

Department Elektrotechnik und Informatik  
der Universität Siegen

Institut für Wissensbasierte Systeme &  
Wissensmanagement

Bachelorarbeit

Gamification von Mathematik in den  
Klassen 9 und 10

---

Gamification of mathematics in class 9  
and 10

Eingereicht von:  
Gerrit Großkopf  
Matrikel-Nr.: 1012091  
Studiengang:  
Informatik B.Sc.

Erstprüfer\*in:  
Prof. Dr.-Ing. Madjid Fathi  
Zweitprüfer\*in:  
M.Sc. Marjan Khobreh

Siegen, den 25.04.2015

## Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, insbesondere keine anderen als die angegebenen Informationen aus dem Internet.

Diejenigen Paragraphen der für mich gültigen Prüfungsordnung, welche etwaige Betrugsversuche betreffen, habe ich zur Kenntnis genommen.

Der Speicherung meiner Diplomarbeit zum Zwecke der Plagiatsprüfung stimme ich zu. Ich versichere, dass die elektronische Version mit der gedruckten Version inhaltlich übereinstimmt.

---

(Unterschrift der Verfassenden Person)

(Ort, Datum)

# 1 Abstrakt / Abstract

## 1.1 english

The topic of this B.A. thesis will be gamification of material contained in - German Gesamtschule - grade nine and ten mathematics curriculum. Since the amount of games for teaching mathematics has dramatically decreased over the years, and companys developing said games, like Terzio[30], have been sold (in this case to Carlsen, wich resulted in the end of their game production), I saw it fit to analyse the potential of „Lernspiele“<sup>1</sup>, and the implementation of such games in Germany.

In order to understand the potential, this thesis will introduce readers to relevant theory about computer game creation to enable them to analyze what existing computer games have to offer. Thanks to a cooperation with the Euregio Gesamtschule Rheine an examination of what students consider a Lernspiel, and what in their opinion makes a good game, was made. After a short definition of the juristical definition of school equipment, which a gamificated game should be, I will talk about the design and implementation of my videogame „Built on solid Seas“, wich is based on my findings in this thesis. There I will examine ideals of design and implementation, and how design and implementation turned out in the context of this thesis.

Lastly, students tested the game, and one of their teachers was interviewed on applicability of such software and „Built on Solid Seas“ was compared to media it competes against. This way, I will first look at the theoretical background of gamification of school contents, then explain the practical design and developement of such, and lastly conclude the thesis with results of my investigation on the current situation of such games.

---

<sup>1</sup>A german term for software gamificating school topics

## 1.2 deutsch

In dieser Bachelorarbeit behandle ich das Thema der Gamification der Klassen 9 und 10 Mathematik. Aufgrund der stark zurückgehenden Menge an Lernspielen und dem Verkauf von ehemals großen Lernspiel-produzierenden Firmen wie Terzio[30] an Carlsen, was in dem Ende ihrer Spiele-Produktion mündete, empfand ich es als wichtig, die Möglichkeiten von Lernspielen, und deren Umsetzung zu untersuchen.

Hierzu beginne ich in der Theorie zum Erstellen von Videospiele, um bewerten zu können, was ein gegebenes Lernspiel bietet. Im Anschluss untersuche ich, was von den Schüler\*innen meiner Partnerschule für die Untersuchung, der Euregio Gesamtschule Rheine[24], als Lernspiel, und was als gutes Spiel wahrgenommen wird.

Nach einer kurzen Definition der rechtlichen Definition eines Lernmediums nach Schulgesetz gehe ich auf die parallel zu der BA stattfindende Entwicklung von meinem, auf diesen Untersuchungen basierenden, Videospiele „Built on solid Seas“ ein. Dort untersuche ich, wie die Entwicklung ablaufen sollte. Zum Schluss lasse ich „Built on solid Seas“ von den Schüler\*innen testen, befrage einen Lehrer zu der Einsetzbarkeit der Software und vergleiche es mit den Medien, mit denen es konkurriert.

So gehe ich zuerst auf den theoretischen Hintergrund eines Lernspiels in der heutigen Zeit ein, zeige dann den praktischen Ablauf der Entwicklung, und schliesse die Bachelorarbeit mit einem Fazit über die Situation dieses Lernspiels in der heutigen Zeit.

## Abbildungsverzeichnis

3	Das Framework „The I’s have it“ . . . . .	10
4	Der Flow-Channel nach Mihaly Csikszentmihalyi . . . . .	11
5	Das Framework „MDA“ . . . . .	12
6	Das Framework „MDE“ . . . . .	13
7	Das Framework „DPE“ . . . . .	14
8	Das Framework „P-III“ . . . . .	16
9	Das Framework „SGDA“ . . . . .	17
10	Der Zusammenhang der Frameworks skizziert . . . . .	19
11	Die Struktur der Klassen in „Built on solid seas“ . . . . .	37
12	Das Hauptmenü von „Built on solid seas“ . . . . .	37
13	Die Charakter-Erstellung in „Built on solid seas“ . . . . .	38
14	Die Spielwelt in „Built on solid seas“ . . . . .	39
15	Die Aufgabenbearbeitung von „Built on solid seas“ . . . . .	40
16	Die Dialoggenerierung in „Built on solid seas“ . . . . .	40
17	Der Export am Beispiel des Inventars . . . . .	41
18	Der Import am Beispiel des Inventars . . . . .	41
19	Das Laden des Spiels zeigt die komplexe Speicherstruktur. . . . .	42
20	Die Aufgabenerstellung in „Built on solid seas“ . . . . .	43
21	Das Fenster zum Bearbeiten von Dialogen . . . . .	46
22	So sind die Ereignisse in der Globals Klasse definiert und aufgerufen . . . . .	46
23	Der Code für die Wassersimulation . . . . .	48

# Inhaltsverzeichnis

<b>1</b>	<b>Abstrakt / Abstract</b>	<b>3</b>
1.1	english . . . . .	3
1.2	deutsch . . . . .	4
<b>2</b>	<b>Einleitung</b>	<b>8</b>
<b>3</b>	<b>Hintergrund</b>	<b>9</b>
3.1	Frameworks . . . . .	9
3.1.1	The I's have it . . . . .	9
3.1.2	Mechanics, Dynamics and Aesthetics . . . . .	12
3.1.3	Mechanics, Dynamics and Emotions . . . . .	13
3.1.4	Design, Play and Experience . . . . .	13
3.1.5	P-III . . . . .	16
3.1.6	Serious Game Design Assessment . . . . .	17
3.1.7	Vergleich . . . . .	19
3.1.8	Erste Befragung der Schüler*innen . . . . .	20
3.2	Vergleich von herkömmlichen Lernspielen . . . . .	22
3.2.1	Tim 7 . . . . .	22
3.2.2	Addy . . . . .	22
3.2.3	Crazy Machines 2 . . . . .	23
3.2.4	S.O.S. Tiefsee . . . . .	23
3.2.5	Phase 6 . . . . .	24
3.2.6	Zusammenfassung . . . . .	24
3.3	Themen als Questpfade (Learning) . . . . .	24
3.3.1	Lehrplan NRW . . . . .	25
3.3.2	Umsetzung der Themen in Büchern . . . . .	25
3.3.3	Umsetzung der Themen in „Built on solid seas“ . . . . .	27
<b>4</b>	<b>„Built on solid seas“</b>	<b>29</b>
4.1	Storytelling . . . . .	30
4.1.1	Character Setting and narrative . . . . .	30
4.1.2	Storytelling . . . . .	31
4.1.3	Story . . . . .	31
4.2	Gameplay . . . . .	31
4.2.1	Mechaniken . . . . .	32
4.2.2	Dynamiken . . . . .	32
4.2.3	Affekt, Emotion, Ästhetik . . . . .	33
4.3	User Experience . . . . .	33
4.3.1	User Interface . . . . .	34
4.3.2	Interactivity . . . . .	34
4.3.3	Engagement . . . . .	35
4.4	Zusammenfassung . . . . .	35
<b>5</b>	<b>Entwicklung von „Built on solid seas“</b>	<b>36</b>
5.1	Szenenstrukturen . . . . .	37
5.1.1	Hauptmenü . . . . .	37
5.1.2	Charaktererstellung . . . . .	38

5.1.3	Spielwelt . . . . .	38
5.1.4	Aufgabenbildschirm . . . . .	39
5.1.5	Dialoggenerator . . . . .	40
5.2	Globale Klassen . . . . .	40
5.2.1	Import-Export . . . . .	41
5.2.2	Savegame . . . . .	41
5.2.3	Item . . . . .	42
5.2.4	Globals . . . . .	42
5.3	Spielelemente . . . . .	42
5.3.1	Dialoge- und Aufgabenerstellung und ihre Bearbeitung . . . . .	43
5.3.2	Spezialereignisse . . . . .	45
5.3.3	Towerdefense . . . . .	46
5.3.4	Motivationselemente . . . . .	49
5.3.5	Einschätzung der Spielenden . . . . .	50
5.3.6	Shading . . . . .	50
<b>6</b>	<b>„Built on solid seas“ im Test</b>	<b>52</b>
6.1	Erste Befragung der Schüler*innen . . . . .	52
6.1.1	Ergebnis . . . . .	52
6.2	Zweite Befragung der Schüler*innen . . . . .	54
6.2.1	Ergebnis . . . . .	54
6.3	Befragung der Lehrer*innen . . . . .	55
6.3.1	Ergebnis . . . . .	56
<b>7</b>	<b>Ergebnisse</b>	<b>56</b>
7.1	Frameworks . . . . .	56
7.2	Vergleich mit Büchern . . . . .	57
7.3	Vergleich mit anderen Lernspielen . . . . .	57
7.4	Zusammenfassung . . . . .	57
7.5	Ausblick . . . . .	57
	<b>Appendices</b>	<b>61</b>

## 2 Einleitung

In dieser Bachelorarbeit werde ich ein unterrichtsbegleitendes Videospiel für die Mathematik der Klassen 9 und 10 entwickeln. Hiermit möchte ich die Fragen “Hat ein Videospiel gegenüber einem Schulbuch bei diesen Themen Vorteile oder Nachteile?” und “Welche Fehler machen herkömmliche Lernspiele, dass sie die Schüler\*innen nicht erreichen?” studieren. In der Bachelorarbeit werde ich mich auf die Themen “Satz des Pythagoras”, “Kreisberechnung”, “Zylinder” und “Pyramiden, Kegel und Kugeln” beschränken, und somit das Thema stark einschränken. Für die Forschung werde ich auf verschiedene Frameworks eingehen, sie vergleichen und ein oder mehrere Frameworks für den Prototypen auswählen, um anhand dessen das Spiel zu entwerfen. Momentan gibt es in Deutschland gerade in den Mathematikkursen sehr große Unterschiede zwischen den Menschen, die die Thematik verstehen, und denen, die die Thematik nicht verstehen. Darüber hinaus sind Firmen wie Coktelvision mit Addy[38], Firmen wie Terzio mit S.O.S. Tiefsee[36] und heureka klett mit Tim 7[19] sind daran gescheitert den Markt der Lernspiele effektiv zu nutzen und existieren in der Form heutzutage nicht mehr. Es ist also sehr wichtig, gerade jetzt zu forschen, wo die Fehler der Hersteller\*innen von Lernspielen liegen, um den Unterricht mit Lernspielen besser unterstützen zu können. Daher werde ich in drei Umfragen mit den Nutzer\*innen, also den Schüler\*innen klären, wie das Spiel aussehen muss, so dass es ihnen Spaß macht. Im Anschluss werde ich in einem Interview mit einem Lehrer klären, wie das Spiel aussehen muss, damit es die Themen gut unterrichtet. Hier unterstützt mich die Euregio Gesamtschule Rheine[24] Zum Abschluss werde ich Interviews mit den Schüler\*innen führen um zu erfahren, ob es Spaß gemacht hat, und mit den Lehrer\*innen, ob es alle Aspekte des Unterrichtsfeldes ausreichend behandelt.

