



universität  
wien

# Diplomarbeit / Diploma Thesis

Titel der Diplomarbeit / Title of the Diploma Thesis

„Digital Game-Based Learning in  
Programmierlernspielen“

Verfasst von / submitted by

Andreas Metz

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

Magister der Naturwissenschaften (Mag. rer. nat.)

Wien, 2016 / Vienna, 2016

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

A 190 884 299

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

Lehramtstudium;  
UF Informatik und Informatikmanagement  
UF Psychologie und Philosophie

Betreut von / Supervisor:

ao. Univ.-Prof. Dipl.-Ing. Dr. Renate Motschnig



## **Vorwort**

Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Alle Personenbezeichnungen gelten gleichermaßen für beiderlei Geschlecht sofern dies nicht dezidiert anders angegeben wird.

## **Danksagung**

Ich möchte mich an dieser Stelle zunächst bei meiner Betreuerin Frau ao. Univ.-Prof. Dipl.-Ing. Dr. Renate Motschnig für die Betreuung und die Ermöglichung dieser Arbeit bedanken. Weiters gilt mein Dank für das geduldige Korrekturlesen und den Verzicht auf den gemeinsamen Urlaub zugunsten der Diplomarbeit meiner Freundin Carina Lauter. Außerdem möchte ich mich bei meinen Eltern bedanken, die mein Studium finanziell überhaupt erst ermöglicht haben. Zuletzt möchte ich mich auch noch bei meinem Studiengefährten Andreas Plank, dessen Verständnis der hohen Kunst der Mathematik mich bis heute in Ehrfurcht und Staunen zu versetzen weiß, für die gemeinsame Zeit bedanken.



# Inhaltsverzeichnis

|       |                                                                |    |
|-------|----------------------------------------------------------------|----|
| 1     | Abstract                                                       | 3  |
| 2     | Einleitung                                                     | 4  |
| 2.1   | Thema                                                          | 4  |
| 2.2   | Zielsetzung                                                    | 4  |
| 2.3   | Verwendete Methoden                                            | 4  |
| 3     | Aktuelle didaktische & psychologische Konzepte                 | 5  |
| 3.1   | Grundlegende Strömungen                                        | 5  |
| 3.1.1 | Behaviorismus                                                  | 5  |
| 3.1.2 | Kognitivismus                                                  | 8  |
| 3.1.3 | Konstruktivismus                                               | 9  |
| 3.1.4 | Humanismus                                                     | 11 |
| 3.2   | Aufmerksamkeit                                                 | 13 |
| 3.2.1 | Aktivierungstheorien                                           | 14 |
| 3.2.2 | Selektive Aufmerksamkeit                                       | 19 |
| 3.3   | Motivation                                                     | 21 |
| 3.4   | Flow Erlebnis                                                  | 26 |
| 4     | Digital game-based Learning                                    | 28 |
| 4.1   | Begriffsdefinition                                             | 28 |
| 4.2   | Kritik am digital game-based Learnings                         | 30 |
| 4.3   | Vorteile des digital game-based Learnings                      | 33 |
| 4.3.1 | Steigerung der Motivation                                      | 33 |
| 4.3.2 | Verbesserung der Problemlösestrategie                          | 34 |
| 4.4   | Eignung von Spielen zur Verwendung im Unterricht               | 35 |
| 4.4.1 | Probleme bei der Verwendung im Unterricht                      | 38 |
| 4.5   | Praktische Beispiele zum digital game-based Learning           | 40 |
| 4.5.1 | Code Combat                                                    | 40 |
| 4.5.2 | Civilization                                                   | 44 |
| 4.5.3 | Comber & Motschnigs Studie zur Entwicklung von Computerspielen | 46 |
| 5     | Grundlagen - Gamedesign                                        | 51 |
| 5.1   | Wichtige Elemente eines Computerspiels                         | 51 |
| 5.2   | Psychologische Phänomene beim Spielen                          | 54 |

|       |                                         |     |
|-------|-----------------------------------------|-----|
| 5.3   | Wie Spiele den Spieler motivieren       | 55  |
| 6     | Beschreibung des erstellten Lernspieles | 58  |
| 6.1   | Lehrplanbezug                           | 58  |
| 6.2   | Bezug zu Lerntheorien                   | 59  |
| 6.3   | Realisierung & Implementierung          | 60  |
| 6.4   | Beschreibung des Spieles                | 61  |
| 6.4.1 | Verwendung im Unterricht                | 61  |
| 6.4.2 | Level 1                                 | 62  |
| 6.4.3 | Level 2                                 | 74  |
| 6.5   | Spieldesign                             | 86  |
| 7     | Qualitative Auswertung                  | 89  |
| 7.1   | Forschungsdesign                        | 89  |
| 7.2   | Ergebnisse                              | 91  |
| 7.3   | Interpretation der Ergebnisse           | 99  |
| 7.4   | Ausblick                                | 100 |
| 8     | Fazit                                   | 101 |

# **1 Abstract**

## **Deutsch**

Digital game-based Learning bekommt im Schulkontext in jüngster Zeit wieder mehr Beachtung. Befürworter dieser Form des Lernens preisen diese als absolut zukunftsweisend, da sie positive Auswirkungen auf die Motivation und die Problemlösefähigkeit hat, während deren Gegner kritisieren, dass diese Lernform langfristig negative Auswirkungen auf die Lernfähigkeit der Schüler hat.

Im Zuge dieser Arbeit wurde ein digitales Programmierlernspiel selbst entwickelt, welches mittels eines Fragebogens qualitativ ausgewertet wurde. Dabei wurde untersucht, wie sich die Verwendung eines digitalen Lernspieles zum Erwerb der Programmiersprache C# im Programmierunterricht auf die Motivation der Schüler auswirkt.

## **English**

Recently digital game-based learning in a school context regained more attention. Proponents of this type of learning praise it as absolutely groundbreaking as it has a positive impact on the motivation and the ability to solve problems while their opponents criticize that this form of learning has long-term negative effects on the learning ability of students.

In the course of this work a proprietary digital learning game for programming was developed, which was qualitatively evaluated using a questionnaire. The study investigated how the use of a digital learning game affected the motivation of the students.

## **2 Einleitung**

### **2.1 Thema**

Digital game-based Learning, also das Lernen mithilfe von Computerspielen, ist derzeit ein brandheißes Thema in der Lernforschung. Befürworter des game-based Learning sprechen von einer bevorstehenden Revolution des Lernprozesses, während dessen Gegner davor warnen, dass die Schüler sich dadurch lediglich passives Wissen aneignen, welches im konkreten Anwendungsfall dann nicht aufgerufen werden kann (vgl. Prensky 2001; vgl. Okhan 2003; vgl. Gebel et al. 2005).

Natürlich ist Lernen mithilfe von Computerspielen vielen Vorurteilen ausgesetzt, da Computerspiele oft mit sinnloser Zeitvergeudung in Verbindung gebracht werden. Diese Arbeit versucht das Thema von beiden Seiten zu beleuchten und konzentriert sich dabei primär auf das Lernen einer Programmiersprache anhand eines den Schülern zur Verfügung gestellten selbstprogrammierten Computerspiels.

### **2.2 Zielsetzung**

Das Ziel dieser Arbeit ist es, die Auswirkungen des Einsatzes eines digitalen Lernspieles für die Programmiersprache C# auf die Schüler zu untersuchen. Dabei liegt der Fokus der Untersuchung primär auf einer qualitativen Evaluierung der Motivation der Schüler.

Es sollen folgende Fragen beantwortet werden:

Empfinden die Schüler das Lernspiel als wirksames Mittel, um etwas zu lernen?

Wie wirkt sich das Lernspiel auf das Interesse der Spieler aus?

### **2.3 Verwendete Methoden**

Um die Wirksamkeit eines Lernspieles für Programmierung zu evaluieren, soll im Rahmen dieser Arbeit ein Computerspiel entwickelt werden, anhand dessen die Programmiersprache C# geübt wird. Wichtig dabei ist, dass dem Einsatz des Programmierspieles eine herkömmliche Theorieeinheit vorausgeht, in der die



Grundprinzipien der Programmierung erläutert werden, sodass das erworbene Wissen im Lernspiel angewandt und geübt werden kann.

Die Grundlage zur Erstellung des Spiels soll eine ausführliche Literaturrecherche bilden, welche sich zunächst an allgemeinen didaktischen Konzepten orientiert, sich jedoch auch auf die Ergebnisse konkreter Implementationen digitaler game-basierter Spiele konzentriert.

Das entwickelte Spiel soll anschließend in einer ersten Klasse der HTL Spengergasse, also der 9. Schulstufe, im Programmierunterricht eingesetzt werden. Am Ende des Unterrichts findet eine qualitative Auswertung der Stunde anhand eines Fragebogens statt.

### **3 Aktuelle didaktische & psychologische Konzepte**

#### **3.1 Grundlegende Strömungen**

##### **3.1.1 Behaviorismus**

Der Behaviorismus ist eine Strömung, die versucht, den Lernprozess möglichst experimentell zu verifizieren. Daher beschränkt sie sich meist lediglich auf die Erklärung beobachtbarer Phänomene. Sie sieht ihr Hauptziel darin, Theorien zur Vorhersage bestimmter Reaktionen in einer gegebenen Situation bereitzustellen (vgl. Hubwieser 2007, S. 3). Gemeinhin wird Lernen im Behaviorismus als „Verhaltensänderung, die auf Erfahrung beruht“ (Lefrancois und Leppmann 1986, S. 30) definiert. Daraus kann man schließen, dass sich die Behavioristen eher mit dem Verhalten befassen haben, als mit einem kognitiven Lernprozess.

##### **Klassische Konditionierung**

Die Grundlagen für diese Strömung lieferte der berühmte russische Forscher Pawlow (1849-1936). Pawlow gilt heutzutage als Begründer der sogenannten klassischen Konditionierung. Pawlow gelang es bei (tierethisch nicht unbedenklichen) Versuchen mit Hunden, ihren Speichelfluss, der normalerweise lediglich durch das Darbieten von Futter angeregt wird, durch das Ertönen einer Glocke herbeizuführen. Er konnte somit beweisen, dass sich das natürliche Verhalten der Hunde durch häufige Wiederholung

Der Versuchsablauf lief folgendermaßen ab: Zunächst wurde einem Hund Futter dargeboten. Bei der Darbietung dieses Futters begannen die Hunde mehr Speichel abzusondern als sonst. Dieses Futter bezeichnete Pawlow als den unconditionierten Stimulus, welcher zum Speichelfluss, also der unconditionierten Reaktion führte (vgl. Trimmel 2011, S. 93).

Fleischpulver

Meßapparat für die Menge des Speichelflusses

Quelle: <http://paedpsych.jk.uni-linz.ac.at:4711/LEHRTEXTE/LERNEN/klassi.htm>

Diese künstlich herbeigeführte Verhaltensänderung kann jedoch auch sehr schnell wieder rückgängig gemacht werden. Bietet man einige Male den konditionierten Stimulus (=Glocke) ohne den unkonditionierten Reiz (=Futter) dar, so wird das erlernte Verhalten wieder vergessen und die Glocke wird wieder zum neutralen Stimulus (Trimmel 2011, S. 94–95). Pawlow spricht hierbei von der Löschung des Verhaltens.

Pawlows Experiment erlangte deshalb einen derart zentralen Status in der Psychologie, weil er dadurch beweisen konnte, dass es möglich ist, Tiere durch die Kombination bestimmter Stimuli dazu zu bringen, durch völlig abstrakte, unnatürliche Stimuli bestimmte Reaktionen zu vollziehen. Klassische Konditionierung ist also, wie bereits in der Definition des Behaviorismus erwähnt, eine Theorie zur Vorhersage des Verhaltens in bestimmten Situationen.

### **Operante Konditionierung**

Sowohl Thorndike als auch Skinner haben basierend auf den Experimenten von Pawlow weitere Versuche mit Katzen, Tauben und Ratten durchgeführt und sind zu dem Schluss gekommen, dass angenehme Reaktionen auf Verhalten zum häufigeren Auftreten des besagten Verhalten führen, während unangenehme Reaktionen auf Verhalten zu einem selteneren Auftreten des Verhaltens führen (vgl. Trimmel 2011, S. 102).

Während bei der klassischen Konditionierung ein bestimmtes Verhalten mit einem gänzlich neutralen Stimulus konditioniert wird, versuchen Skinner und Thorndike bestimmte Verhaltensweisen nicht zu koppeln, sondern nach der Ausführung eines bestimmten Verhaltens eine Belohnung oder eine Bestrafung dafür zu erteilen. Dabei treffen sie die Unterscheidung zwischen einerseits dem Operanten, also dem Verhalten, welcher verstärkt werden soll, und andererseits der Konsequenz, welche beispielsweise in der Form einer Belohnung oder einer Bestrafung für die Ausführung des Operanten verabreicht wird.

Dabei ist essentiell, dass die Konsequenzen in einem räumlich-zeitlichen Kontext zum Operanten stehen, da diese anderenfalls nicht mit dem Operanten assoziiert werden. Laut Trimmel ergibt sich daraus der Leitsatz: „Je unmittelbarer nach dem Operanten die Konsequenz liegt, umso stärker ist ihre Wirkung“ (Trimmel 2011, S. 102).

Nach Thorndike und Skinner gibt es vier Möglichkeiten, wie Konsequenzen zu einer Verhaltensänderung führen können:

|                    | <b>Angenehmer Reiz</b> | <b>Unangenehmer Reiz</b> |
|--------------------|------------------------|--------------------------|
| <b>Hinzugefügt</b> | Positive Verstärkung   | Bestrafung               |
| <b>Entfernt</b>    | Bestrafung             | Negative Verstärkung     |

**Abbildung 2 Verstärkung nach Thorndike & Skinner (Hubwieser 2007, S. 4)**

Hubwieser schreibt bezüglich dieser Möglichkeiten zur Verhaltensänderung folgendes:

*Wie empirische Versuche zeigten, ist Bestrafung weit weniger wirksam als Verstärkung. Erstere führt meist nur zu einer Unterdrückung des Verhaltens in Gegenwart des Bestrafenden. (Hubwieser 2007, S. 4)*

Er folgert daraus, dass es sinnvoll ist, Bestrafungen zu vermeiden und dementsgegenüber kontinuierlich, aber differenziert positiv zu verstärken (vgl. Hubwieser 2007, S. 4). Dies ist auch einer der Gründe, warum man sich von der sogenannten schwarzen Pädagogik, bei der Züchtigung noch an der Tagesordnung stand, mittlerweile weltweit klar distanziert hat. Im schulischen Kontext muss diesbezüglich die oberste Maxime in jedem Fall sein, Abwehrreaktionen und Angstreaktionen der Schüler zu vermeiden.

### **3.1.2 Kognitivismus**

Das Wort Kognition kommt vom lateinischen *cognito*, was so viel bedeutet wie Kenntnis, Erkenntnis oder Begriff, also zusammengefasst das, was wir als Wissen bezeichnen.

Der Kognitivismus kann als Gegenströmung zu den behavioristischen Theorien gesehen werden, der sich mehr für die „inneren“ Prozesse des Gehirns des Lernenden interessiert, als für die Beobachtung von seinem Verhalten.

Gemeinhin wird das Hixon Symposium im Jahre 1948 als Beginn der sogenannten Kognitiven Wende betrachtet. Bei diesem internationalen Symposium stellten die teilnehmenden Forscher fest, dass die mittlerweile etablierten Praktiken des Behaviorismus nicht ausreichen, um menschliches Verhalten hinreichend zu beschreiben (vgl. Trimmel 2013, S. 12). Während der Behaviorismus das Gehirn noch als Black Box versteht, versucht der Kognitivismus diese Black Box zu öffnen und zu beschreiben, welche Vorgänge darin passieren.

Laut Trimmel zählen für Kognitivisten unter anderem folgende Inhalte zu den zentralsten Themen des Lernens: Wahrnehmung, Aufmerksamkeit, interne Repräsentation von Wissen, Begriffsbildung, Problemlösung, Motivation und Emotion (vgl. Trimmel 2013, S. 9–10). Es geht also nicht mehr primär um Verhaltensänderung, sondern wie Wissen aufgenommen, gespeichert und wiedergegeben wird, vor allem aber, wie wir diese Informationen nutzen.

In dieser Arbeit kann selbstverständlich nicht auf alle diese Aspekte eingegangen werden. Da die Vorteile digital game-based Learnings vor allem in der Erhöhung der Aufmerksamkeit und der Motivation liegen, werden diese beiden Aspekte in Kapitel 3.2 und 3.3 etwas ausführlicher behandelt.

### **3.1.3 Konstruktivismus**

Gold formuliert eine knappe Definition des Konstruktivismus folgendermaßen:

*„Lernen wird aus konstruktivistischer Sicht stets als selbstgesteuerte Aktivität des Lernenden verstanden, in deren Verlauf Wissen nicht einfach erworben, sondern individuell (re)konstruiert bzw. sozial ausgehandelt wird.“* (Gold 2003, S. 101)

Auch konstruktivistische Lerntheorien betrachten den Lernprozess also als einen Prozess der Informationsverarbeitung. Allerdings lautet das Grundparadigma des konstruktivistischen Lernens, dass Wissen nicht passiv aufgenommen werden kann, sondern vielmehr selbstständig konstruiert werden muss. Das ist auch der entscheidende Punkt, welcher den Konstruktivismus vom Kognitivismus unterscheidet.

Gold unterscheidet dabei nochmals zwischen individuellem Konstruktivismus, bei dem der Fokus auf der Selbsttätigkeit der Lernenden liegt, und dem sozialen Konstruktivismus, bei dem das soziale Umfeld für den Lernerfolg maßgebend ist (vgl. Gold 2003, S. 100–103).

Hubwieser gibt unter anderen folgende drei Prozessmerkmale zu konstruktivistischem Lernen an, die jeweils kurz in Bezug auf die Verwendung eines digitalen Lernspiels untersucht werden:

- 1. Lernen ist nur über die aktive Beteiligung des Lernenden möglich. Dazu gehört, dass der Lernende zum Lernen motiviert ist und dass er an dem, was er tut und wie er es tut, Interesse hat oder entwickelt. (Hubwieser 2007, S. 10)*

Digitale Lernspiele sind per Definition interaktiv. Der Schüler muss sich somit auch aktiv mit der jeweiligen Aufgabenstellung auseinandersetzen und beteiligen. Auch was die Motivation betrifft, so scheinen Lernspiele ein probates Mittel zu sein, um vor allem jene Schüler zu motivieren, die von sich aus keine große Motivation haben, das Programmieren zu erlernen. Idealerweise entwickelt der Schüler aufgrund des Lernspieles, des einfachen Zugangs zur Programmierung und der schnellen Erfolgserlebnisse eher ein Interesse an der Programmierung als mit herkömmlichen Übungen.

- 2. Bei jedem Lernen übernimmt der Lernende Steuerungs- und Kontrollprozesse. Wenn auch das Ausmaß eigener Steuerung und Kontrolle je nach Lernsituation variiert, so ist doch kein Lernen ohne jegliche Selbststeuerung denkbar. (Hubwieser 2007, S. 10)*

Hierbei weist Hubwieser darauf hin, dass die Freiheit und Selbstwirksamkeit des Lernenden absolut zentrale Voraussetzungen für einen gelungenen Lernprozess sind.

Bei einem normalen Frontalvortrag sind die Möglichkeiten der Lernsteuerung sehr stark begrenzt. Der Schüler hat in diesem Fall dem Tempo des Lehrers zu folgen und keine eigene Kontrolle, was wann gelernt wird. Dies ist wohl auch einer der Gründe, warum aktuell die Reformpädagogik dermaßen boomt. Bei Computerspielen ließe es sich allerdings durchaus einrichten, dass die Schüler bis zu einem gewissen Grad selbst aussuchen können, was sie zuerst lernen möchten.

Außerdem erfahren sich die Schüler bei digital game-based Learning idealerweise als selbstwirksamer, da sie sehr schnell zu kleinen konkreten Erfolgserlebnissen kommen. Im konkreten Fall des Programmierens würde das bedeuten, dass der Schüler nicht eine abstrakte Aufgabe zu erfüllen hat, wie beispielsweise einen Text formatiert in einer Konsole auszugeben. Es können vielmehr konkrete Aufgaben gelöst werden, wie beispielsweise eine Brücke mithilfe von Code herunterzulassen. Dabei erfährt sich der Schüler wesentlich wirksamer, da er auf diese digitale Welt schaffend und verändernd einwirken kann.

*3. Lernen ist in jedem Fall konstruktiv: Ohne den individuellen Erfahrungs- und Wissenshintergrund und eigene Interpretationen finden im Prinzip keine kognitiven Prozesse statt. (Hubwieser 2007, S. 10)*

Demnach ist es wesentlich sinnvoller den Theorieteil deutlich zu kürzen und die dadurch freigewordene Zeit für praktisches Üben und ausprobieren zu verwenden. Bei digital-game based Learning wird normalerweise die Theorie nur relativ kurz behandelt um anschließend an konkreten Situationen die gewonnenen Erkenntnisse auszuprobieren. Dabei ist es absolut essentiell, dass die Schüler dabei die Möglichkeit haben, selbstständig (idealerweise ohne Anweisung) ihr theoretisches Wissen durch diese Übungen noch zu vertiefen und zu erweitern.

### **3.1.4 Humanismus**

Als vierte wichtige Strömung kann der humanistische Ansatz gesehen werden. Bei dieser Strömung wird der Lernende in den Mittelpunkt gestellt, daher wird sie auch oft Lerner-zentrierte Didaktik genannt.

McCombs beschreibt diese Strömung folgendermaßen sehr treffend:

*The person-centered model and its derivatives offer a philosophical foundation focused on the whole person and their perceptions and experiences of learning that are now seen as valid after decades of anti-self-report criticisms by the research community. (McCombs 2013, S. 83)*

Auch die APA schreibt diesbezüglich, dass die Lerner-zentrierten Prinzipien die aktive und reflektierende Natur des Lernens und der Lerner hervorhebt und dass sich die didaktische Praxis nur verbessern kann, wenn der Hauptfokus nicht auf dem Lehrenden oder den Methoden liegt, sondern vielmehr auf dem Lernenden selbst (vgl. APA Board of Educational Affairs 1997, S. 2).

Die APA nennt 14 psychologische Prinzipien, des Lerner-zentrierten Lernens von denen die wichtigsten hier aufgelistet werden:

- „The learning of complex subject matter is most effective when it is an intentional process of constructing meaning from information and experience” (APA Board of Educational Affairs 1997, S. 3).

Die APA geht davon aus, dass erfolgreiche Lernprozesse nur dann stattfinden, wenn diese intentional vollzogen werden und der Lerner selbstständig eine Bedeutung aus seinen Erfahrungen und den vorhandenen Informationen konstruiert.

- “The successful learner can link new information with existing knowledge in meaningful ways” (APA Board of Educational Affairs 1997, S. 3).

Ein weiterer wichtiger Aspekt des humanistischen Ansatzes ist, dass Lernprozesse an bestehendes Wissen des Lerners anknüpfen sollten. In der modernen Pädagogik wird dies durch eine progressive Aufhebung der ehemals strikt getrennten Lehrfächer erzielt. Es soll beispielsweise nicht mehr darum gehen, in einem Fach Geschichte zu lernen und in einem anderen Fach, strikt davon getrennt, Geographie, vielmehr sollten die beiden Fächer ständig Bezug aufeinander nehmen.

- “What and how much is learned is influenced by the motivation. Motivation to learn, in turn, is influenced by the individual's emotional states, beliefs, interests and goals, and habits of thinking” (APA Board of Educational Affairs 1997, S. 4).

Die APA erkennt in der Motivation einen der zentralsten Aspekte des Lernerfolgs. In Kapitel 3.3 wird diese etwas genauer beleuchtet. Was beim humanistischen Ansatz relevant ist, ist, dass die Motivation jedes einzelnen Schülers stark von der Emotion, den Überzeugungen, Zielen und Interessen vor Allem aber auch von den Denkmustern des Lernenden abhängt.



- “The learner's creativity, higher order thinking, and natural curiosity all contribute to motivation to learn. Intrinsic motivation is stimulated by tasks of optimal novelty and difficulty, relevant to personal interests, and providing for personal choice and control” (APA Board of Educational Affairs 1997, S. 5).

Intrinsische Motivation wird als absolut zentral für den Lernerfolg gesehen. Dabei spielen vor Allem die Neugier der Lernenden eine große Rolle. Allerdings kann intrinsische Motivation auch durch selbst anpassbare Schwierigkeitsgrade und die Möglichkeit der Entscheidungsfreiheit stimuliert werden.

- “Learning is influenced by social interactions, interpersonal relations, and communication with others” (APA Board of Educational Affairs 1997, S. 5).

Auch soziale Interaktionen spielen bei den Humanisten eine gewichtige Rolle, da diese den Lernerfolg stark begünstigen können.

- “Learners have different strategies, approaches, and capabilities for learning that are a function of prior experience and heredity” (APA Board of Educational Affairs 1997, S. 6).

Alle Lerner haben im Laufe ihres Lebens aufgrund unterschiedlicher Erfahrungen verschiedene Lernstrategien entwickelt. Es gilt daher auf diese Unterschiede einzugehen und den Lernstoff auf möglichst viele verschiedene Art und weisen zu präsentieren.

Diese Prinzipien veranschaulichen sehr gut, wie vielschichtig die einzelnen Lerner sind und wie wichtig es ist, auf die Unterschiede zwischen den Lernenden Rücksicht zu nehmen.

### **3.2 Aufmerksamkeit**

In der Literatur wird gemeinhin zwischen allgemeiner Aufmerksamkeit und selektiver Aufmerksamkeit unterschieden. Dabei ist mit allgemeiner Aufmerksamkeit die Aktiviertheit bzw. die Wachheit gemeint, während die selektive Aufmerksamkeit eher die Filterung bzw. Fokussierung bestimmter Inhalte meint (vgl. Trimmel 2013, S. 142).

### 3.2.1 Aktivierungstheorien

Trimmel bestimmt die Aktiviertheit folgendermaßen:

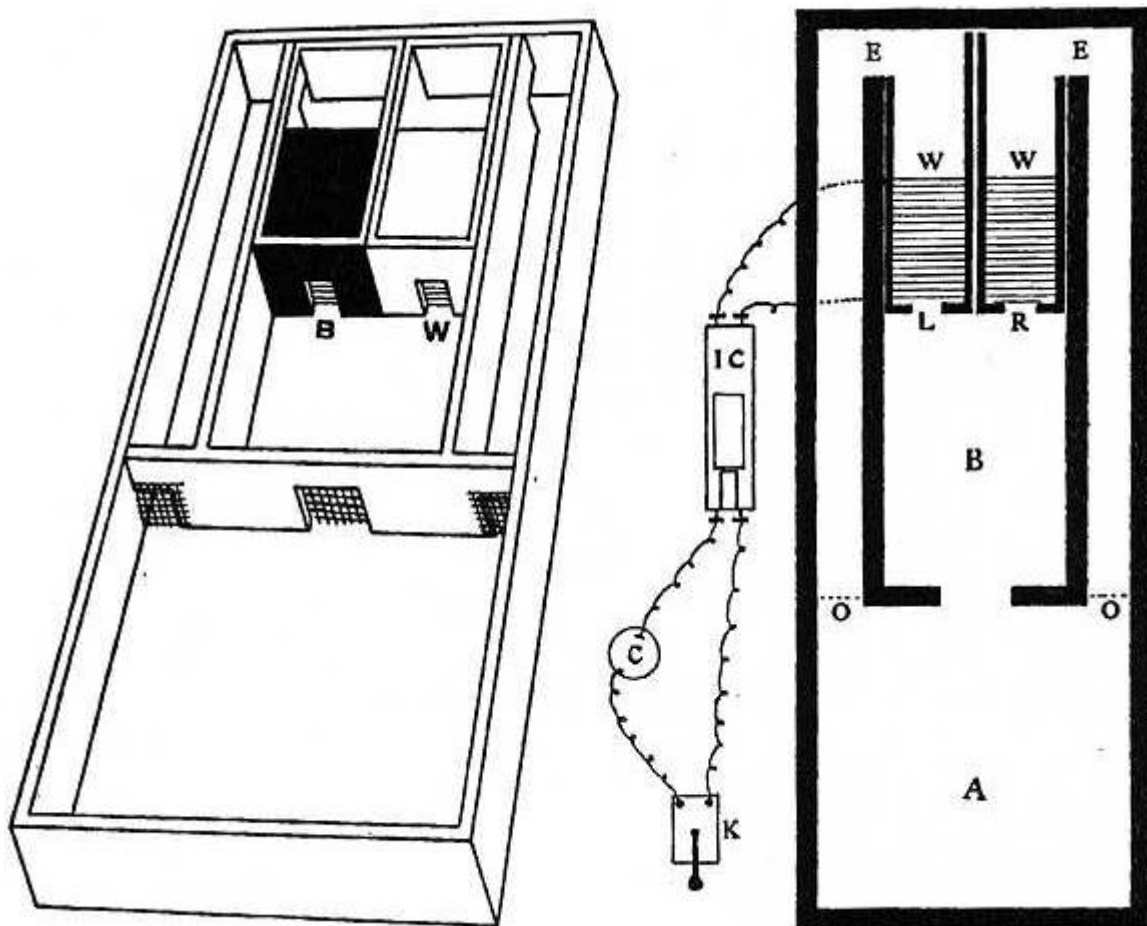
*Die Aktiviertheit stellt eine nicht direkt beobachtbare Eigenschaft des Organismus dar, die eng mit den Begriffen Wachheit, Leistungsbereitschaft und Anregungszustand verbunden ist. Kommt es zu einer erhöhten Aktiviertheit, so liegt eine Aktivierung vor. (Trimmel 2011, S. 73)*

Diesbezüglich gab es im letzten Jahrhundert einige wesentliche Erkenntnisse, die hier in aller Kürze dargestellt werden sollen.

#### **Yerkes & Dodson**

Yerkes und Dodson postulierten 1908 als Folge einer Experimentreihe das nach ihnen benannte Yerkes-Dodson Gesetz, welches den Zusammenhang zwischen Leistung und Erregung beschreiben soll. Sie haben hierfür Ratten in einem konstruierten Käfig von einer Seite auf die andere gedrängt, indem sie die Ratten langsam mithilfe einer bewegbaren Wand voran gescheucht haben.

Die Ratten mussten sich auf der anderen Seite des Käfigs entscheiden, ob sie durch eine schwarze oder durch eine weiße Box laufen. Am Boden dieser Boxen waren elektrische Leiter, die den Ratten einen elektrischen Schlag verabreichten, wenn sie sich für die schwarze Box entschieden. Gemessen wurde, wie lange es dauerte, bis die Ratten lernten, durch die weiße Box zu gehen und die schwarze Box zu meiden.



