von  
24 Leuten die, oder von der Schwierigkeit die einfach Programmieren lernen mit  
25 wenig Computer-Literacy mit sich bringt, zu einer professionellen  
26 Programmierungsumgebung, die aufwändig zum Lernen ist, weil sie komplex ist,  
27 weil man eben viel damit machen kann, bis hin zu Leuten die sich da

28 durchgerungen haben und auch jetzt noch immer sehr interessiert sind am  
29 Programmieren, die eben durch dieses Projekt eigentlich auf den Geschmack  
30 gekommen sind. Also das war sehr unterschiedlich. Also meine Erfahrung war  
31 eben auch unterschiedlich, weil einerseits war ich manchmal stark gefordert,  
32 indem das für mich auch neu war und indem dass wir die Ressourcen erst im  
33 Zuge des Projektes erarbeitet haben, also diese Lernressourcen /  
34 Lernunterlagen und ich eben teilweise gemeinsam mit den Schülerinnen und  
35 Schülern gelernt habe. Und auf der anderen Seite, oder zusätzlich gab es eben  
36 auch Momente, wo dann Spiele herausgekommen sind und das war sehr  
37 spannend. Und ich würde sagen, das faszinierendste an der ganzen  
38 Geschichte war im zweiten Jahr dann, wo wir dann andere Lernunterlagen  
39 hatten, die aufbauender waren. Im ersten Jahr ist es so gewesen, dass sie  
40 zuerst in der Konsole programmiert haben und dann so ein Tutorial  
41 durchgemacht haben für ein Plattformenspiel. Im zweiten Jahr war es so, dass  
42 wir gleich in Unity gearbeitet haben und dort Programmieren gelernt haben.  
43 Und das war Faszinierendste vom ganzen Projekt war eigentlich im zweiten  
44 Durchlauf, wie nach den, also da gab es nachher so Pflichtsachen, da haben  
45 sie dann nicht mehr dieses Plattformenspiel machen müssen bei mir, sondern  
46 sie haben nur die Grundlagen, also so grundlegende Programmiersachen, so  
47 Schleifen, Wenn-Abfragen usw. kennengelernt. Wie wir nachher durch waren,  
48 durch diese Pflichtstunden, haben sie freiwillig Projekte machen können. Und  
49 ich habe immer wieder Projekte gemacht, die letzten Jahre und ich muss  
50 sagen, die Projekte vom letzten Jahr waren wirklich super. Sie haben nachher  
51 nicht alle in Unity Projekte gemacht, aber doch acht Gruppen zirka haben in  
52 Unity Projekte gemacht und die haben aber nachher wirklich neue Spiele  
53 gemacht, oder eben irgendwie Erweiterungen gemacht zu den vorhandenen  
54 Spielen. Und auch die anderen Projekte waren recht cool und recht aufwändig  
55 eigentlich, also im Vergleich zu den vorhergehenden Jahren durchaus  
56 ambitioniert und die meisten sind auch was geworden. Also das habe ich mir  
57 mitgenommen, also irgendwie gerade diese Erfahrung mit Unity und die  
58 Schülerinnen und Schüler waren da stark gefordert auch und viele

59 herausgefordert mit dieser Programmierumgebung. Aber irgendwie da, ich weiß  
60 nicht ob man einen Zusammenhang herstellen kann, aber auf jeden Fall waren  
61 die Projekte nachher sehr super.

62

63 **I:** [0:04:51] Ok. Und das war wirklich eigens von den Schülern quasi die Ideen  
64 dieser Projekte, das waren keine vorgegebenen Aufgaben oder Spiele oder so?

65

66 **B:** [0:05:05] Ja genau, das will ich ja sagen. Nein, nein, nein, genau, das war  
67 ebenso, dass die da im zweiten Semester, also das einzige Mal wo ich das  
68 eben, also ich habe das als Lehrer immer wieder gemacht, außer beim ersten  
69 Durchlauf haben wir das nicht gemacht, aber jetzt im zweiten wieder. Und sie  
70 haben dann wirklich eineinhalb Monate Zeit, dass sie ein selbst gewählten  
71 Projekt machen in Informatik. Also ich habe schon Vorschläge, aber sie wählen  
72 dann selber etwas das sich machen wollen. Das war echt cool, also was da  
73 geliefert wurde letztes Jahr. Also ich meine die acht Spiele waren super, da  
74 waren nachher echte Platformer-Games, also auch mit Kamerabewegung  
75 dabei, da war ein kompletter Mod von einem der Spiele, von diesem Bouncy  
76 Fant Tutorial hat eine Gruppe einen Mod gemacht, wo es nachher um zwei  
77 Fische geht, die herumschwimmen, also wo eine andere Physik war. Dann hat  
78 es gegeben ein 3D-Pong, dass überhaupt nicht irgendwie bei den Tutorials  
79 vorkam. Also es gab wirklich da neue Sachen. Und andere Gruppen haben  
80 einen Algorithmus ausgearbeitet am Papier, der nachher in der Gruppe  
81 besprochen wurde, oder sie haben nachher geschwitcht auf den Rasperry und  
82 haben so ein Whac-a-Mole-Spiel gemacht mit Scratch, aber mit einem externen  
83 Microcontroller, also wo sie nachher mit Leds und mit Buttons dieses Whac-a-  
84 Mole-Spiel gespielt haben. Also das waren echt coole Projekte letztes Jahr und  
85 das nachher in so einer Phase, wo ich schon von vielen Schülerinnen und  
86 Schülern irgendwie mitbekommen habe, dass das sehr anstrengend war für sie  
87 und dass das sehr frustrierend auch, eben in Unity das Programmieren lernen.

88



89 **I:** [0:06:47] Ok. Das war jetzt ohne den Voreinsteigern quasi, ohne dem  
90 Scratch-Tutorial und dem C#-Tutorial?

91

92 **B:** [0:07:02] Das Scratch-Tutorial haben alle gehabt. Also Scratch habe ich  
93 immer gemacht. Nur im ersten Durchlauf war es so, zuerst Scratch, dann in der  
94 Konsole und dann in Unity und beim zweiten Durchlauf habe ich es so gemacht,  
95 zuerst Scratch und dann direkt in Unity starten.

96

97 **I:** [0:07:20] Ok. Dann wechseln wir ein bisschen das Thema. Gehen wir ein  
98 bisschen in eine andere Richtung. Wodurch zeichnet sich Ihrer Meinung nach  
99 ein guter Programmierer aus?