## 3 Hintergrund

### 3.1 Frameworks

Verschiedene Frameworks<sup>2</sup> stellen der Entwicklung dieser Spiele gewisse Richtlinien und Regeln, die das Spiel zum Erfolg führen sollen und den Entwickelnden einen Leitfaden geben, was sie bei der Entwicklung wissen sollten. Verschiedene Frameworks wurden für verschiedene Formen von Spielen entwickelt und geben dadurch für das hier zu entwickelnde Spiel unterschiedliche Empfehlungen. Es ist also wichtig, jene Frameworks besonders zu betonen, die diese explizite Form von Spielen besonders fördern.

In diesem Kontext habe ich mir einige Frameworks genauer angeschaut, und den Zusammenhang der Frameworks „The I’s have it“[5] , „M.D.A.“[31], „M.D.E.“[16], „P-III“[11] und „D.P.E.“[41] als wichtige Elemente betrachtet. Diese Frameworks beachten jeweils bestimmte Aspekte eines Spiels, haben im Bezug auf mein Spiel ihre Vor- und Nachteile, und legen den Fokus auf einige wichtige Elemente. Des weiteren betrachte ich „SGDA“[26], Welches zwar eher den Bereich „Persuasive Games“ bearbeitet und durch das Fehlen der Spielenden-Zentrierung nicht zentral in meinem Fokus liegt, aber auch Inspiration für einige Elemente meines Spiels war.

#### 3.1.1 The I’s have it

Das Framework „The I’s have it“[5] ist ein Spielenden-zentriertes Framework von Leonard A. Annetta aus dem Jahr 2010, das im Zentralen die Identität der Spielenden und ihr Eintauchen in die virtuelle Welt hat. Es teilt sich auf in Identity<sup>3</sup>, Immersion<sup>4</sup>, Interactivity<sup>5</sup>, Increasing Complexity<sup>6</sup>, Informed Teaching<sup>7</sup> und Instructional<sup>8</sup>, die, wie auf dem Bild zu sehen, verschachtelt sind. Dabei legt es sehr großen Wert darauf, dass die Spielenden aus eigener Motivation heraus das Spiel spielen wollen, da intrinsische Motivation[5, S. 106] für das so genannte Flow-Gefühl[10] unabdingbar ist. Wie man auch anhand des Bildes erkennen kann, staffeln sich die I’s nach Wichtigkeit und nach Abdeckung des Gesamtspiels. Identity ist in jedem Element zu beachten, es muss jedoch auch die Möglichkeit bestehen, dass andere Menschen ihre Identität in dem Spiel ausleben können. Somit ist für einen einzelnen Spieler nicht die komplette Immersion abgedeckt. Vielen Spielen gelingt es beispielsweise nicht, weibliche Charaktere zu erstellen, mit denen sich Spielerinnen problemlos identifizieren können. Oftmals wirken Spiele für Spielerinnen nicht erschaffen. So kommt es auch in vielen berühmten Titeln wie Fallout 4[7] dazu, dass für Spielerinnen hin und wieder falsch übertragene Interface-Elemente die Spielerfahrung hemmen.

**Identity** Hier wird empfohlen, besonders darauf zu achten, den Spielenden die Identifikation mit dem Hauptcharakter oder den Nebencharakteren möglichst einfach zu ermöglichen. So lässt sich dieses Framework lediglich auf Rollenspiele anwenden. Es wird eine Studie erwähnt, die belegt, dass Spieler\*innen, die zwischen 100 verschiedenen Avataren<sup>9</sup>

---

<sup>2</sup>Gerüste

<sup>3</sup>Identität

<sup>4</sup>Eintauchen

<sup>5</sup>Interaktivität

<sup>6</sup>steigende Komplexität

<sup>7</sup>informiertes Unterrichten, Unterrichten mit dem Wissen über den Zustand der Lernenden

<sup>8</sup>Ausbildend

<sup>9</sup>Repräsentationen von Spielenden in der Spielwelt

wählen können, viel schneller in das Spiel eintauchen, als jene, die nur zwischen 2 Avataren wählen können (vgl. [5, S. 106,107]).

Wie man hier schon erkennt, verlangt das Framework quasi eine Art Rollenspiel mit Charaktererstellung. Spiele ohne sichtbare Charaktere wie Tetris sind hier nicht im Fokus des Frameworks.

**Immersion** Der zweitwichtigste Punkt des Frameworks ist „Immersion“ das Eintauchen der Spielenden in die Welt. Daher muss die Welt eine in sich schlüssige sein, und kann nicht unterbrochen werden. Hier muss den Spielenden eine Welt geboten sein, in der sie ein Flow-Gefühl[10] entwickeln. Flow ist ein sehr konzentrierter Emotionsstatus, der relativ fragil ist. Hier muss darauf geachtet werden, dass den Spielenden Aufgaben gegeben werden, die sie nicht unterfordern, sie aber auch nicht frustrieren. Dadurch sollen sie in dem sogenannten Flow-Channel von wenig Fähigkeiten und einfachen Anforderungen zu viel Fähigkeiten und viel Anforderungen gebracht werden (vgl. [5, S. 107]). Eine Art Level-System macht hier Sinn, das überprüft, welche Aufgaben die Spielenden bewältigen können und wo ihre Schwächen liegen. Es würde also in meinem Kontext extrem wichtig sein, zu überprüfen, wo die Schwächen und Stärken der Personen sind, um gezielt etwas länger einfachere Sachen zu fordern, falls sie an vielen Aufgaben scheitern. Genauso wichtig ist es aber auch, gezielt schwerere Aufgaben zu stellen, wenn Schüler\*innen die Aufgaben ohnehin schon bewältigt bekommen.



Abbildung 3: Das Framework „The I’s have it“

**Interactivity** Als nächsten Punkt bietet das Framework die Interaktion und meint damit die soziale Interaktion mit anderen Charakteren oder Nicht-Spieler-Charakteren (vgl.[5, p. 108]). Aufgrund der zeitlichen Beschränkung der Bachelorarbeit habe ich mich hier auf Nicht-Spieler-Charaktere beschränkt, um keine Client-Server-Struktur in der Schule zu installieren, welche ohnehin ausgelagert werden müsste, wenn man das Spiel öffentlich zugänglich machen wollte. Es wird ebenfalls empfohlen, dass die Avatare beispielsweise synchron zum Gesprochenen die Lippen bewegen sollten, und Emotionen zei-

gen können sollten.

**Increasing Complexity** Punkt 4 des Frameworks beschäftigt sich mit der steigenden Komplexität. So lassen sich Kinder im Alter von bis zu 2 Jahren mit einfachen Spielen begeistern, Kinder im Alter von etwa 7 Jahren haben die Fantasie, um aus Symbolen ganz andere Objekte zu lesen, und darauf folgen dann Spiele, die Regeln haben. Mit dem Einführen von Regeln zeigt sich dann die Möglichkeit, einfach eine Welt zu erschaffen, in der sich die Spielenden frei nach den Regeln bewegen können (vgl. [5, p. 108-109]). Mit diesen Regeln zeigt sich auch das Problem des Balancing<sup>10</sup>, also dem Feinjustieren von Regeln: Wie schwer ist eine Aufgabe nach den Regeln? Ist sie zu einfach oder doch schon zu schwer? Damit wiederholt Annetta in gewisser Weise die Brisanz des Flow-Gefühls. Schließlich ist das Ziel von Lernspielen, dass die Fähigkeiten der Spielenden sich steigern, und somit muss, um die Aufmerksamkeit nicht abbrechen zu lassen, die Schwierigkeit der Aufgaben sich immer an die Fähigkeiten der Spielenden anpassen.

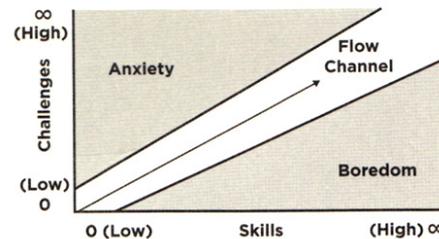


Abbildung 4: Der Flow-Channel nach Mihaly Csikszentmihalyi

**Informed Teaching** Um die Fähigkeiten der Spielenden einzuschätzen, wird im vorletzten Punkt (Informed Teaching) empfohlen, dass anhand von Algorithmen die Fähigkeiten der Spielenden eingeschätzt und beobachtet werden müssen (virtual observations<sup>11</sup>), um nicht aus dem Flow-Gefühl herauszufallen. Hier scheint es mir sinnvoll, die Bearbeitungszeit mit der Schwierigkeit der Aufgaben in Relation zu setzen, um anhand dessen einzuschätzen, was die Spielenden bewältigen können. In diesem Punkt wird auch empfohlen die Lehrpersonen der Spielenden über die Fortschritte der Lernenden zu informieren, was den Rahmen meines Projekts allerdings sprengt.

**Instructional** Im letzten Punkt geht Annetta noch auf die Lehrperson ein. Diese sieht sich seiner Einschätzung nach oftmals einer zu großen Gruppe junger Menschen gegenüber, und lässt schnell die besonders Leistungsstarken oder -schwachen aus der Aufmerksamkeit schwinden. Da ein Mensch unmöglich an mehreren Stellen gleichzeitig existieren kann, ein NPC<sup>12</sup> hingegen schon, findet in genau diesem Bereich viel Forschung statt. Eine Künstliche Intelligenz soll die Lehrperson unterstützen, indem sie die Schüler\*innen begleitet und ihnen zu ihren Fähigkeiten passend Hinweise bzw. Hilfe gibt (vgl. [5, p. 109]). Auch aus diesem Vorteil heraus wird die Möglichkeit empfohlen, Spiele fest in den Lehrplan einzubauen, was zum derzeitigen Stand der Dinge in Deutschland einige Schwierigkeiten geben kann. So muss Lernmaterial erst vom zuständigen Ministerium für die Lernmittelfreiheit in den Lernmittelkatalog aufgenommen werden, was Bedingungen an das Spiel stellt, die außerhalb des Rahmens der Bachelorarbeit sind. So müsste die Schulkonferenz

<sup>10</sup>Ausbalancieren, hier von Schwierigkeitsgraden

<sup>11</sup>virtuelle Beobachtung

<sup>12</sup>nicht-Spieler-Charakter

dem Spiel zustimmen. Hierfür wäre im Falle von Mathematik an Gesamtschulen ein eigenes Gutachter-Verfahren nötig[28]. Der schwierigste Punkt wäre ohnehin, dass das Spiel mindestens ein ganzes Halbjahr die Schüler\*innen mit Informationen zu dem Fach versorgen können soll. Letzteres zu liefern ist meines Erachtens nach in vier Monaten von einer einzelnen Person ohnehin nur schwerlich zu bewältigen, die parallel dazu noch eine Bachelorarbeit schreibt, und bedarf der weiteren Unterstützung mit fachlicher Kompetenz. Dennoch ist dieser Punkt ein entscheidender Vorteil, den Spiele gegenüber Büchern und Lehrern haben können.