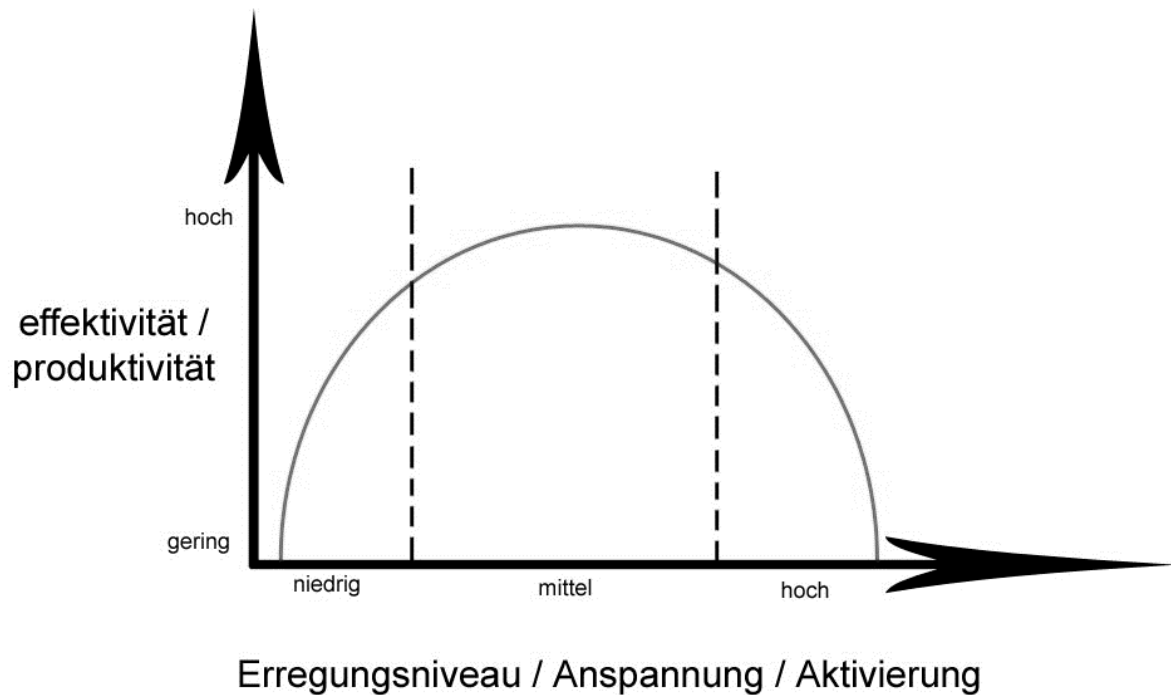
**Abbildung 3 Versuchsaufbau von Yerkes & Dodson**

**Quelle: (www.psychclassics.yorku.ca 2007)**

Yerkes und Dodson führten dabei drei Versuchsdurchgänge durch, welche den Schwierigkeitsgrad der Aufgabe variierten, indem sie die Helligkeit der weißen Box änderten. Bei jedem der drei Durchgänge wurde außerdem die Reizintensität, also die angelegte Spannung variiert.

Als Ergebnis ihrer Forschung stellten sie das Yerkes – Dodson Gesetz auf, welches besagt, dass die Ratten die besten Leistungen erzielten, wenn ein mittlerer Reiz (Spannungsintensität) gegeben ist. Zu schwache und zu starke Reizintensitäten reduzierten die Lernleistung der Ratten. Abbildung 4 veranschaulicht die Ergebnisse von Yerkes und Dodson.

### **Das Yerkes-Dodson-Gesetz**



**Abbildung 4 Yerkes – Dodson Gesetz**

Quelle: <https://de.wikipedia.org/wiki/Yerkes-Dodson-Gesetz>

Nun gehen Psychologen auch heutzutage davon aus, dass sich das Yerkes-Dodson Gesetz ohne Weiteres auch auf Menschen übertragen lässt. Es scheint auch intuitiv nachvollziehbar, dass der Mensch die größte Leistung erbringt, wenn er weder überfordert, noch unterfordert ist, also genau jenes Maß an Aktivierung hat, welches die jeweilige Person als angenehm empfindet.

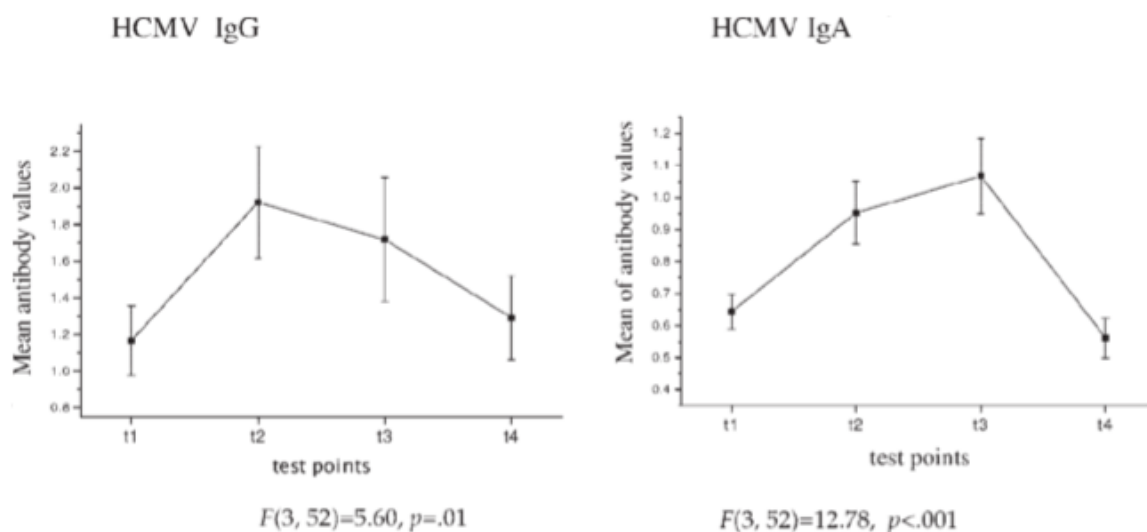
Dieses Experiment ist allerdings lediglich als historische Grundlage zu sehen. Es würde modernen Kriterien selbstverständlich nicht mehr genügen. Allerdings wurde das Yerkes-Dodson Gesetz auch durch modernere Experimente nachgewiesen.

#### **Aktuelle Belege für das Yerkes – Dodson Gesetz**

Beispielsweise konnten Sarid, Anson et al. bei Speicheluntersuchungen von Studenten feststellen, dass zu Prüfungszeiten wesentlich mehr HCMV Antikörper, gebildet wurden als zu Ruhezeiten. Die Forscher postulieren dabei, dass es sich bei der vermehrten HCMV Antikörper Produktion um ein physiologisches Symptom

handelt, das ein Hinweis auf das Yerkes-Dodson Gesetz ist (vgl. Sarid et al. 2004, S. 374–375).

Die Forscher entnahmen im Laufe des Experiments Studenten an vier Zeitpunkten Speichelproben, anhand derer sie den HCMV-Antikörper Gehalt gemessen haben. Dabei konnten sie zeigen, dass sich ein gewisses Maß an Stress positiv auf die Leistung auswirkte. Allerdings ist hierbei festzuhalten, dass in diesem Experiment die Überforderung, also die Leistung bei zu viel Aktiviertheit, nicht gemessen wurde.



**Abbildung HCMV – Antikörper zu verschiedenen Zeitpunkten.**

**Quelle: (Sarid et al. 2004, S. 374)**

t1- 1 Monat nach Semesterbeginn, Die (erstsemestrigen) Studenten konnten sich an die neue Lebenssituation gewöhnen.

t2 – Direkt nach der Physikprüfung

t3 – 2 Wochen nach t2, Direkt nach Epidemiologie Prüfung

t4 – 2 Wochen nachdem die Ergebnisse der Prüfungen bekanntgegeben wurden

An Abbildung lässt sich hervorragend die Ähnlichkeit zum Yorkes Dodson Gesetz ablesen. Zu den stressigeren Zeiten entwickelt der Körper wesentlich mehr HCMV

Antikörper als zu den Ruhezeiten. Dabei korrelierte sowohl die Anzahl der gebildeten HCMV Antikörper bis zum Prüfungstermin als auch die vorhandenen Antikörper direkt nach der Prüfung positiv mit der erreichten Punkteanzahl (vgl. Sarid et al. 2004, S. 374–375). Es ist jedoch darauf hinzuweisen, dass der Zeitpunkt t4 der Messung nicht die Überforderung darstellt, sondern eine weitere Ruhephase. Die Überforderung wird in diesem konkreten Experiment nicht gemessen.

Die Forscher folgerten daraus folgendes:

*It may be suggested that the increase in HCMV [=human cytomegalovirus] salivary antibodies reflect an arousal response to academic stress, which enhanced performance. This interpretation is in line with the common formulation of the Yerkes and Dodson (1908) experiment, which states that some degree of stress is beneficial to functioning, at least for a short period. (Sarid et al. 2004, S. 375)*

Sarid, Anson et al. bestätigten also, dass sich ein gewisses Maß an Stress positiv auf die Leistung auswirkt.

## **Arousal**

Berlyne versucht vor dem Hintergrund der Ergebnisse von Yerkes und Dodson zu erklären, wie der Mensch auf Aktivierung reagiert. Er führte dazu den Begriff des Anregungspotentials ein und postulierte, dass der Mensch stets ein mittleres Anregungspotential anstrebt (vgl. Trimmel 2011, S. 80).

Dabei beschreibt er das Anregungspotential als alles, was wahrgenommen wird, beziehungsweise alles, was auf einen einströmt. Berlyne meint nun, dass jeder Mensch ein individuelles Maß an Anregungspotential anstrebt. So strebt ein Extremsportler, wie zum Beispiel ein Fallschirmspringer, ein wesentlich höheres Anregungspotential an, als ein Schachspieler.

Nach Berlyne ist die Leistung am besten, wenn das vorhandene Anregungspotential dem gewünschten Anregungspotential entspricht (vgl. Trimmel 2011, S. 80–81). Berlynes Theorie erklärt dadurch hervorragend, warum sich sowohl unterforderte, als auch überforderte Schüler im Unterricht oft störend verhalten.

Die Unterforderten verlangen dabei ein höheres Anregungspotential und da es der Lehrer nicht bieten kann, stören sie den Unterricht oder beschäftigen sich mit etwas anderem. Andererseits stören die überforderten Schüler den Unterricht bzw. arbeiten gar nicht mehr mit, da sie, um das hohe Anregungspotential zu reduzieren, den Informationsinput reduzieren müssen, indem sie dem Lehrer nicht mehr zuhören oder etwas anderes, entspannenderes machen.

### **3.2.2 Selektive Aufmerksamkeit**

#### **Zoomlens Modell**

Das wohl bekannteste Aufmerksamkeitsmodell ist wohl das Zoomlens Modell von Eriksen und James. Diese vergleichen die Aufmerksamkeit eines Menschen mit dem Zoomobjektiv eines Fotoapparats. Dabei meinen Eriksen und James, dass die Auflösung des Fotos konstant bleibt, und es darauf ankommt, wie stark man an etwas heranzoomt, ob bestimmte Details dadurch wahrgenommen werden (vgl. Enns 1990, S. 223). Die zur Verfügung stehenden Ressourcen bleiben also gleich, und sie können entweder auf einen kleinen Ausschnitt der Wirklichkeit fokussiert werden, oder aber auf das große Ganze.

Wird dabei nur ein kleiner Ausschnitt fokussiert, so werden zwar winzige Details diesbezüglich bemerkt, dafür können aber Teile, die nicht fokussiert werden, völlig außer Acht gelassen werden. Umgekehrt verhält es sich, wenn man den Fokus auf das große Ganze richtet. Dann werden zwar alle größeren Änderungen bemerkt, dafür entgehen einem aber Änderungen im Detail.

#### **Filtertheorie**

Broadbent postulierte bereits 1958, dass die Aufmerksamkeit eines Menschen durch einen Filter geregelt wird. Dieser Filter soll verhindern, dass kognitive Ressourcen für unnötige Informationsverarbeitung verschwendet werden, indem er basierend auf den physikalischen Charakteristika des Inputs relevante- von nicht-relevanter Information unterscheidet und relevante Informationen bevorzugt zur weiteren Verarbeitung durchlässt (vgl. Eysenck 2001, S. 115–116).

An Broadbents Filtertheorie lässt sich jedoch kritisieren, dass sie relativ unflexibel ist und der Aufmerksamkeitsfilter, der den Flaschenhals (Bottleneck) der gesamten

Informationsverarbeitungstheorie darstellt, ist im Informationsverarbeitungsprozess sehr früh angesiedelt. Spätere Experimente zeigten, dass die Annahme dieses Filters nicht grundsätzlich falsch ist, dass er aber nicht aufgrund physikalischer Merkmale filtert, wie Broadbent noch dachte, sondern erst später im Informationsverarbeitungsprozess angesiedelt ist (vgl. Eysenck 2001, S. 117–118).

### **Abschwächungstheorie**

Treisman entwickelte diesbezüglich eine ähnliche Theorie, die die Filterung von unwichtiger Information beschreibt. In ihrer Abschwächungstheorie geht sie davon aus, dass alles Wahrgenommene bezüglich der subjektiven Relevanz unbewusst bewertet wird und diese Relevanz einen gewissen Schwellenwert überschreiten muss, damit etwas bewusst wahrgenommen werden kann (vgl. Eysenck 2001, S. 118).

Der eigene Name hat dabei beispielsweise einen relativ niedrigen Schwellenwert, sodass er sehr einfach bewusst wahrgenommen wird, ein Gespräch, welches einen nicht betrifft und in einer fremden Sprache geführt wird, hat hingegen einen sehr hohen Schwellenwert, sodass es kaum wahrgenommen wird. Solange etwas diesen Schwellenwert nicht überschreitet, wird die Information nicht verarbeitet, um dadurch keine kostbaren kognitiven Ressourcen zu verschwenden (vgl. Eysenck 2001, S. 118).

Ein Beispiel, welches diese Formen der selektiven Aufmerksamkeit hervorragend demonstriert, liefert Cherry 1953 mit seinem Cocktailparty Phänomen. Er verweist dabei auf das Phänomen, dass auf einer Cocktailparty, bei der ein gewisser Lautstärkepegel herrscht, da alle Personen um einen herum Gespräche führen, man trotzdem in der Lage ist, einer bestimmten Unterhaltung zu folgen (vgl. Trimmel 2013, S. 142–143). Könnten man seine Aufmerksamkeit nicht von der Wahrnehmung aller Geräusche auf diejenigen wenigen fokussieren, die für einen relevant sind, wäre man mit der Informationsverarbeitung hoffnungslos überfordert.

Aus Treismans Theorie kann man schließen, dass die persönliche Relevanz der wahrgenommenen Information absolut zentral für die Aufmerksamkeit ist. Bezüglich des Erwerbes einer Programmiersprache kann ein Großteil der Schüler jedoch zunächst keine besondere persönliche Relevanz entwickeln. Selbst wenn sie sich aber für das Programmieren interessieren, so sind meist die einfachen Konsolenanwendungen, mit denen normalerweise ein Programmierkurs beginnt,



dermaßen abschreckend, abstrakt und erfolgsergebnislos, dass nur die wenigsten Schüler auch tatsächlich einen Bezug dazu herstellen können und dementsprechend eine Begeisterung für das Programmieren entwickeln.

Laut Treismans Abschwächungstheorie wäre es somit aber absolut essentiell für den Lernprozess der Schüler, dem Programmieren bzw. dem Programmierunterricht ein möglichst hohes Maß an Relevanz zuzuordnen. Computerspiele könnten eine Möglichkeit darstellen, genau diese persönliche Relevanz zu verstärken und somit die Aufmerksamkeit, die die Schüler dem Unterricht schenken, erhöhen.

### 3.3 Motivation

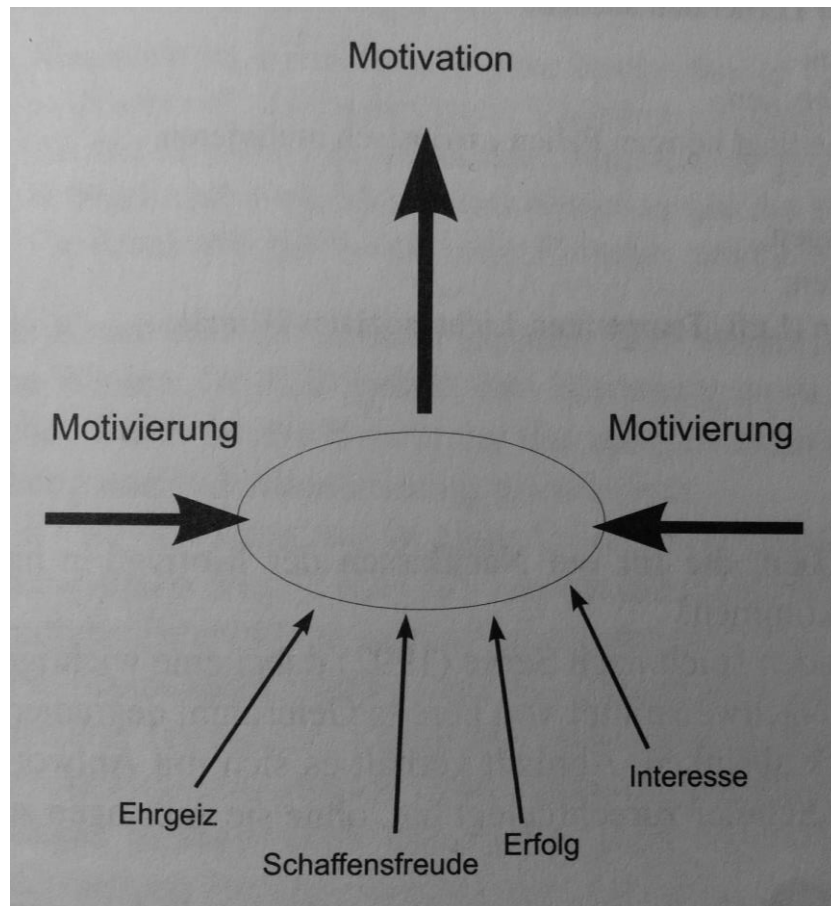
Hubwieser schreibt bezüglich der Motivation folgendes:

*Ohne den durch eine angemessene Motivierung erzeugten Lernwillen ist jedes unterrichtliche Bemühen sinnlos. Deshalb ist Motivierung das vordringlichste Ziel didaktischen Handelns (Hubwieser 2007, S. 15).*

Hubwieser unterscheidet zwischen folgenden drei Aspekten:

- **Motivation:** ein kurz andauernder Zustand des Angetriebenseins.
- **Motivierung:** Ein aktives Bemühen um die Herstellung dieses Zustandes der Motivation.
- **Motive:** Längerfristige Initiierungs- und Lenkungsfaktoren wie Schaffensfreude, Ehrgeiz, Erfolg, usw.

(vgl. Hubwieser 2007, S. 15).



**Abbildung 5 Motivierung, Motive und Motivation**

**Quelle: (Hubwieser 2007, S. 15)**

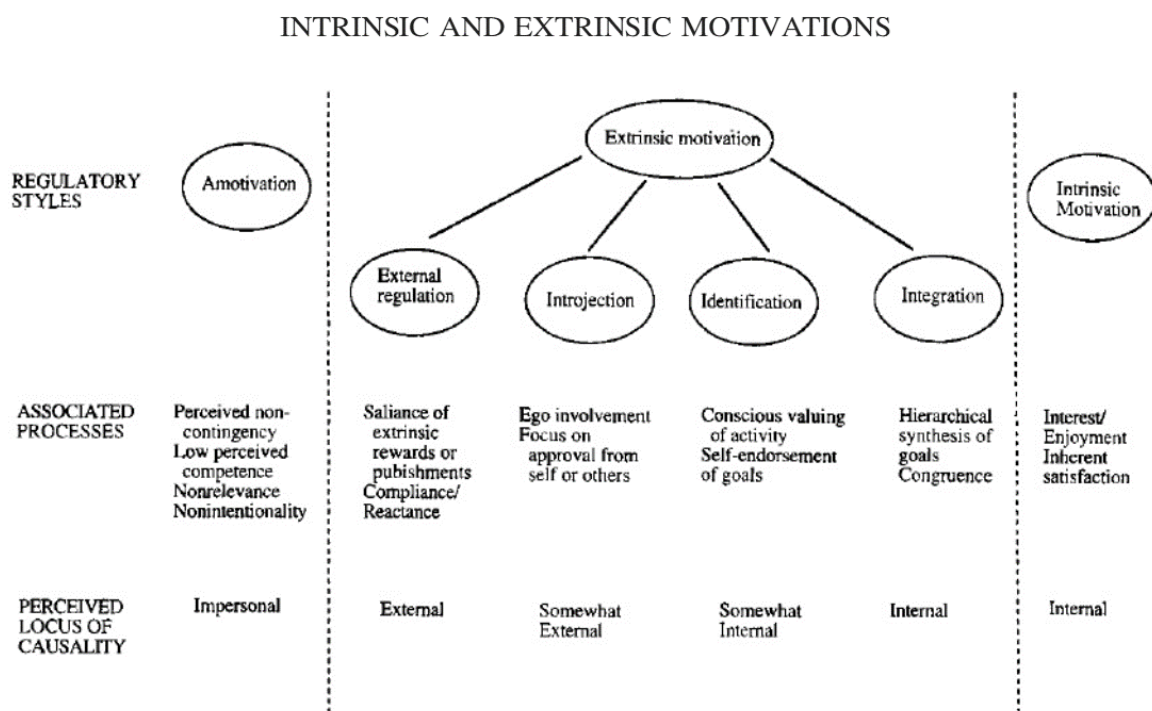
Ryan und Deci schreiben in ihrem Artikel *Intrinsic and Extrinsic Motivations* (Ryan & Deci, 2000), dass es grundsätzlich zwei unterschiedliche Arten von Motivation gibt, nämlich intrinsische und extrinsische.

Auf das Schulsystem bezogen bedeutet intrinsische Motivation, dass sich der Schüler aus Interesse in den jeweiligen Bereich eines Themengebietes vertieft. Der Schüler ist also bereit, sich freiwillig, ohne explizite Anweisung, mit dem Thema zu beschäftigen (vgl. Ryan und Deci 2000, S. 56–60).

Extrinsische Motivation bedeutet hingegen, dass der Schüler einen externen Grund hat, sich mit dem Thema zu beschäftigen. Beispielsweise will er eine gute Note haben, beziehungsweise hat er Angst vor den Konsequenzen einer schlechten Note. Bei extrinsischer Motivation beschäftigt sich der Schüler nicht freiwillig mit dem Thema, sondern aufgrund äußerer Umstände (vgl. Ryan und Deci 2000, S. 60–65).

Laut Ryan und Deci korreliert intrinsische Motivationsregulation direkt mit hochqualitativen Lernerfolgen. Diese ist somit für die Anwendung im Bildungssektor anderen Motivationsregulationstypen überlegen. Die beiden Forscher weisen aber auch darauf hin, dass nicht ganz auf extrinsische Motivation verzichtet werden kann. So werden die Schüler zwangsläufig früher oder später damit konfrontiert, etwas lernen zu müssen, für das sie sich intrinsisch nicht begeistern können. (vgl. Ryan und Deci 2000, S. 60–65)

Weiters konnten die beiden Forscher feststellen, dass es verschiedene Arten extrinsischer Motivation gibt, die sich hauptsächlich in der Autonomie und der Selbstbestimmung unterscheiden. Je autonomer und selbstbestimmter die extrinsische Motivationsregulation ist, umso eher ähnelt sie intrinsischer Motivationsregulation (vgl. Ryan und Deci 2000, S. 62).



**Abbildung 6: Intrinsische und extrinsische Motivationen**

**Quelle: (Ryan und Deci 2000, S. 61)**

In Abbildung 6 werden die verschiedenen Regulationstypen extrinsischer Motivation dargestellt. Diese Regulationstypen bilden einen fließenden Übergang zur intrinsischen Motivation. Ryan und Deci zeigen damit, dass die Motivationsregulation nicht nur entweder intrinsisch oder extrinsisch ist, sondern dass es dazwischen noch weitere Unterstufungen gibt und es sich somit eher um ein Kontinuum handelt.

**Externale Regulation** bedeutet, dass der Schüler primär durch Belohnungen oder Bestrafungen motiviert wird. Die operante Konditionierung von B. F. Skinner arbeitet beispielsweise nahezu ausschließlich mit externaler Regulation, die bezüglich langfristiger Lernerfolge eine eher niedrige Relevanz hat (vgl. Ryan und Deci 2000, S. 61–62).

**Introjizierte Motivation** zielt darauf ab, durch sozial anerkanntes Handeln bei Kollegen einen guten Eindruck zu machen. Die sozialen Normen werden jedoch nicht als persönlich wichtig erachtet, sondern als gesellschaftliche Forderungen verstanden, durch deren Befolgung man soziale Anerkennung erfährt. (vgl. Almut und Müller 2011, S. 4)

**Identifizierte Regulation** wird durch Almut und Müller in ihrem Forschungsbericht folgendermaßen definiert:

*„Identifizierte Regulation zeichnet sich durch Ergebnisorientierung aus – das Motiv für die Handlung liegt in einem Ergebnis, das als persönlich wichtig erachtet wird. Sowohl intrinsische Motivation als auch identifizierte Motivation werden als selbstbestimmt wahrgenommen, die handlungsleitenden Motive lassen sich durch ein ‚ich will‘ beschreiben.“ (Almut und Müller 2011, S. 4)*

Bei identifizierter Motivationsregulation geht es also hauptsächlich darum, Ziele zu erreichen, die sich die Person selbst setzt. Gute Voraussetzungen für die Berufswahl zu schaffen, ist ein typisches Beispiel für identifizierte Motivationsregulation in Schulen.

**Integrierte Regulation** hat den höchsten Grad der Selbstbestimmung. Sie ist das Ergebnis der Integration von Zielen, Normen und Handlungsstrategien, mit denen sich der Schüler identifiziert (vgl. Ryan und Deci 2000, S. 59–61).

Müller et al. haben auf Basis der Erkenntnisse von Ryan und Deci eine großflächige Untersuchung (Müller et al., 2007) an österreichischen Schulen durchgeführt. Die Ergebnisse stimmen mit denen von Ryan und Deci überein.

*Die pädagogische Bedeutung von selbstbestimmtem motiviertem Lernen ist in der Qualität von Bildungsprozessen sowie deren Effekten zu sehen. Selbstbestimmte Lernende sind zufriedener beim Lernen, eignen sich Wissen differenzierter und zusammenhängender an, weisen zumeist längere Behaltensleistungen des Gelernten auf und wenden ihre Wissensbestände eher an. Ferner scheitern intrinsisch motivierte Lernende seltener an den Anforderungen einer Bildungseinrichtung. Sie zeigen – besonders langfristig – bessere Lernleistungen und nehmen sich als kompetenter wahr als Lernende, die sich eher external regulieren. (Müller et al., S. 6)*

Müller et al. bestätigen hiermit die Ergebnisse von Ryan und Deci, dass intrinsisch motivierte Schüler bessere Lernerfolge erzielen als extrinsisch motivierte. Sie untermauern mit ihren Ergebnissen die Theorie, dass sich extrinsische Motivationsregulationen in ihrer Autonomie, und somit ihrem Ähnlichkeitsgrad zu intrinsischer Motivation, von eben dieser unterscheidet.

Die Untersuchung von Müller et al. beweist ebenfalls, dass intrinsische Motivation extrinsische Motivation nicht ausschließt. Zwar kann es, wie es Ryan und Deci beschreiben, zu einer Verlagerung des einen Motivationstyps zum anderen kommen, die beiden Motivationstypen schließen sich jedoch gegenseitig nicht aus. Die Ergebnisse von Müller et al. decken sich diesbezüglich mit der Studie von Lin et al. (vgl. Lin et al. 2001).

Lin et al. konnten in ihrer Studie die Auswirkungen extrinsischer und intrinsischer Motivation auf den Lernerfolg nach der „Cognitive Evaluation Theory“ von Ryan und Deci ebenfalls beweisen. Es stellte sich jedoch heraus, dass nicht nur intrinsische Motivation positiv mit dem Lernerfolg korreliert, sondern auch extrinsische Motivation

bis zu einem gewissen Grad einen positiven Einfluss hat. Weiters konnten sie beweisen, dass extrinsische Motivation intrinsische Motivation nicht ausschließt. So ist es durchaus üblich, dass Schüler sowohl intrinsisch, als auch extrinsisch motiviert sind. Laut dieser Untersuchung ist zwar ein zu niedriges Maß (Unterforderung) oder ein zu hohes Maß (Überforderung) an extrinsischer Motivation schädlich für den Lernerfolg, ein mittleres Maß an extrinsischer Motivation fördert jedoch den Lernerfolg optimal.

Aus diesen Untersuchungen lässt sich schließen, dass intrinsische Motivation der Schlüssel zu effizientem Lernen ist. Extrinsische Motivation kann den Lernerfolg, wenn sie im richtigen Ausmaß erfolgt, allerdings noch weiter anheben. Weiters sind extrinsische Motivationsregulationen effizienter, je eher sie intrinsischer Motivationsregulation ähneln.

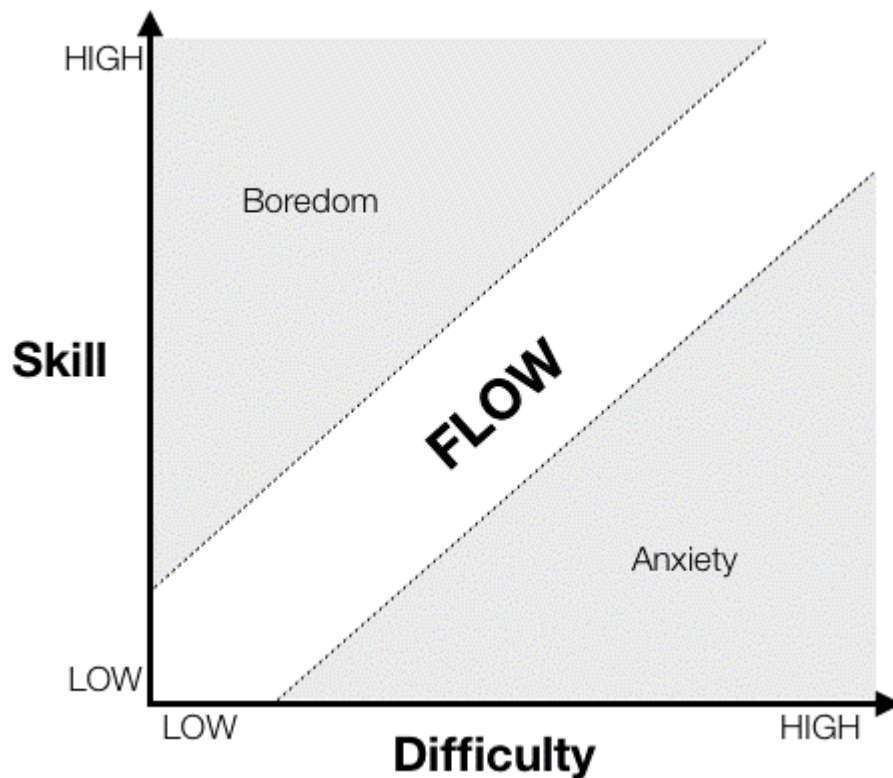
### **3.4 Flow Erlebnis**

Csikszentmihalyi geht bereits davon aus, dass es absolut zentral ist, eine intrinsische Motivation anzustreben, da diese die Grundlage einer guten Arbeitsmoral ist. Er meint, dass der Mensch dermaßen funktioniert, dass extrinsische Motivation zwar ein gewisser Anreiz für etwas sein kann, dieser allein einen jedoch letztlich nicht glücklich machen kann. Butler-Bowdon beschreibt Csikszentmihalyis Flow-Erlebnisse deshalb folgendermaßen:

*Flow activities we do for the sheer enjoyment or intellectual satisfaction, rather than to gain some extrinsic reward. You might want to win a game of Chess for example, but you play it because it engages your mind totally. You might want to become a great dancer, but it is the learning and dancing that are the main reward. (Butler-Bowdon 2007, S. 89)*

In seinen Flow Activities geht es also eher um die Freude oder die intellektuelle Herausforderung, als um extrinsische Belohnungen. Er räumt zwar ein, dass extrinsische Aspekte eine Rolle spielen, wenn er schreibt, dass man zwar ein Schachspiel gewinnen möchte, allerdings meint Csikszentmihalyi, dass das Gewinnen nicht der eigentliche Grund ist, warum man Schach spielt, vielmehr ist es die intellektuelle Herausforderung, die einem die Freude am Schachspiel beschert.