100

101 **B:** [0:07:37] Ein guter Programmierer, eine gute Programmiererin. Also das ist  
102 schwierig zu sagen. Weil ich glaube das Programmieren an sich sehr vielfältig  
103 sein kann, weil, aber ich glaube, oder sagen wir so, meine Erfahrung ist, die  
104 Leute, die ganz gut programmieren können, die waren oft Leute, die Interesse  
105 gehabt haben dafür, dass sie quasi Algorithmen optimieren, was ich so  
106 mitbekommen habe. Meistens waren es Leute, die irgendwie gerne gehabt  
107 haben, dass Programmstruktur einheitlich ist und oft auch irgendwelchen  
108 Designprinzipien gefolgt sind, also dass sie sich eher so an Modell-View-  
109 Controller-Patterns gehalten haben und so. Also ich glaube das sind so  
110 Sachen, die durchaus für professionelles Programmieren stehen. Gleichzeitig  
111 sehe ich, dass das sehr vielfältig sein kann. Es kommt darauf an in welchem  
112 Bereich man programmiert, würde ich sagen, also es gibt ja Frontend, Backend,  
113 es gibt verschiedene Plattformen, es gibt verschiedene Zugänge, also man  
114 kann ja heutzutage sehr viel irgendwie auch mit Frameworks lösen. Also es gibt  
115 Leute die super Userapplikationen machen und mit Hilfe von irgendwelchen  
116 Frameworks die es schon gibt und gleichzeitig gibt es Leute, die sich total  
117 spezialisieren darauf, dass sie irgendwelche Algorithmen verfeinern und das  
118 können alles gute Programmierer sein. Also wahrscheinlich „know-your-tools“  
119 ist würde ich sagen, ist wahrscheinlich etwas, was ein guter Programmierer hat,

120 der weiß mit was er gut arbeiten kann und ist jemand, der immer wieder  
121 dazulernt.

122

123 **I:** [0:09:42] Und wodurch sich ein guter Programmierer auszeichnet, wo sieht  
124 man das vielleicht in Bezug auf digitale Kompetenzen?

125

126 **B:** [0:09:52] Digitale Kompetenzen, meinen Sie so etwas wie, ja, naja,  
127 Modellierungsfähigkeit, also dass man quasi etwas abstrakt irgendwie fassen  
128 kann, dass man Zusammenhänge herstellen kann, dass man so etwas hat wie,  
129 dass man Algorithmen entwickeln kann überhaupt, oder irgendwo auch erkennt,  
130 wo man Algorithmen einsetzen kann. So etwas würde ich sagen. Das wären  
131 digitale Kompetenzen, die ganz praktisch sind, glaube ich.

132

133 **I:** [0:10:36] Ja. Und wo kann dabei die Spieleentwicklung mit Unity einen  
134 Beitrag leisten?

135

136 **B:** [0:10:43] Ja sicher in der Modellierung, also weil bei einem Spiel mit Unity,  
137 also bei beiden was wir jetzt gesagt haben, was man braucht, einerseits die  
138 Fähigkeit, dass man irgendein Modell sich überlegen kann und auch  
139 implementieren kann, weil das ist das was ein Computerspiel quasi ist. Und  
140 man braucht Algorithmen damit man überhaupt die Spiellogik quasi machen  
141 kann.

142

143 **I:** [0:11:18] Denken Sie an einen Musterschüler oder eine Musterschülerin im  
144 Projekt. Und was denken Sie nahm der- oder diejenige für einen etwaigen Job  
145 als Programmierer mit?

146

147 **B:** [0:11:30] Falls die überhaupt in diese Richtung gehen. Was würden sie  
148 mitnehmen? Dadurch, dass sie vorher nie programmiert haben, nehmen sie  
149 sicher mit, dass es nicht so ist wie, dass man einfach irgendetwas  
150 zusammenklickt, sondern das man vorher eben Mal über was, also man muss

151 mitdenken, man muss irgendwie eine Überlegung haben zu dem was man  
152 ausarbeitet. Sie nehmen sich mit, dass man viel in der Doku nachschauen  
153 muss und dass man insofern, also dass das dazu gehört, das man viel lesen  
154 muss, damit man programmieren kann.

155

156 **I:** [0:12:15] Wenn ich da kurz einhaken darf, Lesen, was und wo lesen?

157

158 **B:** [0:12:19] In der Doku lesen. Ich würde behaupten, Programmieren ist achtzig  
159 Prozent Doku lesen.

160

161 **I:** [0:12:23] Ok.

162

163 **B:** [0:12:24] Dann ist zwanzig Prozent von der Programmierung. Von den  
164 zwanzig Prozent ist es so, dass man wahrscheinlich einmal bei dem einen  
165 letzten Prozent einrechnen muss, dass das genauso lange dauert wie die  
166 letzten neunzehn Prozent. Das ist halt Programmieren. Und viel Lesen glaube  
167 ich.

168

169 **I:** [0:12:39] Mit Doku lesen, wohin geht das, in welche Richtung? Was heißt  
170 Doku lesen?

171

172 **B:** [0:12:45] APIs Klassendefinitionen.

173

174 **I:** [0:12:52] Ok. Fällt da auch hinein, Internet-Recherchen?

175

176 **B:** [0:13:01] Ja.

177

178 **I:** [0:13:03] Oder solche Dinge, oder?

179

180 **B:** [0:13:05] Ja, das fällt rein.

181

182 **I:** [0:13:08] Ok. Und was zeichnet die Spieleentwicklung mit Unity in Bezug auf  
183 die Programmierung aus, oder wo sehen Sie den größten Nutzen?

184

185 **B:** [0:13:20] Von Unity?

186

187 **I:** [0:13:21] Mhm.

188

189 **B:** [0:13:22] Naja, Unity hat den riesigen Vorteil, oder den „big advantage“, dass  
190 es in der Industrie verwendet wird. Es ist eigentlich ein Profitool und vom  
191 Programmieren spezifisch her, ist es insbesondere cool, dass man in C#  
192 arbeitet und C# ist einfach eine ganz nette Sprache zum Programmieren. Also  
193 es ist eine objektorientierte Sprache, die doch einige Feinheiten hat, die  
194 weiterentwickelt wird von Microsoft, die Industriestandard ist, die quasi die  
195 Grundlage ist für das Ganze Windows, die ganze Windowsplattform, oder auch  
196 Webplattformen. Also man kann mit .Net ja ziemlich viel machen. Also man hat  
197 eigentlich, wenn man C# lernt eine ziemlich gute Grundlage für alle möglichen  
198 Bereiche zum Programmieren. Es ist eine Hochsprache, es hat Datentypen,  
199 also nicht nur „led“ und „var“, sondern „int“, „float“ usw. Also das sind alles ganz  
200 gute Sachen zum Lernen. Es hat garbage-collection eingebaut, also man muss  
201 sich nicht so viel darum kümmern um das, dass man irgendwelchen Memory  
202 wieder freikommt, usw. Also es ist eine gute Sprache zum Lernen finde ich.

203

204 **I:** [0:14:27] Also und die Kombination mit Unity quasi, ähm.

205

206 **B:** [0:14:30] Es ist an und für sich eine gute Sprache zum Lernen und im Unity  
207 ist das Schöne, dass man noch dazu, ebenso, man bekommt das  
208 Objektorientierte glaube ich noch mehr hinein, dadurch dass in Unity so ein  
209 Editor ist, der quasi Objekte zur Verfügung stellt und man quasi an Objekte  
210 sozusagen ein so ein Skript anhängt, hat man einfach das, also kann ich mir  
211 vorstellen, dass das dadurch irgendwie gleich von vorneherein mehr da ist, der  
212 objektorientierte Zugang. Das würde ich hervorheben. Und man lernt durch

213 Unity natürlich auch andere Skills, die ganz praktisch sind, wenn man in der IT-  
214 Industrie ist, weil man hat eben was mit Grafik zu tun, man muss insofern also  
215 auch etwas wissen über Datentypen, nicht nur über Datentypen, sondern auch  
216 über Dateiformate, also welche Grafiken kann man reinladen, welche nicht, wo  
217 kann man was speichern. Das sind eigentlich so Grundlagen, bei denen viele  
218 Schülerinnen und Schüler in dem Projekt eigentlich teilweise Schwierigkeiten  
219 hatten und das sind auch Sachen, die man da durchaus lernen kann.