### 3.1.2 Mechanics, Dynamics and Aesthetics

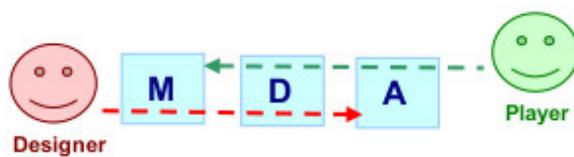


Abbildung 5: Das Framework „MDA“

MDA (Mechanics, Dynamic and Aesthetics) [31] ist ein Framework von Robin Hunicke, Marc LeBlanc und Robert Zubek, das in den Jahren 2001 bis 2004 entstand, und daher als das Älteste von mir behandelte Framework Inspiration für einige andere Frameworks war.

Die Designenden entwickeln Spiele in der Regel über Mechaniken, das MDA Framework

bietet für die generelle Spielentwicklung hier ein Ziel an, auf das es hinarbeiten soll. Dadurch, dass Designende von der Mechanikseite her auf das Spiel Einfluss nehmen, Spielende aber vor allem die Ästhetik sehen, haben unterschiedliche Menschen einen unterschiedlichen Eindruck des Spiels[31, p. 2 MDA as Lens]. Kleinste Veränderungen in der Mechanik können die Ästhetik stark verändern und die Dynamik aus dem Gleichgewicht bringen.

**Mechanics** Als Mechanics werden die Mechaniken des Spiels, also die Algorithmen, die Programmcodes, die Dateien und ihre Kombination bezeichnet, die das Spiel zu dem machen, was es ist[31, p. 3,4]. Die Mechaniken sind also die Seite des Spiels, die die Designenden direkt sehen und beeinflussen können. Aus ihnen entstehen die Dynamiken.

**Dynamics** Dynamics, also Dynamiken, sind die Taktiken und Abläufe des Spiels, das die Spielenden mit der Hilfe der Mechaniken entwickeln. Hier zeigen sich bestimmte Strategien oder Handlungsabläufe, die es den Spielenden ermöglichen, das Spiel möglichst problemlos zu bewältigen[31, p. 3]. Bestimmte Dynamiken lassen sich oftmals durch die Mechaniken feinjustieren. Gelingt die Mechanik nicht, erscheinen einige von den Entwickelnden erwünschte Handlungen zu schwer oder zu einfach und werden daher umgangen.

**Aesthetics** Als Aesthetics oder Ästhetik werden letztendlich dann die Eindrücke bezeichnet, die bei den Spielenden ankommen. Sie entstehen aus den Dynamiken, ob diese ausgeglichen sind, ob und vor allem warum das Spiel Spaß macht. Diese Ästhetik sollten die Designenden als Ziel vor Augen haben, um die Mechaniken dahingehend anzupassen. Designende können auf verschiedene Ziele wie „Sensation<sup>13</sup>“, „Fantasy<sup>14</sup>“, „Narrative<sup>15</sup>“,

<sup>13</sup>positives Gefühl

<sup>14</sup>fantasie

<sup>15</sup>erzählend

„Challenge<sup>16</sup>“, „Fellowship<sup>17</sup>“, „Discovery<sup>18</sup>“, „Expression<sup>19</sup>“ und „Submission<sup>20</sup>“ hinarbeiten[31, p. 2]. Verschiedene Ästhetiken stellen das Spiel in unterschiedlichen Facetten dar. Spiele haben in der Regel Aspekte, die sie besonders hervorheben. Das Spiel „Sims[23]“ beispielsweise betont nach den Autoren des Frameworks besonders Discovery, Fantasy, Expression und Narrative[31, p. 2]. Die Spiele, die von den Lernenden besonders gerne gespielt werden (Minecraft[25], Counter-Strike[37], Call of Duty[39], Fifa[9], World of Tanks[40] und League of Legends[15]), betonen besonders den Aspekt Challenge, und weiterhin Fantasy, Fellowship und Expression. Auch das am liebsten gespielte Lernspiel Quizduell[2] betont die Wichtigkeit von Challenge und Expression. Es scheint den Lernenden also besonders darum zu gehen, gefordert zu werden, und sich miteinander zu messen und weniger darum, eine Geschichte zu erleben, wie es mit dem Aspekt Narrative der Fall wäre.

### 3.1.3 Mechanics, Dynamics and Emotions

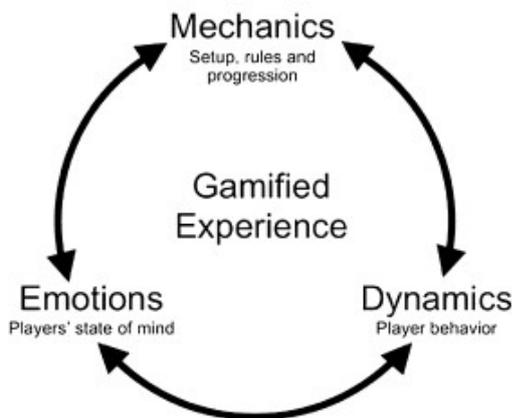


Abbildung 6: Das Framework „MDE“

MDE (Mechanics<sup>21</sup>, Dynamics<sup>22</sup> and Emotions<sup>23</sup>) [16] ist ein Framework von Karen Robson, Kirk Plangger, Jan H. Kietzmann und Ian McCarthy und Leyland Pitt aus dem Jahr 2015. Somit ist es das aktuellste Framework, das ich hier behandle, und daher ist es nicht verwunderlich, dass es einem anderen Framework nachempfunden ist. MDE stellt eine Weiterentwicklung des Frameworks MDA dar(vgl. [16, p. 4]), das auf Lernspiele spezialisiert ist, statt generell alle Spiele zu behandeln. Mechaniken und Dynamiken sind hier gleich. Der Unterschied liegt in dem letzten Punkt, „Emotions“.

**Emotions** Die Emotionen hingegen sind etwas, das in dem MDE Framework anders ist.

Hier wird nicht mehr gefragt „Warum macht das Spiel Spaß?“, sondern „Wie fühlen sich die Spielenden?“ Schließlich würden Spielende ein Spiel nicht spielen, wenn es ihnen nicht gefällt[16, p. 7].

### 3.1.4 Design, Play and Experience

Das „DPE“ („Design<sup>24</sup>, Play<sup>25</sup> and Experience<sup>26</sup>“) -Framework [41] ist eine weitere Spezialisierung des „MDA“ -Frameworks. DPE ist entwickelt von Brian M. Winn im Jahr

<sup>16</sup>herausfordernd

<sup>17</sup>Kamerad\*innenschaft beinhaltend

<sup>18</sup>erforschend

<sup>19</sup>Emotionen ausdrückend

<sup>20</sup>unterbreitend

<sup>21</sup>Mechaniken

<sup>22</sup>Dynamiken

<sup>23</sup>Emotionen

<sup>24</sup>Darstellung

<sup>25</sup>Spielerei

<sup>26</sup>Erfahrung

2009. Es ist auf Lernspiele spezialisiert und unterscheidet sich damit schon von dem zuvor entwickelten MDA[31]. MDA ist ein Framework, das lediglich gegenüberstellt, welche Anforderungen die Spielenden und die Entwickelnden an die drei Aspekte des Spiels stellen. DPA hingegen splittet die Anforderungen in die vier Abschnitte „Learning<sup>27</sup>“, „Storytelling<sup>28</sup>“, „Gameplay<sup>29</sup>“ und „User-Experience<sup>30</sup>“ auf, die jeweils aus der Design-, der Play- und der Experience-Sicht bestehen.

**Learning** Der Lern-Aspekt des Spiels (Learning) stellt gegenüber, dass die Designen- den in das Design verständlicherweise der Thematik entsprechende Inhalte liefern müssen (Content<sup>31</sup> and Pedagogy<sup>32</sup>), die auf der anderen Seite von den Spielenden gelernt werden soll (Learning). Dazwischen steht das, was das Spiel selbst leisten können muss, nämlich den Inhalt beizubringen (Teaching<sup>33</sup>)[41, p. 7,8]. Im Falle eines Spiels für eine Schule ist der Content schon von der Thematik her sehr klar definiert, und kann höchstens noch von Lehrpersonen oder von Nachhilfelehrer\*innen absegnet werden, was eine interdisziplinäre Zusammenarbeit nach dem P-III Framework fordert. Von Seite der Lernenden her stellt sich gerade im Bereich Mathematik eine besondere Schwierigkeit dar, da viel Stoff aufeinander aufbaut, und sich ein Versagen einer einzelnen Lehrperson, meiner Erfahrung als Nachhilfelehrer nach, oft bis zum Ende der Schulzeit bemerkbar macht. Ebenso wichtig ist es, hier, wie auch von Schulbüchern oft empfohlen, Dinge zu wiederholen, die für die Themen bekannt sein müssten. So finden sich auch eventuell vorher liegende Wissenslücken, deren Behebung extrem wichtig für das Fortschreiten in der Thematik sind. Auch rein rechtlich ist es sinnvoll, je nach Bundesland die Inhalte mit dem jeweiligen Lehrplan abzustimmen, da dieser genau definiert, was von den Lernenden zu lernen ist.

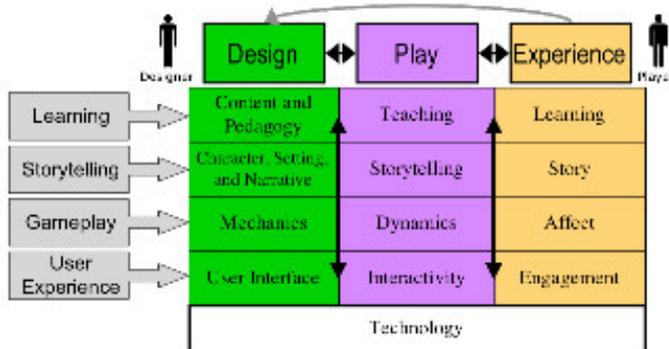


Abbildung 7: Das Framework „DPE“

**Storytelling** Den Storytelling-Aspekt spaltet man in der Regel in die Geschichte der Entwickler\*innen, und der Spieler\*innen auf. Die Designer\*innen entwickeln eine, mehrere, oder gar keine Geschichten, die sie in das Spiel fest integrieren, und stellen die Umgebung, die Problematiken sowie die anderen Charaktere, mit denen sie diese durchleben (Character, Setting and Narrative). Die Spielenden allerdings

entwickeln im Laufe des Spielens eine ganz eigene Geschichte, nämlich die, die sie während des Spielens erleben (Story). Die Aspekte, die nun von Spielenden am häufigsten miterlebt werden, ergeben zu guter Letzt die Geschichte des Spiels (Storytelling)(vgl. [41, p.

<sup>27</sup>lernen

<sup>28</sup>erzählen

<sup>29</sup>Spiel

<sup>30</sup>Benutzer\*innen-Erlebnis

<sup>31</sup>Inhalt

<sup>32</sup>Pedagogik

<sup>33</sup>Unterrichten

8,9]). Spannend ist an diesem Punkt zu erwähnen, dass eine Umfrage bei der Zielgruppe ergab, dass sehr oft die Spielenden Spiele bevorzugen, die keine, oder kaum narrative Elemente vorweisen. Wie schon im Zusammenhang MDA erwähnt, wird also kein Wert auf Narrative gelegt. Offenbar gefällt es der Zielgruppe mehr, ihre eigene Geschichte anhand der gegebenen Umstände und Charaktere zu realisieren, als eine von den Designer\*innen vorgegeben zu bekommen. Die Aufgabe der Designenden beschränkt sich also in dieser Ebene fast komplett darauf, das Setting und die Charaktere zu entwickeln, mit denen die Spielenden interagieren können.