Was den Lernprozess anbelangt, so ist dieses Flow-Erleben genau das, was im Unterricht idealerweise angestrebt werden sollte. Trimmel schreibt, dass es genau dann zu einem Flow – Erleben kommt, wenn die Anforderungen mit den Fähigkeiten einer Person möglichst übereinstimmen (vgl. Trimmel 2011, S. 136–137).



**Abbildung 7 Flow – Erleben**

**Quelle:** [http://gamasutra.com/db\\_area/images/feature/166972/figure1.png](http://gamasutra.com/db_area/images/feature/166972/figure1.png)

Sind die Anforderungen zu hoch, so bekommt die betroffene Person Angst. Im schulischen Kontext sind dabei besonders schwächere Schüler betroffen, die dann aufgrund des nicht-Übereinstimmens der Fähigkeiten und der Anforderungen auch jegliche Freude am Lernen verlieren. Der Lernprozess im jeweiligen Fach degradiert dann zu einem bloß extrinsischen kurzfristigen Aneignen von Fakten.

Das andere Extrem wäre, dass die Anforderungen zu niedrig sind und das eigene Können zu hoch ist. In diesem Fall langweilt sich der Schüler und hat demnach auch keine große Freude an der jeweiligen Aufgabe.

Die Fähigkeit, diesen Balanceakt zwischen einerseits den Fähigkeiten der Schüler und andererseits der Schwierigkeit der Aufgaben im Unterricht zu meistern, macht ein gutes didaktisches Konzept aus. Dabei ist selbstverständlich darauf zu achten, dass der jeweilige Stoff einerseits einfach genug ist, sodass ihn möglichst alle Schüler verstehen, andererseits sollte es genügend Herausforderungen für begabte Schüler geben, damit diese sich nicht langweilen.

## **4 Digital game-based Learning**

### **4.1 Begriffsdefinition**

Die Verwendung von Computerspielen zur Wissensvermittlung kam bereits in den 80er Jahren auf. Damals sprach man jedoch von Edutainment. Dieser Begriff umfasst allerdings nicht nur Lernspiele, sondern auch lehrreiche Fernsehsendungen und Ähnliches (vgl. Breuer 2010, S. 7–8).

Da das Format eines Films aufgrund der fehlenden interaktiven Komponente allerdings ein gänzlich anderes ist, als das eines Computerspieles, sind meistens auch jegliche pädagogischen Handlungsrichtlinien jeweils nur für eine dieser beiden Medienformen gültig. Daher bedarf es einer klaren Unterscheidung zwischen Computerspielen und Filmen. Der Begriff des digital game-based Learnings grenzt sich vom Edutainment genau dadurch ab, dass er ausschließlich für Computerspiele gilt und somit etwas präziser ist.

Laut Breuer wurde in den 80er Jahren das Entertainment oft als Belohnung für braves Lernen gedacht. Heute geht der Trend in eine andere Richtung, Computerspiele versuchen nicht eine Belohnung für klassisches Lernen bereitzustellen, sondern es wird versucht, das zu Lernende bereits in den Spielmechaniken zu verankern (vgl. Breuer 2010, S. 7–18). Das Computerspiel soll also nicht mehr in einen Lernbereich und einen Entertainment Bereich aufgeteilt werden, sondern die spielerische Komponente des Spieles soll dermaßen gestaltet sein, dass der Spieler bereits durch den Prozess des Spielens neues Wissen erwirbt.

Ein weiterer Begriff, der oft verwendet wird, um Lernspiele zu bezeichnen und in etwas jüngerer Literatur oft Anwendung findet, lautet Serious Games. Für diesen Begriff gibt



es allerdings noch keine eindeutige Definition, da dieser noch sehr jung ist. Eine der verbreiteteren Definitionen ist die von Sawyer, die folgendermaßen lautet:

*Any meaningful use of computerized game/game industry resources  
whose chief mission is not entertainment (Sawyer 2007, S. 12).*

Diese Definition von Serious Games umfasst also all jene Spiele, deren Hauptziel nicht ist, den Spieler zu unterhalten.

Meistens wird der Begriff im Zusammenhang mit der Vermittlung von Wissen oder Weltanschauungen verwendet. Der Begriff des Serious Games wird aber, wie der Name bereits anklingen lässt, eher im Zusammenhang mit ernsten Themen verwendet. So werden klassischerweise jene Spiele als Serious Games kategorisiert, die dem Spieler beispielsweise näherbringen, was es bedeutet, an der Armutsgrenze zu leben, aus einem Land zu flüchten oder Ähnliches. Es geht dabei also häufig darum, ein Bewusstsein für sehr ernste Themen zu schaffen.

Die Begriffe des Serious Gamings und des digital game-based Learnings sind somit nahezu deckungsgleich, allerdings wird in dieser Arbeit dennoch der Begriff des digital game-based Learnings verwendet, da es sich bei Lernprozessen nicht zwingendermaßen um ernste Themen handelt. Im Gegenteil, idealerweise ist der Lernprozess mithilfe von Computerspielen gerade eben nichts Ernstes, sondern etwas Auflockerndes, Unterhaltsames, das den ernststen Schulalltag etwas abwechslungsreicher gestalten kann.

Ein weiterer Begriff, der in der wissenschaftlichen Literatur Anwendung findet ist das sogenannte game-enhanced Learning. Dieser basiert auf dem Begriff technology-enhanced Learning und baut auf der Infrastruktur eines Lernmanagementsystems auf (vgl. Pfeiffer und Motschnig 2015, S. 167). Mithilfe dieses Lernmanagementsystems werden spielerische Elemente und eine Story dazu verwendet, um die Lernenden zu involvieren (vgl. Pfeiffer und Motschnig 2015, S. 167).

Prensky definiert digital game-based Learning als “any marriage of educational content and computer games” (Prensky 2001, S. 145). Während bei game-enhanced Learning lediglich spielerische Elemente in einem Lernmanagementsystem eingesetzt werden, handelt es sich beim digital game-based Learning um ein richtiges Computerspiel.

Zuletzt ist in diesem Zusammenhang noch der Begriff der Gamification zu nennen, welcher von Deterding et al. folgendermaßen definiert wird: “Gamification” is the use of game design elements in non-game contexts” (Deterding et al., S. 10). Dieser Begriff bezeichnet also den Prozess, irgendeinen Kontext mit spielerischen Elementen auszustatten.

## 4.2 Kritik am digital game-based Learnings

Jedes neue Medium, das zur Wissensvermittlung eingesetzt werden kann, muss sich zunächst bewähren und unter Beweis stellen, dass es tatsächlich einen Mehrwert für Lernprozesse herbeiführen kann. Bereits in der Antike kritisierte Platon durch die Figur des Sokrates das grundlegendste Medium überhaupt, nämlich die Schrift.

Sokrates verwendet dabei exakt dasselbe Argument, das auch gegen Lernen mithilfe von Filmen, Lernsoftware und nun auch gegen das Lernen anhand von Computerspielen eingesetzt wird. Er schreibt im Phaidros folgendes:

*Als er aber an die Buchstaben gekommen, habe Teuth gesagt: Diese Kunst [die Schrift], o König, wird die Ägypter weiser machen und gedächtnisreicher, denn als ein Mittel für Erinnerung und Weisheit ist sie erfunden. Jener [Thamus] aber habe erwidert: [...] O kunstreichster Theuth, einer weiß, was zu den Künsten gehört, ans Licht zu bringen; ein anderer zu beurteilen, wie viel Schaden und Vorteil sie denen bringen, die sie gebrauchen werden. So hast auch du jetzt, als Vater der Buchstaben, aus Liebe das Gegenteil dessen gesagt, was sie bewirken. Denn diese Erfindung wird den Seelen der Lernenden vielmehr Vergessenheit einflößen aus Vernachlässigung der Erinnerung[...] (Schleiermacher 2011)*

Sokrates argumentiert hierbei, dass die Menschen, wenn sie alles niederschreiben, sich nicht mehr die Mühe machen, das Niedergeschriebene auch zu verinnerlichen und im Kopf zu behalten. Laut Sokrates erleichtert die Schrift zwar das Festhalten von Wissen, sie führt aber gleichsam dazu, dass die Menschen dümmer werden, weil sie das Wissen nicht mehr im Kopf behalten müssen.

Die Geschichte hat allerdings gezeigt, dass es sehr wohl sinnvoll ist, Wissen niederzuschreiben, da mündliche Quellen im Laufe der Zeit enorm verzerrt werden, wenn es keinen Urtext gibt, auf den Bezug genommen werden kann.

Auch heute werden neue Formen des Wissenserwerbes kritisiert und das ist auch gut so, da es der notwendigen wissenschaftlichen Überprüfung bedarf, ehe ein neues Medium breitflächig eingesetzt wird, um sicherzugehen, dass das neue Medium mehr Vor- als Nachteile mit sich bringt. Neue Methoden müssen sich somit zuerst bewähren und im Zuge wissenschaftlicher Untersuchungen zeigen, ob sie wirklich einen Mehrwert bringen und ob dieser Mehrwert den Aufwand rechtfertigt, den sie mit sich bringen.

Granic et al schreiben nicht nur euphorisch über die Potentiale von digital game-based Learning, sondern sie warnen durchaus auch vor den Gefahren, die das Computerspielen mit sich bringt.

Sie warnen diesbezüglich vor Depression, Aggression und vor allem vor dem Suchtpotential diverser Spiele, wobei sich die Studien diesbezüglich natürlich nicht auf Lernspiele beziehen, sondern auf Spiele, die lediglich der Unterhaltung dienen (vgl. Granic et al. 2014, S. 9).

Laut einer Studie von Gentile können etwa 8% der Jugendlichen der Vereinigten Staaten als pathologische Spieler eingestuft werden (vgl. Gentile 2009). Diese pathologischen Spieler verbrachten doppelt so viel Zeit mit Computerspielen, als andere Studienteilnehmer, hatten schlechtere Noten in der Schule und größere Aufmerksamkeitsprobleme (vgl. Gentile 2009).

Aus Studien wie dieser könnte man natürlich ableiten, dass Computerspiele grundsätzlich eher zu einer schlechteren Leistung in der Schule führen. Allerdings muss bei Studien wie dieser immer beachtet werden, dass hierbei lediglich Unterhaltungsspiele berücksichtigt werden. Die Aussagekraft dieser Studie bezüglich Lernspielen hält sich somit in Grenzen, allerdings darf diese trotzdem nicht völlig außer Acht gelassen werden.

Denn was die Aufmerksamkeitsprobleme betrifft, so gibt es hierbei durchaus ein Argument, das von Befürwortern des digital game-based Learnings nur schwer von der Hand zu weisen ist. Gemeint ist der zunehmende Verlust der Fähigkeit langfristiger

Aufmerksam der Schüler, um einem Unterricht zu folgen, der eben nicht mithilfe Unterhaltungsmedien gestaltet wird.

Okhan stellt sogar die Frage, ob es denn überhaupt Sinn macht, den Lernprozess lustig und unterhaltsam zu gestalten (vgl. Okhan 2003, S. 258). Er fragt, ob der übermäßige Gebrauch von unterhaltsamen Lernmitteln dann nicht gleichsam dazu führt, dass jene Themengebiete, die nicht durch eine dermaßen unterhaltsame Methode, wie sie ein Lernspiel darstellt, vermittelt werden, dann schlechter gelernt werden. Wenn die Schüler also einmal keinen Spaß am Lernen haben, haben sie auch kein Interesse daran, ein Themengebiet zu lernen und werfen schneller das Handtuch (vgl. Okhan 2003, S. 258).

Wenn man Okhan hierbei zustimmt, so muss natürlich auch eingestanden werden, dass es absolut essentiell ist, diverse Themengebiete zu verinnerlichen, die in sich nicht unterhaltsam sind. Das Lesen eines Fachbuches ist im Normalfall auch nicht unbedingt unterhaltsam, dennoch sollte dieses Fachbuch, was den Wissenserwerb betrifft, beispielsweise einem Youtubetutorial vorgezogen werden.

In diesem Punkt muss man Okhan zweifelsohne Recht geben, eine großflächige Verwendung von unterhaltsamen Lernspielen könnte durchaus dazu führen, dass dadurch der Lernprozess von weniger unterhaltsamen Stoffgebieten in Mitleidenschaft gezogen werden könnte. Diesbezüglich gibt es allerdings noch keine Studien, welche diesen Effekt genauer untersucht hätten, es bleibt also lediglich bei einer Hypothese.

Diesem von Okhan beschriebenen Effekt kann man zwar nicht gänzlich von der Hand weisen, allerdings kann man ihm sehr wohl entgegen, dass digital game-based Learning nicht den Anspruch erhebt, den gesamten Unterricht zu ersetzen, sondern lediglich als eine Methode unter vielen anderen gesehen werden muss und nur gelegentlich angewendet werden sollte. Ein Lernspiel ist somit nichts anderes, als ein Werkzeug im Werkzeugkasten des Lehrers, welches er je nach Situation einsetzen kann.

Ein weiterer Kritikpunkt an digitalen Lernspielen ist das Fehlen von langfristig angelegten Studien. Dieser Sachverhalt liegt zwar natürlich daran, dass es erst seit den 80er Jahren ernstzunehmende Lernspiele gibt, allerdings ist er nicht von der Hand zu weisen (vgl. Granic et al. 2014, S. 9–10).

Der Großteil der Studien beschäftigt sich lediglich mit kurzfristigen Lernleistungen und überprüft keine Langzeiterneffekte. Dadurch verhärten sich natürlich auch die Fronten zwischen den Befürwortern und den Gegnern des digital game-based Learnings, da es somit der Interpretation des jeweiligen Forschers überlassen ist, ob Lernspiele auch langfristig sinnvoll sind oder nicht.

## **4.3 Vorteile des digital game-based Learnings**

### **4.3.1 Steigerung der Motivation**

Christa Gebel et al schreiben, dass es doch genial wäre, wenn Spiele einen Mehrwert für die Weiterbildung hätten und somit die ins Spiel versunkenen Stunden auf das Konto der Weiterbildung gebucht werden könnte (vgl. Gebel et al. 2005, S. 241).

Auch Gebel et al sehen eine der größten Stärken des digital game-based Learnings in der Steigerung der Motivation der Lernenden da sie Folgendes schreiben:

*Gewinnversprechend stellt sich z.B. das Ziel dar, die Faszinationskraft und Flexibilität der Computerspiele für den Transport von Lerninhalten und zur Vermittlung von (z. B. Denk-) Techniken zu nutzen, indem multimediale Lernprogramme mit den Motivationsprinzipien konstruiert werden, wie sie auch in Computerspielen Anwendung finden. (Gebel et al. 2005, S. 241)*

Sie schreiben außerdem, dass sich bei digital game-based Learning unter anderen die folgenden drei Elemente positiv auf die Motivation des Lernenden auswirken: Interaktivität, Involvement und Ästhetik (vgl. Gebel et al. 2005, S. 268, 361).

Auch Granic et al sehen ein gewaltiges Potential bezüglich der Motivationsförderung in Computerspielen. Ihre Beschreibung von Gamedesignern könnte man ebenso als Beschreibung eines guten Lehrers verwenden:

*Game designers are wizards of engagement. They have mastered the art of pulling people of all ages into virtual environments, having them work toward meaningful goals, persevere in the face of multiple failures, and celebrate the rare moments of triumph after successfully completing challenging tasks. (Granic et al. 2014, S. 5)*

Spiele sollen also die Spieler (im Sinne des Involvements) fesseln, ihnen sinnvolle Ziele geben, aus Fehlern lernen lassen und den Mut dadurch nicht verlieren zu lassen, vor allem aber Erfolgserlebnisse bieten, wenn der Spieler eine Aufgabe gemeistert hat.

Sie schreiben, dass gute Spiele die Verteilung zwischen Herausforderung und Frustration einerseits und Erfolgserlebnissen andererseits optimal implementieren und dadurch ein hohes Maß an Motivation im Spieler hervorgerufen wird (vgl. Granic et al. 2014, S. 6). Hier verweisen Granic et al auf die Wichtigkeit, das in Kapitel 3.4 beschriebene Flow- Erlebnis beim Designen von Lernspielen anzuwenden.

Granic et al gehen hierbei sogar so weit, dass sie sagen, dass der Lernende durch die Verwendung von Computerspielen ein inkrementelles Verständnis von Intelligenz entwickelt und dadurch wesentlich besser mit Misserfolgen umgehen kann als jemand, der dieses inkrementelle Intelligenzverständnis nicht hat. Die Forscher meinen mit einem inkrementellen Intelligenzverständnis, dass der Lernende Intelligenz nicht als etwas angeborenes Unveränderliches betrachtet, sondern vielmehr als etwas Kultivierbares sieht, das er durch ein ausreichendes Maß an Training verbessern kann.

Wenn diese These stimmt, würde das ein gewaltiges Potential für überforderte Schüler darstellen, die bereits relativ demotiviert sind, da sie das Gefühl haben, nicht intelligent genug zu sein, um mit den Klassenkollegen mithalten zu können.

#### **4.3.2 Verbesserung der Problemlösestrategie**

Granic et al weisen darauf hin, dass Menschen, die verhältnismäßig viele Computerspiele spielen, dazu tendieren, bessere Problemlösestrategien zu entwickeln (vgl. Granic et al. 2014, S. 4).

Die Forscher führen diesen Umstand darauf zurück, dass Spieler in Computerspielen meistens nicht einfach durch lineare Instruktionen lernen, wie sie bestimmte Aufgaben lösen können, sondern durch Experimentieren beziehungsweise Trial and Error (vgl. Granic et al. 2014, S. 4).

Schell geht hierbei noch weiter indem er Spiele als Problemlöseaktivität, welche mit einer spielerischen Einstellung angegangen wird definiert (vgl. Schell 2010, S. 37).

Allerdings räumen Granic et al dabei sehr wohl ein, dass es darauf ankommt, welche Form von Computerspielen gespielt wird. Ein klassischer First Person Shooter führt beispielsweise zu keinen besseren Problemlösefertigkeiten, dafür aber zu einer erstaunlichen Augen-Hand Koordination. Strategiespiele oder Rollenspiele, bei denen Rätsel gelöst werden müssen und Strategien langfristig vorausgeplant werden, führen allerdings zu einer deutlichen Steigerung der Problemlösekompetenz.

Besonders imposant ist hierbei ein Durchbruch in der Biologie, der durch die effiziente Nutzung der Problemlösefähigkeit von Computerspielern herbeigeführt wurde.

Dabei entwickelte die Universität von Washington ein Online-Spiel namens Foldit (Cooper et al. 2010), in der die Spieler ein Protein möglichst gut falten mussten. Dazu waren keine Vorkenntnisse im Bereich der Biologie notwendig. Nach einem dreiwöchigen Wettbewerb hatten die besten Spieler geniale Lösungsansätze und die Wissenschaftler konnten die Ergebnisse dieses Wettbewerbs dazu verwenden, um das seit über zehn Jahren bestehende, scheinbar unlösbare Problem zu lösen.

Granic et al möchten hier aber den Mehrwert von digital game-based Learning noch nicht zu hoch anpreisen und räumen ein, dass definitiv noch mehr Forschung in diese Richtung betrieben werden muss, um festzustellen, ob und wie stark sich diese Form des Lernens tatsächlich positiv auf die Lernleistung auswirkt und welche Nebenwirkungen es mit sich bringt.

#### **4.4 Eignung von Spielen zur Verwendung im Unterricht**

Maya Pivec, eine Befürworterin des digital game-based Learnings, fasst die ihrer Ansicht nach wesentlichen Vorteile von Lernspielen im Unterricht folgendermaßen zusammen:

*Games can provide the motivation to learn, increasing the likelihood that the desired learning outcomes will be achieved. Learning is defined as the acquisition of knowledge or skills through experience or practice, and what better way to learn than through a game. (Pivec 2007, S. 422)*

Pivec sieht dabei den größten Vorteil von Lernspielen in der Erhöhung der Motivation der Lernenden. Dabei stimmen ihr sowohl die Befürworter als auch die Gegner des digital game-based Learnings zu.

Julie Wood, eine Professorin der Harvard Universität, schreibt in ihrem Werk „Can software support children’s vocabulary learning“ (Wood 2001, S. 184), dass neue Technologien besonders im Bereich des Vokabellernens gewaltiges Potenzial haben. Durch die Entwicklungen im IT-Bereich können neue mediale Möglichkeiten genutzt werden, die mit Printmedien in dieser Form nicht realisiert werden können. Wood verweist dabei besonders auf interaktive Animationen, da sie darin den Hauptvorteil von spielebasiertem Lernen sieht.

Sie behauptet weiter, dass spieleähnliche Medien die Aufmerksamkeit der Kinder effektiver stimulieren als Schul- oder Übungsbücher. Durch externe Stimulationen wie Grafiken, Animationen und Belohnungen werden die Kinder motiviert, sich intensiver mit den Themen auseinanderzusetzen (Wood 2001, S. 184).

Es kann davon ausgegangen werden, dass Wood einen eher behavioristischen Zugang zur Thematik hat und daher Lernsoftware als ausgezeichnetes Mittel für instrumentelle Konditionierung sieht. Instrumentelle Konditionierung fördert jedoch eine sehr externale Motivationsregulation. Die Forschungsergebnisse von Ryan und Deci zeigen jedoch, dass Lernen aufgrund intrinsischer Lernmotivation bessere Ergebnisse hervorbringt, als Lernen aufgrund externaler Motivation (vgl. Ryan und Deci 2000, S. 56–65).

Da Lin et al allerdings festgestellt haben, dass ein gewisses Maß an externaler Motivation sehr wohl sinnvoll ist, ist die Argumentation Woods zwar valide, allerdings sollte man sich davor hüten, Animationen und Grafiken einen zu hohen Stellenwert zu geben.

### **Fehlende Langzeituntersuchungen**

Da es bezüglich der Verwendung von Lernspielen zum Erwerb von Programmiersprachen noch relativ wenige langfristig angesetzte Untersuchungen gibt, wird in diesem Kapitel zunächst der Fokus auf dem Erlernen einer Fremdsprache anhand von Spielen gelegt, da dieser Bereich etwas besser untersucht ist, um



ausgehend davon auf ähnliche Effekte im digital game-based Learning schließen zu können.

Jentges führte 2007 eine Langzeituntersuchung zum Thema „Spielebasierter Fremdsprachunterricht“ (Jentges 2007) durch. Sie stellte bei ihren Untersuchungen fest, dass durch Spiele zwar kurzfristig signifikant bessere Behaltensleistungen erzielt wurden, langfristig konnten jedoch keine deutlich besseren Erfolge bezüglich der Behaltensleistungen festgestellt werden.

Jentges verwendete in ihrer Untersuchung analoge Spiele. So hat sie beispielsweise Memory-Spiele entworfen, bei denen die Schüler Bilder mit Wörtern verknüpfen. Weiters verwendete sie bei der Experimentalgruppe ausschließlich Spiele und hat gänzlich auf konventionelle Lerntechniken verzichtet. Idealerweise werden Spiele im Lernprozess jedoch nur unterstützend eingesetzt, um die Theorie auf vielfältige Weise zu festigen.

Folglich kann man aus ihren Untersuchungen schließen, dass spielebasiertes Lernen kein Allheilmittel ist. Rein behavioristische Lernmechanismen, wie sie im Spiel Memory, das Jentges unter anderen in ihrer Studie verwendet hat, vorkommen, führen langfristig zu keinen besseren Lernprozessen. Obwohl durch behavioristische Lernspiele zwar kurzfristig bessere Behaltensleistungen erzielt werden, wird langfristig ungefähr derselbe Lernerfolg erzielt wie bei herkömmlichen Unterrichtsmethoden.

Jentges setzte sich im Zuge ihrer Arbeit intensiv mit der Frage auseinander, inwieweit Lernspiele einen Platz im Unterricht haben sollen. Sie kommt zu dem Schluss, dass der Einsatz von Lernspielen für den Lernerfolg weder eine Verbesserung noch eine Verschlechterung mit sich bringt. Allerdings haben Lernspiele den Vorteil, dass sie den Unterricht auflockern und die Schüler etwas lernen, ohne dass diese es bewusst wahrnehmen (vgl. Jentges 2007, S. 115–122).

## **Fazit**

Die Verwendung von Lernspielen im Unterricht führen bei den Lernenden zu einer verbesserten Motivation und Problemlösestrategie.

Aus den Erfahrungen und Berichten von Jentges lässt sich ableiten, dass Lernspiele ein sehr gutes Mittel sind, um den tristen Schulalltag aufzulockern und den Unterricht

abwechslungsreicher zu gestalten, den Schülern aber dennoch etwas beigebracht wird. Demnach liegen die Vorteile von Lernspielen nicht primär im besseren Lernerfolg, sondern vielmehr in der Abwechslung zum Schulalltag.

In jedem Fall warnen die meisten Forscher, die sich mit dieser Thematik auseinandergesetzt haben, vor einem übermäßigen Einsatz von Lernspielen, da ansonsten das Spiel zur Arbeit wird.

Shakespeare bringt diese Problematik bereits in „König Heinrich IV“ mit folgendem Satz sehr treffend auf den Punkt:

„Wenn alle Tag‘ im Jahr gefeiert würden, so würde Spiel so lästig sein wie Arbeit“ (Shakespeare 2015, S. 12).

#### **4.4.1 Probleme bei der Verwendung im Unterricht**

Das Hauptproblem, warum so wenige digital game-based Learning Spiele im Unterricht verwendet werden, ist, dass es schlicht und einfach kaum qualitativ hochwertige Angebote gibt, die sich zudem auch noch für den Unterricht eignen. Granic et al verweisen außerdem darauf, dass die wenigen Spiele, die es gibt, leider zu selten wissenschaftlich evaluiert werden, wenn sie Folgendes schreiben:

*The most important reason is that very few of the games developed to improve health and learning outcomes are ever scientifically evaluated. As a result, it is still unknown how much more effective these games are in changing behavioural and health outcomes than are conventional approaches. (Granic et al. 2014, S. 9)*

Die Forscher meinen, dass es sehr schwer ist, Studien über die Verwendung von Computerspielen im Bildungssektor zu finden. Als eine Folge daraus ist es auch noch relativ schwierig, etwas über die langfristigen Auswirkungen der Verwendung von Computerspielen im Unterricht zu sagen.

Hinzu kommt, dass es auch nur relativ wenige wirklich professionell implementierte Spiele gibt, welche sich zur Verwendung im Unterricht eignen. Das Problem dabei ist, dass Firmen, die Computerspiele entwickeln, selbstverständlich gewinnorientiert arbeiten. Gleichzeitig ist aber das Bildungsbudget jedes Jahr sehr knapp geschnürt und in Österreich vergeht kaum ein Jahr, in dem man nicht in der Presse lesen muss,

dass das Bildungsministerium sein Budget wieder um eine halbe Millionen Euro überschritten hat. Die Schulen haben somit normalerweise keine Ressourcen für Lernsoftware.

Dadurch entsteht natürlich ein Teufelskreis: Die Schulen haben keine finanziellen Mittel für Lernsoftware und dadurch auch keine sonderlich große Nachfrage nach Lernsoftware. Dadurch wird natürlich auch keine entsprechende Software entwickelt, da keine größere Firma das Risiko eingehen möchte, ein Produkt ohne entsprechenden Markt zu entwickeln. Aus diesem Grund entsteht der Großteil der gängigen Lernsoftware auch bloß aus Hobbyprojekten von engagierten Lehrern, die zufälligerweise auch programmieren können.

Diesbezüglich gibt es einige wenige Ausnahmen, welche in Kapitel 4.5 genauer dargestellt werden.

Ein weiterer wesentlicher Grund, warum es so wenige qualitativ hochwertig entwickelte Spiele für den Unterricht gibt, ist laut Van Eck, dass diejenigen, die diese Spiele entwickeln, sehr oft keine Gamedesigner sind (vgl. van Eck 2006, S. 3).

Unglücklicherweise fehlt diesen Praktikern dann sehr oft ein Gefühl dafür, was ein Spiel interessant und unterhaltsam macht. Van Eck spricht hierbei von „akademisierten“ Lernspielen, wie sie in den vergangenen Jahrzehnten entstanden sind (vgl. van Eck 2006, S. 3). Er meint damit Spiele, die keinen Spaß mehr machen und dadurch die schlimmsten Eigenschaften von Spielen, also die Ablenkung, und Lernen, die Anstrengung, in sich vereint (vgl. van Eck 2006, S. 3).

Granic et al vergleichen diese Spiele im folgenden Zitat sehr treffend mit schokoladeüberzogener Brokkoli:

*In an effort to pull together a set of valid principles or lessons, games for health and education often end up with the “chocolate-covered broccoli” problem— the games look great, they are good for you, but they ultimately fail to work because the creative game dynamics that induce transportation and immersion are missing, making them simply not fun. (Granic et al. 2014, S. 9)*

Wenn das entwickelte Spiel keinen Spaß macht, dann ist es nicht sehr viel mehr als eine herkömmliche Lernsoftware. Der größten Vorteile von Computerspielen, die Motivationssteigerung und das Involvement, gehen dadurch dann allerdings natürlich verloren.