220

221 **I:** [0:15:37] Hat die Spielentwicklung mit Unity einen Einfluss auf das  
222 Verständnis der SchülerInnen eines Algorithmus?

223

224 **B:** [0:15:46] Das ist zu allgemein, die Frage für mich. Also, das kommt auf das  
225 Spiel darauf an, das man implementiert und auf die Geschichte, die man  
226 produziert. Also ein Bekannter von mir, der unterrichtet an einer anderen Schule  
227 mit Unity zum Beispiel „artificial intelligence“ momentan, er baut quasi ein  
228 Szenario in Unity, wo Schüler dann etwas weiter programmieren müssen und  
229 Schülerinnen und dadurch kommt, kann man quasi AI-Algorithmen entwickeln.  
230 Also, klar kann man mit Unity recht gut wahrscheinlich sogar modellieren, wo  
231 Algorithmen sinnvoll sind und wie Algorithmen ausschauen, aber per se muss  
232 das nicht sein. Also man kann auch Unity machen, ohne dass man ein großes  
233 Verständnis hat von Algorithmen, würde ich behaupten. Ich meine es kommt  
234 darauf an, sobald man glaube ich einsteigt und wirklich selbstständig zu  
235 arbeiten beginnt und wirklich selbst etwas entwickeln will, kommt man nicht  
236 umhin, dass man sich beschäftigt mit Algorithmen und mit den mathematischen  
237 Grundlagen, wie zum Beispiel dieser Physiks-Body funktioniert oder so. Aber  
238 wenn man nur so, wie wir das gemacht haben, eben Tutorials macht und die  
239 quasi so abschreibt, da wird wahrscheinlich noch nicht so viel hängen bleiben,  
240 was das angeht.

241

242 **I:** [0:17:15] Ok, da noch nicht, und dann bei den eigenen Projekten die die  
243 SchülerInnen dann haben, da auch nicht?

244

245 **B:** [0:17:23] Also da haben sie sich auch beschäftigt, mit allen möglichen  
246 Sachen, also von ähm, ich weiß nicht, beim 3D-Pong war oft das Thema so, wie  
247 geht das jetzt mit der X-Koordinate, Y-Koordinate, dass man da irgendetwas  
248 bewegen kann und so, also das war zum Beispiel so ein Thema. Oder bei dem  
249 anderen wo quasi die Fische so herumschwimmen, da ging es darum, wie kann  
250 man quasi dieses Springen umprogrammieren, sodass die Fische  
251 herumschwimmen können. Also da wird so, ich weiß nicht ob man es schon  
252 algorithmisches Denken nennen kann, aber so Problemlösungszugang, das auf  
253 jeden Fall kommt da viel stärker. Also bei den eigenen Projekten kam das sehr  
254 stark. Da gab es dann bestimmte Probleme und dann wurde versucht mit „trial-  
255 and-error“ oder mit Fragen wie kann man da weitermachen.

256

257 **I:** [0:18:16] Ok. Wird da explizit darauf eingegangen, dass man sich mit  
258 Algorithmen beschäftigt, also Handlungsanleitungen?

259

260 **B:** [0:18:23] Ja, das ist gleich am Anfang gekommen. Das war die zweite  
261 Unterrichtsstunde und ist dann in diesem Tutorial, diesem proGrAME-Tutorial  
262 eigentlich gleich anfangs, also das Programmieren eigentlich so eine Art  
263 Problem lösen ist und dass es da um Algorithmen geht und was ein Algorithmus  
264 ist, also das wurde auf jeden Fall erklärt. Und ich meine im Prinzip, wenn man  
265 diese Spiele durchgeht, macht man ja das, dass man Algorithmen  
266 implementiert. Nur ich glaube es ist die eine Sache ob man das mitmacht und  
267 nachmacht, so wie man das eben macht, wenn man noch nie vorher  
268 programmiert hat und dann mit Unity konfrontiert wird. Also wenn man versteht,  
269 was das bedeutet, was man da gerade macht. Das sind glaube ich zwei  
270 verschiedene Sachen und ich glaube, dass es in meinen Klassen auf jeden Fall  
271 ein paar [Leute] gegeben hat, die das versucht haben zu verstehen und viele  
272 einfach nicht. Weil es einfach prinzipiell zu anspruchsvoll war, weil man eben  
273 viele Grundlagen wissen muss schon vorher, bevor man überhaupt dort

274 hinkommt, dass man verstehen kann, was da in so einem C#-Code eigentlich  
275 ausgedrückt wird.

276

277 **I:** [0:19:26] Ok. Können die Schülerinnen nach der Spieleentwicklung mit Unity  
278 wesentliche Aspekte der Programmierung, also zum Beispiel Funktionen,  
279 Objektorientierung nennen und erläutern?

280

281 **B:** [0:19:42] Also die Idee wäre gewesen ja, aber da kommt jetzt der andere  
282 Steam bei meinen Gruppen hinein, dass wenn man das lernen würde, und  
283 wirklich wiederholen würde und zu Hause sich Zeit nehmen würde, Sie  
284 bemerken meinen Konjunktiv, dann könnte das so sein, dass man das nachher  
285 erklären kann, nach diesem Tutorial, weil diesen Aspekte werden ja durchaus  
286 explizit erwähnt und durchaus explizit behandelt, nur muss man halt die  
287 Schulrealität sehen, wo auf jeden Fall die Schülerinnen und Schüler in meiner  
288 Schule, also in meiner Schule hat Informatik, ist nicht so ein zentrales Fach. Ich  
289 bin auch der einzige Informatiklehrer dort. Wir haben keinen dezidierten  
290 Informatikraum und die Schülerinnen und Schüler haben netto in der fünften  
291 Klasse Informatik und vorher nicht wirklich und nachher nicht wirklich. Das  
292 heißt, sie kommen wirklich mit minimaler Computer-Literacy oft in die fünfte  
293 Klasse. Also es ist tatsächlich jedes Jahr immer wieder Thema, jetzt speichere  
294 ich das auf einen USB-Stick und es bleibt lange Thema, das man überhaupt  
295 solche Sachen lernt, wie das Dateisystem funktioniert. Und wenn nachher Leute  
296 die irgendwo eine Schwierigkeit haben damit, das Windowsdateimanagement  
297 zu verstehen, also wo sie jetzt irgendwelche Dateien speichern und so, dann  
298 Unity programmieren, das ist, da sind ganz ehrlich, ganz viele einfach  
299 ausgestiegen, weil das einfach zu viel Voraussetzung gebraucht hat für die  
300 Jugendlichen. Also insofern ist die Frage, ja sicher, man kann das lernen, wenn  
301 man einmal schon so weit ist, dass man die Grundlagen von dem wie ein  
302 Betriebssystem funktioniert hat und sich wirklich auf das konzentrieren kann  
303 und wenn man weiß wie zum Beispiel Grafiken funktionieren und so und  
304 Dateieindungen und was die für Bedeutungen haben. Also wenn man diese

305 ganzen Grundlagen hat, dann kann man sich auf Unity wahrscheinlich  
306 einlassen und dann kann man wahrscheinlich das schon gut lernen, was eine  
307 Funktion ist. Wobei ich weiß nicht, ob es nicht einfachere Tools gibt, um zu  
308 lernen wie Programmieren funktioniert, das muss ich schon ganz ehrlich  
309 gestehen. Aber eben der große Vorteil von Unity wie gesagt ist, dass man  
310 Spiele programmiert und dass das eine industriell verwendetes Tool ist. Also  
311 das ist der große Vorteil von Unity. Ob es didaktisch das beste Tool ist, das  
312 würde ich durchaus kritisieren. Aber man kann es natürlich lernen, was  
313 Funktionen sind und was objektorientiertes Programmieren ist.