**Gameplay** Die Gameplay-Ebene bricht sich herunter auf Mechaniken, Dynamiken und Emotion. Hier fällt auf, dass explizit die 3 Aspekte des MDE Frameworks genannt sind. Im Englischen steht hier zwar als Emotion „Affekt“ und im MDE Framework „Emotion“, aber die Bedeutung beider Worte lässt sich auf verschiedene Aspekte des Wortes Emotion zurückführen. Mechaniken sind die von den Designenden implementierten Möglichkeiten und Ergebnisse im Verlauf des Spiels. Die Emotion der Spielenden definiert, wie sie sich beim Spielen fühlen, ob sie motiviert sind, ob sie Lust haben weiterzuspielen[41, p. 9]. Hier greift auch das Konzept von Flow, dass die Designer\*innen darauf achten müssen, die Aufmerksamkeit der Spielenden nicht zu verlieren, sie nicht zu überfordern, sie aber auch nicht zu unterfordern[41, p. 10]. Diese Ebene entspricht im ehesten wie gesagt dem MDE Framework. Der Unterschied liegt hier darin, dass der Bereich der Emotion im DPE Framework als Affekt behandelt wird. Es handelt sich also hier auch um die zentrale Überschneidung mit dem vorher entstandenen MDA Framework, aus dem es sich schließlich auch spezialisiert hat[41, p. 6]. Mit der Wandlung von Aesthetics zu Affect<sup>34</sup> konzentriert sich das Spiel nicht darauf, möglichst schön zu sein, sondern eher, möglichst die Aufmerksamkeit und das Flow-Gefühl der Spielenden zu ergattern und zu behalten. Des weiteren wird ein Reward System benötigt, das ständig die Spielenden für ihre Leistungen belohnt, und ihnen die Möglichkeit gibt, während des Lernens ihren Erfolg zu sehen.

**User Experience** Auf der User Experience Ebene basiert das Interface und die Steuerung (User Interface). Diese sollen möglichst intuitiv sein, damit die Spielenden durch das Interface möglichst tief in die konstruierte Welt eintauchen können (immersion[41, p. 12]). Diese Ebene entspricht in einigen Aspekten dem Framework „The I’s have it“. Die Spielenden tauchen hier in das Spiel ein, das die Designenden ihnen bieten, und interagieren mit ihm in den ihnen gegebenen Arten (Interactivity). Daher ist diese Ebene die Ebene, die von den Spielenden am intensivsten wahrgenommen wird.

**Interaction between Layers** Die vertikalen Pfeile in der Skizze (siehe 7) des Frameworks lassen schon vermuten, was hier an Interaktion stattfindet. es mag zwar die User Experience Ebene mit der Interaction sein, in die die Spielenden eintauchen, dennoch beeinflusst diese stark das Gameplay und die Spielenden erleben mit dem Storytelling dann den Lerneffekt. Genauso funktioniert das Ganze auch anders herum: Lernen die Spielenden etwas dazu, eröffnen sich andere Bereiche der Story, eventuell neue Gameplay-Mechaniken und andere Formen von User Experience[41, p. 12]. Eine Ebene funktioniert nur dadurch, dass die anderen Ebenen mit ihnen interagieren. Das bedeutet natürlich nicht, dass jedes Lernspiel beispielsweise eine Story implementiert haben muss. Schliesslich ist das, was die Spielenden als Story erleben, oft lediglich die Geschichte die sie sich selbst erschaffen, da-

---

<sup>34</sup>berühren

durch dass die Entwickler\*innen ihnen das Setting und die anderen im Spiel vorhandenen Charaktere geben.

### 3.1.5 P-III

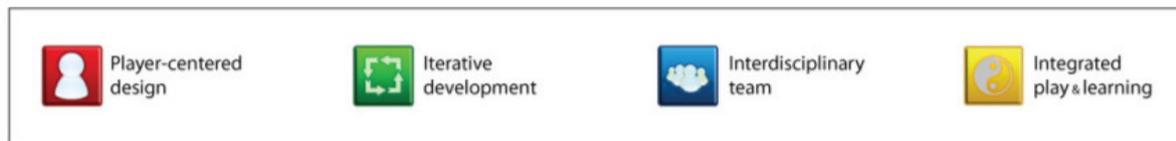


Abbildung 8: Das Framework „P-III“

Das Framework „P-III“ [11] ist ein, im Jahr 2012 von Vero Vanden Abeele, Bob De Schutter, Luc Geurts, Stef Desmet, Jeroen Wauters, Jelle Husson, Lieven Van den Audenaeren, Frederik Van Broeckhoven, Jan-Henk Annema, und David Geerts entwickeltes Framework, das das MDA-Framework lediglich in einem Abschnitt erwähnt und viel mehr auf den zeitlichen Ablauf der Entwicklung eingeht, der es zulässt, möglichst oft mit Experten und Spielenden zu kommunizieren. P-III teilt sich in ein P und drei I's auf, das P steht hier für Player-Centered Design<sup>35</sup>, die I's der Reihe nach für iterative Entwicklung, interdisziplinäre Zusammenarbeit und Integration von Spielspaß und Lernerfahrung.

**Player-Centered Design** Beim P-III Framework kommt es ganz besonders im Player-Centered Design darauf an, die Designenden daran zu erinnern, dass sie nicht die Spielenden sind. Die Menschen, die Videospiele designen, haben oftmals einen akademischen Hintergrund und andere Interessen und Fähigkeiten als jene, die das Spiel nachher spielen sollen (vgl. [11, p. 82]). Daher ist es essenziell, zwischendurch die Spielenden zu befragen, ob das Spiel in den verschiedenen Phasen der Entwicklung den Wünschen der Zielgruppe entspricht.

**Iterative Development** Die iterative Entwicklung stellt sicher, dass der Vorgang der Entwicklung zeitlich strukturiert ist. So wird diese aufgeteilt in Concept Design<sup>36</sup>, Game Design<sup>37</sup> und Game Development<sup>38</sup>, die jeweils mit einer Expert\*inneneinbindung beendet werden. Im Punkt Game Development befindet sich die Entwicklung in einer Schleife, in der nach Playtests<sup>39</sup> und vielen Prototypen immer wieder Design- und Software-Entwicklung stattfindet (vgl. [11, p. 83]).

**Interdisciplinary Teamwork** Interdisziplinäre Zusammenarbeit betont genau diese Expert\*innen, mit denen sich zur Entwicklungszeit immer wieder ausgetauscht werden sollte, da für ein Lernspiel egal welcher Kategorie die Entwickelnden in den seltensten Fällen das Wissen der Lehrenden haben. So darf auch das Wissen dieser und anderer Menschen als wichtiger Pfeiler dieses Frameworks nicht unterschätzt werden (vgl. [11, p. 84]). Rechtlich betrachtet ist die Kennzeichnung von Lernmaterial ohnehin an gewisse Vorgaben gebunden und muss dem jeweiligen Lehrplan entsprechen (vgl. [28]).

<sup>35</sup>spielendenzentriertes Design

<sup>36</sup>Konzept-Entwurf

<sup>37</sup>Spiel-Entwurf

<sup>38</sup>Spiel-Entwicklung

<sup>39</sup>Spieltests

**Integration of Play and Learning** In diesem Punkt geht das Framework auf die Spielenden ein und nennt hier die motivierende Qualität des Spiels, die mit dem Flow Gefühl erreicht werden soll. An diesen motivierenden Play-Aspekt muss möglichst nahe auch der Learning-Aspekt angegliedert sein, um das Wissen zu vermitteln.

### 3.1.6 Serious Game Design Assessment



Abbildung 9: Das Framework „SGDA“

Serious Game Design Assessment [26] ist ein Framework von Konstantin Mitgutsch und Narda Alvarado aus dem Jahr 2013. Dieses Framework ist eher für die Entwicklung von Serious Games, also ernstesten Spielen da, und spezialisiert sich in diesem Bereich mehr auf „Persuasive Games“<sup>40</sup> und nicht Lernspiele. Der Unterschied liegt hier darin, dass ernste Spiele einen Oberbegriff sind, der nicht nur klassische Lernspiele beschreibt, sondern auch solche Spiele, die eine Meinung oder einen Anstoß zur Debatte transportieren sollen (Persuasive Games)(vgl. [26, p. 2]). In diesem Framework werden die Beispiele „ICED“ [8] und „Sweatshop“ [22] genauestens analysiert und es wird versucht zu beschreiben, warum und wie sie funktionieren.

**Purpose** Was alle Serious Games vereint, und von den regulären Spielen unterscheidet, ist der Purpose<sup>41</sup>. Serious Games versuchen zu überzeugen oder zu unterrichten. Sie haben ein Ziel, was sie bei den Spielenden erreichen wollen. Dieses Ziel unterscheidet sich zwar immer ein wenig von dem, was sie bewirken. Die Schnittmenge jedoch wird als der Zweck des Spiels bezeichnet(vgl. [26, p. 3]). Im Falle des von mir entwickelten Spiels ist das Ziel das perfekte Unterrichten der gewählten Thematiken aus der Mathematik. Je nachdem, wie gut verschiedene Methoden die Thematiken unterrichten, wird natürlich der Zweck ein größerer oder weniger großer Teil des Ziels sein.

**Content and Information** Mit Content und Information bezeichnet dieses Framework jegliche Information, die das Spiel darstellt. Zusammenfassend ist alles, was die Spielenden wahrnehmen sollen, Content und Information. Einige Inhalte sind dem Informationszweck entsprechend, z.b. bei „Sweatshop“ die Gespräche und Zitate aus der realen Welt, die Problematiken von Bränden, die billigen Löhne, die Grafiken usw.. Andere Inhalte dienen lediglich zur Unterhaltung oder um die Geschichte plausibel erscheinen zu lassen, beispielsweise die Design-Entscheidung, die bei Sweatshop ein eher fröhliches, buntes Design darstellt(vgl. [26, p. 4]).

<sup>40</sup>Überzeugung generierende Spiele

<sup>41</sup>Zweck

**Game Mechanics** Unter Game Mechanics wird zusammengefasst, wie die Spielenden auf das Spiel Einfluss nehmen. Hier werden die Handlungen als „Verben“ eingeführt, im Fall von „Sweatshop“ also beispielsweise einstellen, feuern, bilden usw. (vgl. [26, p. 4]). Hier unterscheidet sich das SGDA Framework von MDA oder MDE, bei denen Mechanics die Gesamtheit der von den Entwickelnden gegebenen Arbeiten darstellen. Dadurch, dass SGDA Design-zentriert und nicht Spielenden-zentriert ist, teilt es das Design deutlich stärker auf, als es das mit dem Purpose, also dem Zweck des Spiels, macht.