## **4.5 Praktische Beispiele zum digital game-based Learning**

### **4.5.1 Code Combat**

Eines der wahrscheinlich interessantesten Projekte im Bereich des digital game-based Learnings für Programmiersprachen ist zweifelsohne Code Combat. Code Combat ist ein Spiel, basierend auf HTML5, welches online auf der Website [www.codecombat.com](http://www.codecombat.com) gratis gespielt werden kann.

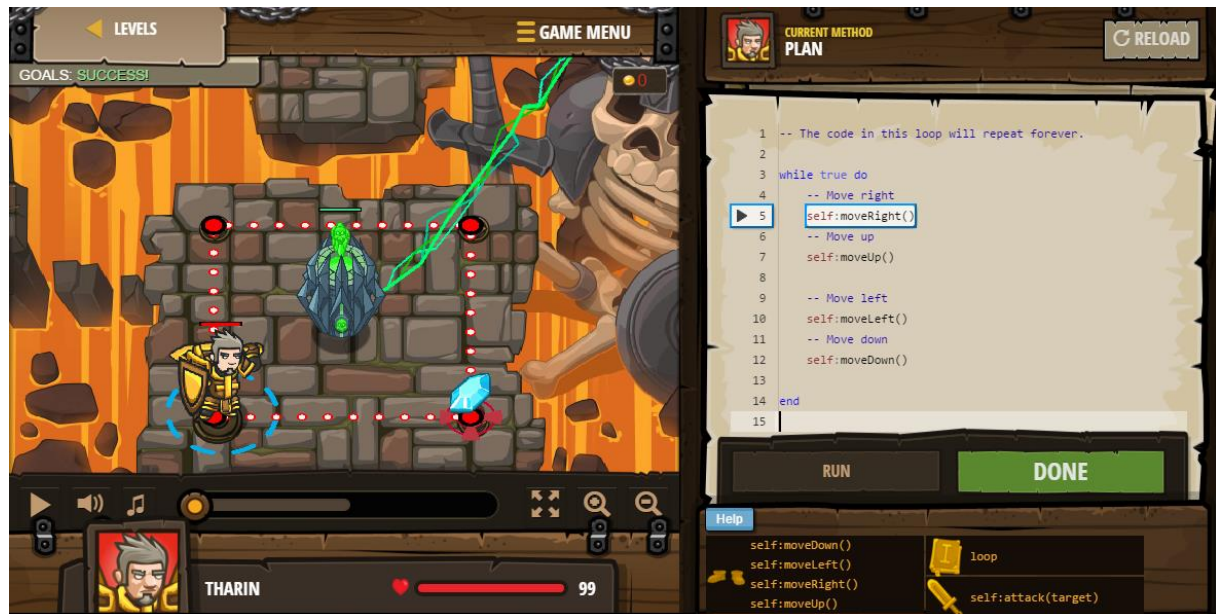
Die Entwickler beschreiben das Spiel folgendermaßen:

*“CodeCombat is a platform for students to learn computer science while playing through a real game”. (CodeCombat)*

Es ist durchaus interessant, dass die Entwickler des Spiels hierbei dezidiert darauf hinweisen, dass es sich dabei um ein richtiges Spiel handelt. Sie verweisen deshalb so klar darauf, weil der Großteil der Lernspiele in Wirklichkeit keine Spiele sind, sondern vielmehr, um es mit den Worten von Granic et al zu sagen, schokoladeüberzogene Brokkoli, also herkömmliche Lernsoftware mit hübschen Grafiken (vgl. Granic et al. 2014, S. 9).

Codecombat fällt definitiv nicht in die Kategorie des „schokoladeüberzogenen Brokkoli“, da das Spiel tatsächlich sehr interessante Spiel- und Motivationsmechaniken hat, die den Spieler dazu ermutigen, den Code möglichst optimal zu schreiben.

So gibt es beispielsweise für jedes Level eine Belohnung in Form von einer Währung, um welche man sich im Laufe des Spiels Ausrüstung kaufen kann. Jedes Level hat aber zusätzlich zur Standardbelohnung noch weitere Belohnungen. Wenn der Spieler beispielsweise nur sehr wenige Zeilen Code benötigt oder bestimmte Zusatzaufgaben erledigt wurden, bekommt er am Ende des Levels mehr Währung gutgeschrieben.



**Abbildung 8 Code Combat- Übersicht**

**Quelle: [www.codecombat.com](http://www.codecombat.com)**

In Abbildung 8 wird ein Level des Spiels veranschaulicht. Dabei sieht man, dass der linke Teil des Bildschirms die konkrete Aufgabenstellung darstellt, während im rechten Teil des Bildes der Code eingegeben wird, um die Spielfiguren zu steuern. Dem Spieler wird dabei zu Beginn des Levels immer die Aufgabenstellung erklärt und der Spieler muss die Spielfiguren mittels Code die jeweiligen Aufgaben lösen lassen.

Dabei gibt es Methoden, wie `moveRight()` oder `moveLeft()`, welche den Spieler nach rechts, bzw. nach links bewegen. Das Spiel beginnt also sehr intuitiv und wird zunehmend immer komplexer. Dabei wird das Flow-Prinzip, welches bereits in Kapitel 3.4 genauer erläutert wurde, angewandt.

Der Spieler lernt das Programm also Schritt für Schritt und der Schwierigkeitsgrad steigt immer nur langsam an. In jedem Level wird entweder ein neues Konzept vorgestellt oder ein bestehendes Konzept in einem etwas neuen Licht betrachtet.



Abbildung 9 Code Combat – Aufgabenbeschreibung

Quelle: [www.codecombat.com](http://www.codecombat.com)

Abbildung 9 veranschaulicht die Hilfestellungen, die einem das Spiel zu Beginn eines Levels anzeigt. Dabei werden die Schüler darauf hingewiesen, dass sich in diesem Beispiel die Anwendung von Schleifen sehr gut eignet, um das Problem zu lösen.

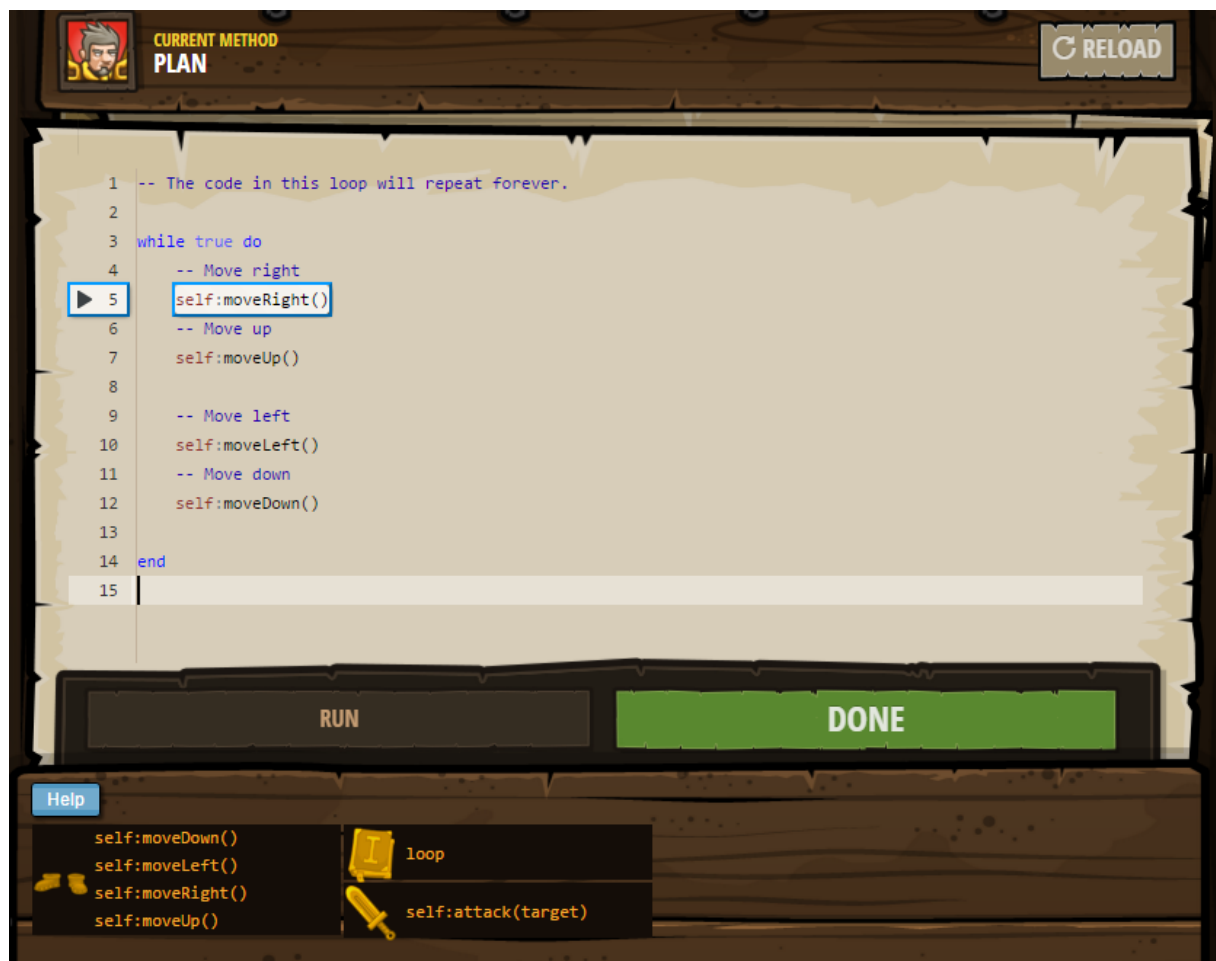
Wichtig dabei ist, dass das jeweilige Level auch geschafft werden kann, ohne Schleifen zu verwenden, allerdings bekommt man in diesem Fall weniger Punkte.

Die Belohnungsmechanismen dieses Spieles sind generell sehr gut implementiert, da es bei jeder Aufgabe Challenges gibt, die dem Spieler Bonuspunkte einbringen. So

gibt es bei vielen Levels die Challenge, z.B. mit nur 9 Zeilen Code die Aufgabe zu lösen.

Diese Mechanik der Bonusaufgaben ist eine exzellente Möglichkeit, wie man mit demselben Lernstoff sowohl eher herausgeforderte als auch begabte Schüler gut fördern kann, ohne die jeweils anderen Schüler zu benachteiligen.

Mit den erworbenen Punkten kann man sich bessere Ausrüstung für die Charaktere kaufen, um später schwierigere Levels schaffen zu können.



**Abbildung 10 Code Combat – Codefenster**

**Quelle: [www.codecombat.com](http://www.codecombat.com)**

In Abbildung 10 sieht man das Codefenster, in welchem der Spieler programmiert, welche Aktionen die Charaktere des Spiels ausführen sollen. Dabei werden dem Spieler im unteren Teil des Codefensters Hilfestellungen bereitgestellt. Wenn der

Spieler die Maus über diese Hilfestellungen bewegt, wird ihm ein Beispiel zur jeweiligen Funktion angezeigt. Diese Beispiele sind allerdings nicht immer ideal gewählt, damit Neueinsteiger in die Programmierung damit etwas anfangen können.

Grundsätzlich ist diese Idee sehr gut, da dadurch die Lernenden auch gleich, wie es in der neuen Matura auch gefordert wird, eine Transferleistung erbringen müssen, um den abstrakt formulierten Code auf die konkrete jeweilige Situation anzupassen. Allerdings sind diese Beispiele teilweise etwas zu komplex bzw. unerklärt, als dass sie ein Programmierneuling verstehen würde.

Das Spiel wirbt auf der Startseite damit, dass es sich sehr gut für den Einsatz im Klassenraum eignet, was auch definitiv der Fall ist, allerdings muss der didaktischen Umsetzung des Spieles entgegengehalten werden, dass es unglücklicherweise in sehr vielen Levels so ist, dass bereits ein Teil des Codes gegeben ist, den man lediglich kopieren muss und anschließend Variablennamen ändert.

So führt dies leider dazu, dass man die wesentlichen Befehle der Programmiersprache nicht unbedingt lernt, sondern eigentlich nur kopiert und ersetzt, um das Ziel des Levels zu erreichen. Hierbei wäre es sehr sinnvoll, wenn weniger Code vorgegeben wäre, sodass die Schüler mehr selbst überlegen und lernen müssen, um die Aufgaben zu meistern.

#### **4.5.2 Civilization**

Civilization ist ein Rundenstrategiespiel, bei dem der Spieler eine Zivilisation aus der Steinzeit bis hin ins Informationszeitalter führt. Dabei geht es primär darum, Städte zu bauen und diese möglichst effektiv und produktiv zu entwickeln.





**Abbildung 11 Civilization 5 – Übersicht**

**Quelle:** <http://www.civilization.com/de/games/civilization-v-brave-new-world/>

Laut McCall kann dieses Spiel hervorragend im Unterricht eingesetzt werden, um Geschichte oder Social Studies zu unterrichten (vgl. McCall 2011, S. 1–2).

Bezüglich der Sinnhaftigkeit im Geschichtsunterricht verweist er darauf, dass jede Nation des Spieles einer real existierenden Nation entspricht, eine historische Führungsperson als Vertreter hat und gegründete Städte die Namen realer Städte dieser Nation haben (vgl. McCall 2011, S. 47–48).

So gibt es beispielsweise Washington als Vertreter der USA oder Napoleon als Vertreter Frankreichs. Die Spieler sind dadurch ständig einerseits mit historischen Figuren, andererseits mit den Namen der Städte dieser Nation in Berührung wodurch sie diese unterbewusst verinnerlichen. Das Spiel hat außerdem eine eingebaute Civlopedia, in der die Spieler historische Fakten über diverse Nationen, Weltwunder, historische Persönlichkeiten und Ähnliches nachlesen können.

McCall gibt den Schülern dabei immer einige Fragestellungen auf, welche sie im Lauf der Spielsession bedenken und anschließend in einem Essay beantworten sollen. So fragt er beispielsweise, wie die verschiedenen Regierungsarten (Despotismus,

Monarchie, Republik und Demokratie) in Civilization dargestellt werden und was die Vor- und Nachteile der jeweiligen Regierungsform ist (vgl. McCall 2011, S. 103). Er fragt auch, welchen Einfluss die geographische Lage und der Zugang zu wertvollen Ressourcen in Civilization auf eine Zivilisation hat (vgl. McCall 2011, S. 103). Die Schüler sollen dann recherchieren, inwiefern das Spiel die Realität abbildet und wo die Grenzen der Simulation des Spiels liegen.

Bezüglich politischer Bildung verweist er darauf, dass die Spielmechaniken dermaßen funktionieren, dass der Spieler immer versuchen muss, die Bedürfnisse seiner Bevölkerung zu erfüllen, da eine unzufriedene Bevölkerung schlechter forschet und weniger produziert (vgl. McCall 2011, S. 1–2). Dabei spielen die unterschiedlichsten Faktoren eine Rolle, wie glücklich eine Bevölkerung ist. Beispielsweise kann der Spieler Handel betreiben, um Luxusressourcen zu importieren, welche die Bevölkerung sehr hoch schätzen, sodass diese dadurch glücklicher werden und die Forschung und Produktion gleichsam wieder angekurbelt werden.

Da die Ressourcen in Civilization 5 sehr knapp sind, erfährt der Spieler das ganze Spiel über, was es bedeutet, dass die Ressourcen der Menschheit limitiert sind. Hat ein Spieler beispielsweise keinen Zugang zu Eisen oder Aluminium, so ist er gezwungen, diese Ressourcen relativ teuer zu importieren, was oft auch dazu führen kann, dass die betroffene Nation eher dazu neigt, ein angrenzendes Territorium einer anderen Nation militärisch einzunehmen, um die dort liegenden Ressourcen verwenden zu können. Die Schüler bekommen somit ein Gefühl dafür, was es bedeutet, in einer Welt begrenzter Ressourcen zu leben und welche Konsequenzen das hat (vgl. McCall 2011, S. 47–48).

#### **4.5.3 Comber & Motschnigs Studie zur Entwicklung von Computerspielen**

Comber führte 2015 in Österreich eine enorm interessante Studie durch, bei der sie einer Gruppe von Schülern zunächst die Grundlagen der Programmierung beibrachten und diese Gruppe anschließend in zwei Untergruppen aufteilten. Eine Gruppe sollte die Grundlagen mithilfe eines Gamedevelopment Projektes üben, die andere Gruppe entwickelte ein normales, selbst überlegtes Softwareprojekt (vgl. Comber und Motschnig 2015, S. 1899).

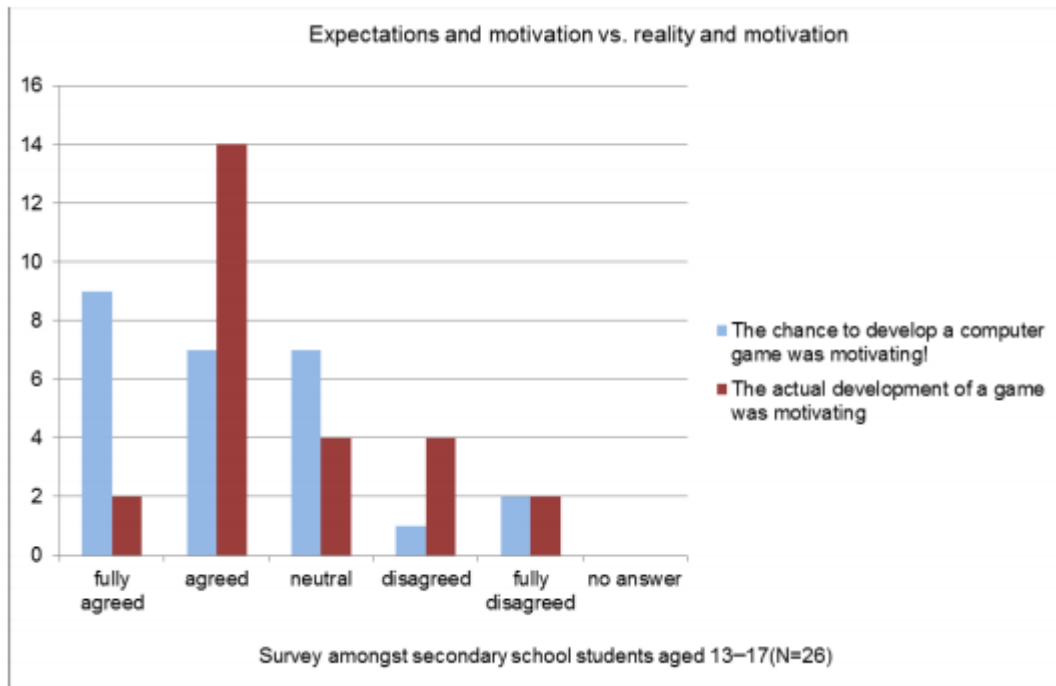
Im Anschluss an die Projekte mussten die Schüler sowohl einen Test schreiben als auch einen Selbstbewertungsfragebogen ausfüllen. Die Ergebnisse dieser beiden Evaluationen zeigen, wie positiv sich der Einsatz von Computerspielen im Unterricht auswirken kann.

Allerdings muss man bezüglich dieser Untersuchung beachten, dass die Untersuchungsgruppe sehr klein war und dadurch auch keine allgemein gültige Aussage getroffen werden kann, allerdings kann sehr wohl, wie Comber und Motschnig schreiben, eine Tendenz festgestellt werden (vgl. Comber und Motschnig 2015, S. 1899).

Es ist weiter darauf hinzuweisen, dass in dieser Studie Computerspiele nicht zum Unterrichten eingesetzt wurden, sondern es wurde im Zuge eines Projektes ein Computerspiel programmiert. Es ging in dieser Studie also nicht um game-based Learning, sondern eher darum, welchen Einfluss die Verwendung von Computerspielen allgemein im Unterricht hat.

Abbildung 12 veranschaulicht den Vergleich zwischen der Motivation, die die Schüler angaben, als ihnen die Möglichkeit geboten wurde, ein Computerspiel zu entwickeln und der tatsächlichen Motivation, als sie am Spiel entwickelten.

Dabei sieht man, dass die Schüler eine etwas höhere Motivation erwarteten, als sie dann tatsächlich wahrnahmen, allerdings war die Motivation insgesamt dennoch sehr hoch.

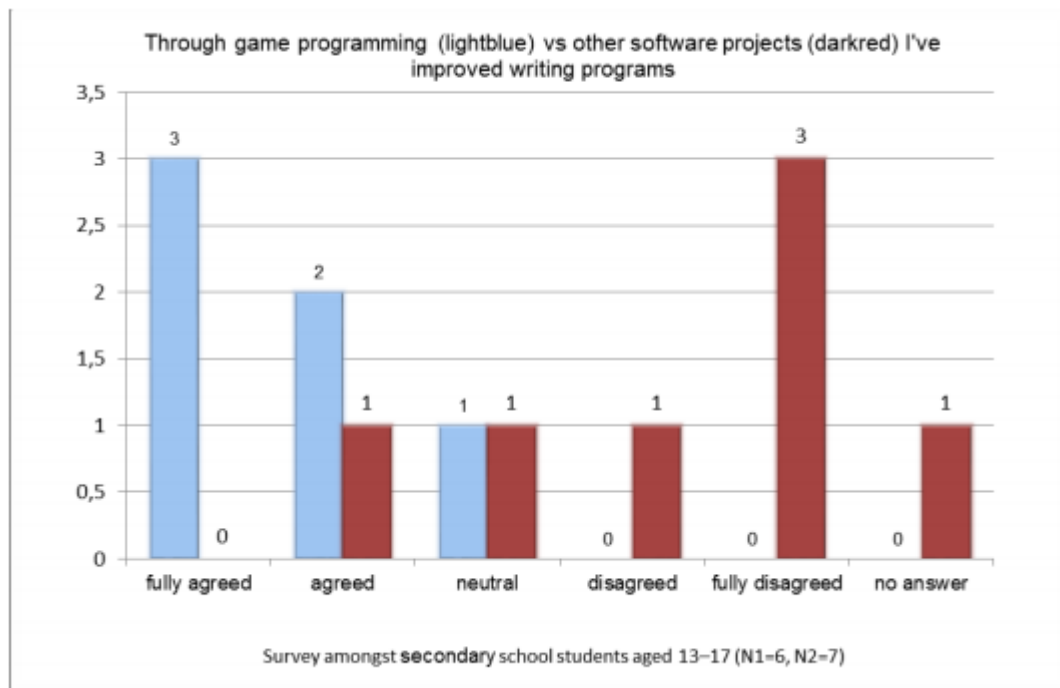


**Abbildung 12 Erwartete und tatsächliche Motivation der Schüler**

**Quelle: (Comber und Motschnig 2015, S. 1897)**

Abbildung 13 veranschaulicht, wie sich die Schüler selbst einschätzten, als sie gefragt wurden, wie sehr sie sich in ihren Programmierfähigkeiten verbessert haben. Dabei ist klar zu erkennen, dass sich die Schüler, die das Spiel programmierten, deutlich besser einschätzten, als diejenigen Schüler, die ein normales Softwareprojekt entwickelten. Der schriftliche Test stimmte hierbei mit der Selbsteinschätzung überein.

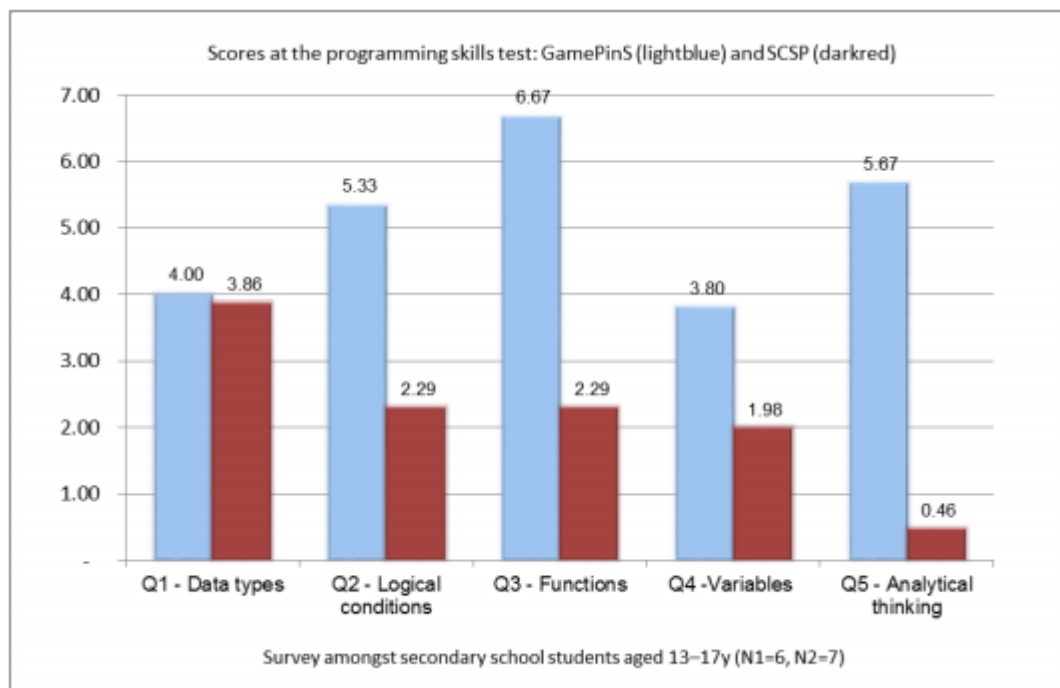
Der Unterschied könnte stärker kaum sein. Die Verwendung von Computerspielen scheint in diesem Fall eine enorm positive Auswirkung auf die Selbstevaluation der Schüler gehabt zu haben, da die Referenzgruppe, die ein normales Softwareprojekt entwickeln musste, angab, dass sich ihre Programmierfähigkeit dadurch kaum verbessert hat.



**Abbildung 13 Selbst wahrgenommene Verbesserung der Programmierskills**

**Quelle: (Comber und Motschnig 2015, S. 1900)**

Abbildung 14 veranschaulicht zuletzt noch die wichtigste Erkenntnis dieser Untersuchung, nämlich, dass die Schüler, die ein Computerspiel programmierten, in allen Fragen des Theorietests deutlich besser, mit durchschnittlich mehr als doppelt so vielen Punkten, abgeschnitten haben als die Kontrollgruppe (vgl. Comber und Motschnig 2015, S. 1902).



**Abbildung 14 Punkte am Programmiertest**

**Quelle: (Comber und Motschnig 2015, S. 1901)**

Wie bereits erwähnt, untersuchten Comber und Motschnig zwar nicht den Einsatz von Computerspielen als Lernmittel, sondern die Entwicklung von Computerspielen als Anreiz für Projektarbeiten, aber man kann die zumindest kurzfristigen positiven Auswirkungen der Verwendung von Computerspielen im Unterricht hierbei nicht leugnen.

Die Einbindung von Computerspielen und damit der direkte Bezug zur Welt der Schüler und Schülerinnen, hatte in dieser Studie einen enorm positiven Einfluss auf den Lernerfolg der Lernenden. Diese Studie bestätigt, dass sich Computerspiele auch im Programmierunterricht hervorragend eignen, um den Lernerfolg zu steigern.

## 5 Grundlagen - Gamedesign

### 5.1 Wichtige Elemente eines Computerspiels

Jesse Schell beschreibt in seinem Buch *The Art of Gamedesign* (Schell 2010), dass es nicht die Aufgabe eines Gamedesigners ist, ein Spiel zu machen, sondern es geht vielmehr darum, eine Erfahrung für die Spieler zu entwerfen (vgl. Schell 2010, S. 10). Er weist weiter darauf hin, dass das Spiel nicht der Erfahrung gleichgesetzt werden darf. Das Spiel selbst ist also nur ein Mittel zum Zweck. Schell bringt dies mit folgender Aussage zur Geltung:

*If we could, through some high-tech magic, create experiences for people directly, with no underlying media – no game boards, no computers, no screens – we would do it. [...] For now, we live in the present, where all we can do is create artifacts (rule sets, game boards, computer programs) that are likely to create certain kinds of experiences when a player interacts with them. (Schell 2010, S. 11)*

Dieser Ansatz ist bei der Entwicklung eines Lernspiels absolut zentral. Es geht nicht darum ein Spiel zu entwickeln, sondern vielmehr darum, eine bestimmte Form der Erfahrung für die Schüler zu entwickeln.

Schell nennt einige wichtige Elemente, die ein Computerspiel laut der Literatur ausmachen. Unter anderen nennt er folgende grundlegende Eigenschaften von Spielen: (vgl. Schell 2010, S. 34)

1. Sie werden freiwillig gespielt
2. Sie haben Ziele
3. Sie beinhalten Konflikte
4. Sie haben Regeln
5. Sie können gewonnen oder verloren werden
6. Sie sind interaktiv
7. Sie bieten eine Herausforderung
8. Sie haben ihren eigenen internen Wert
9. Sie fesseln den Spieler

## 10. Sie sind geschlossene formale Systeme

Schell versucht aus diesen unüberschaubaren Eigenschaften die wesentlichsten zusammenzufassen und nennt letztendlich seine, zugegebenermaßen stark verkürzte, Definition von Spielen als Problemlöseaktivität, die mit einer spielerischen Einstellung angegangen wird (vgl. Schell 2010, S. 37).

Etwas konkreter betrachtet besteht ein Computerspiel bei Schell aus 4 grundlegenden Elementen, die sich idealerweise gegenseitig ergänzen sollten:

### **Mechaniken**

Mechaniken beschreiben die Regeln, Möglichkeiten, sowie das Ziel des Spieles. Mechaniken sind jenes Merkmal, welches das Spiel von anderen Medien unterscheidet, da sowohl Filme, Bücher oder ähnliche lineare Medien normalerweise keine Interaktion mit dem Medium bieten (vgl. Schell 2010, S. 41).

### **Geschichte**

Jedes Spiel erzählt eine Geschichte, die den Spieler fesseln und interessieren sollte und damit den Spieler involviert.

### **Ästhetik**

Die Ästhetik eines Spieles wirkt sich neben den Mechaniken wohl am deutlichsten auf die Erfahrung des Spielers aus (vgl. Schell 2010, S. 52). Selbst bei gleichbleibenden Mechaniken kann eine andere Ästhetik die Erfahrung des Spieles völlig verändern.

### **Technologie**

Die Technologie bestimmt, auf welche Art und Weise der Spieler mit dem Spiel interagiert. Steuert er den Charakter mit einem Lenkrad, einer Tastatur oder einem Joystick? Läuft das Spiel auf einem 5 Zoll Smartphone oder auf einem 30 Zoll Bildschirm? Die verwendete Technologie ist somit ein wesentlicher Aspekt, der die Erfahrung des Spieles maßgeblich beeinflusst.

Schell schreibt, dass sich gute Spiele dadurch auszeichnen, dass alle vier Aspekte das Leitmotiv des Spieles unterstützen (vgl. Schell 2010, S. 52).