314

315 **I:** [0:22:10] Ok. Lernen die SchülerInnen im Verlauf der Spielentwicklung mit  
316 Unity unterschiedliche Programmiersprachen und Produktionsabläufe kennen?

317

318 **B:** [0:22:22] Wie unterschiedliche Programmiersprachen? Nein, also  
319 Programmiersprachen haben sie bei mir C# kennen gelernt.  
320 Produktionsabläufe, ich weiß nicht was Sie damit meinen. Was meinen Sie mit  
321 Produktionsabläufen?

322

323 **I:** [0:22:36] Naja, wie zum Beispiel, dass es nicht nur aus dem Coden zum  
324 Beispiel besteht, Programmierung und so weiter.

325

326 **B:** [0:22:46] Ja klar. Programmieren per se würde ich mal behaupten ist gleich  
327 Coden. Also ich kenne da jetzt nicht so den Unterschied. Aber dass man mit  
328 Unity andere Sachen macht auch, wie zum Beispiel Animationsabläufe erstellen  
329 und so, ja das lernen sie auch. Aber das ist nicht notwendigerweise  
330 Programmieren für mich.

331

332 **I:** [0:23:10] Ok. Erwerben die SchülerInnen die Fertigkeit Aufgaben mit Mitteln  
333 der Informatik zu modellieren?

334



335 **B:** [0:23:17] Ja. Wieder im Konjunktiv, bis zu einem gewissen Grad ja, wäre so  
336 oder ist so, wäre so. Wenn man sich damit auseinandersetzt, ist das so, dann  
337 kann man diese Fähigkeit erwerben. Also Unity ist so wie andere auch dazu  
338 geeignet, dass man so etwas erwerben könnte und kann, wenn man sich damit  
339 auseinandersetzt. Aber Voraussetzung ist, das man selber motiviert ist und das  
340 man das gerne machen will und das war auch so in meinen Klassen, dass ich  
341 definitiv auch Schüler hatte, die dieses Spiele-programmieren nicht so  
342 interessiert hat und ich glaube das war ein großer Faktor, oder die überhaupt in  
343 Informatik irgendwie nicht so, die das nicht so erkannt haben als etwas  
344 besonders Wichtiges in ihrem Leben und dadurch auch wenig Interesse oder  
345 wenig Aufmerksamkeit bezüglich Unity gehabt haben und diese insofern auch  
346 nicht so viel mitgenommen haben vermutlich. Aber wieder eben, wenn man die  
347 Leute, die sich darauf eingelassen haben, die Leute, die versucht haben, ein  
348 eigenes Spiel nachher zu machen in den eigenen Projekten, von denen würde  
349 ich sehr wohl behaupten, dass die genau das mitgenommen haben, ja.

350

351 **I:** [0:24:28] Mhm. Und gibt es da vielleicht irgendein Beispiel wo man das sieht,  
352 wo das der Fall war?

353

354 **B:** [0:24:34] Naja eben ein eigenes Spiel, also zum Beispiel das 3D-Pong, das  
355 überhaupt nicht in irgendeiner Form in den Tutorials vorkommt. Ein so ein  
356 eigenes Spiel sich zu überlegen und dann zu implementieren mit den ganzen  
357 Arbeitsabläufen, die in Unity dafür notwendig sind, nämlich dass man nicht eine  
358 2D-Umgebung, sondern eine 3D-Umgebung verwendet, die wir im Tutorial nie  
359 verwendet haben. Dass man dort 3D-Objekte gestaltet, dass man diese noch  
360 dazu bringt, dass sie sich bewegen, dass man sie bewegen kann als Spieler,  
361 also das sind ja eindeutig Zugänge wo man quasi Probleme hat und wo man  
362 quasi das lernt, wie man da einen Problemlösungszugang entwickeln kann.

363

364 **I:** [0:25:10] Und sehen Sie das als Voraussetzung für die Spielentwicklung mit  
365 Unity oder erst als dadurch erworben?

366

367 **B:** [0:25:17] Ich glaube es ist ein Prozess. Ich glaube man fängt an mit dem.  
368 Also ich würde überhaupt behaupten, ist es nicht so, dass wenn man Software  
369 schreibt, dass man meistens mit irgendeiner Vorstellung herangeht, die sich  
370 dann irgendwie im Prozess des Schreibens dann auch revidiert und dann hat  
371 man irgendwie eine andere, also man lernt, also mir geht es so, dass ich lerne  
372 während ich programmiere, wie das dann geht und wie ich das wirklich nachher  
373 machen muss. Ich würde behaupten, man geht mit einer Vorstellung ran an das  
374 Problem und dann versucht man das mit den Möglichkeiten, die man hat  
375 umzusetzen. Und im Zuge dessen lernt man aber auch wieder neue Sachen  
376 dazu, wie man das vielleicht machen kann, oder vielleicht lernt man überhaupt  
377 etwas anderes, auf das man draufkommt währenddessen.

378

379 **I:** [0:26:11] Und damit man effektiv mit Unity Spiele entwickeln kann, ist es da  
380 eine Voraussetzung oder ist es auch absolut sinnvoll, wenn man es erst in der  
381 Spielentwicklung mit Unity erwirbt, die Fertigkeit Aufgaben mit Mitteln der  
382 Informatik zu modellieren?

383

384 **B:** [0:26:37] Naja, es ist insofern ein Voraussetzung, als dass es eben echt gut  
385 ist, das habe ich gemerkt mit dem Projekt eben, dass man weiß, wie das  
386 Dateisystem auf Windows zum Beispiel funktioniert, oder wie eben  
387 Grafikformate eingebettet werden können in das Spiel usw. Also das würde ich  
388 als Voraussetzung sehen, damit man lernen kann mit Unity.

389

390 **I:** [0:26:54] Das jetzt schon, aber ich meine jetzt konkret, dass man etwas  
391 modellieren kann mit Mitteln der Informatik.

392

393 **B:** [0:27:06] Nein, das muss man vielleicht nicht als Voraussetzung sehen, weil  
394 dafür ist ja gerade dieser Editor nicht schlecht, dass man irgendwo einmal so  
395 ein Game, also Spielszenen modelliert eben, so dass man das lernen kann,  
396 auch irgendwie ganz gut wie man so etwas modelliert etwa.

397

398 **I:** [0:27:23] Ok. Und können die SchülerInnen nach der Spielentwicklung mit  
399 Unity Daten-, Klassen- und Objektstrukturen in einem Algorithmus verwenden?

400

401 **B:** [0:27:41] Das ist jetzt dieselbe, also ich kann Ihnen dieselbe Antwort geben,  
402 wie vorhin. Ich bin der Überzeugung, dass es nicht vom Tool abhängt, sondern  
403 wie motiviert man ist, sich mit dem Tool zu beschäftigen. Also wenn Schüler  
404 eben motiviert sind, Unity zu lernen und das machen wollen, dann ja, dann  
405 werden sie mehr auf das kommen, wie wichtig das ist und werden die  
406 Zusammenhänge klar. Einfach nur durch die Verwendung des Tools, nein. Aber  
407 das würde ich auch jeder anderen Programmierumgebung unterstellen. Wenn  
408 ich nur abschreibe was der Lehrer oder die Lehrerin vorgibt, dann werde ich  
409 nicht mitnehmen was das bedeutet, was ich da schreibe.