**Fiction and Narrative** Fiction und Narrative sind die Elemente des Spiels, die nicht aus der realen Welt kommen, und die Geschichte des Spiels definieren (vgl. [26, p. 5]). Sie formen die virtuelle Welt und die Charaktere in ihr. Hier beschreibt es quasi die gleichen Punkte wie das DPE Framework es im Storytelling Bereich des Designs zusammenfasst. Im Falle von „Sweatshop“ wie auch in meinem Falle sind alle Charaktere, Weltelemente und die Geschichte an sich natürlich frei erfunden, viele Spiele müssen dies speziell angeben, da sonst bestimmte Firmen oder Personen sie verklagen könnten. Im Falle von „Sweatshop“ kommen hier Firmennamen vor, die Ähnlichkeit zu realen Firmen haben. In solchen Fällen sollte die Fiktion der Charaktere und Handlungen zur rechtlichen Sicherheit noch einmal betont werden.

**Aesthetics and Graphics** Sobald die Geschichte und die Formen der Interaktion definiert sind, ist es wichtig, eine dem Spiel entsprechende Grafik zu wählen. Um die Thematik von Kinderarbeit beispielsweise zu bearbeiten, ist in diesem Punkt die Entscheidung bei „Sweatshop“ auf eine sehr kindliche, sehr bunte Grafik gefallen (vgl. [26, p. 6]), die das ernste Thema kontrastreich sarkastisch erscheinen lässt. Bei einem Educative Game ist es meiner Meinung nach essenziell, dass die Grafik nicht zu rechenaufwendig wird, da Bildung jeder Person unabhängig der Finanzstärke zuzustehen hat, und eine realistische Grafik schnell teure Grafik-Hardware fordert. Daher setzt mein Spiel ebenfalls auf eine relativ simple Grafik, in der Hoffnung, auch auf älteren Geräten noch zu funktionieren.

**Framing** In diesem Framework wird das Framing<sup>42</sup> erst hier eingeführt (vgl. [26, p. 6]). Daher scheint hier die Relevanz der Zielgruppe niedrig gehalten, denn erst in diesem Punkt wird die Zielgruppe angesprochen. Das Framing beinhaltet auch mehr das Herausfinden der Fähigkeiten und Schwächen der Zielgruppe. Im Falle meines Spiels sind die Fähigkeiten im Lehrplan des Landes NRW definiert. Es gibt allerdings auch oft die Fälle, dass einige Elemente des Lehrplans nicht zu den Fähigkeiten der Spielenden gehören, sei es aus Lehrperson- oder Schüler\*innen- Versagen. Diese Schwächen sorgen oftmals dafür, dass Schüler\*innen dem Unterricht nicht mehr folgen können und nicht motiviert sind. Meine langjährige Erfahrung als Nachhilfelehrer bestätigt dies. Daher ist es essenziell wichtig, auch diese Schwächen aufzudecken und zu beheben, um irgendwie im Flow-Channel zu bleiben.

**Coherence and Cohesiveness of the Game-system** In dem Punkt „Coherence and cohesiveness of the game system“<sup>43</sup> wird darauf eingegangen, dass das Spiel an sich abgestimmt sein muss. Die einzelnen Elemente, Inhalt, Mechaniken, Geschichte usw. könnten zwar relativ unabhängig voneinander entwickelt werden. Es ist jedoch extrem wichtig, dass

---

<sup>42</sup>das Zuschneiden der Inhalte auf die Zielgruppe

<sup>43</sup>Stimmigkeit und Geschlossenheit des Spielbaren Systems

die Geschichte nicht eine ist, die der Thematik beispielsweise widerspricht. Es wird in diesem Punkt das Zusammenführen zu einem „ganzheitlichen“ Design gefordert(vgl. [26, p. 6,7]). Für mein Spiel würde das bedeuten, dass die Aufgaben in einem für sie passenden Kontext stattfinden sollen, es also nicht dazu kommt, dass ohne Kontext beispielsweise Popup-Fenster erscheinen die die Spielenden auffordern eine Aufgabe zu lösen. Dadurch, dass ich die einzige Person bin, die das Spiel entwickelt, ist es für mich relativ einfach, die verschiedenen Elemente des Spiels abzustimmen.

### 3.1.7 Vergleich

Die Frameworks „The I’s have it“, „Design, Play, and Experience“, „Mechanics, Dynamics and Aesthetics“, „Mechanics, Dynamics and Emotions“ sowie „P-III“ stehen in enger Verbindung zueinander, wenn auch „P-III“ und „The I’s have it“ nicht im Kontext der anderen entwickelt wurden. Trotzdem zeigen sich inhaltliche Ähnlichkeiten, die dazu führen, dass diese Frameworks in gewisser Weise miteinander in Zusammenhang gebracht werden können.

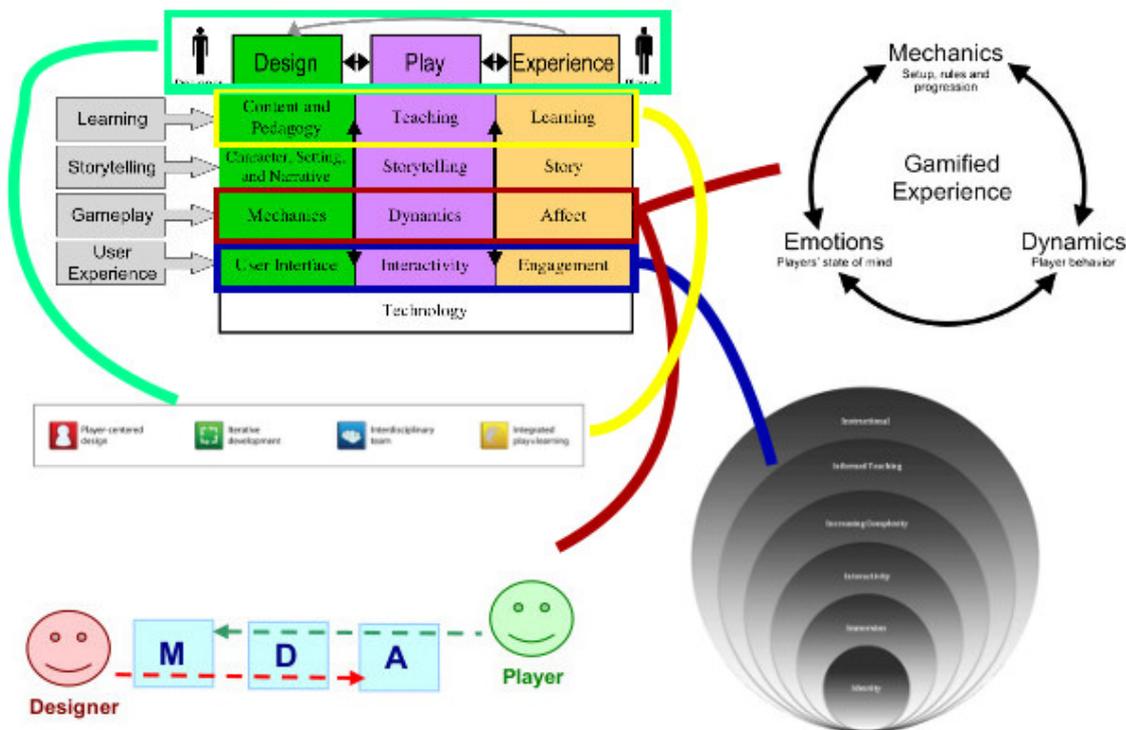


Abbildung 10: Der Zusammenhang der Frameworks skizziert

So ließe sich in „DPE“[41] als Gameplay-Ebene sowohl „MDA“[31] als auch „MDE“[16] einfügen, was dem geschichtlichen Verlauf dieser Frameworks geschuldet ist. In diesem Aspekt erachte ich es als sinnvoll, sich von beiden Frameworks inspirieren zu lassen, und die Wichtigkeit von Integrated Play and Learning[11] gerecht zu werden, wie sie „P-III“[11] fordert. Einerseits ist es wichtig, die Ästhetik[31] des Spiels nicht aus den Augen zu verlieren, damit das Spiel als Spiel angenommen und gerne gespielt wird. Andererseits ist es aber auch wichtig, die Emotion[16] der Spielenden nicht zu vergessen und sie in den Zustand von Flow[10] zu bringen, und sie dort zu halten solange sie lernen. „The I’s have it“[5] splittet die User-Experience Ebene noch einmal genauer auf und geht

ebenfalls auf den Fokus ein, den die Designenden auf die Spielenden haben sollen. Die Ebenen Identity und Immersion betonen den Aspekt des Engagements, Instructional betont Aspekte des User Interfaces, und Increasing Complexity sowie Informed Teaching stellen gemeinsam mit dem gleichnamigen Begriff Interactivity wichtige Aspekte der Interactivity dar.

„P-III“[11] adressiert verschiedene Aspekte von „DPE“, gibt der Ebene der Designenden die iterative Entwicklung als Zeitplan, stellt für das Teaching die interdisziplinäre Zusammenarbeit auf, und betont die Wichtigkeit der zusammenarbeitenden Aspekte Play und Learning. Darüber hinaus existiert mit dem Player-centered Design der Pfeil im Modell von „DPE“ von Experience nach Design, der ebenso durch das Iterative Development betont wird.

„SGDA“[26] hingegen hat als Serious Games Framework mit dem Fokus auf Persuasive Games auf der Experience-Seite nur wenige Überschneidungen. Es betont ganz besonders den Bereich Design und selten die Seite der Experience, da Persuasive Games nicht das Ziel haben, eine Information zu vermitteln, sondern von einer Meinung zu überzeugen. Dadurch, dass „SGDA“ mit dem Titel „Purposefull by Design“ schon die Wichtigkeit des Designs betont, stellt es kein Spielenden-zentriertes Framework dar.

Für dieses Spiel scheint es also sinnvoll, sich auf „The I’s have it“, „MDE“, „MDA“, „P-III“ und, in der Basis, „DPE“ zu beziehen. Dies stärkt im DPE-Bereich vor allem die Bereiche „Learning“, „Gameplay“ und „User-experience“, und lässt den „Storytelling“-Aspekt etwas nebensächlicher erscheinen. Genau dies ist von der Zielgruppe offenbar sogar gewollt. Schließlich werden von dieser wie schon vorher erwähnt, sehr häufig und gerne Spiele wie Minecraft, Call of Duty, World of Tanks, League of Legends und Counterstrike gespielt, die sehr wenig Wert auf eine Story im Sinne von Narrative legen, bzw. den Spielenden die Entwicklung einer solchen anhand von Characters und Context[41, S. 8] überlassen. Der Player-Centered Aspekt ist darüber hinaus in jedem dieser Frameworks zentral und fast jedes dieser Frameworks betont die Wichtigkeit eines Flow-Gefühls bei den Spielenden. Daher gliedert sich auch das Designkapitel („Built on solid Seas“) in viele der Ebenen und Ebenen-Elemente von „DPE“.