Bisher wurden allerdings nur grundlegende Bausteine eines Computerspieles behandelt. Prensky listet in seinem Buch digital game-based Learning unter anderen folgende Eigenschaften eines guten Computerspiels auf (vgl. Prensky 2001, S. 134–136):

1. Eine klare Vision des Spieles
2. Ein konstanter Fokus auf der Erfahrung des Spielers
3. Eine starke Struktur
4. Sehr adaptiv
5. Leicht zu lernen, schwer zu meistern
6. Führt zu einer Flow-Experience
7. Bietet regelmäßige Belohnungen, keine Strafen
8. Beinhaltet Erkundung und Entdeckung
9. Lösen einzelner Probleme führt dazu, dass andere Probleme auch einfacher werden
10. Sinnvolles Interface
11. Möglichkeit den Spielstand zu speichern

Prensky versucht damit, einen groben Überblick über die wichtigsten Eigenschaften von guten Spielen zu geben. Dabei gibt er selbst an, dass diese Liste natürlich keinen Anspruch auf Vollständigkeit hat.

### **Guidelines zur Erstellung eines Lernspieles**

Elisabeth Weißenböck hat bezüglich digitaler Lernspiele unter anderen folgende Guidelines zur Erstellung eben dieser Spiele ausgearbeitet (Weißenböck 2014, S. 146–147):

- Die Grafik eines Lernspiels soll die Zielgruppe ansprechen und zum Thema passen.
- Informationen können auf unterschiedliche Weise in ein Spiel integriert werden: durch Finden von Objekten, durch Ausprobieren, durch Dialoge etc.
- Jedes Lernziel hat unterschiedliche Anforderungen, der gewählte Typ des Spiels soll das Lernziel unterstützen.

- Narration, Spiel, Spaß und Lernprozess sollen so miteinander verflochten sein, dass das Lernen nicht in den Vordergrund tritt und dies somit für die Lernenden nicht offensichtlich ist.
- Nachdem eine Aufgabe gelöst/bearbeitet wurde, soll Feedback erfolgen, das den/die Lernende/n unterstützt.
- Das Feedback muss an das Wissen des/der SpielerIn angepasst sein und dazu anregen zu reflektieren.
- Komplexe Lernspiele benötigen ein Tutorial bzw. Introduction levels zu Beginn. Reines Lernen durch Bedienungsanleitung soll vermieden werden.
- Das Verhältnis Qualifikation zu Unterhaltung soll ausgeglichen sein.

Weißböck liefert damit sehr sinnvolle Guidelines, anhand welcher in Kapitel 6.5 - *Spieldesign* das Lernspiel, das im Zuge dieser Arbeit entstanden ist, analysiert wird.

## 5.2 Psychologische Phänomene beim Spielen

Jason Vandenberghe, ein Creative Director bei Ubisoft, einer der größten Computerspiele Publisher weltweit, versuchte in einem Vortrag auf der Game Developers Conference 2013 die Big Five der Psychologie auf Gamedesign anzuwenden (vgl. Vandenberghe 2013, 0:00 - 0:05).

Die Big Five sind eine unter Psychologen weithin akzeptierte Persönlichkeitstheorie, die die Persönlichkeit eines Menschen in diese fünf Dimensionen aufteilt. Diese fünf Dimensionen heißen in der deutschen Übersetzung Neurotizismus, Extraversion, Offenheit für Erfahrungen, Gewissenhaftigkeit und Verträglichkeit (vgl. McAdams 1997, S. 25).

Vandenberghe untersuchte, welche Bereiche von Computerspielen verschiedenen Personen zusagten und stellte dabei fest, dass diese Big Five mit fünf Bereichen von Computerspielen korrelierten. Er stellt diese folgendermaßen gegenüber:

- Openness of Experience – Novelty
- Conscientiousness – Challenge
- Extraversion – Stimulation
- Agreeableness – Harmony
- Neuroticism – Threat

(vgl. Vandenberghe 2013, 0:02)

Indem Vanderberghe zeigte, dass bestimmte Spielbereiche mit Persönlichkeitstypen korrelieren, kann man versuchen, diese verschiedenen Bereiche der Spiele gezielt einzubauen, sodass die Spiele ein möglichst breites Publikum ansprechen.

### 5.3 Wie Spiele den Spieler motivieren

Vanderberghe postuliert, dass sich Spieler in ~4 Kategorien unterteilen lassen, welche mit vier Persönlichkeitstypen der Big Five korrelieren (Neuroticism korreliert unglücklicherweise nur schlecht mit bestimmten Spielbereichen und wird deshalb ausgelassen) (vgl. Vandenberghe 2013). Er schreibt, dass jeder dieser vier Persönlichkeitstypen durch jeweils zwei primäre Motivatoren Spaß an Spielen entwickelt (vgl. Vandenberghe 2013, 0:20 - 0:25). Vanderberghe beschreibt diese folgendermaßen (vgl. Vandenberghe 2013, 0:25 - 0:50):

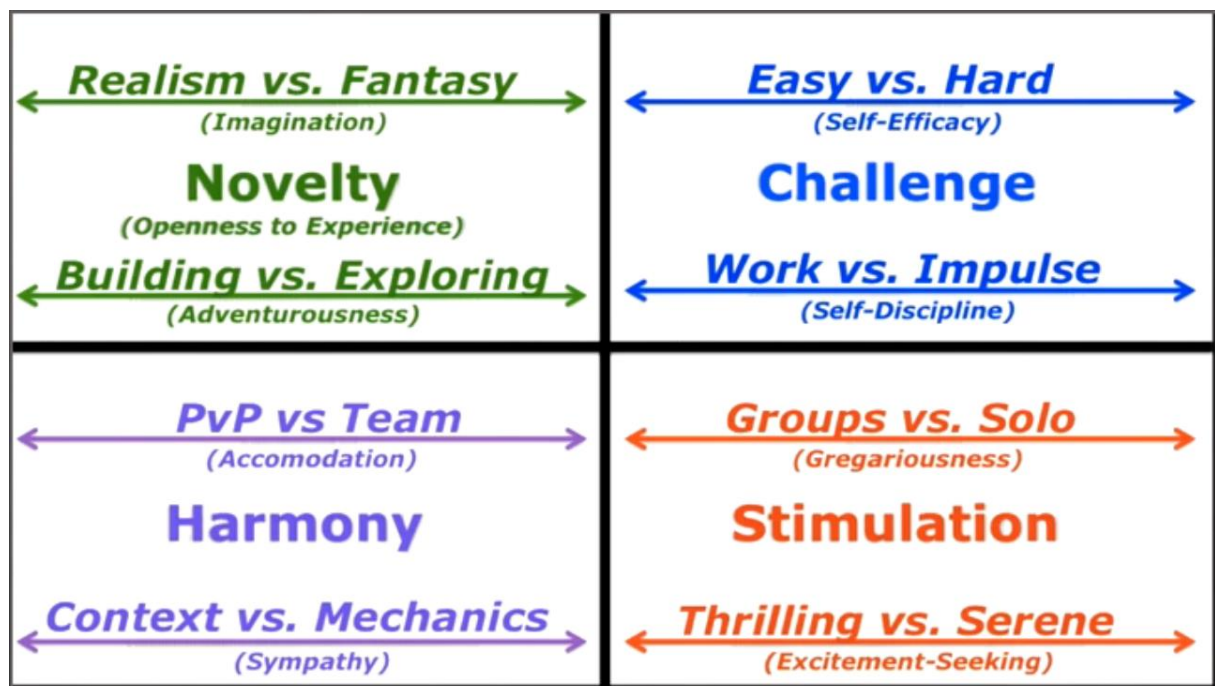


Abbildung 15 Vandenberghes Domains of Play

Quelle: (Vandenberghe 2013, 0:25)

**Novelty:** Menschen, die hohe Werte im Bereich der Openness of experience haben, beziehen ihre Motivation meistens aus den folgenden beiden Gegenüberstellungen:

- Realismus vs Fantasy – Leute, die realistische Spiele mögen, schätzen Simulatoren, die die Realität möglichst genau abbilden, während Fantasy-affine Spieler eher in völlig unbekannte Welten eintauchen möchten (vgl. Vandenberghe 2013, 0:26 - 0: 31).
- Building vs Exploration – Einerseits gibt es Menschen, die gerne etwas Bleibendes schaffen und einen Einfluss auf die virtuelle Welt nehmen möchten, das sind die Builder. Die Explorer hingegen beziehen ihre Motivation primär aus dem Erkunden von Welten bzw. dem Aufdecken von Geheimnissen (vgl. Vandenberghe 2013, 0:26 - 0: 31).

**Challenge:** Menschen mit hohen Conscientiousness-Werten beziehen ihre Motivation meist aus den Dimensionen

- Easy vs Hard – Manche Spieler möchten ein Spielerlebnis möglichst ohne Rückschläge, während andere Spieler auf eine echte Herausforderung bestehen, an welcher man, wenn man auch nur kleine Fehler begeht, sofort scheitert (vgl. Vandenberghe 2013, 0:31 - 0:36).
- Work vs Impuls – Vor allem im asiatischen Raum beziehen Spieler einen großen Teil der Motivation daraus, dass man im Spiel für bestimmte Belohnungen lange hinarbeiten muss. Andere Spieler möchten jedoch lieber eher impulsiv agieren und auch für kleine Aufgaben belohnt werden (vgl. Vandenberghe 2013, 0:31 - 0:36).

**Harmony:** Menschen, die ausgeprägte Werte im Bereich der Agreeableness aufweisen, werden durch folgende Gegenüberstellungen motiviert:

- Spieler gegen Spieler vs Teamplay – Einerseits gibt es Spieler, denen es wichtig ist sich mit anderen Spielern zu messen, andererseits gibt es Spieler, die Motivation daraus beziehen, dass sie im Team gemeinsam etwas schaffen (vgl. Vandenberghe 2013, 0:41 - 0:50).
- Context vs Mechanics – Contextbezogene Spieler brauchen Gründe und Kontexte, warum sie bestimmte Handlungen ausführen. Diese Spieler beziehen einen Großteil ihrer Motivation aus der Geschichte des Spiels. Mechanics-

bezogene Spieler legen keinen Wert darauf, warum sie etwas machen, vielmehr sollte die Mechanik, bzw. die Aufgabe, in sich interessant sein (vgl. Vandenberghe 2013, 0:41 - 0:50).

**Stimulation:** Eher extrovertierte Menschen lassen sich durch diese Dimensionen motivieren:

- Groups vs Solo – Gruppenbezogene Spieler versuchen eher mit anderen Spielern zu interagieren während Solospieler lieber auf soziale Interaktionen verzichten (vgl. Vandenberghe 2013, 0:36 - 0:41).
- Thrilling vs Serene – Thrilling-affine Spieler beziehen Motivation daraus, dass ständig etwas Spannendes auf dem Bildschirm passiert. Serene-affine Spieler streben eher die Ruhe und Entspannung an (vgl. Vandenberghe 2013, 0:36 - 0:41).

Ein Verständnis dieser verschiedenen Bereiche von Spielen und die Auswirkungen auf die Motivation der Spieler ist laut Vandenberghe essentiell, um ein Spiel an die jeweilige Zielgruppe anzupassen (Vandenberghe 2013).

Möchte man die Zielgruppe also möglichst weit fassen, wie es im Fall von Lernspielen aufgrund der wachsenden Heterogenität in den Schulen zwingend notwendig ist, so muss versucht werden, so gut es geht immer einen Mittelweg zwischen den beiden Enden der verschiedenen Skalen zu finden. Beispielsweise sollte versucht werden, einen Kompromiss aus Kontext und Mechaniken zu finden. Damit einerseits die Mechaniken interessant und spannend sind, diese aber gleichzeitig in einen sinnvollen Kontext eingebettet werden.

## 6 Beschreibung des erstellten Lernspieles

### 6.1 Lehrplanbezug

Der Lehrplan für Informatik sieht in der 5. Klasse der AHS folgenden Lehrstoff vor:

*Einblicke in wesentliche Begriffe und Methoden der Informatik, ihre typischen Denk- und Arbeitsweisen, ihre historische Entwicklung sowie ihre technischen und theoretischen Grundlagen gewinnen und Grundprinzipien von Automaten, Algorithmen und Programmen kennen lernen. (bmbf, S. 2)*

Dieser Punkt kommt unglücklicherweise sehr oft in den Schulen zu kurz, da der Fokus (zurecht) auf dem Erwerb grundlegender Kenntnisse in der Bürosoftware liegt. So hat ein Großteil der Schüler leider nur sehr wenig Ahnung, wie ein Computerprogramm überhaupt funktioniert.

Durch dieses Lernspiel erwerben die Schüler ein grundlegendes Verständnis, wie die Denk- und Arbeitsweisen in der Informatik ablaufen. Außerdem bekommen sie einen kurzen Einblick in simple Algorithmen.

Das Spiel ist aber eigentlich für Schüler konzipiert, die nicht nur einen kurzen Einblick in die Programmierung machen, sondern für Schüler, die die Grundlagen der Informatik lernen sollen. Dabei soll das Lernspiel den Einstieg etwas interessanter und unterhaltsamer gestalten.

Deshalb macht es für dieses Spiel eher Sinn, sich auf den Lehrplan einer Informatik HTL zu beziehen. Es heißt beispielsweise im Lehrplan der HTL Spengergasse bezüglich des Kompetenzbereiches Programmiertechniken folgendermaßen:

*Die Schülerinnen und Schüler*

- kennen die Elemente und die grundlegenden Techniken des Programmierens;[...]*
- können die richtigen Programmiertechniken für konkrete Probleme auswählen und umsetzen;*

*- können Problemlösungen für konkrete Aufgabenstellungen konzipieren und Programme selbständig entwerfen und realisieren.*

*Lehrstoff: I . Jahrgang : Anweisungen; Kontrollstrukturen; Datentypen; Funktionen, Prozeduren; Fehlerbehandlung.*

*(HTL Spengergasse, S. 8)*

Hier deckt das Lernspiel bereits einige der geforderten Punkte ab: Die Schüler lernen dadurch grundlegende Programmiertechniken, sie können damit konkrete Aufgabenstellungen lösen und vertiefen ihr Verständnis von Anweisungen, Datentypen und bis zu einem gewissen Grad auch Funktionen.

Das Spiel kann somit in diesem Fach sehr gut als Einstieg in die Programmierung verwendet werden.

## **6.2 Bezug zu Lerntheorien**

Das Spiel verwendet unterschiedliche Aspekte der verschiedenen Lerntheorien. Grundsätzlich war es das Ziel des Spiels, einen konstruktivistischen Zugang zu finden, indem dem Spieler möglichst viel Freiraum gelassen wird und ihm die Möglichkeit gegeben wird, Aufgaben auf verschiedene Arten zu lösen. Deshalb wurde auch ein Laufzeitcompiler entwickelt, der dem Spieler erlaubt sämtliche Funktionalitäten der Programmiersprache C# zu nutzen, um die verschiedenen Aufgaben zu lösen.

Allerdings stellte es sich als sehr schwierig heraus, Aufgaben zu entwickeln, welche dem Spieler zwar einen großen Freiraum bezüglich des Lösungsweges lassen, sodass er frei experimentieren kann, er aber trotzdem gefordert ist, die neu erworbenen Erkenntnisse anzuwenden.

Bei Aufgabe 5 des ersten Levels ist die Verwendung der konstruktivistischen Lerntheorie am augenscheinlichsten, da dem Spieler darin sehr viele Lösungsmöglichkeiten offenstehen und er dazu aufgefordert ist zu experimentieren.

Allerdings sind besonders die ersten Aufgaben eher restriktiv was die Experimentiermöglichkeiten betrifft, da der Spieler zunächst die Grundlegenden Funktionalitäten des Spiels und jene der Programmierung erwerben muss.

Was das 2. Level betrifft, so wurde dort großer Wert darauf gelegt, die Aufgaben nach dem derzeitigen Kompetenzmodell der berufsbildenden höheren Schulen zu gestalten. So sind die Schüler bei den Aufgaben hauptsächlich gefordert, Transferleistungen zu erbringen, indem die abstrakten Programmierkonzepte, die im Theorieunterricht zuvor erlernt wurden, in konkreten Situationen selbstständig anzuwenden.

### **6.3 Realisierung & Implementierung**

Das Spiel wurde mithilfe der Unity3D Engine (Unity3D 2016) entwickelt, die besonders für kleine Teams und Prototypen gut funktioniert, da man sehr schnell zu Ergebnissen kommt.

Als Programmiersprache wurde C# verwendet, da sich C# bei der Entwicklung von Unity3D – Projekten als Industriestandard etabliert hat. Es wäre zwar auch möglich gewesen, das Spiel in Javascript zu programmieren, allerdings bringt die Verwendung von C# einige wichtige Vorteile, wie beispielsweise bessere Performance, bessere vorgefertigte Libraries etc., mit sich.

Es wurde zunächst ein Prototyp entwickelt, der das Grundkonzept des Spieles veranschaulichen sollte. Für diesen Prototyp wurde zunächst die Scriptsprache LUA verwendet, die zwar relativ einfach und simpel, allerdings für die Verwendung in der Praxis eher irrelevant ist. Daher wurde basierend auf dem C# Interpreter von Tiaan Geldenhuys (Geldenhuys 2016) ein Laufzeitcompiler entwickelt, der es ermöglicht, den C# Code, den die Spieler im Spiel eingeben, zu kompilieren und auszuführen.

Der Interpreter von Geldenhuys musste allerdings auf die neueste Version von Unity3D aktualisiert werden, außerdem wurden mit Geldenhuys Einverständnis einige kleine Bugs behoben. Insgesamt wurde diese Aufgabe allerdings zeitmäßig deutlich unterschätzt. Alleine die Anpassung an die neue Unity3D Version und die Implementierung in das Lernspiel nahmen ungefähr zwei der vier Wochen Entwicklungszeit in Anspruch.

Der Großteil der Spiele, die Programmierfertigkeiten vermitteln, funktioniert nicht mit einem Compiler, sondern lediglich durch Texterkennung und Vergleichen mit den richtigen Lösungen.



Da dem Spieler allerdings die größtmögliche Freiheit im Spiel gewährt und kreative Problemlösungen gefördert werden sollen, war es ein besonderes Anliegen, dass die Schüler, sofern sie die notwendigen Skills haben, in der vom Spiel bereitgestellten Lernumgebung alles machen können, was sie wollen und die Aufgaben auf die verschiedensten Arten lösen können sollen.

Die Spieler können also, wenn sie bereits Vorkenntnisse haben, die Aufgabenstellungen auf völlig andere Arten lösen, als es die Aufgabe vorgibt. In den meisten Lernspielen gibt es diese Möglichkeit nicht, da normalerweise nur genau ein richtiger Codesatz als richtig erkannt wird.

Dieser Laufzeitcompiler funktioniert dank des Monoframeworks wunderbar mit Unity3D und erlaubt es den Spielern mit exakt denselben Tools zu arbeiten, wie der Entwickler selbst.

Als Versionskontrollsystem wurde GIT mit Sourcetree als Client und Bitbucket als Webhost verwendet.

Die Entwicklung des Spiels wurde in vier Wochen mit je ~50 Stunden abgeschlossen, wobei ungefähr zwei Wochen für die Entwicklung des grundlegenden Prototyps und den Laufzeitcompiler aufgebracht und jeweils eine Woche für die Entwicklung des Gameplays und der Grafik aufgewandt wurden.

Insgesamt ist das Ergebnis in Anbetracht der aufgewandten Zeit sehr gut geworden, allerdings hätte dem Spiel noch ein zusätzlicher Monat Entwicklungszeit sehr gut getan, da der Schwierigkeitsgrad zwischen den Aufgaben leider sehr rapide ansteigt und eine etwas flachere Lernkurve dem Lernerfolg der Schüler sehr gut getan hätte.

## **6.4 Beschreibung des Spieles**

### **6.4.1 Verwendung im Unterricht**

Zunächst muss klar festgehalten werden, dass das Spiel so konzipiert ist, dass es im Anschluss an eine Theorieeinheit zum Thema Datentypen eingesetzt wird. Es dient dabei primär zum Üben der gelernten Techniken und zur Auflockerung des ansonsten sehr trockenen Programmierunterrichts.

Idealerweise startet der Lehrende das Spiel und projiziert es mit einem Beamer an die Tafel. Die ersten beiden Aufgaben werden dann noch mit dem Lehrer gemeinsam gemacht, wobei er den Schülern erklärt, wie das Spiel funktioniert. Ab Aufgabe 3 in Level 1 sollten die Schüler allerdings selbstständig arbeiten.

Dabei sollte der Ablauf dann so aussehen, dass die Schüler zwar selbstständig arbeiten, allerdings lässt der Lehrer nach einer bestimmten Zeit, je nachdem wie schwierig die Aufgabe ist, einen der Schüler die Aufgabe am Beamer lösen, damit alle wieder am selben Stand sind. Danach wird wieder Zeit für die nächste Aufgabe gegeben.

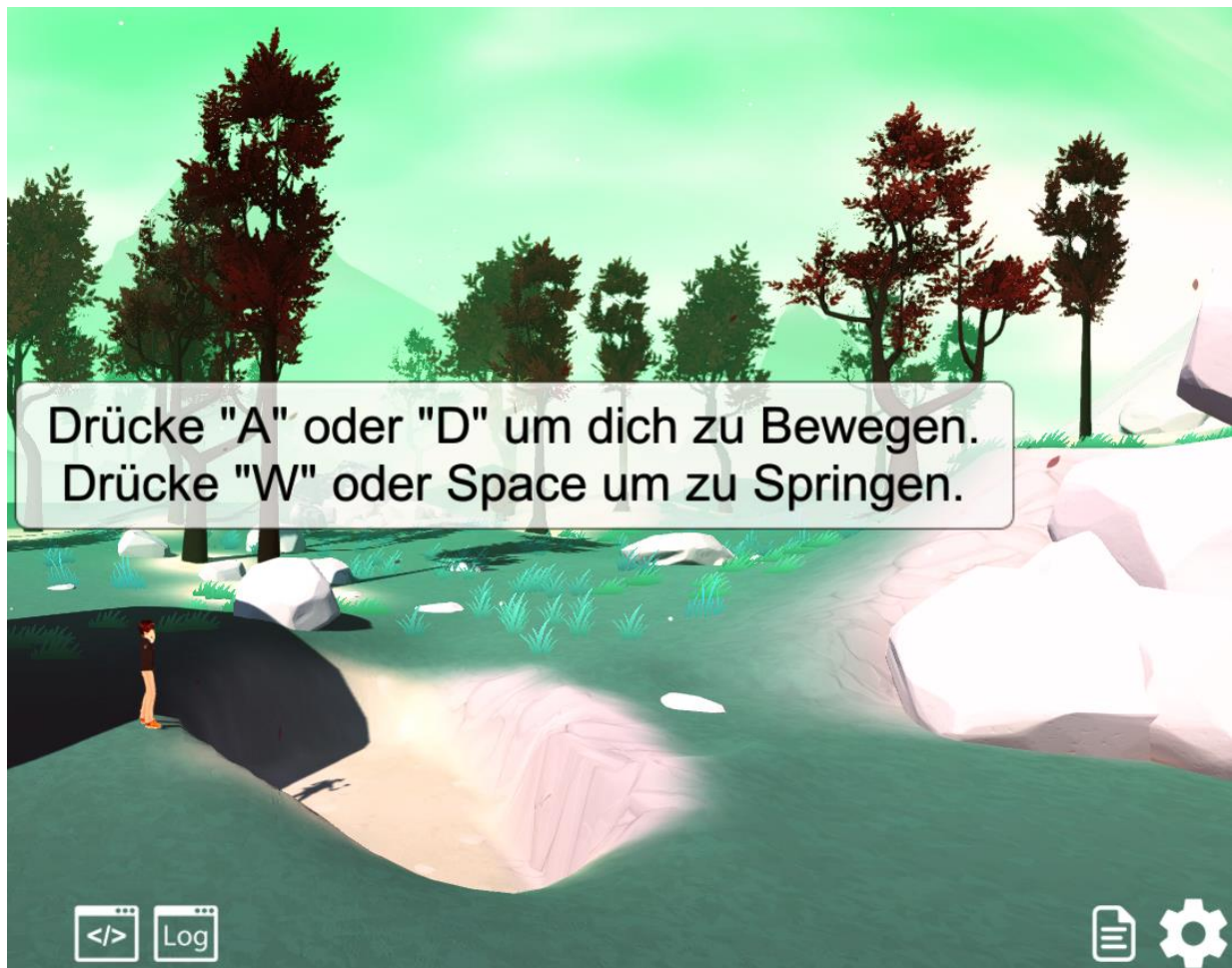
Das Lernspiel kann, sofern es bereits auf den Schülerrechnern installiert ist, innerhalb von ungefähr zwei Einheiten durchgespielt werden und dient zur Übung des gelernten Stoffes.

#### **6.4.2 Level 1**

##### **Tutorial**

**Lernziel:** Der Spieler kann die Spielfigur nach rechts beziehungsweise nach links bewegen und springen.

Im ersten Level wird der Spieler zunächst an die Steuerung und die grundlegende Funktionsweise des Spiels herangeführt. Zuerst wird dem Spieler erklärt, dass er sich mit der „A“ – Taste nach links, und mit der „D“ – Taste nach rechts bewegen kann. Außerdem springt der Charakter, wenn man die Space Taste drückt.



**Abbildung 16 Programmierlernspiel - Erstes Hindernis**

Hat der Spieler diese Grube überwunden, trainiert er die Bewegungs- und Sprungmechaniken noch kurz, indem er einen Steinhaufen erklimmen muss. Diese kleine Sprungaufgabe ist lediglich dazu da, um den Spieler an die Bewegungs- und Sprungmechanik zu gewöhnen, sodass er die Spielerfigur ausreichend gut steuern kann, um die folgenden Aufgaben lösen zu können.

**Sicherstellung des Lernziels:** Der Spieler muss diese erworbenen Kenntnisse anwenden, um die Grube überwinden und den Steinhaufen erklimmen zu können.



**Abbildung 17 Programmierlernspiel – Üben der Sprungmechanik**

## **Level 1 - Aufgabe 1**

### **Lernziele:**

- Der Spieler kann das Codefenster öffnen, es mit einem Objekt verbinden.
- Der Spieler kann Code kompilieren.
- Der Spieler kann den Code zurücksetzen.

Direkt nach diesem Steinhaufen kommt der Spieler bereits zu der ersten richtigen Aufgabe. Das Ziel dieser Aufgabe ist, den Spieler mit dem Codefenster vertraut zu machen und ihm zu erläutern, wie er das Verhalten von Objekten beeinflussen kann.

Die Aufgabe besteht darin, dass eine Felswand den Weg blockiert. Über der Aufgabe ist ein Hinweistext (siehe Abbildung 18, Abbildung 19), der dem Spieler Schritt für

Schritt erklärt, wie er sein Codefenster öffnet, es mit der Aufzugsplattform verbindet und anschließend Code kompiliert.

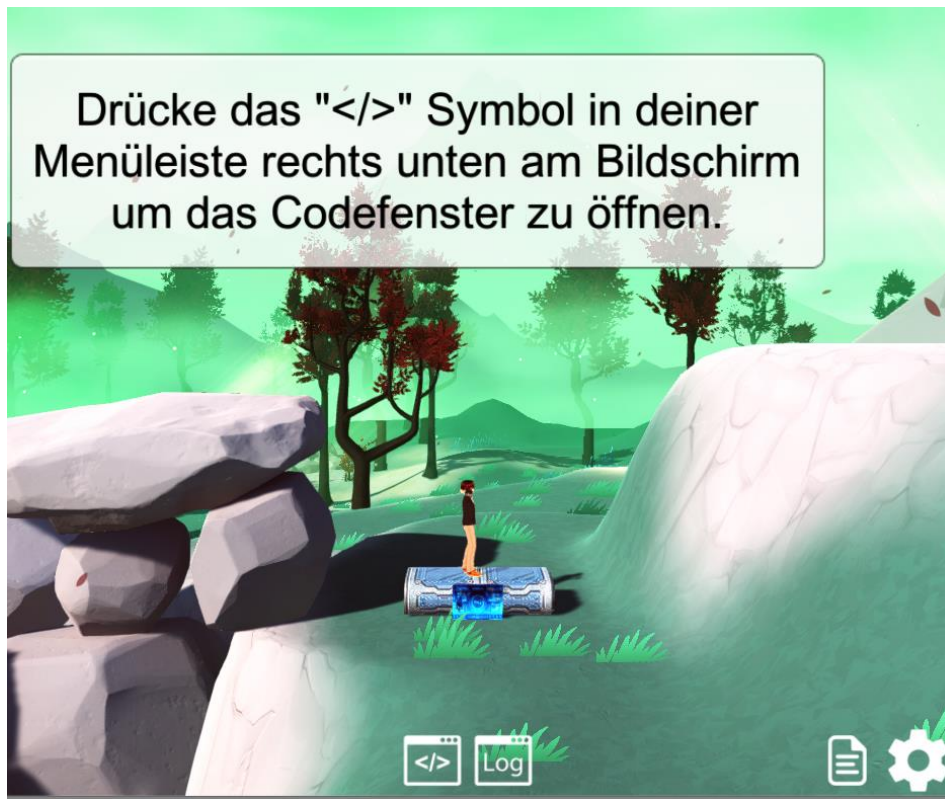


Abbildung 18 Programmierlernspiel Level 1 - Aufgabe 1

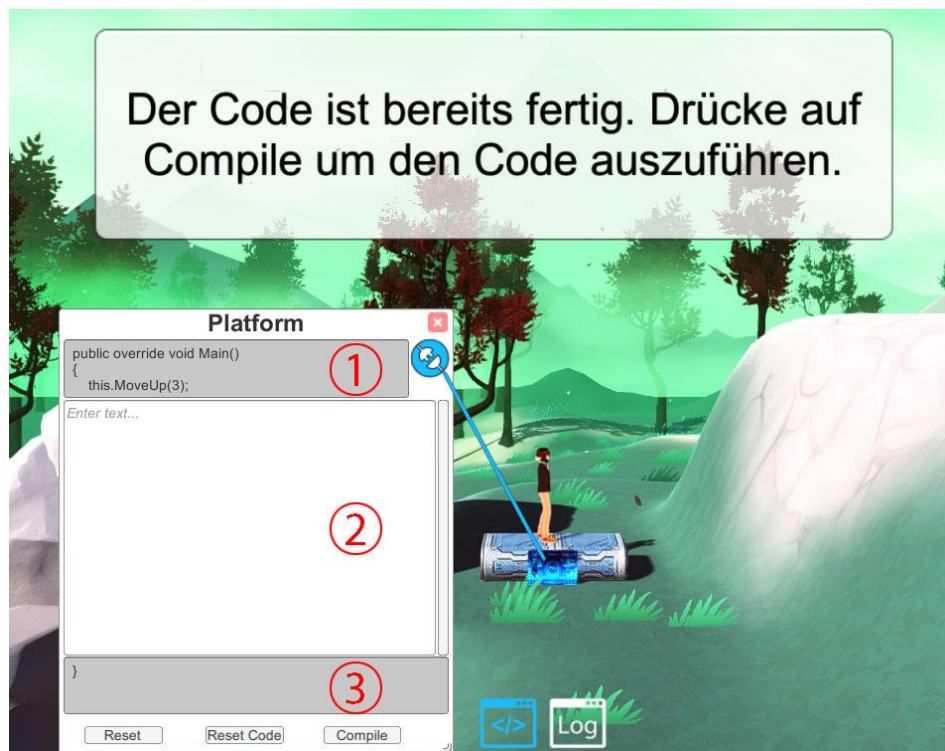


Abbildung 19 Programmierlernspiel Level 1 - Aufgabe 1 mit Codefenster



Dabei sieht der Spieler auch erstmals das Codefenster, in dem er den Code, durch welchen er die Welt manipulieren kann, für das Spiel programmiert.