410

411 **I:** [0:28:24] Ich würde fast sagen, gehen wir wirklich, spezialisieren wir uns auf  
412 diejenigen, die das interessiert, weil ich, wenn ich so daran denke, es ist  
413 eigentlich in jedem Fach so, wenn man sich nicht dafür interessiert, kann man  
414 Dinge auch recht simpel ohne ein Verständnis dafür, würde ich sagen  
415 hinbekommen irgendwie, aber dass man tatsächlich da in die Tiefe geht, oder  
416 Dinge mitnimmt, wie Kompetenzen oder so, das ist ja eigentlich in jedem Fach  
417 nur für die Interessierten, oder? Kann man das so sagen?

418

419 **B:** [0:29:09] Ich zögere gerade so, da so direkt zuzustimmen, weil ich glaube,  
420 dass gerade beim Programmieren so etwas ganz wichtig ist, dass man eben  
421 wirklich versucht, sich selber damit auseinanderzusetzen.

422

423 **I:** [0:29:22] Ah ok.

424

425 **B:** [0:29:24] Also es ist eben was, dass man nicht auswendig lernen kann. Also  
426 in anderen Fächern, man kann in Geschichte und in Psychologie, also ich weiß  
427 jetzt einfach in Psychologie und ich unterstelle es jetzt einfach Mal der

428 Geschichte manches auswendig lernen und dann weiß man auch ein bisschen  
429 was, unabhängig davon, ob man ein Interesse dafür hat oder nicht, aber man  
430 kann viele Sachen auswendig lernen und dann ist das so. Das ist aber beim  
431 Programmieren nicht der Fall. Also man kann zwar auch irgendetwas lernen,  
432 also was Datentypen sind und wie die heißen und so, aber was es bedeutet und  
433 wie man die jetzt wirklich verwendet und so, das ist etwas, das lernt man erst im  
434 Machen, also das ist eben was Praktisches. Man muss programmieren glaube  
435 ich, damit man das hineinbekommt, was das jetzt bedeutet, wenn ich eine  
436 Klasse habe oder, weil nur rein diese abstrakte Erklärung, das bringt dann  
437 eigentlich nur mittelmäßig etwas. Aber wenn wir von den interessierten  
438 Schülerinnen und Schülern ausgehen, dann würde ich auch sagen, Unity ist  
439 bedingt ein gutes Tool, um wirklich objektorientiertes Programmieren zu lernen.  
440 Also es ist ganz ok würde ich sagen, aber Unity hat seine Eigenheiten habe ich  
441 den Eindruck. Also ich finde nicht, dass es besonders intuitiv ist, sondern man  
442 muss eben viel in der API nachschauen, wie was funktioniert und Unity hat  
443 eigene Konventionen, die man lernen muss. Also es ist nicht etwas, was C#-  
444 Konvention ist, sondern es gibt Unity-Sachen, wie zum Beispiel, dass Dateien  
445 genau so heißen müssen wie die Klassen, weil sonst gibt es Schwierigkeiten  
446 und so weiter. Also ich meine das ist eh eine gute Konvention, aber es gibt halt  
447 Konventionen, an die man sich halten muss, die man lernen muss, die man  
448 irgendwo lesen muss. Also wenn es nur darum geht, dass man das Konzept  
449 vom objektorientierten Programmieren lernt, würde ich sagen, ja klar kann man  
450 die lernen, aber ich würde sagen, es ist jetzt nicht in irgendeiner Form leichter  
451 als in irgendeiner anderen Programmierumgebung.

452

453 **I:** [0:31:20] Sind die SchülerInnen nach der Spielentwicklung mit Unity fähig, ein  
454 Softwareprojekt zu planen und durchzuführen, oder sogar zu reflektieren?

455

456 **B:** [0:31:30] Offensichtlich. Also in meinen Klassen haben sie eindeutig Projekte  
457 gemacht und sie haben, und zwar das ist das Spannende, sie sind, ich würde  
458 jetzt einmal sagen, der Großteil der Schüler war wirklich frustriert mit diesem

459 Unity, weil das wirklich für viele überfordernd war und sie haben sich nicht  
460 ausgekannt usw. und haben sich dann aber durchgeboxt sozusagen durch  
461 diese Pflichtstunden in Unity. Aber nachher, wie sie das selbstgewählte Projekt  
462 gemacht haben, haben sie gesagt, also ich glaube drei Projekte sind nichts  
463 geworden von allen sechs Gruppen, das heißt fast alle Projekte sind etwas  
464 geworden und eigentlich alle Projekte, die etwas geworden sind, waren alle  
465 wirklich interessant und super. Also ich würde behaupten, also das war für mich  
466 so etwas wo ich mir gedacht habe, das ist das wo glaube ich die größte  
467 Lernsituation geschaffen wurde, weil so dieses Umgehen mit Frustration und  
468 auch irgendwie so diese, also ich weiß nicht, also sie haben nachher auf jeden  
469 Fall die Projekte wirklich verantwortungsvoll selbst ausgearbeitet. Und zwar  
470 manche in Unity, manche in anderen Sachen, aber es waren alle relativ  
471 aufwendig und sie haben wirklich eine Zeit investiert in diese anderen Projekte.  
472 Also das war cool. Also das würde ich behaupten, das haben sie gelernt.

473

474 **I:** [0:33:03] Also, dass man das wirklich als Softwareprojekte bezeichnet und  
475 dass sie das geplant und durchgeführt haben?

476

477 **B:** [0:33:10] Ich würde sagen Projekte, es waren nicht unbedingt  
478 Softwareprojekte, aber es waren Projekte. Und sie haben das ganz gut  
479 hinbekommen dann, ja.

480

481 **I:** [0:33:17] Und auch zu reflektieren? Das sie wirklich überlegen, was war gut,  
482 was war schlecht daran? Das bewerten können usw.?

483

484 **B:** [0:33:27] Also ich finde das ist eine super spannende Frage, mir kommt nur  
485 gerade, dass das irgendwie nichts mit Unity zu tun hat, sondern es hat einfach  
486 etwas zu tun mit wie Unterricht gestaltet wird, oder? Also wenn man die Zeit  
487 gibt, dass die Schülerinnen und Schüler das Reflektieren lernen können. Also  
488 von sich aus, einfach nur durch Unity verwenden, lernt man wenig was so  
489 etwas angeht, aber wenn man die Zeit hat und wenn das irgendwie klar ist,

490 dass es darum auch geht, dann kann man das schon lernen. Ich würde jetzt  
491 nicht sagen, dass das alle gemacht haben oder das jetzt alle reflektiert haben  
492 und dass sie die Projekte alle reflektiert hätten und so, aber sie haben alle  
493 einen Bericht abgegeben, wo drinnen gestanden ist, was sie gemacht haben.  
494 Also das könnte man sagen ist eine Reflexion in irgendeiner Form, weil es eben  
495 auch von mir verlangt wurde. Ich glaube das muss man schon auch so, also  
496 den Unterricht auch dementsprechend gestalten. Also wenn einem das wichtig  
497 ist, dass sie dann auch reflektieren können, dann muss man den Unterricht so  
498 gestalten, in welcher Form auch immer, dass sie reflektieren können auch. Das  
499 hat aber nichts mit Unity zu tun.