### 3.1.8 Erste Befragung der Schüler\*innen

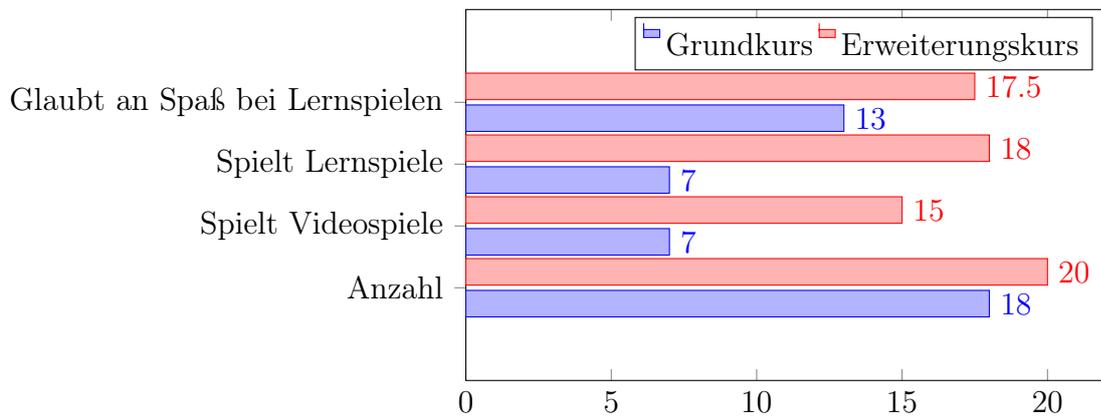
Um zu erfahren, welche Interessen die Schüler\*innen haben, und wie ein Spiel aussehen muss, damit es Playercentered[5]<sup>44</sup> ist, war es nun nötig, die Spielenden zu befragen. Hierzu fragte ich, ob sie Computerspiele spielen, ob sie Lernspiele spielen, und wenn ja, ob sie glauben dass ihnen Lernspiele Spaß machen können. Dann erfragte ich die beliebtesten Spiele und die am häufigsten gespielten Lernspiele mit ihrer Bewertung.

**Ergebnis** Da Computerspiele oftmals das logische Denken fördern, war es für mich kaum verwunderlich, dass im Erweiterungskurs der Mathematik mehr Menschen Computerspiele spielten als im Grundkurs (vgl. mit den Umfrage-Ergebnissen im Anhang).

---

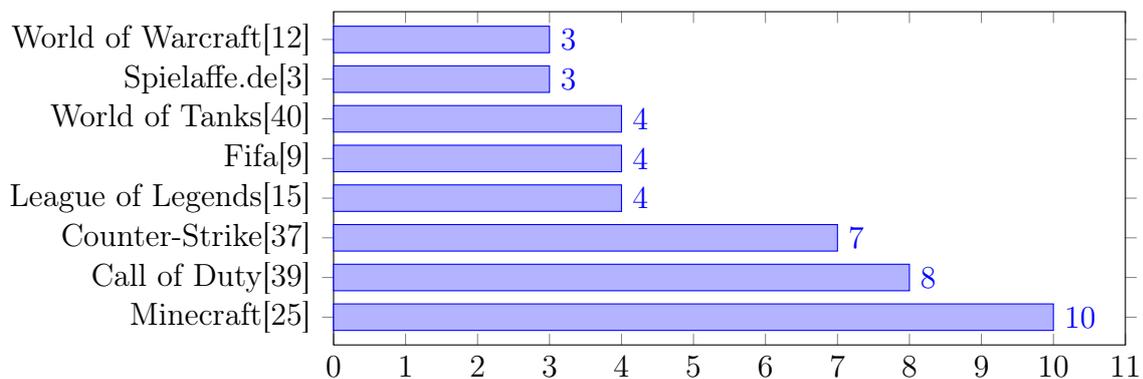
<sup>44</sup>spielenden-zentriert

Spielt Videospiele

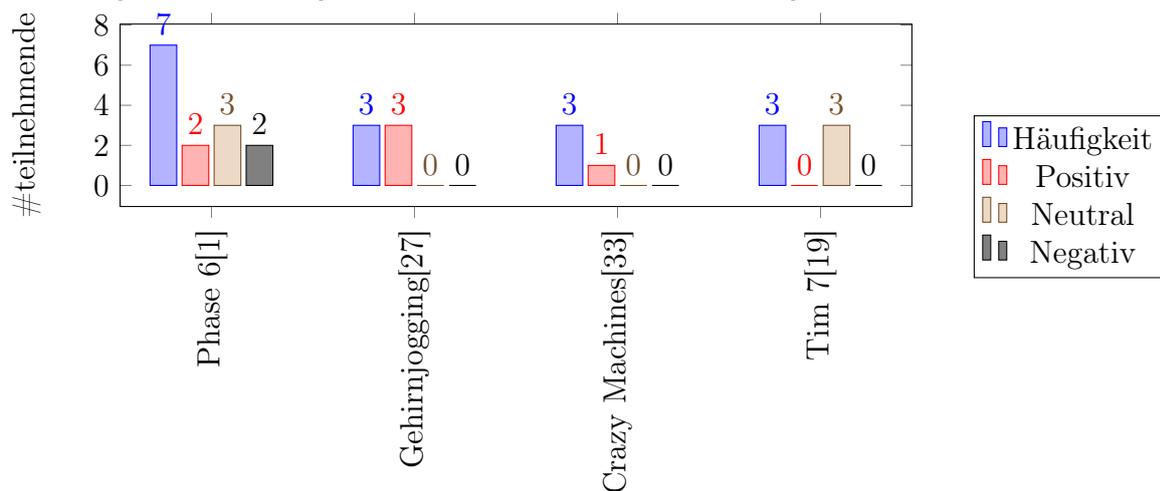


Dann stellte sich heraus, welche Spiele diese Menschen spielen, auch hier vermutete ich nicht viel anderes, besonders zu erkennen ist hier die große Anzahl so genannter „MO-BA“<sup>45</sup>- Spiele.

Spielt Folgende Spiele:



Und zu guter Letzt zeigte sich, welche Lernspiele sie wie gerne spielten



Bei diesen Umfragen kam es hin und wieder dazu, dass Befragte ihre Bögen nicht vollständig ausfüllten. Daher kommt es hier zu Inkonsistenzen.

<sup>45</sup>Multiplayer online battle arena - Spiele die zum Großteil daraus bestehen, dass Spielende im Internet gegeneinander antreten

## 3.2 Vergleich von herkömmlichen Lernspielen

In diesem Kapitel möchte ich auf verschiedene Lernspiele eingehen. Hier möchte ich überprüfen, ob die Firma hinter dem jeweiligen Spiel mit Computerspielen oder Lernsoftware eher bewandert ist, welche Mechaniken[31, S. 3,4][16, S. 5,6] die Spiele den Spielenden der verschiedenen Spielendertypen[6] bieten, wie diese Spiele das Eintauchen[5] versuchen zu ermöglichen und wie das Interface<sup>46</sup> für das Erscheinungsdatum gebaut ist.

### 3.2.1 Tim 7

Tim 7[19] ist eines der Spiele, das die Schüler\*innen dieser Schule häufiger nannten und gespielt hatten. Mit dem Erscheinungsdatum von etwa 1996 durch Heureka Klett und Ubisoft (verschiedene Varianten für verschiedene Klassen unterscheiden sich hier) und der DOS basierten Programmierung von Windows bis Windows 98 stellte es, aufgrund von schwerwiegenden Kompatibilitätsproblemen, eine kleine Herausforderung dar, dies auf modernen PC's zu installieren, dennoch fand es bei den Spielenden die es offenbar gespielt hatten den wenigsten Anklang und war dementsprechend ein wichtiges Spiel zum Untersuchen. Die Version die ich untersuchte war „Tim 7 und die Erinnerungen eines Vergessenen“.

Dieses Spiel bietet den Spielenden einen festgelegten jungen männlichen Hauptcharakter und eine kleine Anzahl an Charakteren, die allesamt deutlich älter als die Zielgruppe sind. Die Kamera ist statisch und die Steuerung entspricht einem regulären Point & Click-Adventure<sup>47</sup>

Die Mechaniken beschränken sich relativ stark darauf, ohne viel Kontext die sozialen Probleme der Menschen anzugehen, und hin und wieder eine große Anzahl an Aufgaben zu erledigen. Der Fortschritt wird in Prozent angezeigt und es erscheint eine Animation wenn genügend Aufgaben erledigt sind. Dann gibt es noch die Möglichkeit außerhalb des Spiels Aufgaben zu erledigen.

Diese Mechaniken fördern anhand der Prozentzahlen, die man abarbeiten kann, die Achiever-Typen[6]<sup>48</sup>, und zum großen Teil, mit der Story und den Sozialen Problemen, die Socializer-Typen[6]<sup>49</sup>. Dennoch scheinen nur wenige Motivationsmechaniken um eine relativ feste Lernsoftware gebaut zu sein. Auch passt ein männlicher Charakter eher dazu, dass sich männlich sozialisierte Spielende mit ihm identifizieren sollen. Die vielen Socializermechaniken<sup>50</sup> motivieren jedoch eher weiblich sozialisierte Spielende. Beides sind Einschränkungen, die möglicherweise mitverantwortlich für die schlechtere Bewertung des Spiels durch die Zielgruppe ist. Auch sind mir keinerlei Mechaniken aufgefallen, die einen Explorer-Typen[6]<sup>51</sup> oder einen Killer-Typen[6]<sup>52</sup> motivieren könnten.

### 3.2.2 Addy

Addy[38] ist ein Rollenspiel, von Coctel Games unter Havas Interactive 2001 erschienen. Addy weist einige fortschrittlichen Technologien auf. Es gibt einen langen und intensiven

---

<sup>46</sup>Schnittstelle, hier Grafik und Steuerung

<sup>47</sup>Zeigen und Klicken

<sup>48</sup>Leistungs-Menschen

<sup>49</sup>sozialisierende Spielende

<sup>50</sup>Mechaniken für Socializer-Typen

<sup>51</sup>Forscher

<sup>52</sup>Mörder-Typen, Spieler die auf andere Charaktere agieren

Prozess, der den Charakter erstellt, so dass man sich theoretisch mit diesem Charakter identifizieren könnte. Man sieht jedoch nur den Kopf des Charakters auf einem Bild, ansonsten ist der eigens aufwändig erstellte Charakter im Spiel nicht zu sehen. Als andere Charaktere gibt es noch einen männlichen Alien, einen Hund, und einen Roboter. Das Spiel baut auf einem einfachen Prinzip auf. Es gibt jeweils eine Grundstruktur eines Raumschiffs und je nach Schulklasse gibt es die Möglichkeit, verschiedene andere Raumschiffe zu besuchen, in denen dann Aufgaben bereitgehalten werden. Dann gab es ein Onlineportal und freischaltbare Spiele, die man mit einer bestimmten Punktzahl freischalten kann. Zusammengefasst gibt es einiges an Mechaniken für Achiever-Typen (Points, Badges, Levels and Leaderboards<sup>53</sup>), einiges an Allgemeinwissen für Explorer-Typen zu erforschen, (was jedoch schnell ein Ende findet, wenn jeder Raum gesehen wurde gibt es hier kaum Wiederholungswert, auch ist dieser Teil des Spiels ohne Belohnungen für Achiever-Typen) und dann relativ wenig bis gar keine Mechaniken für Killer-Typen und Socializer-Typen, was bedeutet, dass sich ein purer Achiever-Typ wohlfühlen kann, die anderen Typen jedoch kaum, oder nur übersichtlich die für sie nötigen Mechaniken finden.