Um ein Verständnis für das Codefenster zu entwickeln, ist eine Lehrperson unabdinglich. Diese Lehrperson sollte genau an dieser Stelle erläutern, dass beim Codefenster der Code jedes Objektes aus drei Teilen besteht (siehe Abbildung 19). Die Bereiche, die mit 1 und 3 markiert sind, stehen dabei für das Präfix und das Suffix, welche fix vorgegebene Teile des Codes sind, an denen die Schüler nichts verändern können. Der mit 2 markierte Teil ist jener Teil, in welchen die Schüler ihren Code schreiben müssen. Beim Kompilieren werden natürlich automatisch alle drei Teile des Codes zusammengeführt.

Durch diese unveränderlichen Prä- und Suffixe des Codes wird die Aufmerksamkeit des Schülers auf den jeweils wichtigen Teil der Aufgabe gelenkt und es hilft den Schülern vor allem bei den ersten Aufgaben, dass sie sich zunächst nur auf die eine Zeile Code konzentrieren können, die eine Variable deklariert bzw. definiert.

Am unteren Rand des Codefensters finden sich dann noch 3 Buttons: Reset, Reset Code und Compile. Auch diese Buttons sollten vom Lehrer kurz erläutert werden.

- **Reset** setzt die gesamte Aufgabe, und alles, was man daran bisher gemacht hat, auf den Ausgangszustand zurück.
- **Reset Code** setzt nur den Code des aktuell selektierten Objektes auf den Ausgangszustand zurück.
- **Compile** kompiliert den gesamten Code.

Bei Aufgabe 1 muss der Schüler allerdings noch gar keinen Code schreiben, da der Code im Präfix bereits ausreicht, um die Plattform zu bewegen. Es reicht somit, wenn er lediglich auf den Compile-Button klickt, um die Plattform nach oben zu bewegen, sodass der Spieler über die Klippe springen kann.

In dieser Aufgabe geht es noch lediglich darum, dem Spieler das Codefenster vorzustellen und ihm zu zeigen, wie er den Code ausführen kann. Neuen Code zu schreiben, lernt er in der nächsten Aufgabe.

### Sicherstellung der Lernziele:

- Der Lehrer erläutert die Funktionalitäten des Codefensters
- Solange der Spieler den Code nicht kompiliert, kann er den Felshang nicht erklimmen.

### Level 1 - Aufgabe 2



Abbildung 20 Programmierlernspiel Level 1 - Aufgabe 2

### Lernziele:

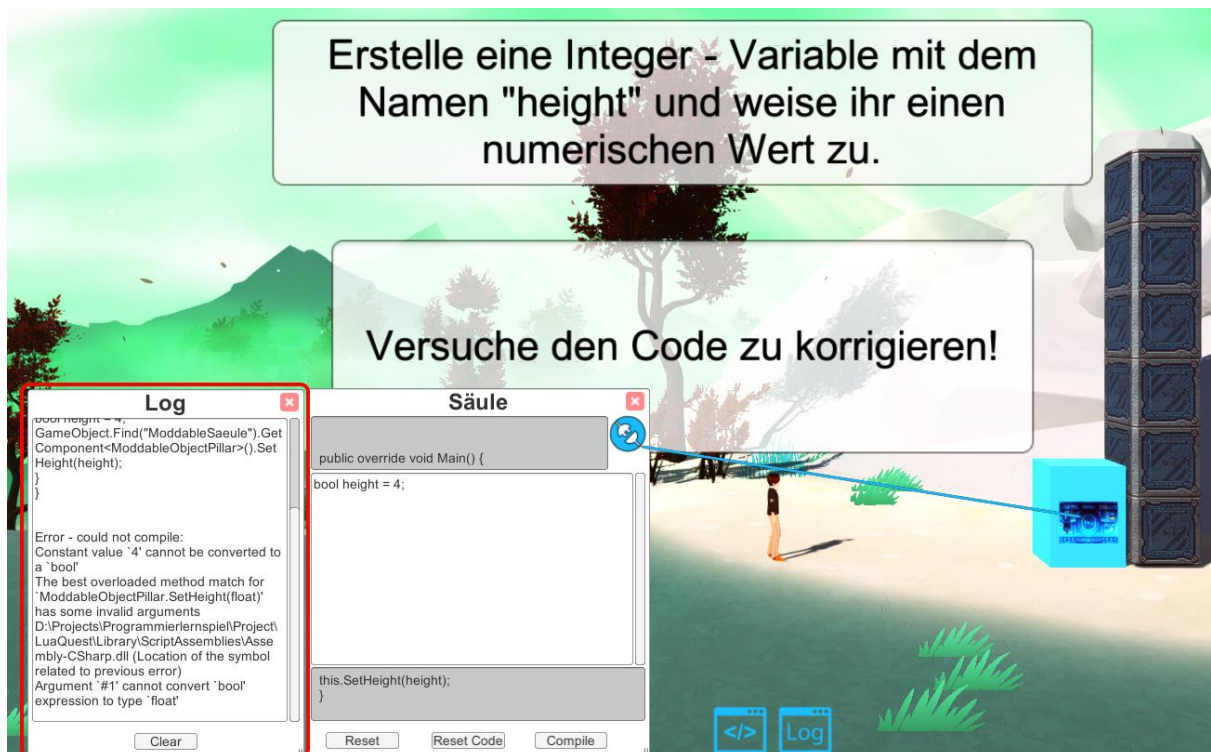
- Der Spieler kann Fehlermeldungen mithilfe des Logfensters grob analysieren.
- Der Spieler kann die Datentypen *Bool* und *Integer* korrekt anwenden, um Variablen zu definieren.

Bei Aufgabe 2 muss der Spieler eine Integer Variable mit dem Namen *height* angeben. Bei dieser Aufgabe empfiehlt es sich, dass sie vom Lehrer gemeinsam mit den Schülern gemacht wird, sodass er sie auf die Besonderheiten des Logfensters aufmerksam machen kann.

Es ist sinnvoll, wenn der Lehrer zunächst die Schüler darauf hinweist, dass diese Variable, die sie hier anlegen müssen, im Suffix folgendermaßen verwendet wird:

`this.SetHeight(height)`. Hier lohnt es sich, die Schüler kurz zu fragen, was diese Zeile Code macht und darüber zu brainstormen.

Wenn die Schüler das Codefenster mit der blauen Säule verbinden, so wird im beschreibbaren Bereich des Codefensters folgende Zeile Code automatisch geschrieben: `bool height = 4;` Natürlich kompiliert diese Zeile nicht, da der Datentyp *Bool* keine Zahlen fassen kann. Der Spieler wird nun von der Tutorialnachricht dazu aufgefordert, das Logfenster zu öffnen.



**Abbildung 21 Programmierlernspiel Level1 – Aufgabe 2 Logfenster**

Beim Logfenster sollte der Lehrer ebenfalls kurz die relevanten Zeilen erläutern, sodass die Schüler zumindest eine grobe Vorstellung davon haben, was das Problem des Codes ist. Die Fehlermeldungen, die hier ausgegeben werden, kommen direkt vom Compiler und sind somit für Einsteiger nur sehr bedingt hilfreich.

Im Zuge des Spieles reicht allerdings immer lediglich die letzte Zeile des Logfensters, um auf das Problem zu schließen. In diesem Fall bekommt man folgende Fehlermeldung:



Argument #1 cannot convert ,bool‘ expression to type ,float‘

Wie man unschwer erkennen kann, ist diese Fehlermeldung ziemlich aussagekräftig. Hierbei können die Schüler durchaus selbstständig arbeiten und versuchen, den Fehler im Code zu beheben. Die Lösung ist natürlich, dass man den Datentyp der Variable *height* von *bool* auf *float* ändert.

Ist dies geschehen, so skaliert sich die blaue Box, sodass sie eine Höhe von 4 Metern hat und damit dem Spieler erlaubt, über das Hindernis zu springen.

### **Sicherstellung der Lernziele:**

- Das Logfenster muss vom Lehrer kurz erläutert werden.
- Der Code kann nur kompiliert werden, wenn alle Fehler aus dem Code entfernt wurden, der Spieler kommt also nicht weiter solange er den Fehler nicht behoben hat.

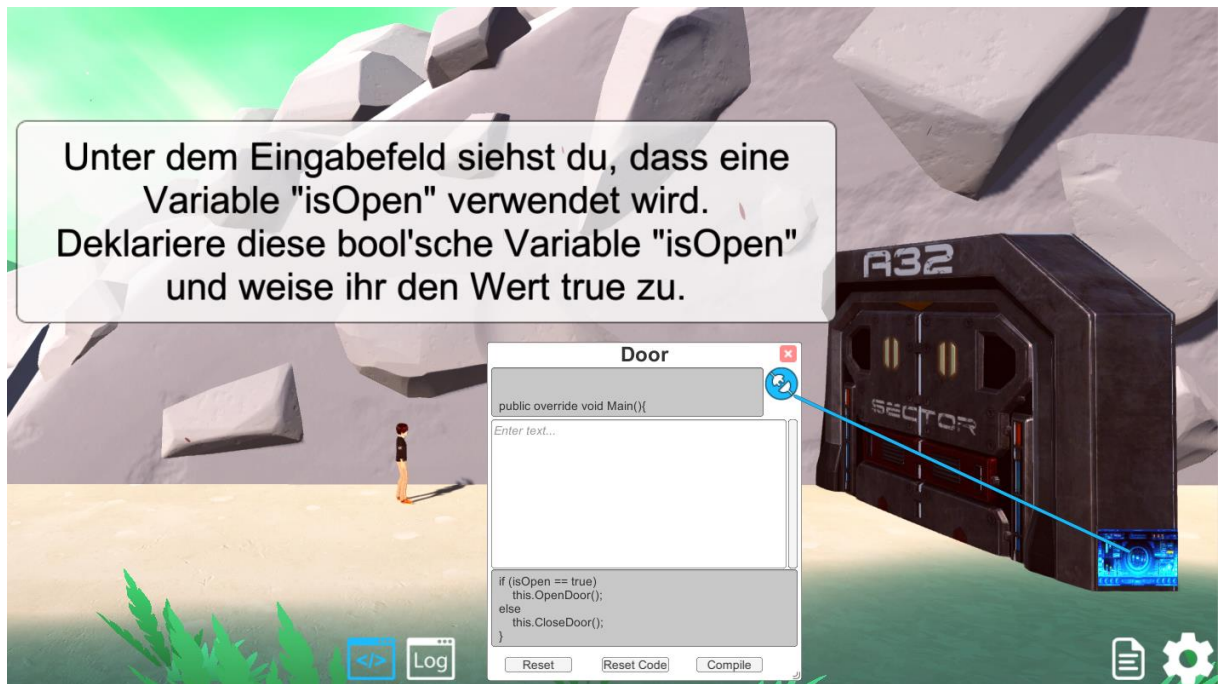
### **Level 1 - Aufgabe 3**

#### **Lernziel:**

- Der Schüler kann selbstständig eine Variable mit dem Datentyp *Bool* deklarieren und ihr einen Wert zuweisen.

Die dritte Aufgabe des Lernspieles kann bereits völlig selbstständig von den Schülern gelöst werden. Dabei ist das Ziel dieser Aufgabe, eine Variable vom Datentyp *Bool* zu erstellen. Die Schüler bekommen diesbezüglich auch in dieser Aufgabe noch sehr klare Handlungsanweisungen, sodass sie zunächst die Deklaration von Variablen ausreichend verstehen und beherrschen.

In späteren Übungen werden die Handlungsanweisungen immer weniger, sodass die Schüler selbst überlegen müssen, wie die jeweilige Aufgabe am besten gelöst werden kann.



**Abbildung 22 Programmierlernspiel Level 1 – Aufgabe 3**

An dieser Aufgabe kommen die Vorteile der unveränderbaren Präfixe und Suffixe des Codefensters zum Vorschein. Die Schüler müssen lediglich eine Zeile Code schreiben, in der die Variable `isOpen` vom Datentyp `Bool` definiert wird. Der Rest der Aufgabe wird im Suffix ausgeführt.

Hat der Schüler die Variable erstellt und klickt auf `Compile`, öffnet sich die Türe und der Weg zur nächsten Aufgabe ist frei.

#### **Sicherstellung der Lernziele:**

- Erst wenn der Spieler die Variable deklariert hat und den Wert auf *true* gesetzt hat, öffnet sich die Tür damit der Spieler zur nächsten Aufgabe kommt.

#### **Level 1 - Aufgabe 4**

##### **Lernziele:**

- Der Spieler kann für Variablen, die bereits bestimmte Werte vorgegeben haben die entsprechenden Datentypen zuweisen.

Bei Aufgabe 4 sind die Anweisungen, die der Spieler bekommt, bereits deutlich kompakter und dadurch auch weniger anweisend. Es wird lediglich gefordert, dass es eine Variable *isActive* mit dem Wert *true* gibt und eine Variable *elevatorSpeed* mit dem Wert 3. Der Schüler muss sich jetzt bereits selbst überlegen, welche Datentypen sich für die jeweilige Variable am besten eignet. Die Aufgabe verlangt vom Schüler somit bereits wesentlich mehr Selbstständigkeit.

Wenn der Spieler die Aufgabe korrekt gelöst hat, beginnt sich die Aufzugsplattform langsam nach oben zu bewegen und der Spieler gelangt zur nächsten Aufgabe.



**Abbildung 23 Programmierlernspiel Level 1 – Aufgabe 4**

#### **Sicherstellung der Lernziele:**

- Der Aufzug aktiviert sich nur, wenn der Code kompiliert, somit ist ihm der Weg zur nächsten Aufgabe versperrt, solange er diese nicht lösen kann.

## Level 1 – Aufgabe 5

### Lernziele:

- Der Spieler kann Variablen nicht nur fixe Werte zuweisen, sondern auch den Wert einer anderen Variable darin abspeichern.
- Der Spieler erkennt, dass es in der Programmierung oft mehrere Möglichkeiten gibt, um den gewünschten Effekt zu erzielen, diese jedoch immer mit verschiedenen Vor- und Nachteilen verbunden sind.

Aufgabe 5 fordert die Schüler nun ganz schön heraus. Die Aufgabe besteht darin, dass beide Autos gleichzeitig am Felsen auftreffen. Die Schwierigkeit dieser Aufgabe besteht allerdings darin, dass eines der beiden Autos eine zufällige Geschwindigkeit hat.

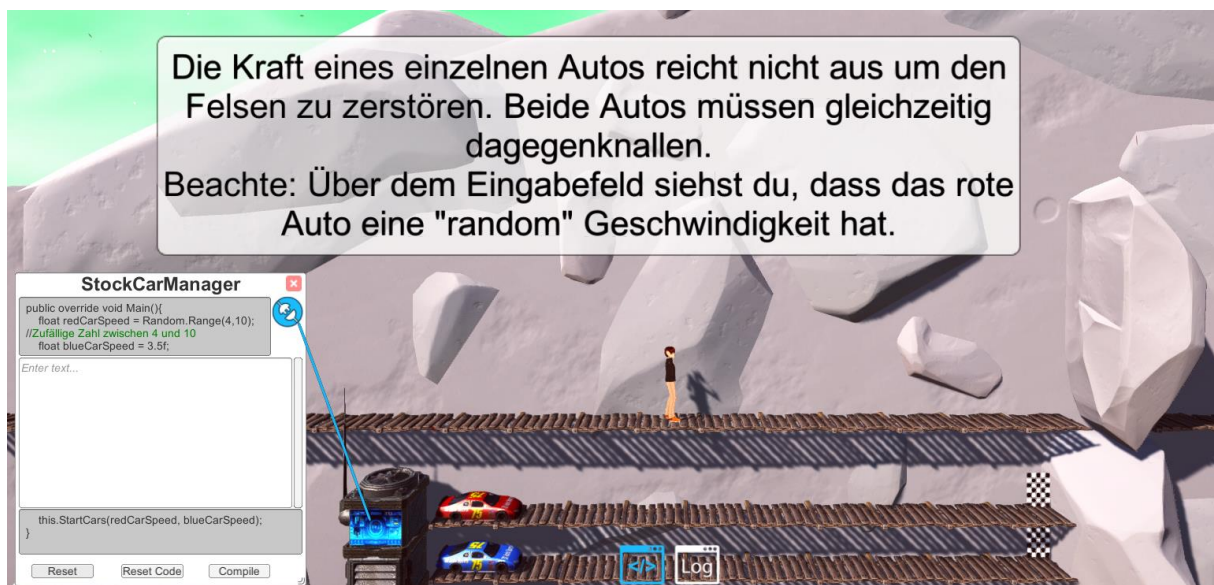


Abbildung 24 Programmierlernspiel Level 1 – Aufgabe 5

Die Schüler müssen hier bereits alles, was sie in den vorangegangenen Aufgaben gelernt haben, selbstständig anwenden können und sich überlegen, wie sie die Aufgabe lösen.

Außerdem gibt es für diese Aufgabe nicht nur einen korrekten Lösungsweg, sondern zumindest drei. Hierbei ist vom Lehrer auch darauf hinzuweisen, dass sich die Schüler, die etwas schneller fertig sind, alternative Lösungsmöglichkeiten überlegen sollen.

Dadurch sind sowohl die langsameren, als auch die fortgeschrittenen Schüler bei dieser Aufgabe herausgefordert.

Außerdem wäre es bei dieser Aufgabe essentiell, dass der Lehrer alle vier Möglichkeiten mit den Schülern durchbespricht und auf die jeweiligen Vor- und Nachteile dieser Lösungen eingeht:

1. Die Geschwindigkeit beider Autos auf einen bestimmten Wert setzen

```
redCarSpeed = 5f;  
blueCarSpeed = 5f;
```

Das ist eine sehr einfache Lösung, die den Nachteil hat, dass die ursprüngliche Geschwindigkeit beider Autos überschrieben wird.

2. Die Geschwindigkeit des roten Autos hardgecodet auf 3.5f setzen

```
redCarSpeed = 3.5f;
```

Diese Lösung ist auch noch relativ einfach gehalten. Sie hat den Nachteil, dass die Geschwindigkeit hardgecodet auf denselben Wert gesetzt wird wie jene des blauen Autos. Besser wäre es aber, wenn die Geschwindigkeit des roten Autos genau auf die Geschwindigkeit des blauen Autos gesetzt wird, unabhängig davon, ob diese 3.5f ist oder nicht (Siehe Lösung 3).

3. Die Geschwindigkeit des roten Autos auf die Geschwindigkeit des blauen Autos setzen

```
redCarSpeed = blueCarSpeed;
```

Im Gegensatz zu Aufgabe 2 ist es in diesem Fall egal, welche Geschwindigkeit das blaue Auto hat, es werden in jedem Fall beide gleich schnell sein. Diese Lösung ist also änderungsrobuster.

4. Die Geschwindigkeit des blauen Autos auf die Geschwindigkeit des roten Autos setzen

```
blueCarSpeed = redCarSpeed;
```

Auch hier gilt dasselbe wie bei Lösungsansatz 3, allerdings ist das Schöne an diesem Ansatz, dass die zufällige Geschwindigkeit des roten Autos bewahrt bleibt und das blaue Auto diese zufällige Geschwindigkeit annimmt.

Nach dem erfolgreichen Abschluss dieser Aufgabe kann der Spieler das Level abschließen und somit Level 2 beginnen.

#### **Sicherstellung der Lernziele:**

- Der Schüler muss zumindest eine der 4 Lösungsmöglichkeiten herausfinden, um an dem Felsbrocken vorbei zu kommen.
- Der Lehrer weist die Schüler darauf hin, dass sie mehrere Möglichkeiten finden sollen, um die Aufgabe zu lösen und bespricht anschließend alle Möglichkeiten mit den Schülern durch.

### **6.4.3 Level 2**

Level 2 setzt beim Schwierigkeitsgrad dort an, wo Level 1 abgeschlossen hat. Die Schüler müssen sich in diesem Level bei jeder Aufgabe selbst überlegen, wie sie diese bewältigen können. Außerdem beginnt Level 2 im Gegensatz zu Level 1 damit, eine kleine Geschichte zu erzählen.

#### **Level 2 – Aufgabe 1**

##### **Lernziele:**

- Der Spieler versteht, dass das Binärsystem keine anderen Zeichen außer 1 und 0 beinhaltet

Zu Beginn des Levels bekommt der Spieler gleich den Hinweis, dass sich der Spieler den Umstand zunutze machen sollte, dass Roboter grundsätzlich sehr exakt



berechnende Wesen sind und mit unvorhersehbaren Ereignissen nicht umgehen können. Kurz darauf stößt der Spieler auf den Roboter *BinaryBot*, der ihm den Weg blockiert.

*BinaryBot* möchte den Spieler nicht weiterlassen, da er im Begriff ist, das Roboterland zu betreten, was organischen Lebewesen nicht erlaubt ist.



**Abbildung 25 Programmierlernspiel Level 2 – Aufgabe 1**

Da der Spieler bereits weiß, dass der Roboter nicht mit unvorhergesehenen Ereignissen umgehen kann, muss er versuchen, sich diesen Umstand zunutze zu machen, um an *BinaryBot* vorbeizukommen.

Den Schülern werden in dieser Aufgabe diverse Hinweise gegeben, dass *BinaryBot* nur binär, also mit Nullen und Einsen, denken kann. Da Roboter also nicht mit

unvorhersehbaren Ereignissen umgehen können und BinaryBot nur binär denken kann, müssen die Schüler daraus schließen, dass man ihn aus der Fassung bringen kann, indem man ihm eine andere Zahl als 0 oder 1 einspeist.

Hier wird es erstmals wichtig für die Schüler das Präfix und das Suffix des Codes etwas genauer zu betrachten. So erkennt man bei diesen Teilen des Codes, dass *BinaryBot* regelmäßig eine zufällige Zahl 0 oder 1 generiert und diese evaluiert.

Die Lösung dieser Aufgabe ist es also, diese zufällige Zahl zu verändern und sie beispielsweise auf 2 oder eine andere Zahl, die größer als 1 ist, zu setzen, sodass *BinaryBot* damit nicht mehr umgehen kann.

*BinaryBot* kann damit nicht umgehen und explodiert, nachdem er versucht, diese nicht-binäre Zahl zu evaluieren. Dem Spieler steht daraufhin der Weg ins Roboterland frei.

#### **Sicherstellung der Lernziele:**

- Der Roboter versperrt den Weg, solange die Aufgabe nicht richtig gelöst wurde. Je nachdem, welches Vorwissen die Schüler bereits haben, muss der Lehrer bei dieser Aufgabe erklärend eingreifen.





**Abbildung 26 Programmierlernspiel Level 2 – Aufgabe 1 BinaryBot ausgetrickst**

## **Level 2 – Aufgabe 2**

### **Lernziele:**

- Der Spieler kann arithmetische Operationen im Code ausführen.

In der zweiten Aufgabe üben die Schüler arithmetische Operationen in Programmiersprachen. Nachdem BinaryBot nun besiegt wurde, stellt *CalcBot* dem Spieler die Aufgabe, drei Rechenoperationen auszuführen, um zu beweisen, dass er kein Roboter ist.



**Abbildung 27 Programmierlernspiel Level 2 – Aufgabe 2**

Jede bestandene Aufgabe öffnet eine Tür. Erst wenn alle drei Türen offen sind, kann der Spieler weiterkommen.

Die erste Aufgabe ist noch recht simpel. Der Spieler soll  $54321 - 12345$  berechnen und in der Variable result speichern. Bei der zweiten Aufgabe muss bereits 12345 mit 54321 multipliziert werden. Die letzte Aufgabe ist allerdings dann etwas komplizierter.

Hier muss 12345 durch 25 dividiert werden. Der Spieler bekommt dabei den Hinweis, dass er floating Point Zahlen verwenden muss, um die Kommazahl darzustellen. Das wäre grundsätzlich auch nicht sonderlich schwierig, allerdings kommt, wenn man lediglich die Zeile `float result = 12345 / 25;` schreibt, ein falsches Ergebnis heraus, weil C# sowohl 12345 als auch 25 als Integer interpretiert und somit glaubt, dass das Ergebnis der Division ebenfalls den Datentyp Integer besitzt. Da Integer aber keine Kommazahlen darstellen kann, wird die Zahl einfach abgerundet und es kommt somit ein falsches Ergebnis heraus.

Die Lösung besteht darin, eine der beiden Zahlen zum Datentyp float zu konvertieren, da C# dann auch das Ergebnis der Berechnung als float interpretiert und somit die Nachkommastellen berechnet. Die korrekte Lösung lautet also folgendermaßen:

```
float result = 12345 / 25f;
```

Hier ist selbstverständlich der Lehrer gefragt, dieses Problem den Schülern zu erläutern, da nur die wenigsten Schüler selbstständig das Problem finden werden. Diese Aufgabe hat auch den Zweck, dass die Schüler lernen, dass man in der Programmierung sehr genau arbeiten muss und ein Verständnis dafür entwickeln muss, wie der Computer im Hintergrund arbeitet.

### **Sicherstellung der Lernziele:**

- Die Türen öffnen sich nur, wenn die Berechnungen korrekt durchgeführt wurden. Da die meisten Rechnungen nicht im Kopf zu lösen sind, kommt der Spieler nicht darum herum, arithmetische Operationen anzuwenden.

### **Level 2 – Aufgabe 3**

#### **Lernziele:**

- Der Spieler kann arithmetische Operationen auch in Kombination mit Variablen verwenden.

Bei dieser Aufgabe müssen die Schüler ebenfalls eine arithmetische Rechenoperation durchführen, allerdings verwenden sie dazu diesmal keine konkreten Werte, sondern Variablen.

Außerdem gibt es bei dieser Aufgabe wieder etwas weniger Anleitung durch Hinweiskfelder.

Der Spieler kommt zu einem kleinen verseuchten See, in dem ein Boot schwimmt, das von zwei Robotern gezogen wird. Dabei versucht der eine Roboter das Boot nach links zu ziehen, während der andere Roboter das Boot nach rechts zu ziehen versucht (siehe Abbildung 28).

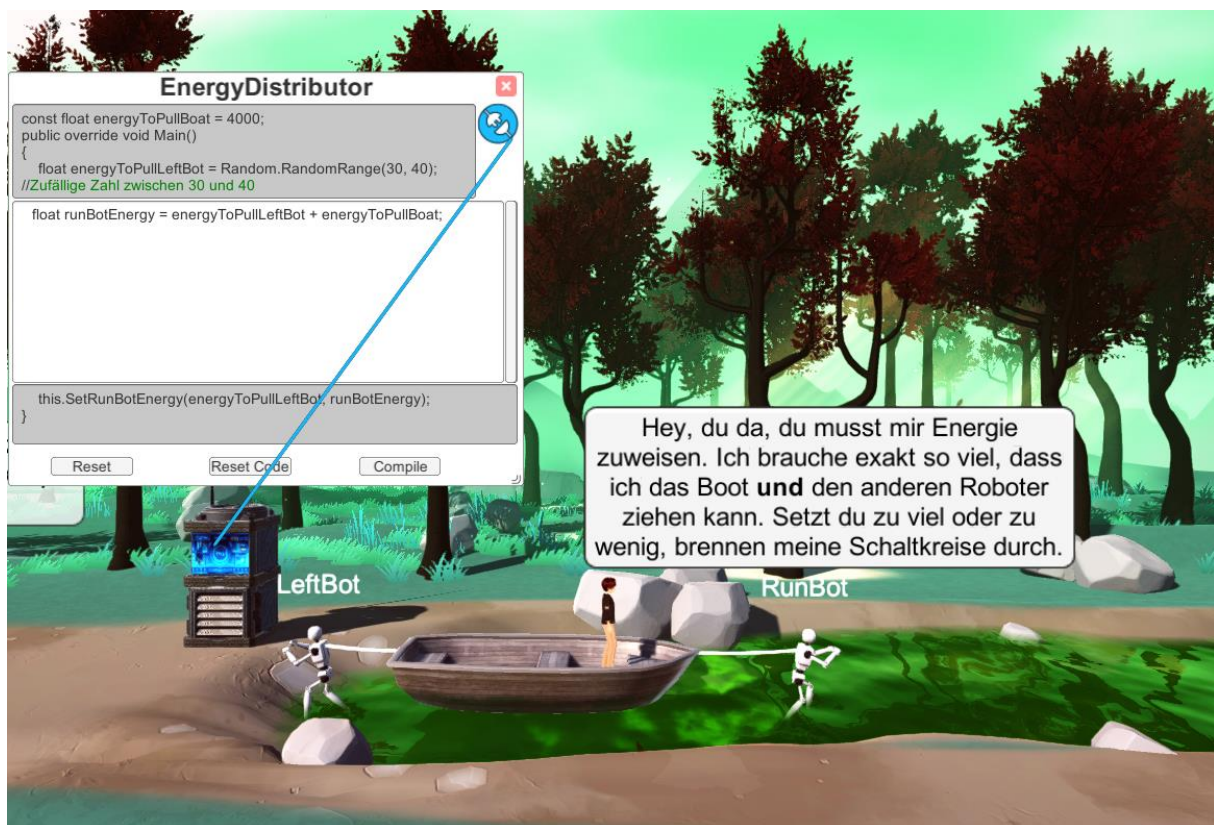
Der rechte Roboter trägt den Namen RunBot und fordert den Spieler auf, ihm exakt so viel Energie zuzuweisen, dass er sowohl das Boot, als auch den linken Roboter ziehen

kann. Er warnt den Spieler aber, dass zu viel Energie seine Schaltkreise verschmort, daher muss die zugewiesene Energie exakt berechnet werden.

Auch in dieser Aufgabe ist der Spieler gefordert, den Präfix des Codes zu lesen und zu verstehen, um zur Lösung der Aufgabe zu kommen.

Im Präfix wird eine konstante Variable angegeben, die beschreibt, wie viel Energie benötigt wird, um das Boot zu ziehen. Außerdem ist hier eine Variable deklariert, die die Energie angibt, die benötigt wird, um den linken Roboter zu ziehen.

Die Lösung der Aufgabe besteht also darin, diese beiden Variablen zu addieren und das Resultat in die Variable *runBotEnergy* zu speichern.