500

501 **I:** [0:34:40] Und das ist jetzt, wenn man jetzt die Unity-Tutorials und so  
502 hernimmt, lernen sie da das auch schon oder ist es wirklich eher danach, bei  
503 den eigenen Projekten?

504

505 **B:** [0:34:57] Ja, also das lernen sie nicht durch die Tutorials. Die Tutorials sind  
506 sehr fachlich toolorientiert und programmierlernorientiert, würde ich behaupten.  
507 Also so habe ich das erlebt, also so habe ich es auch mitgeplant eigentlich.

508

509 **I:** [0:35:20] Lernen die SchülerInnen die Angemessenheit von  
510 Entwicklungswerkzeugen grob einzuschätzen?

511

512 **B:** [0:35:25] Meine Schüler sicher nicht, weil die haben keine Ahnung gehabt,  
513 von unterschiedlichen Entwicklungswerkzeugen. Ich glaube, dass kann man  
514 erst dann, wenn man mehrere Entwicklungswerkzeuge kennt und auch versteht  
515 worum es geht, dann kann man sie einschätzen. Vorher kann man einfach nur  
516 einmal sagen, ja ok, wir könnten das, also der Lehrer hat gesagt wir müssen  
517 Unity machen, also machen wir Unity, der Lehrer hat gesagt wir machen  
518 Scratch, also machen wir Scratch, gut passt, es ist lustig, es ist nicht so lustig.  
519 Also, dass sie irgendwie einschätzen können, wofür sie jetzt was brauchen  
520 können und ob sie jetzt lieber Unity verwenden oder Cocos2D oder doch lieber

521 in Python programmieren, da muss man sich mit dem allen ein bisschen  
522 auseinandergesetzt haben, vorher kann man das nicht wirklich einschätzen,  
523 würde ich behaupten.

524

525 **I:** [0:36:05] Ja. Entdecken die SchülerInnen Gemeinsamkeiten und Regeln, also  
526 Muster in den Handlungsanleitungen bzw. Algorithmen?

527

528 **B:** [0:36:16] Die, die sich damit auseinandersetzen wahrscheinlich, aber ich  
529 habe jetzt keine Beispiele oder so, aber die wahrscheinlich und die anderen  
530 nicht.

531

532 **I:** [0:36:28] Lässt sich das irgendwo vielleicht trotzdem konkretisieren, fällt Ihnen  
533 da nichts ein, vielleicht?

534

535 **B:** [0:36:35] Mm.

536

537 **I:** [0:36:36] Ok. Können die SchülerInnen durch die Spielentwicklung mit Unity  
538 die Bedeutung von Algorithmen erkennen?

539

540 **B:** [0:36:51] Ja, das schon.

541

542 **I:** [0:36:55] Ok und wo sieht man das?

543

544 **B:** [0:36:59] Naja, dadurch dass eben auch Schülerinnen und Schüler nach  
545 dem Projekt im zweiten Durchlauf auch Algorithmen ausgearbeitet haben, die  
546 nicht nur in Unity waren. Eine Gruppe hat einen eigenen Dating-Algorithmus  
547 selbst entwickelt. Eine andere Gruppe hat den Dijkstra-Algorithmus sich genau  
548 angeschaut und ausgearbeitet in einer quasi wirklich Unterrichtsstunde könnte  
549 man fast sagen und vorgestellt. Also das würde ich behaupten, dass haben sie  
550 mitbekommen durch das was wir da gemacht haben.

551

552 **I:** [0:37:33] Und können sie dann auch die Effizienz von einem Algorithmus  
553 bewerten?

554

555 **B:** [0:37:37] Nein. Das ist wieder etwas, wo ich sagen würde, da muss man den  
556 Unterricht dementsprechend ausrichten. Also, wenn einem das wichtig ist, dass  
557 die Schüler in der AHS fünften Klasse die Effizienz eines Algorithmus  
558 bestimmen können, dann kann man natürlich dementsprechend Beispiele  
559 machen. Ich würde auch sagen, dass da Unity ungeeignet ist dafür. Das sollte  
560 man in einer Programmierumgebung, oder ich würde überhaupt sagen, für  
561 diese ganzen Programmiergeschichten, also für das wie jetzt Programmieren  
562 funktioniert per se, ist Unity sekundär ein gutes Tool. Es ist ein super Tool zum  
563 Spiele-programmieren, dafür ist es gemacht und dafür ist es auch super. Und  
564 man kann natürlich alle möglichen Algorithmen in Unity implementieren und  
565 man kann natürlich auch das mit der Effizienz in Unity darstellen. Da ist es  
566 sicher nicht das einfachste Tool dafür und es ist sicher auch jetzt nicht, also die  
567 Effizienz von Algorithmen zu bewerten oder das in irgendeiner Form das zu  
568 thematisieren überhaupt, ist auch die Entscheidung der Lehrperson, ob das das  
569 ist, was man in der fünften Klasse Informatik Pflichtfach machen will. Also ich  
570 würde es anders formulieren, also was zum Beispiel ein Lerneffekt war, also ich  
571 habe nachher so Reflexionsrunden gemacht, nach diesem Unity mit meinen  
572 Schülerinnen und Schülern und eine Schülerin hat in einer von diesen Runden  
573 aber gesagt, ich habe es irgendwie super gefunden, weil jetzt verstehe ich was  
574 ein „Glitch“ bedeutet bei einem Computerspiel und jetzt weiß ich, dass das ein  
575 Bug ist und jetzt weiß ich dass das ur viele Stunden dauert, dass man so etwas  
576 überhaupt schreiben kann. Also das waren die Erfahrungen die SchülerInnen  
577 gehabt haben. Also, das hat zwar nichts mit der Effizienz von Algorithmen zu  
578 tun, aber es hat etwas zu tun mit dem was es bedeutet einen Algorithmus zu  
579 schreiben und was ein Fehler im Code ist.

580

581 **I:** [0:39:54] Ok. Dann kommen wir jetzt zum zweiten Cluster. Der ist jetzt  
582 deutlich kürzer. Bei SchülerInnen treten bei einem neuen Lehrstoff immer



583 wieder einmal Schwierigkeiten auf. Erinnern Sie sich einmal an solche  
584 Situationen. Was waren da die Schwierigkeiten?

585

586 **B:** [0:40:12] Bei Unity?

587

588 **I:** [0:40:14] Ja.

589

590 **B:** [0:40:16] Vielfältig. Die Schwierigkeiten waren vor allem, also es waren, die  
591 SchülerInnen haben immer wieder rückgemeldet, es war eine hochkomplexe  
592 Benutzerumgebung. Also sie haben sich nicht ausgekannt mit den vielen  
593 Einstellungen, die es gibt. Ganz schwierig war das irgendein Häkchen irgendwo  
594 setzen gleich irgendwas bedeutet und gleich irgendwas schrotten kann, ohne  
595 dass man es wieder rückgängig machen kann, gescheit, oder ohne, dass sie  
596 gewusst haben, wie man es rückgängig machen kann, gescheit. Sie haben  
597 eben, wie schon erwähnt ganz viel Schwierigkeiten gehabt mit dem wie man  
598 Dateien speichert und wo man Dateien findet und wo man irgendetwas hin lädt  
599 usw. Sie haben nicht genau gewusst, wie wir nachher programmiert haben, was  
600 war für sie dieses Grundtemplate von C#, wo eben die Klassendefinitionen  
601 drinnen stehen und alles am Anfang unverständlich und insofern auch  
602 überfordernd. Solche Schwierigkeiten gab es.