### 3.2.3 Crazy Machines 2

Crazy Machines[33] war ein Spiel, das bei der Zielgruppe beliebt war. Es ist ein Puzzlespiel, ehemals von Fakt-Software und Pepper-Games entwickelt, und wie dies schon vermuten lässt, ist es kein Lernspiel. Es wurde von den Schüler\*innen jedoch als solches oft wahrgenommen, da die Schüler\*innen hier experimentieren können und bestimmte Fähigkeiten wie das logische Denken oder eine grobe Idee von physikalischen Gesetzen in Experimenten gelernt werden können. Crazy Machines folgt jedoch keinem Lehrplan und hat in Teilen sogar unrealistische Physik, obwohl es mit dem wissenschaftlichen Design wahrscheinlich den Anschein erwecken soll, physikalische Kenntnisse zu vermitteln. Es handelt sich hier größtenteils um ein klassisches Achiever-Spiel mit Punkten, Auszeichnungen, Leveln und Ranglisten, fasst sich etwas kurz im sozialen Aspekt. In dem Spiel gibt es nur einen Charakter, der erwachsen ist, mit dem sich die Spielenden nicht identifizieren können. Hier wird jedoch stattdessen die vierte Wand<sup>54</sup> durchbrochen und die Spielenden werden direkt angesprochen, was sie stark in das Spielgeschehen integriert. Andernfalls gibt es noch ein paar Charaktere, über die gesprochen wird, die jedoch nie zu sehen sind. Für die Killer-Typen gibt es dann noch die Möglichkeit, Level zu erstellen und damit Freunde über das Internet herauszufordern. Crazy Machines spricht somit eher eine männlich sozialisierte Gruppe von Spielenden an, muss hier jedoch nicht alle ansprechen, da es kein Lernspiel ist.

### 3.2.4 S.O.S. Tiefsee

S.O.S. Tiefsee[36] ist eine wichtige Quelle von Inspiration für das Spielprojekt. Es handelt sich hierbei um ein Lernspiel von Terzio und Levande Böcker, das im Jahr 2001 erschienen ist. Dieses Lernspiel ist eins, das für alle Spielendotypen eine Menge an Mechaniken bietet. Für Achiever gibt es Punkte, Forschungsfortschritt, einen Finanzstand und den Fortschritt in der Story. Dann gibt es auch hier das Zerbrechen der vierten Wand, das die Spielenden direkt einbezieht und eine sich entwickelnde Anzahl an Charakteren, die nach und nach noch einmal Möglichkeiten geben, sich mit ihnen zu identifizieren, oder

---

<sup>53</sup>Punkte, Abzeichen, Level und Ranglisten

<sup>54</sup>„Die vierte Wand durchbrechen“ steht für „die Zuschauenden direkt ansprechen und involvieren“ und ist eine bekannte Technik in Theater, Film und Spiel

als Sozializer-Typ mit ihnen zu interagieren. Auch der einprogrammierte Protest, wenn man einen Charakter zu lange stehen lässt, durchbricht die vierte Wand immer häufiger und erhält somit die Aufmerksamkeit der Spielenden. Für die Explorer bietet es die Möglichkeit, viele neue Mechaniken zu finden und ständig neue Inhalte zu entdecken, und für die Killer-Typen die Möglichkeit, die Story nach den eigenen Regeln zu formen. Es ist damit ein Lernspiel, das es schafft, für alle Spielendotypen genügend Motivation bereit zu halten, definiert jedoch nicht genau die Einordnung in die Lehrpläne.

### 3.2.5 Phase 6

Phase 6[1] ist eine Lernsoftware, von sämtlichen Schulbuchverlagen gemeinsam entwickelt, die das Lernen von Vokabeln erleichtern soll. Unter den Schüler\*innen erhielt es mal gute, mal mittelmäßige und sogar schlechte Bewertungen. Es ist im Abo monetarisiert und stellt daher einen größeren Kostenfaktor für die Langzeitnutzung dar. Es bietet an Mechaniken das ständige, spezialisierte Wiederholen von Vokabeln mit Bewertungsmechanismen. Damit motiviert es stark die Achiever-Typen. Für Socializer-, Explorer- und Killer- Typen bietet es keinerlei Mechaniken. Es versucht nicht, sich als ein Spiel darzustellen, sondern viel eher als ein Hilfsmittel, als das es auch mal mit mehr, mal mit weniger Erfolg wahrgenommen wird.

### 3.2.6 Zusammenfassung

Von einigen Schüler\*innen wurden als gute Lernspiele Quizspiele und Puzzlespiele wie Crazy Machines und Gehirnjogging genannt und stark als positiv bewertet. Diese Spiele folgen keinem Lehrplan, könnten jedoch Kenntnisse verschiedener Fächer vermitteln. Lernspiele die unter anderem von Firmen erstellt wurden, die sich auf Lernmaterialien spezialisieren, tendieren oft dazu, wenig zu motivieren (Tim 7 / Addy), nur einige Typen zu motivieren (Phase 6), oder den Lehrplänen nicht so genau zu folgen. Ein Punkt, warum es heutzutage kaum mehr Lernspiele gibt, scheint dieser zu sein. Lernspiele schaffen es derzeit nicht, alle Schüler\*innen zu motivieren, da ihnen hin und wieder Mechaniken oder Identifikationsfiguren fehlen, und die Monetarisierung eines einfachen Kaufes dazu verleitet, dass die Lernenden die Spiele vorwiegend gebraucht kaufen. Monetarisierung in Abo-Systemen schreckt jedoch schnell ab. Spiele, die so viel Motivieren, dass sie einen Wiederspiel-Wert haben, scheinen weniger dem Lehrplan zu folgen, da sie auch Abwechslung bieten. Diese werden dann eher gekauft und behalten, was für die Lernspiele nach Lehrplan eine massive Schwierigkeit in einem kapitalistischen System wie heutzutage in Deutschland darstellt, eine solche Software produzieren zu lassen, ohne große Verluste zu machen.

## 3.3 Themen als Questpfade (Learning)

In diesem Abschnitt setze ich mich explizit mit dem Bereich Learning (siehe DPE) auseinander. Hier kommt es auf interdisziplinäre Zusammenarbeit zur Integration von Spiel (im Sinne von Play<sup>55</sup>) und Learning (siehe P-III). Damit versuche ich über die Bereiche von Inhalt und pädagogischen Methoden die Bereiche Lehren und Lernen abzudecken. Wie schon in MDA erwähnt, gibt es hier die Sicht der Spielenden, die lernen, gegenübergestellt

---

<sup>55</sup>Play steht für Spielen für den Spaß, Game dagegen auch für Herausforderung und Ernsthaftigkeit im Spiel

zur Sicht der Entwickelnden, die Inhalt und pädagogische Methoden zur Verfügung stellen. Wie in dem Kontext bereits erwähnt, stellt die Thematik hier den Bereich des Inhalts schon sehr präzise. Um als Lernspiel in einem Bundesland zu gelten, muss ein Spiel dessen Lehrplan entsprechen. Diese sind von Land zu Land unterschiedlich. Im Bereich Geometrie stellt das Land NRW folgenden Lehrplan zur Verfügung (die Kursiv geschriebenen Textteile sind nur im Erweiterungskurs zu finden):

### 3.3.1 Lehrplan NRW

Der Lehrplan NRW sieht viele Themen für die Schüler\*innen der Klasse 9 und 10 vor. In dem aktuellen Lehrplan finden sich dort zum Thema Geometrie die Fähigkeiten,

- Körper (Zylinder, Pyramiden, Kegel, Kugeln) benennen, charakterisieren und in ihrer Umwelt zu identifizieren
- Schrägbilder zu skizzieren, Netze von Zylindern, Pyramiden und Kegeln zu entwerfen, und die Körper herzustellen
- einfache Figuren maßstabsgetreu zu verkleinern und zu vergrößern
- Umfänge und Flächeninhalte von Kreisen und zusammengesetzten Flächen sowie Oberflächen und Volumina von Zylindern, Pyramiden, Kegeln und Kugeln zu schätzen und zu bestimmen
- geometrische Größen zu berechnen und dazu den Satz des Pythagoras zu verwenden, Ähnlichkeitsbeziehungen und die Definition von Sinus, Kosinus und Tangens zu erkennen und Eigenschaften von Figuren mit Hilfe des Satzes von Thales zu begründen

[20] Im Rahmen dieser Bachelorarbeit behandle ich lediglich diesen Geometrie-Abschnitt. Es ist jedoch wichtig anzuerkennen, dass in diesen Jahren noch einiger anderer Stoff gefordert wird.

Um Körper zu identifizieren dienen in Büchern oft Aufgaben mit Bildern. Skizzen und Zeichnungen von Objekten werden im Unterricht oft mit Stift und Papier hergestellt, aber auch Geometrie-Software wie das Opensource Projekt Geogebra[4] kommt hier zum Einsatz.

Das Vergrößern und Verkleinern von Objekten wird mit dem Strahlensatz überprüft. Die Berechnung von Oberflächen und Volumina erfolgt über einfache Formeln, der Satz des Pythagoras zeigt sich oft in unterschiedlichsten Variationen, und Sinus, Cosinus sowie Tangens werden meistens über Triangulierung definiert.

### 3.3.2 Umsetzung der Themen in Büchern

Für die Umsetzung der Themen kommen für mich nur Schulbücher in Frage, und diese sind nach Gesetz geregelt und einheitlich gehalten. Im Bereich der Mathematik in der Klasse 9 und 10 stehen hier Bücher der Verlage Klett, Cornelsen, Westermann und Schroedel zur Auswahl. Jedes andere Buch müsste eine extra Bestätigung haben, daher werden in den Schulen in den meisten Fällen die Bücher dieser Verlage verwendet. Die Euregio Gesamtschule Rheine, mit der dieses Spiel in Zusammenarbeit entstand, nutzt die Bücher des Cornelsen-Verlags, in denen die Erweiterungskurs-Bücher die exakten Aufgaben der Grundkurs-Bücher enthalten und einige Aufgaben mehr bieten. Da in diesen Büchern die

gesamte Thematik der Geometrie in der zweiten Hälfte des Buches „Zahlen und Größen 9“ [42] die bearbeitet wird, und dies auch in der Regel dem zweiten Halbjahr der Klasse 9 entspricht, ist dieses Buch meine Hauptquelle im Bezug auf die Umsetzung in Büchern. Zum Geometrieteil des Buchs wird erst einmal eingeleitet, indem die bereits bekannten Formen (Rechtecke, Quadrate, Dreiecke, Kreise und Würfel) gezeichnet werden sollen. Dann sollen Maßstäbe anhand von Karten erklärt werden. Darauf folgen noch Flächen- und Umfangs-Berechnungen von Rechtecken, Parallelogrammen, und Dreiecken. Die Begriffe Radius und Durchmesser sowie das Distributivgesetz werden wiederholt.