**Abbildung 28 Programmierlernspiel Level 2 – Aufgabe 3**

Wenn der Spieler dem Roboter zu viel Energie zuweist, läuft dieser zwar los, brennt aber kurz darauf durch und fällt um. In diesem Fall muss der Spieler den Reset-Button betätigen, um die Aufgabe zurückzusetzen. Wenn der Spieler versucht, in das

verseuchte Wasser zu springen, wird die Aufgabe ebenfalls zurückgesetzt und der Spieler wird am Ufer platziert.

Hat der Spieler die Berechnung korrekt durchgeführt, so zieht RunBot das Boot mitsamt dem Spieler ans rechte Ufer und der Weg zur nächsten Aufgabe ist frei.

#### **Sicherstellung der Lernziele:**

- Wenn der Spieler den Energiebedarf von RunBot falsch berechnet gibt ihm RunBot Rückmeldungen, die dem Spieler helfen auf die Fehlerquelle zu kommen.
- Solange RunBot nicht die richtige Energie bekommt, kommt er nicht weiter als ein paar Schritte und die Aufgabe muss wiederholt werden.

#### **Level 2 – Aufgabe 4**

##### **Lernziele:**

- Der Spieler kann Strings in Variablen speichern.

Aufgabe 4 des zweiten Levels behandelt das Thema Strings. Dabei musst der Spieler beim Raumschiff die Variable *targetOfShip* den Namen des *SlaveMasterBot* eintragen, damit es auf ihn zusteuert, ihn damit zerstört und RunBot dadurch befreit.





**Abbildung 29 Programmierlernspiel Level 2 – Aufgabe 4**

Auch hier ist der Spieler gefordert, selbst nachzudenken, wie die Aufgabe funktionieren könnte. Als Hinweis steht im Präfix als Kommentar, dass in die Variable *targetOfShip* der Name eines Charakters gespeichert werden muss.

Wenn das Ziel des Raumschiffes korrekt eingegeben wurde, fliegt es auf den *SlaveMasterBot* zu, zerstört ihn und befreit damit *Runbot*, der sogleich auf seinem Laufrad rechts aus dem sichtbaren Spielfeld läuft.

RunBot rammt mit seinem Laufrad rechts des hier abgebildeten Ausschnittes einen Baum, der umfällt und dadurch den Weg zur letzten Aufgabe des Spiels ebnet.

### Sicherstellung der Lernziele:

- Der Weg wird durch einen Baum blockiert, der durch die Lösung der Aufgabe umfällt und dem Spieler hilft über einen Fluss zu kommen.
- Das Raumschiff gibt dem Spieler Rückmeldungen, wenn der String falsch eingegeben wurde beziehungsweise, wenn der Spieler ein falsches Ziel eingegeben hat.



Abbildung 30 Programmierlernspiel Level 2 – Aufgabe 4

### Level 2 – Aufgabe 5

#### Lernziele:

- Der Spieler kann arithmetische Operationen mithilfe von Konstanten und Variablen ausführen und diese anhand einer konkreten Aufgabenstellung anwenden.
- Der Spieler kann Strings definieren und in Variablen speichern.

Aufgabe 5 stellt das große Finale dieses kleinen Lernspieles dar. Ähnlich wie in der ersten Aufgabe dieses Levels versperrt wieder ein Roboter den Weg. Dieses Mal ist es allerdings ein großer roter *GuardBot*, der den Endgegner dieses Spieles darstellt.

Bei diesem Endgegner muss der Spieler alles, was er in diesem Level gelernt hat, einsetzen, um die Aufgabe zu lösen. Bei dieser Aufgabe muss zunächst die Energie auf das Raumschiff und *RunBot* aufgeteilt werden (vgl. Level 2 Aufgabe 3), außerdem muss das Ziel des Raumschiffes und von *RunBot* gesetzt werden (vgl. Level 2 Aufgabe 4). Nur wenn diese vier Variablen richtig gesetzt sind, starten die beiden und können *GuardBot* besiegen.



Abbildung 31 Level 2 – Aufgabe 5: Endboss

Im Präfix des Codefensters ist bei dieser Aufgabe die verfügbare Energie durch die Variable *availableEnergy* und die Energie, die das Raumschiff benötigt, um fliegen zu können, durch die Variable *requiredEnergyShip* gegeben.

Da *RunBot* dem Spieler den Hinweis gibt, dass er so viel Energie wie möglich braucht, muss also die Energie, die das Raumschiff benötigt, von der verfügbaren Energie abgezogen werden. Dieser Wert muss anschließend *RunBot* zugeführt werden. Außerdem muss das Ziel der beiden auf „*GuardBot*“ eingestellt werden.



Ist die Aufgabe richtig gelöst, so schließt sich das Codefenster automatisch und die Hintergrundmusik spielt Freude schöner Götterfunke aus Beethovens 9. Sinfonie, um die bevorstehende Zwischensequenz entsprechend zu untermalen. Das Spiel wechselt sogleich auf eine Slow Motion, in der das Raumschiff und *RunBot* auf den *GuardBot* zustürmen und ihn überwältigen, sodass der Weg zum Ende des Spiels frei wird.

### **Sicherstellung der Lernziele:**

- Der große GuardBot blockiert den Weg, solange die Aufgabe nicht gelöst wurde.
- Bei dieser Aufgabe muss der Spieler beweisen, dass er alle bisherigen Aufgaben auch wirklich verstanden hat, da sie in dieser letzten Aufgabe nochmals Anwendung finden.



**Abbildung 32 Programmierlernspiel Level 2 – Aufgabe 5 Endboss - Endsequenz**

## 6.5 Spieldesign

In Kapitel 5.1 - *Wichtige Elemente eines Computerspiels* wurde bereits auf Weißenböcks Guidelines zur Evaluierung digitaler Lernspiele (vgl. Weißenböck 2014, S. 146–147) hingewiesen. Das entwickelte Spiel soll nun darauf untersucht werden, ob es sich an die wesentlichsten Guidelines hält und wie diese erfüllt werden.

- „Die Grafik eines Lernspiels soll die Zielgruppe ansprechen und zum Thema passen“ (vgl. Weißenböck 2014, S. 146–147)

Da Spielebudgets mittlerweile teilweise größer sind als viele Hollywood Blockbuster Budgets (vgl. wccfttech.com 2016) sind die Jugendlichen heutzutage daran gewöhnt, relativ hochwertige Grafiken in Spielen vorzufinden. Da ich allerdings keinerlei grafische Vorbildung habe, versuchte ich mit möglichst wenigen Grafikassets ein möglichst gutes Gesamtbild hinzubekommen.

So habe ich aus dem Assetstore von Unity3D einige Assets gekauft und mit diesen wenigen Assets in Kombination von diversen Imageeffects wie Antialiasing, Chromatic Aberration und Color Correction ein ästhetisch durchaus ansprechendes Spiel entwickelt.

Allerdings muss dazugesagt werden, dass alleine die grafische Gestaltung des Spieles nahezu ein Viertel der gesamten Entwicklungszeit des Spieles in Anspruch genommen hat.

- „Informationen können auf unterschiedliche Weise in ein Spiel integriert werden: durch Finden von Objekten, durch Ausprobieren, durch Dialoge etc.“ (vgl. Weißenböck 2014, S. 146–147)

Die Informationen im Spiel sind größtenteils in Textform integriert. Allerdings werden die Spieler darauf aufmerksam gemacht, dass diese Informationen je nach Aufgabe in den Tutorialtexten, den Aufgabentexten, den Bemerkungen der NPCs oder aber im Code selbst versteckt sind.

- „Jedes Lernziel hat unterschiedliche Anforderungen, der gewählte Typ des Spiels soll das Lernziel unterstützen“ (vgl. Weißenböck 2014, S. 146–147)

Das Spiel ist am ehesten dem Genre der Adventuregames zuzuordnen und versucht, Programmierkenntnisse zu vermitteln. Sommeregger schreibt in seiner Arbeit *Digital Game-Based Learning am Beispiel eines Adventure Lernspieles*, dass sich Adventurespiele besonders in Bereichen zur Wissensvermittlung eignen, in denen es um Problemlösekompetenzen geht (vgl. Sommeregger 2012, S. 39–41).

Zu Beginn des Designprozesses gab es zwar die Überlegung, das Spiel eher als Jump & Run zu entwickeln, allerdings wurde diese Idee aufgrund der Erkenntnisse der Literaturrecherche dermaßen abgeändert, dass das Spiel den Fokus nicht mehr auf den Bewegungsmechaniken, sondern vielmehr auf dem Adventure Aspekt legt.

- „Narration, Spiel, Spaß und Lernprozess sollen so miteinander verflochten sein, dass das Lernen nicht in den Vordergrund tritt und dies somit für die Lernenden nicht offensichtlich ist.“ (vgl. Weißenböck 2014, S. 146–147)

Eines der großen Ziele dieses Spieles war es, die Geschichte des Spiels mit den Mechaniken sinnvoll zu verbinden. Dies war auch der Grund dafür, dass das Spiel in einem Roboter-Setting angesiedelt ist. Der Spieler kann im Spiel einige der Roboter programmieren, um die Aufgaben zu lösen, was den Aspekt des Programmierens im Kontext der Geschichte sehr sinnvoll einbettet und den Spielern auch direkt grundlegende Anwendungsbereiche der Programmierung näherbringt.

Unglücklicherweise ist es beim Programmieren sehr schwer, den Spaß bzw. die Mechaniken direkt in die Ausübung der Programmierung einzubetten. Es gibt zwar Spiele, die genau das versuchen, allerdings werden dabei nicht Programmiersprachen erlernt, sondern lediglich grundlegendste Funktionsweisen der Programmierung (vgl. code.org).

Da es allerdings das Ziel des Spieles war, ein grundlegendes Verständnis der Programmiersprache C# zu entwickeln, wurde der Kompromiss eingegangen, die Motivation der Spieler mit interessanten und teilweise lustigen Zwischensequenzen zu erhöhen und dadurch den Unterricht etwas aufzulockern.

Allerdings muss klar festgehalten werden, dass es bei diesem Spiel schon sehr deutlich ist, dass es einen Lernprozess anstoßen möchte und bloß reine Unterhaltung darstellt.

- „Nachdem eine Aufgabe gelöst/bearbeitet wurde, soll Feedback erfolgen, das den/die Lernende/n unterstützt.“ (vgl. Weißenböck 2014, S. 146–147)

Auf diesen Aspekt wurde bei der Entwicklung des Lernspiels ein besonderes Augenmerk gelegt. So gibt es nach nahezu jeder Aufgabe eine kleine interessante Sequenz, die abgespielt wird und die Geschichte des Spiels vorantreibt.

Wenn der Spieler eine Aufgabe korrekt löst, reagieren die Roboter auf die Anweisungen, die der Spieler ihnen gegeben hat und laufen, fliegen oder explodieren sogar.

- „Das Feedback muss an das Wissen des/der SpielerIn angepasst sein und dazu anregen zu reflektieren.“ (vgl. Weißenböck 2014, S. 146–147)

Das Spiel behandelt Datentypen, also jenes Thema das am Anfang nahezu aller Programmierkurse steht. Somit bedarf es relativ wenig Vorwissen von Seiten der Schüler. Allerdings ist definitiv ein Lehrer notwendig, der den Schülern erläutert, was es zum Beispiel bedeutet, ein Programm zu kompilieren.

Was das Feedback anbelangt, so bekommt der Spieler eine Compileerror-Meldung, wenn eine Zeile Code falsch eingegeben wurde. Natürlich können die Schüler, wenn sie gerade erst mit dem Programmieren begonnen haben, damit nicht gleich sehr viel anfangen, allerdings ist es hierbei die Aufgabe des Lehrers, diese Fehlermeldungen kurz zu erklären.

Mit grundlegenden Englischkenntnissen können dann auch die verschiedenen Fehlermeldungen, nachdem der Lehrer auf die relevanten Stellen hingewiesen hat, von den Schülern selbst interpretiert werden.

Wenn es im Spiel allerdings nicht zu Kompilierfehlern, sondern zu logischen Fehlern kommt, geben normalerweise die beteiligten Roboter in Form von Sprechblasen Feedback, welches dem Spieler einen Hinweis auf die Fehlerquelle gibt.

- „Komplexe Lernspiele benötigen ein Tutorial bzw. Introduction levels [sic] zu Beginn. Reines Lernen durch Bedienungsanleitung soll vermieden werden.“ (vgl. Weißenböck 2014, S. 146–147)

Die ersten drei Aufgaben des ersten Levels dienen ausschließlich dazu, den Spieler mit der Steuerung und der Funktionalität des Codefensters, sowie des Logfensters vertraut zu machen. Dabei wird dem Spieler noch sehr detailliert erklärt, wie er sich bewegt, verschiedene Objekte umprogrammiert und Fehler analysieren kann.

Anschließend wird der Spieler langsam dazu hingeführt, immer selbstständiger die Aufgaben zu lösen. So wird dem Spieler bei den ersten Aufgaben noch dezidiert angegeben, welche Datentypen er zu verwenden hat, und welche Werte darin gespeichert werden müssen, während der Spieler gegen Ende des Spiels sich genau diese Aspekte selbstständig überlegen muss.

## **7 Qualitative Auswertung**

### **7.1 Forschungsdesign**

Das Spiel wurde im Zuge des regulären Informatikunterrichts der 1. Klasse (= 9. Schulstufe) des Zweiges für Mediendesign/Gamedesign der HTL Spengergasse im 5. Wiener Gemeindebezirk zur Übung der Themen Datentypen eingesetzt.

Im Anschluss an die Verwendung des Spiels wurde ein Fragebogen (siehe Abbildung 33) ausgeteilt, den die Schüler anonym ausfüllten. Der Fragebogen maß qualitative Parameter, wie die Schüler den Unterricht wahrgenommen haben.

Das Spiel wurde dabei zunächst vom Lehrer kurz vorgestellt und erklärt und anschließend von den Schülern selbst gestartet. Die Schüler sollten dann Aufgabe 1-3 selbstständig lösen und ab Aufgabe 3 wurde dann immer ein Schüler gebeten, den anderen Schülern die Lösung der Aufgabe am Lehrercomputer vorzuführen und zu erklären.

Unglücklicherweise war es nicht möglich, eine zweite Klasse als Kontrollgruppe für die Untersuchung zu gewinnen, da die betroffenen Lehrer aus der Parallelklasse nicht daran teilnehmen wollten und der Informatikraum der Schule bei weitem zu klein war, um darin zwei Gruppen parallel zu unterrichten. So gibt es bezüglich des Fragebogens leider keine Vergleichswerte.

## Fragebogen zur Evaluation der Einheit

Lieber Schüler,

Lesen Sie bitte die folgenden Aussagen durch und geben Sie anhand der vorgegebenen Skala an, in welchem Maße sie jeweils auf die soeben gehaltene Einheit trifft. Sind Sie der Ansicht, dass Sie zu einer der Aussagen keine Stellung beziehen können, kreuzen Sie bitte nichts an.

|                                                                            | trifft gar<br>nicht zu   | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | trifft<br>völlig zu      |
|----------------------------------------------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 1. Die Lernziele der Einheit wurden mir klar erklärt                       | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2. Diese Lernziele wurden erreicht                                         | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 3. Im Laufe der Einheit hat sich mein Interesse am Thema erhöht            | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 4. Der Lehrer hat viel zu meinem Lernerfolg beigetragen                    | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 5. Das Lernspiel hat viel zu meinem Lernerfolg beigetragen                 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 6. Der Lehrer hat sich verständlich ausgedrückt                            | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 7. Der Lehrer hat ausreichend Raum für Fragen gelassen                     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 8. Der Lehrer hat die Beteiligung der Teilnehmer/innen gefördert           | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 9. Der Lehrer hat zu viel geredet                                          | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 10. Insgesamt fand ich die Leistung des Lehrers gut                        | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 11. Die Einheit war                                                        |                          |                          |                          |                          |                          |
| a) zu schwer                                                               |                          |                          |                          | <input type="checkbox"/> |                          |
| b) mittelschwer                                                            |                          |                          |                          | <input type="checkbox"/> |                          |
| c) zu leicht                                                               |                          |                          |                          | <input type="checkbox"/> |                          |
| 12. Das Lernspiel hat mir das Lernen erleichtert                           | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 14. Ich habe nach der Erklärung verstanden, wie das Lernspiel funktioniert | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 15. Meine Vorkenntnisse für die Einheit waren groß                         | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 16. Ich bin mit meiner eigenen Leistung im Lernspiel zufrieden             | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 17. Ich habe in der Einheit viel gelernt                                   | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 18. Ich würde das Lernspiel meinen Freunden weiterempfehlen                | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

**Was hat Ihnen am Lernspiel besonders gefallen, was hat Sie besonders gestört?**

Gut gefallen hat mir:

.....

.....

Gestört hat mich:

.....

.....

**Sonstige Anmerkungen**

.....

.....

Vielen Dank für Ihre Mitarbeit!

**Abbildung 33 Qualitativer Fragebogen für Schüler**

## **7.2 Ergebnisse**

### **Zugeständnisse**

Bevor die Ergebnisse genauer betrachtet werden können, muss auf einige Störfaktoren, welche die Ergebnisse der Evaluation beeinflusst haben, hingewiesen werden.

Zunächst umfasste die Klasse 31 Schüler und Schülerinnen, von denen auszugehen ist, dass sie relativ spieleaffin sind, da sie den Zweig Mediendesign/Gamedesign besuchen.

Die Unterrichtsstunden fanden in der 10. und 11. Einheit, zwischen 16:15 und 17:55 statt. Da die Schüler also an diesem Tag zuvor bereits 9 Einheiten Unterricht hatten, ist davon auszugehen, dass diese bereits relativ erschöpft waren und nur schwierig ausreichend Konzentration aufbringen konnten, um dem Unterricht zu folgen.

In der 8. und der 9. Einheit hatten die Schüler noch regulären Programmierunterricht, bei dem sie zunächst in einer Theorieeinheit die wichtigsten Datentypen und Methoden bearbeiteten.

Das Spiel wurde während der Einheit mit den Schülern auf den Rechnern installiert. Dabei kam es unglücklicherweise zu einer etwa 20-minütigen Verspätung, da das Schulnetzwerk durch das Kopieren der Spieldaten auf die lokalen Rechner etwas überfordert war. Außerdem haben einzelne Schüler nicht alle notwendigen Ordner mitkopiert, wodurch das Spiel bei diesen Schülern zwar gelaufen ist, aber im Spiel kein Code kompiliert werden konnte.

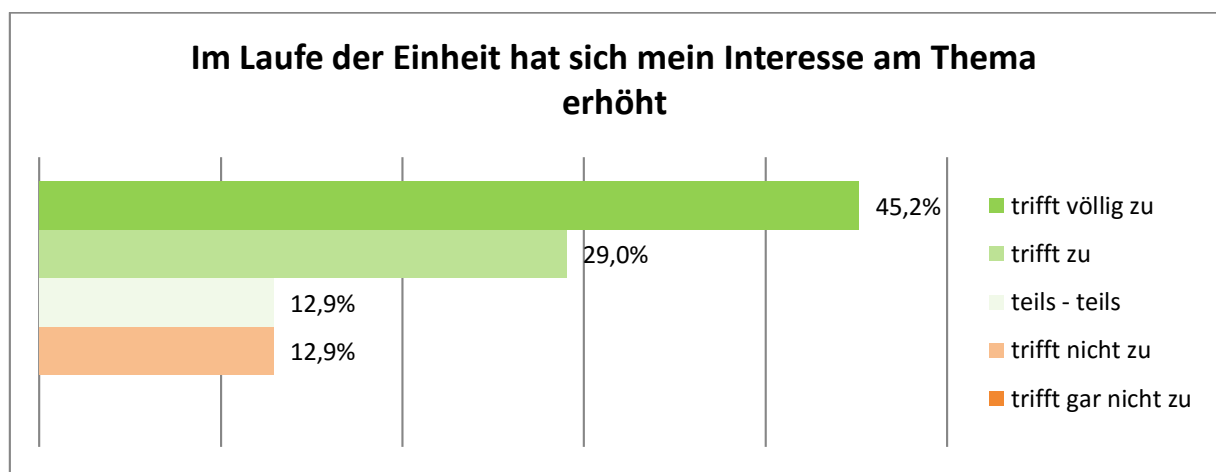
All jene Schüler, die irgendwelche Probleme dieser Art hatten, wurden gebeten, bei ihren Nachbarn mitzuschauen, sich aber dennoch an der Problemlösung zu beteiligen.

Weiters lief das Spiel nicht auf allen Rechnern flüssig. So mussten viele Schüler das Spiel mit ca. 15 - 25 Frames pro Sekunde spielen, was dem Spielerlebnis selbstverständlich nicht unbedingt zugutekam.

## Fragebogenergebnisse

Die Schüler gaben an, dass ihnen die Lernziele der Einheit klar erklärt wurden, und ein Großteil davon auch erreicht wurde.

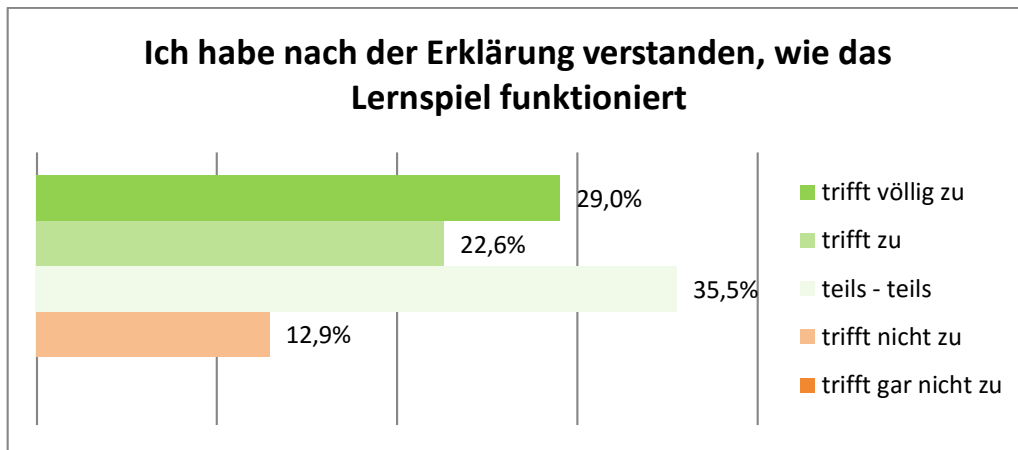
Zunächst wurde versucht festzustellen, wie sich das Spiel auf die Motivation der Schüler ausgewirkt hat. Dabei sollte allerdings keine extrinsische Motivation festgestellt werden, sondern eher intrinsische. Deshalb wurden die Schüler gefragt, ob sich im Laufe der Einheit ihr Interesse am Thema erhöht hat.



**Abbildung 34 Fragebogen – Interessenssteigerung im Laufe der Einheit (N=31)**

Wie Abbildung 34 deutlich zeigt, hat das Interesse bei einem Großteil der Schüler deutlich zugenommen. So ist davon auszugehen, dass das Spiel zumindest kurzfristig das Interesse der Schüler am Thema deutlich erhöht hat und somit zu einer Verschiebung von extrinsischer hin zu intrinsischer Motivation geführt hat.

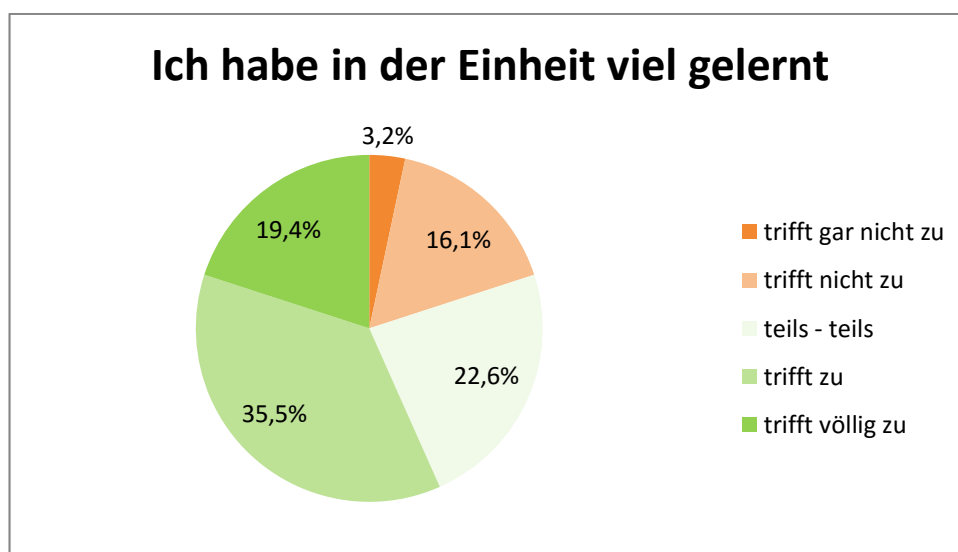




**Abbildung 35 Fragebogen – Verständnis der Funktionsweise des Spiels (N=30)**

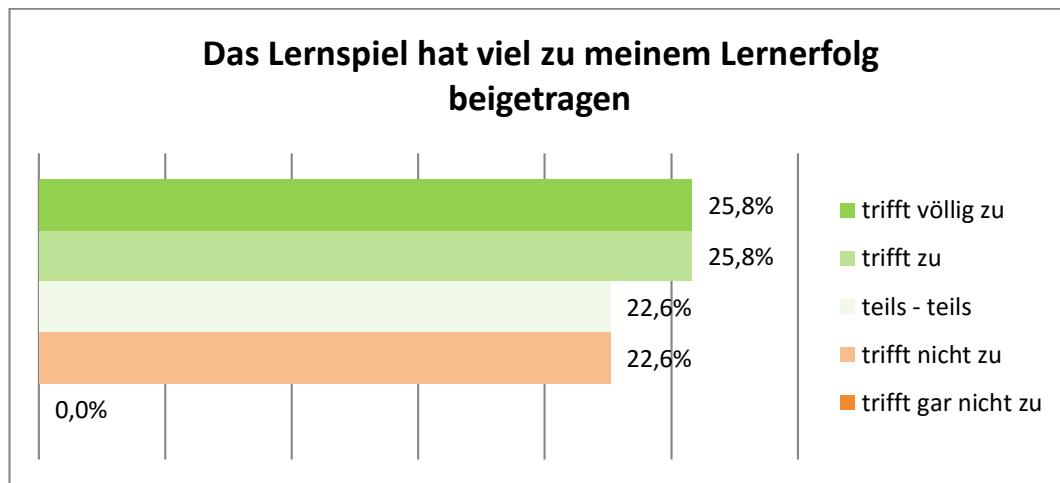
Ein großes Manko gab es allerdings bei der Erklärung des Lernspiels. Nahezu die Hälfte der Schüler gab an, dass sie nach der Erklärung nicht genau verstanden hat, wie das Spiel funktioniert.

Hierzu ist zu sagen, dass das Spiel unvorteilhafterweise den Schülern erklärt wurde, während diese das Spiel noch vom Netzwerk herunterkopierten. So waren die Schüler einerseits nicht mit der vollen Aufmerksamkeit bei der Sache und andererseits konnten sie die Funktionalitäten nicht gleich selbst ausprobieren.



**Abbildung 36 Fragebogen – Selbsteinschätzung des Gelernten (N=30)**

Ungefähr 60% der Schüler gaben an, dass sie in dieser Einheit viel gelernt haben, während lediglich ~15% der Ansicht waren, nicht viel gelernt zu haben. Dieses Ergebnis ist auch durchaus sehr positiv ausgefallen, allerdings wäre diesbezüglich der Vergleich mit einer Kontrollgruppe sehr interessant gewesen.

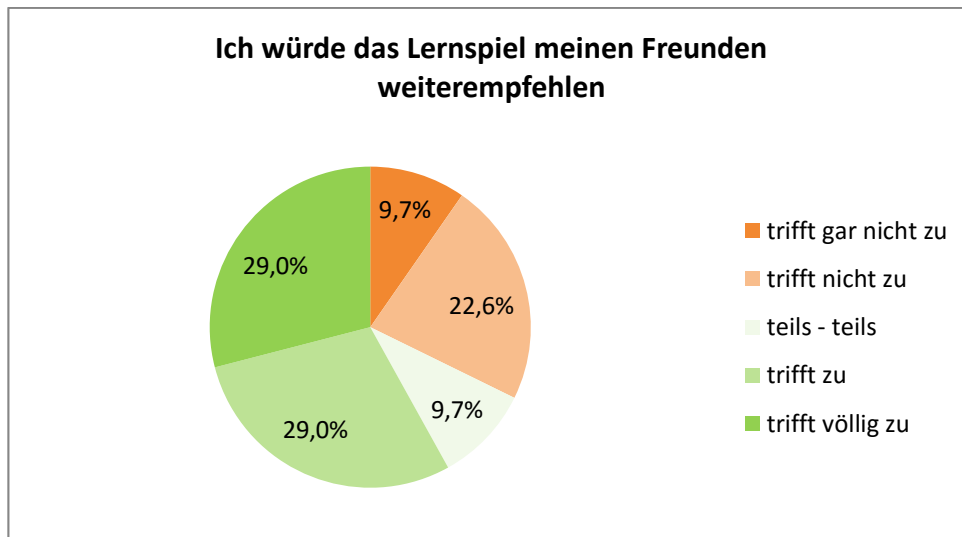


**Abbildung 37 Fragebogen - Beitrag zum Lernerfolg (N=30)**

Die zentrale Frage, ob das Lernspiel zum Lernerfolg der Schüler beigetragen hat, fällt unglücklicherweise etwas schlechter aus. Etwas mehr als die Hälfte der Schüler stimmt zu, während etwa ein Fünftel der Schüler meint, dass das Spiel nicht zum Lernerfolg beigetragen hat.

Diese Frage wäre zwar eigentlich der zentralste und wichtigste Teil des Fragebogens, allerdings muss dieses Ergebnis mit großer Vorsicht betrachtet werden. Wie bereits bei den Zugeständnissen erwähnt, konnten einige Schüler das Spiel nicht auf ihren Rechnern spielen und bei fast allen war die Performance relativ schlecht, sodass das Spiel nur relativ mühsam gespielt werden konnte. Da das Spiel natürlich nicht zum Lernerfolg beiträgt, wenn man es aufgrund der niedrigen Framerate auf manchen Rechnern kaum spielen konnte, ist selbstverständlich.