603

604 **I:** [0:41:13] Und was müsste den SchülerInnen fehlen, damit die Situation noch  
605 schlimmer wird?

606

607 **B:** [0:41:18] In erster Linie die Lehrperson und in zweiter Linie die Unterlagen,  
608 die irgendwie erklärend sind. Also ohne API, ohne Tutorial, ohne  
609 Lehrer/Lehrerin, das wäre nachher ganz, ein sehr mutiger Zugang zu Unity und  
610 Programmieren.

611

612 **I:** [0:41:44] Ok und wenn man das jetzt in Richtung digitaler Kompetenzen mehr  
613 drehen, was müsste da den SchülerInnen fehlen?

614

615 **B:** [0:41:58] Naja eh, das sind eh digitale Kompetenzen, so Umgang mit  
616 Computer, das man irgendwie weiß, wie das Ding einigermaßen zu bedienen  
617 ist. Was müsste ihnen fehlen? Lesefähigkeit, noch mehr Lesefähigkeit. Und  
618 Lesefähigkeit ist etwas, das sehr nahe ist bei den Grundkompetenzen, was  
619 überhaupt Lernen und Leben und so angeht. Und ich glaube diese  
620 Grundkompetenz ist gerade beim Programmieren eine ganz eine wichtige. Und  
621 ich glaube, das ist eine die sehr stark in das Gewicht fällt beim Programmieren,  
622 weil es schwierig ist für Schüler und Schülerinnen, also nicht nur Lesen im  
623 Sinne von API lesen, aber ich meine wir haben es mit Programmier-"Sprachen"  
624 zu tun, also insofern ist das immer etwas, das eigentlich etwas mit Sprachen-  
625 lernen zu tun hat. Das ist sicher eine Fähigkeit, also wenn sie da irgendwo  
626 weniger haben, tun sie sicher noch schwerer. Also Lesen, Sprachverständnis.  
627 Und damit zusammenhängend, logisches Denken.

628

629 **I:** [0:43:09] Ok. Und was würde man zur Bewältigung dieser Schwierigkeiten  
630 benötigen?

631

632 **B:** [0:43:17] Hahaha. Also in der Volksschule anfangen damit, Bücher zu lesen.  
633 Die Handys am besten in die Ecke werfen und hin und wieder Bücher lesen,  
634 diese besprechen mit irgendwelchen Erziehungsberechtigten. In der Unterstufe  
635 den Fokus darauflegen, dass die Leute Lesen, Schreiben, Rechnen können.  
636 Vor allem eben auch Rechnen und vor allem eben auch bei Mathematik die  
637 Angaben verstehen, also das man eben Textbeispiele irgendwie versucht  
638 nachzuvollziehen und dass das ein ganz ein wichtiger Beitrag ist. Und wenn da  
639 eine gute Grundlage geschaffen wird, also dass Leute Lesen können, also auch  
640 mehrere Sätze sinnerfassend lesen, möglicherweise sogar auf Englisch, dann  
641 ist Programmieren ein guter nächster Schritt.

642

643 **I:** [0:44:13] Wie könnte man die Schwierigkeiten bewältigen für die  
644 Spieleentwicklung mit Unity?

645

646 **B:** [0:44:28] Man kann Schritte setzen in diese Richtung. Also bewältigen? Also  
647 ich habe hoffentlich jetzt aufgezeigt, dass ich das ein bisschen, also den  
648 Kontext ein bisschen weiter fasse als nur den Informatikunterricht der fünften  
649 Klasse, aber im Unterricht selber kann man eben genau diese Fähigkeiten  
650 fördern, indem man liest, indem man ihnen manchmal Aufgaben gibt, die im  
651 Rahmen, im Sinne von, also Unity ist eben sehr, sehr, da kann man sehr viel  
652 auf einmal machen und ich bin dann nach Unity, also nach diesem Projekt  
653 wieder übergegangen, mit Tools zu arbeiten, die jetzt nicht so viel bieten, oder  
654 die ein bisschen intuitiver funktionieren, einerseits weil man dann glaube ich  
655 sich so annähern kann an diese Grundkompetenzen, dass man das lernt mehr  
656 und andererseits glaube ich eben auch dass es ganz wichtig ist, gerade in  
657 Informatik, wenn man es nur in der AHS in einer Klasse erlebt, diese Inspiration  
658 und die Motivation aufrecht zu erhalten. Und ich habe schon gemerkt, dass  
659 Unity für ganz Vieles sehr demotivierend letzten Endes ist, obwohl es eine  
660 coole Idee war.

661

662 **I:** [0:45:45] Was hatten die SchülerInnen an sich, die nicht auf diese  
663 Schwierigkeiten gestoßen sind, mit der Spieleentwicklungsumgebung Unity?

664

665 **B:** [0:45:57] Es gab niemanden in meinen Klassen, die nicht irgendwo  
666 Schwierigkeiten hatten, aber die Motivation war von den Schülern teilweise  
667 unterschiedlich. Also die Leute, die von vorneherein irgendwie gesagt haben,  
668 sie möchten lernen Spiele zu programmieren und sind interessiert an Informatik  
669 und das hat sie schon immer interessiert, das waren die Leute, die mit diesen  
670 ganzen Frustrationen die Unity mit sich brachte, die damit sich irgendwie  
671 arrangiert haben. Das war irgendwie sehr herausstechend. Also die Leute, die  
672 einfach wirklich Interesse gehabt haben, die haben das auch gemacht. Für die  
673 anderen war es eher schwierig, die haben dann auch weniger gemacht und  
674 haben sich auch weniger motiviert und haben eher gesagt, sie fühlen sich da

675 irgendwie überfordert oder sie haben es nicht ordentlich gelernt, oder was weiß  
676 ich.

677

678 **I:** [0:47:08] Angenommen morgen beginnt ein neues Schuljahr und Ihre  
679 SchülerInnen sind bestens gerüstet um direkt mit der Spielprogrammierung mit  
680 Unity zu beginnen. Woran erkennen Sie dieses optimale Rüstzeug?

681

682 **B:** [0:47:23] Ok. In erster Linie daran, dass sie echt gerne Spiele  
683 programmieren wollen. Also, sie wollen Spiele programmieren. Das ist was sie  
684 total beschäftigt, das ist wo sie Spaß daran haben. Und das wäre das optimale  
685 Rüstzeug. Sie müssten wahrscheinlich gar nicht einmal gut programmieren  
686 können. Ich glaube sie müssten nur diesen Wunsch haben, ich mag Spiele  
687 programmieren. Das wäre das optimale Rüstzeug.

688

689 **I:** [0:47:57] Ok. Und jetzt eine hypothetische Frage. Was wäre, wenn Sie die  
690 Zeit für Sie und ihre jeweilige Klasse anhalten könnten, was würden Sie dabei  
691 den SchülerInnen vor der Spielentwicklung mit Unity beibringen wollen?

692

693 **B:** [0:48:17] Den SchülerInnen, die ich jetzt gehabt habe mit Unity?