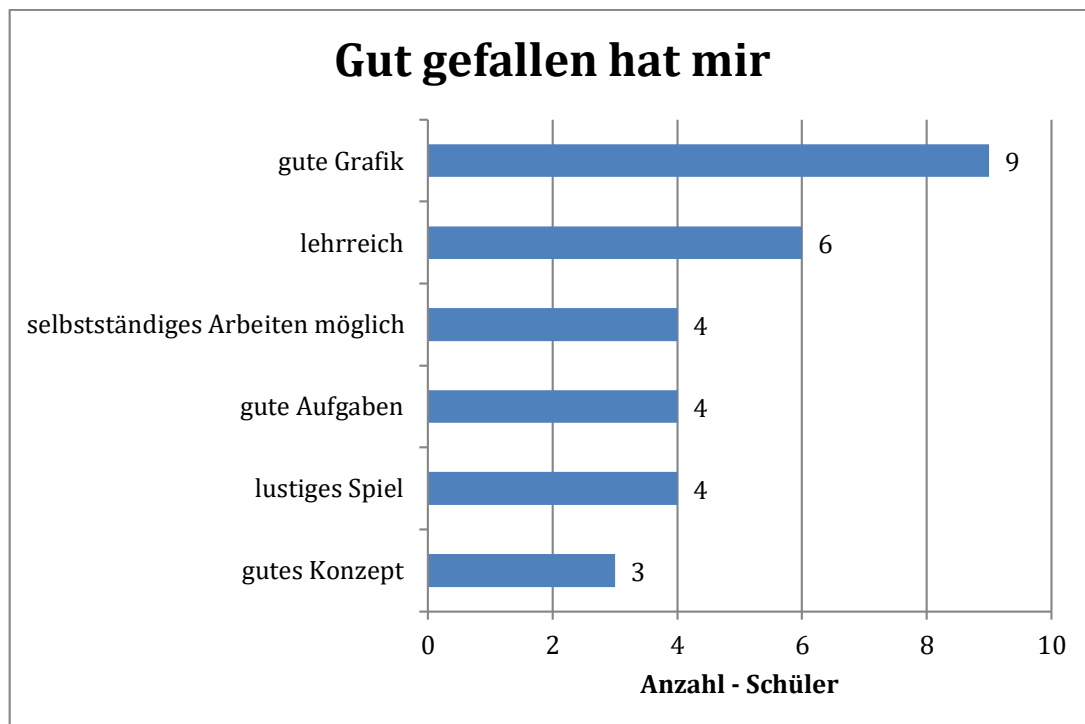
Dennoch sprechen sich etwas mehr als die Hälfte für den positiven Beitrag zum Lernerfolg aus, was definitiv ein Argument für die Schüler ist.



**Abbildung 38 Fragebogen – Würde Lernspiel den Freunden empfehlen (N=31)**

Zuletzt wurden die Schüler noch gefragt, ob sie das Lernspiel ihren Freunden empfehlen würden. Hierbei gaben ungefähr 60% der Schüler an, dass sie das durchaus tun würden. In etwa 30% würden das Spiel jedoch nicht weiterempfehlen. Auch bei dieser Frage ist zu bedenken, dass das Spiel auf vielen Rechnern sehr schlecht gelaufen ist.

Zuletzt wurde den Schülern noch die Möglichkeit gegeben, mit eigenen Worten rückzumelden, was ihnen gut und was ihnen schlecht gefallen hat. Für die Auswertung wurden ähnliche Rückmeldungen unter bestimmten Sammelbegriffen zusammengefasst. Abbildung 39 veranschaulicht die positiven Rückmeldungen der Schüler.



**Abbildung 39 Fragebogen – Gut gefallen hat mir am Spiel (N=31, Mehrfachnennungen möglich)**

Am Augenscheinlichsten ist, dass die Schüler vor allem positiv auf die Grafik reagiert haben. Wenn es nach Wood ginge, wäre das Spiel also schon ein voller Erfolg (vgl. Wood 2001). Einige Schüler geben noch an, dass sie es für lehrreich halten und heben den Humor, das selbstständige Arbeiten und gute Aufgabenstellungen hervor.

Folgende drei Rückmeldungen sollen noch etwas genauer betrachtet werden:

*Erstes Lernspiel, das Spaß gemacht hat*

Der Schüler, der diesen Kommentar dazugeschrieben hat, kannte offensichtlich bereits andere Lernspiele, welche unter die Kategorie „schokoladeüberzogener Brokkoli“ von (vgl. Granic et al. 2014) fallen.

*Das [sic] ich dadurch auch meine Handlungen bzw. Lösungen wirklich verstanden habe*

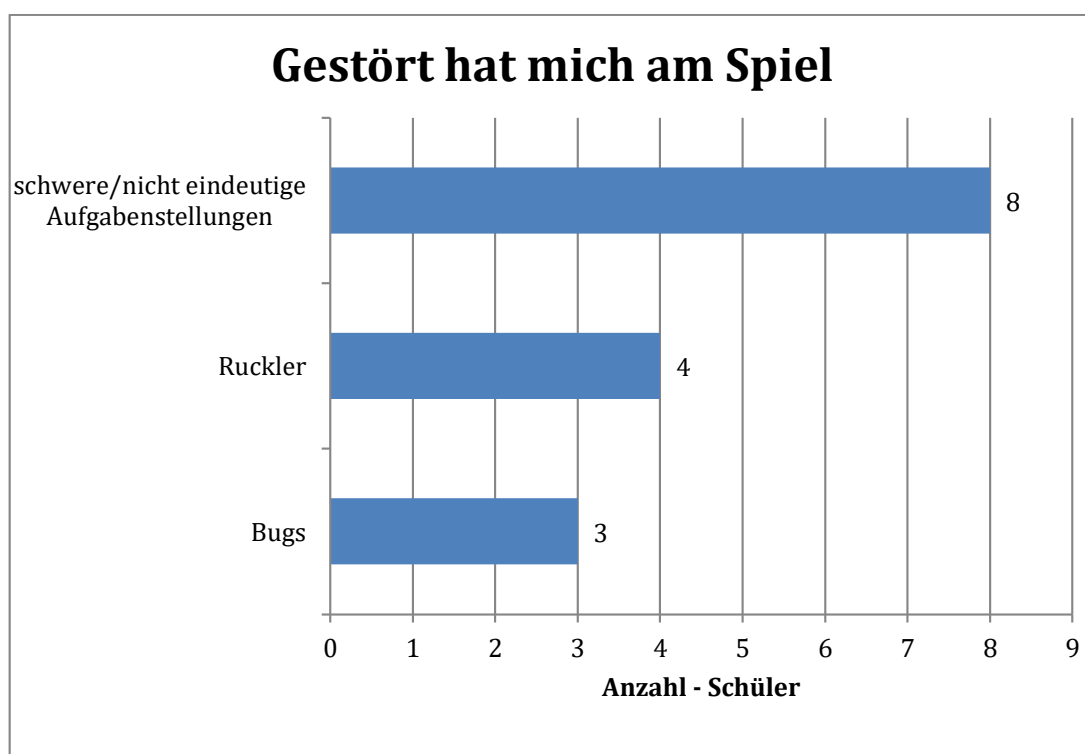
Der Autor dieser Rückmeldung hat erst durch das Üben mit dem Lernspiel den Stoff wirklich verstanden. Das Lernspiel hat also zumindest diesem Schüler davor bewahrt,

das Thema Datentypen als totes Wissen, wie es Pädagogen heutzutage gerne nennen, zu lernen.

*Dass es einen zum Nachdenken gebracht hat. Also dass es nicht zu einfach war sondern man überlegen musste bevor man etwas hinschreibt und man darüber auch was gelernt hat*

Dieser Schüler hebt hervor, dass er erst durch das selbstständige Üben in dieser Aufgabe herausgefordert war. Er dürfte die Herausforderung, die das Spiel bietet, als guten Anreiz erfahren haben, sich mit dem Thema intensiv zu beschäftigen.

Abbildung 40 veranschaulicht die wesentlichsten Punkte, die die Schüler am Spiel gestört haben.



**Abbildung 40 Fragebogen – Nicht gefallen hat mir (n=31, Mehrfachnennungen möglich)**

Wie an der Abbildung unschwer zu erkennen ist, war das größte Problem die Schwierigkeit der Aufgabenstellungen. Zwar gaben vier Schüler an, dass ihnen die

Aufgabenstellungen sehr gut gefallen haben, aber wenn gleichzeitig acht Schüler eben diese kritisieren, deutet das klar darauf hin, dass diesbezüglich Handlungsbedarf besteht.

Vier Schüler kritisierten die Ruckler, also die schlechte Performance des Spiels. Allerdings gaben zwei dieser vier Schüler unter den sonstigen Anmerkungen an, dass das Spiel „eigentlich ganz cool ist“ und „ziemlich was drauf hat“.

Zuletzt gab es noch drei Schüler, die die Bugs am Spiel kritisierten. Hierbei ist allerdings dazu zu sagen, dass es sich dabei wohl weniger um tatsächliche Bugs im Spiel handelt als vielmehr darum, dass manche Schüler nicht alle erforderlichen Dateien aus dem Netzwerk kopiert haben. So ist das Spiel zwar gelaufen, allerdings konnte dann kein Code kompiliert werden. Diese Schüler haben leider beim Erklären der Installation des Spiels etwas versäumt.

### **Erkenntnisse für künftige Auswertungen**

Es wäre absolut essentiell, das Spiel bereits auf den Rechnern zu installieren, damit die Schüler nicht 20 Minuten Leerlauf haben, in denen das Spiel vom Netzwerk heruntergeladen werden muss. In dieser Zeit kann das Spiel zwar kurz erklärt werden, allerdings müssten die Schüler das Spiel direkt ausprobieren, um zu verstehen, wie es funktioniert. Eine Erklärung im Sinne eines Frontalvortrages ist für Computerspiele natürlich sehr unvorteilhaft.

Da die Performance des Spiels auf den Schulrechnern unglücklicherweise nicht sonderlich gut war, müsste die Grafik des Spiels hinuntergeschraubt werden. Die Schüler haben die Grafik zwar sehr gelobt, gleichzeitig war aber eines der größten Mankos in der Auswertung die niedrige Framerate des Spiels.

Der Schwierigkeitsgrad im Spiel steigt viel zu schnell an. Es wäre wichtig, den Schülern für jeden Übungstyp zumindest drei bis vier Aufgaben zu geben. Vor allem in Level 2 wurde den Schülern zu viel abverlangt, da besonders die Aufgaben 1,3 und 5 einen zu hohen Schwierigkeitsgrad aufwiesen. Aus Zeitgründen konnten leider nicht mehr Übungen eingebaut werden, da die Entwicklung einer einzigen Übung ungefähr zehn Stunden in Anspruch nahm. Für die Weiterentwicklung der Software wäre es wichtig,

die Anzahl der Übungen bei gleichbleibendem Stoffumfang auf 20 Stück zu verdoppeln.

Ein leider sehr grundlegendes Problem ist, dass die Prä- und Suffixe im Codefenster einige Schüler verwirrt haben, da sie nicht wussten, ob diese zum Code gehören oder nicht. Hierbei müsste das Interface etwas abgeändert werden, sodass einerseits die Zusammengehörigkeit des Codes besser zur Geltung kommt, andererseits sollte aber auch klar sichtbar sein, dass nur der mittlere Teil des Codes editierbar ist.

### **7.3 Interpretation der Ergebnisse**

Trotz der beträchtlichen Zugeständnisse, die die Schüler machen mussten (schlechte Performance, lange Installationszeit, zu schwierige Aufgaben) nahmen sie die Verwendung des Lernspieles größtenteils positiv auf.

Ungefähr 75% der Schüler gaben an, dass sich ihr Interesse am Thema erhöht hat, was zunächst zu einer erhöhten intrinsischen Motivation führt, welche wiederum zu verbesserten Lernerfolgen führt. Diesbezüglich scheint das Lernspiel ein voller Erfolg gewesen zu sein.

Da jeweils ~60% der Schüler angaben, dass sie in dieser Einheit viel gelernt haben und ungefähr ebenso viele Schüler meinten, dass sie das Lernspiel weiterempfehlen würden ist davon auszugehen, dass das Spiel insgesamt verhältnismäßig gut angekommen ist.

Allerdings muss gleichzeitig beachtet werden, dass ~30% der Schüler das Spiel nicht weiterempfehlen würden und ~22% angaben, dass das Lernspiel nicht zu ihrem Lernerfolg beigetragen hat. Hierbei wäre es wichtig die Untersuchung unter verbesserten Bedingungen (bessere Performance, vorinstalliertes Spiel) zu wiederholen, da aufgrund der relativ allgemeinen Fragestellung nicht auszumachen ist, ob diese Schüler aufgrund der organisatorischen bzw. strukturellen Probleme dem Lernspiel kritisch gegenüberstehen oder ob es tatsächlich das Lernspiel selbst ist, an dem sie etwas auszusetzen haben.

Insgesamt überwiegen jedoch die positiven Rückmeldungen der Schüler und die großen Kritikpunkte der Schüler betreffen weniger das Spielen an sich, sondern vielmehr Probleme in der Implementation. So sind sowohl die Bugs, als auch die

Performanceprobleme Dinge, die relativ einfach behoben werden können. Auch der Schwierigkeitsgrad der Aufgaben ist natürlich ein Problem, welches noch adaptiert werden müsste.

## **7.4 Ausblick**

Es wäre absolut essentiell, eine ähnliche Evaluation unter besseren Bedingungen durchzuführen. Die Grafik des Spiels müsste reduziert werden, sodass das Spiel flüssig auf allen Schulrechnern läuft. Außerdem sollte das Spiel bereits vor Beginn der Stunde vorinstalliert sein, damit die Schüler während der Erklärung des Spiels direkt mitarbeiten können.

Was das Forschungsdesign betrifft, so wäre diesbezüglich ein Vergleich mit einer Kontrollgruppe sehr vorteilhaft, damit die Ergebnisse mehr Gewicht bekommen. Außerdem wäre ein quantitativer Teil im Sinne eines kleinen Tests, der das Verständnis zur Programmierung abprüft, sehr sinnvoll, um den Mehrwert der Lernsoftware bezüglich des Lernfortschritts festzustellen.

Eine neue Version des Spiels, bei der viele Aspekte grundlegend verändert werden, ist bereits in Arbeit. Diese neue Version des Spiels reduziert die Grafik und ändert das Gameplay ziemlich drastisch dermaßen ab, dass der Spieler mehr Freiheiten hat und sich aussuchen kann, welche Aufgabe er zuerst macht. Dabei orientiert sich die neue Version des Spiels an dem unter Jugendlichen sehr beliebten Spiel Minecraft (Mojang). Außerdem werden die Prä- und Suffixe entfernt. Anstatt den Prä- und Suffixen wird es nur noch einen Codebereich geben, der bei manchen Aufgaben bereits vorgefertigten Code enthält.

Da diese Änderungen jedoch ein nicht unbeachtliches Maß an Arbeit mit sich bringen, wird die nächste Evaluation unglücklicherweise nicht mehr im Schuljahr 2016/17 durchgeführt werden können, sondern erst im darauffolgenden Jahr.



## **8 Fazit**

### **Beantwortung der Forschungsfragen**

Empfinden die Schüler das Lernspiel als wirksames Mittel, um etwas zu lernen?

Die Schüler sind diesbezüglich gespaltenen Meinung, ungefähr 50% denken, dass das Lernspiel etwas zu ihrem Lernerfolg beigetragen hat. ~25% der Schüler sind der Meinung, dass sich Lernspiel nicht positiv auf den Lernerfolg auswirkt. Da allerdings bei der Durchführung der Untersuchung einige Störvariablen hinzugekommen sind (siehe 7.2 - Ergebnisse), welche diese Bewertung dieser Frage tendenziell verschlechtert hat, sind diese Werte nur bedingt aussagekräftig. Hierbei bedürfte es einer Wiederholung der Untersuchung im Schuljahr 2017/18, um diese Frage sicher beantworten zu können.

Wie wirkt sich das Lernspiel auf das Interesse der Schüler aus?

Das Interesse der Schüler hat sich beim Großteil der Schüler im Laufe des Unterrichts laut ihren Angaben gesteigert, somit wirkt sich das Lernspiel definitiv positiv auf das Interesse der Schüler aus.

### **Allgemeines Fazit**

Digital game-based Learning ist durchaus ein probates Mittel, um den Programmierunterricht etwas aufzulockern und spannender zu machen. Den Schülern ist die Verwendung eines Lernspiels eine willkommene Abwechslung zum tristen Schulalltag.

Der Großteil der Schüler steht der Verwendung von Lernspielen im Unterricht sehr positiv gegenüber. Das Interesse und die Motivation der Schüler wird durch digital game-based Learning deutlich verbessert, was wiederum zu besseren Lernerfolgen führt.

Allerdings darf der zusätzliche Aufwand, den die Verwendung eines Lernspiels im Unterricht mit sich bringt, keinesfalls außer Acht gelassen werden. Die Vorbereitungszeit für den Unterricht erhöht sich definitiv, da die Lernsoftware auf den

Schulrechnern installiert werden muss und der Lehrer das Lernspiel zunächst selbst ausprobieren muss.

Außerdem darf Okhans Kritik (Okhan 2003) nicht außer Acht gelassen werden, dass sich die breitflächige Verwendung von Unterhaltungsmedien im Unterricht negativ auf andere Fächer auswirken könnte, die diese Medien nicht verwendet. Diesbezüglich bedarf es noch Langzeitstudien, die den langfristigen Mehrwert von digital game-based Learning feststellt und die Auswirkungen auf andere Fächer misst.

# Abbildungsverzeichnis

|                                                                                    |    |
|------------------------------------------------------------------------------------|----|
| Abbildung 1 Pawlowscher Hund .....                                                 | 6  |
| Abbildung 2 Verstärkung nach Thorndike & Skinner (Hubwieser 2007, S. 4) .....      | 8  |
| Abbildung 3 Versuchsaufbau von Yerkes & Dodson .....                               | 15 |
| Abbildung 4 Yerkes – Dodson Gesetz .....                                           | 16 |
| Abbildung 5 Motivierung, Motive und Motivation .....                               | 22 |
| Abbildung 6: Intrinsische und extrinsische Motivationen .....                      | 23 |
| Abbildung 7 Flow – Erleben .....                                                   | 27 |
| Abbildung 8 Code Combat- Übersicht .....                                           | 41 |
| Abbildung 9 Code Combat – Aufgabenbeschreibung .....                               | 42 |
| Abbildung 10 Code Combat – Codefenster .....                                       | 43 |
| Abbildung 11 Civilization 5 – Übersicht .....                                      | 45 |
| Abbildung 12 Erwartete und tatsächliche Motivation der Schüler .....               | 48 |
| Abbildung 13 Selbst wahrgenommene Verbesserung der Programmierskills .....         | 49 |
| Abbildung 14 Punkte am Programmiertest .....                                       | 50 |
| Abbildung 15 Vandenberghes Domains of Play .....                                   | 55 |
| Abbildung 16 Programmierlernspiel - Erstes Hindernis .....                         | 63 |
| Abbildung 17 Programmierlernspiel – Üben der Sprungmechanik .....                  | 64 |
| Abbildung 18 Programmierlernspiel Level 1 - Aufgabe 1 .....                        | 65 |
| Abbildung 19 Programmierlernspiel Level 1 - Aufgabe 1 mit Codefenster .....        | 65 |
| Abbildung 20 Programmierlernspiel Level 1 - Aufgabe 2 .....                        | 67 |
| Abbildung 21 Programmierlernspiel Level1 – Aufgabe 2 Logfenster .....              | 68 |
| Abbildung 22 Programmierlernspiel Level 1 – Aufgabe 3 .....                        | 70 |
| Abbildung 23 Programmierlernspiel Level 1 – Aufgabe 4 .....                        | 71 |
| Abbildung 24 Programmierlernspiel Level 1 – Aufgabe 5 .....                        | 72 |
| Abbildung 25 Programmierlernspiel Level 2 – Aufgabe 1 .....                        | 75 |
| Abbildung 26 Programmierlernspiel Level 2 – Aufgabe 1 BinaryBot ausgetrickst ..... | 77 |
| Abbildung 27 Programmierlernspiel Level 2 – Aufgabe 2 .....                        | 78 |
| Abbildung 28 Programmierlernspiel Level 2 – Aufgabe 3 .....                        | 80 |
| Abbildung 29 Programmierlernspiel Level 2 – Aufgabe 4 .....                        | 82 |
| Abbildung 30 Programmierlernspiel Level 2 – Aufgabe 4 .....                        | 83 |
| Abbildung 31 Level 2 – Aufgabe 5: Endboss .....                                    | 84 |
| Abbildung 32 Programmierlernspiel Level 2 – Aufgabe 5 Endboss - Endsequenz .....   | 85 |

|                                                                                                |    |
|------------------------------------------------------------------------------------------------|----|
| Abbildung 33 Qualitativer Fragebogen für Schüler .....                                         | 90 |
| Abbildung 34 Fragebogen – Interessenssteigerung im Laufe der Einheit (N=31) .....              | 92 |
| Abbildung 35 Fragebogen – Verständnis der Funktionsweise des Spiels (N=30) .....               | 93 |
| Abbildung 36 Fragebogen – Selbsteinschätzung des Gelernten (N=30).....                         | 93 |
| Abbildung 37 Fragebogen - Beitrag zum Lernerfolg (N=30) .....                                  | 94 |
| Abbildung 38 Fragebogen – Würde Lernspiel den Freunden empfehlen (N=31) .....                  | 95 |
| Abbildung 39 Fragebogen – Gut gefallen hat mir am Spiel (N=31, Mehrfachnennungen möglich)..... | 96 |
| Abbildung 40 Fragebogen – Nicht gefallen hat mir (n=31, Mehrfachnennungen möglich).....        | 97 |

## 9 Literaturverzeichnis

Almut, E. T.; Müller, H. F. (2011): Skalen zur motivationalen Regulation beim Lernen von Schülerinnen und Schülern. Klagenfurt: Alpen-Adria-Universität Klagenfurt.

APA Board of Educational Affairs (1997): Learner - Centered Psychological Principles. A Framework for School Reform & Redesign.

bmbf: AHS-Lehrpläne Oberstufe neu. Online verfügbar unter [https://www.bmbf.gv.at/schulen/unterricht/lp/lp\\_neu\\_ahs\\_14\\_11866.pdf?4dzgm2](https://www.bmbf.gv.at/schulen/unterricht/lp/lp_neu_ahs_14_11866.pdf?4dzgm2), zuletzt geprüft am 18.03.2016.

Breuer, J. (2010): Spielend Lernen. Eine Bestandsaufnahme zum (Digital) Game-Based Learning. Landesanstalt für Medien Nordrhein-Westfalen (LfM). Düsseldorf.

Butler-Bowdon, T. (2007): 50 Psychology Classics. Who we are, how we think, what we do: insight and inspiration from 50 key books. London, Boston: Nicholas Brealey Pub.

code.org: Frozen - Programmierlernspiel. Online verfügbar unter <https://studio.code.org/s/frozen/stage/1/puzzle/1>, zuletzt geprüft am 22.03.2016.

CodeCombat: CodeCombat. Learn to Code by Playing a Game. Online verfügbar unter <http://codecombat.com/>, zuletzt geprüft am 15.03.2016.

Comber, O.; Motschnig, R. (2015): Challenges and opportunities in employing game development in computer science classes. In: *EdMedia: World Conference on Educational Media and Technology 2015* (1), S. 1892–1900.

Cooper, S.; Khatib, F.; Treuille, A.; Barbero, J.; Lee, J.; Beenen, M. et al. (2010): Predicting protein structures with a multiplayer online game. In: *Nature* 466, S. 756–760.

Deterding, S.; Dixon, D.; Khaled, R.; Nacke, L.: From game design elements to gamefulness. In: Artur Lugmayr, Heljä Franssila, Christian Safran und Imed Hammouda (Hg.): the 15th International Academic MindTrek Conference. Tampere, Finland, S. 9–15.

Enns, J. T. (1990): The Development of attention. Research and theory: Elsevier Science Ltd (Advances in psychology, 69).

Eysenck, M. W. (2001): Principles of cognitive psychology. 2nd ed. Hove, UK, Philadelphia: Psychology Press (Principles of psychology series).

Gebel, C.; Gurt, M.; Wagner, U. (2005): Kompetenzförderliche Potenziale populärer Computerspiele. In: *Arbeitsgemeinschaft Betriebliche Weiterbildungsforschung* (92), S. 241–376.

Geldenhuis, T. (2016): C# Interpreter Console for Unity 3D | blog.Tiaan.com. Online verfügbar unter <http://blog.tiaan.com/link/2010/03/15/csharp-interpreter-unity-plugin-console-debugger?page=1>, zuletzt aktualisiert am 15.03.2016, zuletzt geprüft am 15.03.2016.

Gentile, D. (2009): Pathological Video-Game Use Among Youth Ages 8 to 18. A National Study. In: *Psychological Science* 20 (5), S. 594–602.

Gold, A. (2003): Lernen. In: Siegfried Preiser und Markus Dresel (Hg.): Pädagogische Psychologie. Psychologische Grundlagen von Erziehung und Unterricht. Weinheim: Juventa (Grundlagentexte Pädagogik), S. 99–124.

Granic, I.; Lobel, A.; Engels, Rutger C. M. E. (2014): The benefits of playing Video Games. In: *American Psychologist* 69 (1), S. 66–78.

HTL Spengergasse: Lehrplan Informatik. Online verfügbar unter <http://www.spengergasse.at/sites/default/files/pdf/LehrplanInformatik.pdf>, zuletzt geprüft am 18.03.2016.

Hubwieser, P. (2007): Didaktik der Informatik. Grundlagen, Konzepte, Beispiele. 3., überarb. und erw. Aufl. Berlin: Springer.

Jentges, S. (2007): Effektivität von Sprachlernspielen. Zur Theorie und Praxis des Spieleinsatzes im Deutsch-als-Fremdsprache-Unterricht. Baltmannsweiler: Schneider-Verl. Hohengehren (Perspektiven Deutsch als Fremdsprache, 21).

Lefrancois, G. R.; Leppmann, P. K. (1986): Psychologie des Lernens. Mit 13 Tabellen. 2., vollkommen überarb. u. erg. Aufl. Berlin [u.a.]: Springer.

Lin, Y.-G.; McKeachie, W. J.; Kim, Y. C. (2001): College student intrinsic and/or extrinsic motivation and learning. In: *Learning and Individual Differences* 13 (3), S. 251–258.

McAdams, D. P. (1997): A Conceptual History of Personality Psychology. In: Robert Hogan, John Johnson und Stephen R. Briggs (Hg.): *Handbook of personality psychology*. San Diego: Academic Press, S. 3–39.

McCall, J. B. (2011): *Gaming the past. Using video games to teach secondary history*. New York: Routledge.

McCombs, B. L. (2013): The Learner-Centered Model: From the Vision to the Future. In: Jeffrey H. D. Cornelius-White, Renate Motschnig-Pitrik und Michael Lux (Hg.): *Interdisciplinary Applications of the Person-Centered Approach*. New York, NY: Springer New York, S. 83–113.

Mojang: Minecraft. Online verfügbar unter <https://minecraft.net/>, zuletzt geprüft am 24.03.2016.

Müller, F. H.; Hanfstingl, B.; Andreitz, I.: Skalen zur motivationalen Regulation beim Lernen von Schülerinnen und Schülern. (Adaptierte und ergänzte Version des Academic Self-Regulation Questionnaire nach Ryan & Connell). Klagenfurt: Alpen-Adria Universität Klagenfurt (Wissenschaftliche Beiträge aus dem Institut für Unterrichts- und Schulentwicklung, 1).

Okhan, Z. (2003): Edutainment. is learning at risk? In: *British Journal of Educational Technology* 34 (3), S. 255–264. Online verfügbar unter [http://www.savie.ca/SAGE/Articles/1084\\_453-Okan-2003.pdf](http://www.savie.ca/SAGE/Articles/1084_453-Okan-2003.pdf), zuletzt geprüft am 08.03.2016.

Pfeiffer, D.; Motschnig, R. (2015): Enhancing e-learning with gaming. Case-study of a virtual course on 3D-printing for teachers. In: Francesco Pisanu (Hg.): *Ricerca*

(Six-monthly Journal on Learning, Research and Innovation in Education, Vol. 7 Nr. 2), S. 165–187.

Pivec, M. (2007): Games for Learning and Learning from Games. In: *Informatica* 31, S. 419–423.

Prensky, M. (2001): Digital game-based learning. New York: McGraw-Hill.

Ryan, R.; Deci, E. (2000): Intrinsic and Extrinsic Motivations. Classic Definitions and new Directions: Academic Press (Contemporary Educational Psychology, 25).

Sarid, O.; Anson, O.; Yaari, A.; Margalith, M. (2004): Academic stress, immunological reaction, and academic performance among students of nursing and physiotherapy. In: *Res. Nurs. Health* 27 (5), S. 370–377.

Sawyer, B. (2007): The "Serious Games" Landscape. Online verfügbar unter [internet2.rutgers.edu/pres/speaker6-sawyer-final.ppt](http://internet2.rutgers.edu/pres/speaker6-sawyer-final.ppt), zuletzt geprüft am 08.03.2016.

Schell, J. (2010): The art of game design. A book of lenses. Reprinted. Amsterdam [etc.]: Elsevier; Morgan Kaufmann publishers.

Schleiermacher, F. (2011): Phaidros: Fischer.

Shakespeare, W. (2015): König Heinrich IV.: Erster und zweiter Teil: Hofenberg.

Sommeregger, P. F. (2012): Abenteuer Lernspiel. Digital Game-Based Learning am Beispiel eines Adventurespiels. Diplomarbeit.

Trimmel, M. (2011): Motivation, Emotion und Lernen. Allgemeine Psychologie 1. Wien: Facultas Verlag.

Trimmel, M. (2013): Kognitive Psychologie. Allgemeine Psychologie 2. Wien: Facultas Verlag.

Unity3D (2016): Unity - Game Engine. Online verfügbar unter <http://unity3d.com/>, zuletzt aktualisiert am 15.03.2016, zuletzt geprüft am 15.03.2016.

van Eck, R. (2006): Digital Game-Based Learning. It's Not Just the Digital Natives Who Are Restless... In: *Educause Review* 41 (2).

Vandenberghe, J. (2013): Applying the 5 Domains of Play. Acting like Players, 2013. Online verfügbar unter <http://www.gdcvault.com/play/1017876/Applying-the-5-Domains-of>, zuletzt geprüft am 21.03.2016.

wccfttech.com (2016): GTA V Most Expensive Video Game in History - Budget More than High Budget Hollywood Films. Online verfügbar unter <http://wccfttech.com/gta-v-most-expensive-video-game-history/>, zuletzt aktualisiert am 22.03.2016, zuletzt geprüft am 22.03.2016.

Weißböck, E. (2014): Guidelines zur Evaluierung digitaler Lernspiele. Eine Beurteilung der Anwendbarkeit. Masterthesis.

Wood, J. (2001): Can software support childrens vocabulary development. In: *Language Learning and Technology* 5 (1), S. 166–201.

www.psychclassics.yorku.ca (2007): Classics in the History of Psychology -- Yerkes & Dodson (1908). Online verfügbar unter <http://psychclassics.yorku.ca/Yerkes/Law/>, zuletzt aktualisiert am 13.09.2007, zuletzt geprüft am 06.05.2016.