694

695 **I:** [0:48:21] Zum Beispiel, ja.

696

697 **B:** [0:48:22] Ja eh die Sachen, von denen ich geredet habe, also eben  
698 Lesefähigkeit trainieren. In gewisser Hinsicht glaube ich, würde ich ganz vielen,  
699 ganz starken Wert darauf legen, ihnen den Wert von der Informatik in unserer  
700 heutigen Gesellschaft näher zu bringen, also dass ihnen das sehr klar ist, dass  
701 wir in einer hochindustrialisierten Gesellschaft leben, wo gerade die Informatik  
702 ein wesentlicher technologischer Stützpfeiler ist momentan. So dass diese  
703 Motivation da mehr gefördert werden könnte, dass man sich mit so etwas  
704 auseinandersetzen kann und dass es durchaus sinnvoll ist, auch wenn man  
705 denkt, man will vielleicht Rockstar oder man will Medizin studieren, oder so,

706 oder Rockstar werden, oder irgendwie so. Und dass es trotzdem nicht schlecht  
707 ist, wenn man eine Ahnung hat von Informatik, weil wer weiß was man eben  
708 tatsächlich dann studieren wird. Also das wäre glaube ich etwas, wo ich  
709 dahinter wäre. Ja und vom Fachlichen her natürlich eben so etwas wie  
710 Filemanagement würde ich mit ihnen öfter üben, so etwas wie dass man mit  
711 Grafiken ganz viel vorher arbeiten, also dass wir wirklich lernen Gimp gut zu  
712 verwenden, damit sie schöne Grafiken machen können, damit sie so ein wenig  
713 lernen wie diese Layering-Geschichte funktioniert, weil die brauchen sie dann in  
714 Unity auch wieder. Dass sie lernen, dass man vielleicht eben vorher so in  
715 Scratch länger Zeit verbringt, um diesen objektorientierten Zugang und dieses  
716 Algorithmische mehr hineinbekommt, damit es danach vielleicht hoffentlich  
717 weniger Schwierigkeiten macht, wenn man nachher dieses Textbasierte liest.  
718 Das waren wahrscheinlich Sachen, die gerade spannend waren.

719

720 **I:** [0:50:25] Mhm. Und jetzt noch fünf so Bereichsfragen. Benötigen  
721 SchülerInnen Ihrer Ansicht nach für die Spielentwicklung mit Unity Kenntnisse  
722 und Fertigkeiten aus dem Bereich Verantwortung, Datenschutz und  
723 Datensicherheit?

724

725 **B:** [0:50:47] Ja sicher, wenn sie Grafiken verwenden, die urheberrechtlich  
726 geschützt sind und so weiter, Musik usw. Brauchen sie. (...) Ich bin auf das  
727 nicht so eingegangen, weil ich glaube, das ist einfach das Einfachste irgendwie  
728 zum Lernen, von den ganzen Sachen, die es bei Unity zum Lernen gibt.

729

730 **I:** [0:51:14] Ok. Also Sie würden sagen, es ist keine Voraussetzung, kann man  
731 durchaus mit Unity erlernen?

732

733 **B:** [0:51:20] Ja. Achso, ja. Ich meine es ist nicht schlecht, wenn man es vorher  
734 weiß, weil dann weiß man welche Grafiken man aussucht usw. Also insofern  
735 hat es jetzt wieder nichts per se mit Unity zu tun eigentlich, aber es ist nicht  
736 schlecht, wenn man es vorher weiß.

737

738 **I:** [0:51:40] Und damit man effektiv mit Unity Spiele programmiert, ist es da gut,  
739 wenn man es vorher weiß oder nicht?

740

741 **B:** [0:51:50] Damit man effektiv Spiele programmiert mit Unity ist es egal. Aber  
742 wenn man jetzt Grafiken verwenden will, die im Spiel vorkommen, die nicht  
743 selber gemacht worden sind, ist es sehr wichtig.

744

745 **I:** [0:52:08] Ok. Wie sieht das Ganze bezüglich Kenntnisse und Fertigkeiten im  
746 Bereich der technischen Grundlagen und Funktionsweisen aus?

747

748 **B:** [0:52:14] Was ist technische Grundlagen und Funktionsweisen?

749

750 **I:** [0:52:17] Wie zum Beispiel der Computer funktioniert. Peripheriegeräte. Wie  
751 diese zusammen funktionieren usw.

752

753 **B:** [0:52:26] Das ist eigentlich für das Spiele-programmieren per se, für das  
754 Anfangen auf jeden Fall ziemlich egal, glaube ich. Also wenn man nachher  
755 einmal sich beschäftigt damit, dass man irgendeine Virtual Reality für Oculus  
756 Rift schreiben will, dann ist es wahrscheinlich spannend, aber für des wo man  
757 anfängt mit Unity ein wenig herum zu tun, ist das uninteressant.

758

759 **I:** [0:52:49] Mhm. Dann im Bereich der Betriebssysteme und Software?

760

761 **B:** [0:52:54] Ganz wichtig. Eben wie Dateimanagement funktioniert, wie  
762 Dateistrukturen usw. Das hatte ich eh schon vorher öfter gesagt.

763

764 **I:** [0:53:10] Fallen da vielleicht auch Dinge hinein wie Datensicherung,  
765 Datenwiederherstellung?

766

767 **B:** [0:53:14] Das habe ich glaube ich eh auch gesagt, also wo man etwas  
768 speichert, wo man etwas wiederfindet. Genau, also damit meine ich  
769 Datensicherung und Datenwiederherstellung.

770

771 **I:** [0:53:20] Ok. Aus dem Bereich Produktion digitaler Medien? Text, Bild, Ton,  
772 die Bearbeitung, die Produktion davon.

773

774 **B:** [0:53:27] Also, das habe ich eh vorher auch schon angeschnitten, wenn man  
775 mehr Zeit hätte, dann würde es natürlich mehr Spaß machen, wenn Leute, oder  
776 ich glaube es würde den Leuten auch mehr Spaß machen, wenn sie mehr  
777 Erfahrung haben mit dem, wie sie quasi selber Medien gestalten. Weil mehr  
778 individualisiert ist und so. Aber ist es eine dringende Voraussetzung? Nein. Weil  
779 man kann irgendwelche Grafiken nehmen von irgendwo.

780

781 **I:** [0:53:47] Ok. Und zuletzt noch im Bereich Suche, Auswahl und Organisation  
782 von Information?

783

784 **B:** [0:53:59] Suche, Auswahl oder. Ja, man lernt es glaube ich „on-the-way“,  
785 aber es ist auf jeden Fall nicht schlecht, wenn man schon vorher weiß, wie man  
786 DuckDuckGo einsetzen kann und wie man einen Finder oder Explorer so  
787 einsetzen kann, dass man irgendwelche Dateien sucht und so, das ist sicher  
788 spannend und sicher hilfreich. Man wird es notgedrungen lernen, wenn Unity  
789 verwendet wird, solche Sachen.

790

791 **I:** [0:54:32] Ok. Gut, dann sind wir am Ende des Interviews angelangt. Gibt es  
792 noch irgendetwas, das sie loswerden oder ergänzen möchten?

793

794 **B:** [0:54:41] Ja, mich würde natürlich Ihre Sichtweise interessieren.

795 [Der Rest des Interviews beinhaltet nichts Interessantes für diese Studie]

796