**Satz des Pythagoras** Für den Satz des Pythagoras wird wiederholt, wie man Seiten und Winkel misst, wie man verschiedene Einheiten addiert, welche Dreiecke jeweils gezeichnet werden und wie man sie konstruiert. Dann wird das Distributivgesetz wiederholt, lineare Gleichungen werden gelöst und Wurzeln werden gezogen, anhand von Quadraten mit Flächengröße und der Suche nach der Kantenlänge. Die Klammersetze, Prozentrechnung und Flächenarten werden ebenfalls wiederholt. Dann werden Quadratzahlen und Wurzeln anhand von Flächeninhalt und Kantenlänge eines Quadrats genauer beleuchtet. Einfache Wurzeln werden gezogen, über 8 Seiten werden dann durch Kacheln, Fliesen und Mosaiksteinchen Quadratwurzeln erklärt. Es folgt dann eine Seite mit verschiedenen Aufgaben zu Intervallschachtelungen anhand von Tabellenkalkulation, und eine mit direkten und indirekten Beweisen. Abschließend wird der Satz des Pythagoras anhand eines Modells des Potsdamer Exploratorium erklärt, in dem Sand aus dem Hypotenusenquadrat die Kathetenquadrate auffüllt. Hierzu gibt es allerdings auch andere Animationen, die das Ganze einfacher zeigen. Nun werden über Leitern, Warnhütchen, Winkelwerkzeug, umgeknickten Bäumen, Dachschrägen, Löffeln in Tassen, Seilbahnstrecken, Stiften in kubischen Verpackungen, Blickweiten, Straßenpläne und Schilfrohren die verschiedenen Längen der Dreiecke berechnet. Auf der Folgeseite wird der Satz des Thales definiert (hier könnte man eventuell ein Halbkreis mit einem rechteckigen Dreieck zeichnen). Eine Seite später kommt dann der Kathetensatz und der Höhensatz von Euklid. Kurz darauf wird angefangen zu wiederholen. Quadratzahlen und Wurzeln, Leitern, Pythagoras, Schussbahnlängen über Diagonalen, Seillängen an einem abgetriebenen Ballon, Grabentiefen und Segelboote dienen dazu, den Satz des Pythagoras immer häufiger anzuwenden.

**Kreisberechnung** Für die Kreisberechnung werden Dreiecke, Drehsymmetrie und Achsensymmetrie wiederholt. Darüber hinaus werden Flächeninhalte, Innenwinkel, Flächeninhalt von Rechtecken, Radius im Vergleich zum Durchmesser und Kreisdiagramme angesprochen.

Die darauffolgenden n-Eck Flächenberechnungen weisen bis auf einige Kacheln und Verkehrsschilder und das Pentagon bzw. die Europaflagge keine Beispiele auf. Über Archimedes Annäherungsverfahren wird dann der Umfang eingeführt. Ein Beispiel mit Geschenkbund soll im Anschluss den Übergang von einer Strecke zu einem Kreis bieten. Ab diesem Punkt folgen viele Beispiele: Laufstrecken im Stadion, Baumumfänge, Äquatorlängen, Riesenräder, Gleislängen und Kuchenstücke, die das Berechnen von Umfängen erklären sollen. Interessant ist hier, dass auch Kreisabschnitte berechnet werden.

Im nächsten Punkt wird mit einem Blumenbeet die Flächenberechnung für Kreise und Kreisringe eingeführt. Im Anschluss werden über zwei Seiten Flächen über Beispiele wie Straßen, Gebäude, Blasendiagramme (Radius vs. Flächeninhalt), Pizza und geographischer Flächenschätzung sowie Bäume/Funkgebiete gezeigt. Dann kommt ein Part, in dem über Tabellenkalkulation die Annäherung von Pi besprochen wird. In den vermischten

Übungen über 4 Seiten werden Fußbälle, CDs, Torbögen, Kacheln und Dartscheiben benutzt, um Flächen zu berechnen, und Räder, Bögen in der Baukunst und Planeten werden benutzt um Umfänge zu berechnen.

**Zylinder** Für die Zylinderberechnung werden Einheitenumrechnungen, Schrägbilder, Volumen und Oberflächen von Prismen und Schnittvolumen wiederholt. Dann werden noch Einheiten für Volumen, Berechnungen von Prismenvolumen /-oberflächen, Kreisumfangberechnung und Kreisflächenberechnung angesprochen. Über 4 Seiten werden im Anschluss mit Paketen, Laternen, Dosen, Geldstücken, Stiften, Batterien, Geräteschuppen, und Litfaßsäulen Mantel- und Oberflächenberechnung von Zylindern geübt. Im Anschluss werden über Schrägbilder, und Versuche mit Flüssigkeiten, Rundhölzern und Kerzen die Volumenformeln eingeführt. Im Anschluss werden mit Getränken in zylinderförmigen Gläsern, Tunneln, Werkstücken, Regenrinnen und Getreidesilos einfache Volumenberechnung auf 4 Seiten ausführlich geprobt. Danach werden Hohlzylinder anhand von Stahlzylindern und Schornsteinen auf 2 Seiten gezeigt. Auf den nächsten 4 Wiederholungs und Testseiten werden mit Strohzyindern, Litfaßsäulen, Palisaden, Mühlsteinen, Fässern, Kerzen, Gläsern, Eishockeypucks, Werkstücken und Konservendosen die verschiedenen Berechnungen (Oberflächen, Volumen und Hohlzylindervolumen) wiederholt.

**Pyramiden, Kegel, Kugeln** Für diesen Abschnitt werden erst einmal nur Pyramiden und Kegel definiert. Bevor damit begonnen werden kann, werden allerdings wieder Einheiten, Prismen, Flächeninhalte und Strahlensatz wiederholt. Kurz werden noch der Satz des Pythagoras, und die Unterschiede zwischen Quadrat - Rechteck, Mantel - Oberfläche eines Zylinders und Radius - Durchmesser angesprochen.

Kurz darauf werden Eigenschaften von Pyramiden und Kegeln angesprochen, sie identifiziert und gezeichnet. Hierzu werden die Pyramiden von Gizeh und andere berühmte Bauwerke verwendet. Im Anschluss werden über 4 Seiten Mantel und Oberfläche einer Pyramide behandelt. Hierfür werden wieder bestimmte Gebäude und ein Briefbeschwerer verwendet. Die folgenden 4 Seiten befassen sich mit Mantel und Oberfläche eines Kegels. Hierfür werden wieder Gebäude, Silos und die kegelförmigen Becher von Wasserspendern als Beispiel genannt. Im Anschluss folgen 4 Seiten mit Volumenberechnung, über die Füllmenge mit Flüssigkeit definiert, und mit Beispielen wie Trinkpäckchen, Würfelstapel, Gebäuden, Werkstücken, und Salzhaufen. Der vorletzte Teil beschäftigt sich dann mit Kugeln, und unterteilt nicht mehr zwischen Oberfläche und Volumen. Hier wird direkt beides behandelt. Hierfür werden Sputnik, Bälle, Planeten, Gebäude, Eiskugeln, Lungenblasen und verschiedene Werkstücke als Beispiele eingeführt. Zuletzt werden noch auf 4 Seiten anhand von Gebäuden, Gläsern, Sputnik, Heißluftballons und Eiskugeln die verschiedenen Formen berechnet. Hier zeigt sich, dass das Erschaffen von dreidimensionalen Objekten oft schon Motivation genug ist.

### 3.3.3 Umsetzung der Themen in „Built on solid Seas“

Die Plattform eines Videospiele bietet ganz andere Möglichkeiten, als sie ein Buch bietet. So können die Spielenden in einem Videospiele mit Personen interagieren, ihre Reaktionen direkt erfahren, Objekte herstellen, und ihre Siege direkt auskosten, während ein Buch erst eine korrigierende Person braucht, und dadurch die Motivation schwierig aufrechtzuerhalten ist. In „Built on solid Seas“ werden für das technische Zeichnen der selbst-geschaffenen

Bauwerke auf den Bauplänen noch einige Werte ausgelassen, die die Spielenden nachtragen müssen, wie z.B. Querstangen der Segel. Zum Abschluss des Ganzen können die Spielenden dann endlich den Müll einsammeln und die Insel vergrößern. Hin und wieder zerbrechen natürlich Teile der Ausrüstung. Der Schrott kann dann nachher aus dem Kanal und aus dem Auffangbecken der Mühlen aufgesammelt werden. Für jede neue Mühle müssen weniger Werte nachgetragen werden.

Zuerst wird selbstverständlich auch wieder wiederholt. Im Spiel werden über Overlays Geodreiecke zur Verfügung gestellt. Diese sollen helfen, in technischen Zeichnungen Winkel zu messen, die je nach Bildschirm verzerrt sein könnten. Anschließend sollen verschiedene Berechnungen in das Thema „Satz des Pythagoras“ einführen. Zum Beispiel benötigt ein Charakter einige Längen um einen Tisch zu produzieren. Die Spielenden beginnen diese Aufgabe dann, berechnen sie und geben sie ab. Einige andere Aufgaben zeigen dann den Satz des Pythagoras genauer, auch hier kann man dann den  $90^\circ$  Winkel nachmessen. Hier könnten die Spielenden die Fähigkeit freischalten, in der Design-Sicht mehrere Winkel direkt angezeigt zu bekommen.

Verschiedene Aufgaben schalten dann unterschiedliche Spielelemente frei und können Belohnungen bereithalten.

Die generelle Struktur einer Aufgabe ist also die, dass die Person mit einem NPC einen Dialog hat, eine Aufgabe angeboten bekommt, diese akzeptiert, sie berechnet, und sie wieder abgibt, um die Belohnung einzustreichen.

**Satz des Pythagoras** Als Nächstes sollen für verschiedene Personen kleine Aufgaben erledigt werden, die sie überzeugen, bei der Konstruktion einer Wassermühle zum Aussortieren von Schrott aus dem Ozean zu helfen, womit man auch die ersten Plastikvorräte sammeln kann. Hier bieten sich Aufgaben an wie:

- „Wie weit muss ich eigentlich von dem Haus weg stehen, damit ich nicht von der Leiter falle?“
- „Wie sollen wir eigentlich die Solaranlagen auf die Dächer bringen?“
- „Wie viel Fläche haben wir eigentlich zur Verfügung?“
- „Wie lang muss ein Kabel über den Damm sein?“

Mithilfe einer Konstruktion anhand des Satz des Thales könnten dann zum Schluss noch effektivere Kanäle ausgegraben werden. Was natürlich Fragen aufwirft wie „Wenn ich eine gerade Mauer bauen möchte, wie weit müssen die Steine auf jeder Ebene mindestens reichen?“ (konstanter Abstand, Ebenen einzeln abfragen). Auch das nötige Tor für die Spielfläche soll hier konstruiert werden.

**Kreisberechnung** Auch die Kreisberechnung stellt sich relativ einfach dar. Hier lassen sich Rohre, Kanäle, Leuchttürme und Behälter konstruieren. Des weiteren steigern sich natürlich die Möglichkeiten, die Türme zu reparieren, und zu verbessern.

**Zylinder** Die Zylinderberechnung verschafft auch neue Upgrade-Möglichkeiten. Die Kreisberechnungen lassen sich nun wiederverwerten, um Zylinder zu berechnen, um zum Beispiel Kolben für Pumpen zu erstellen, Schiffe für den Kontakt mit „Der Firma“, Volumenmaximierung bei Wassertanks und einige Story-relevante Aufgaben, wie die Bedrohung durch die Plastikkontinente, und eine Legende über alles rettende Baupläne am Boden des Ozeans tauchen auf.

**Pyramiden, Kegel, Kugeln** Die Berechnung von Pyramiden, Kegeln und Kugeln dient größtenteils dazu, die Baupläne vom Grund des Ozeans zu bergen. Verschiedene Teile eines U-Boots sollen gebastelt werden und die Untersuchungen mit dem U-Boot sollen auslösen, dass man Pläne findet, die vorausberechnen, dass man möglichst schnell eine Plastik-Kuppel erschaffen sollte, um die Insel und ihre Bewohner zu retten. In diesem Moment zeigt sich „Die Firma“ auch besorgt, man könnte es überleben, und nur mithilfe der Eltern der Protestperson lässt sich ein Eingreifen der Firma verhindern, und die Kuppel wird fertig, bevor der Plastik-Kontinent die Insel überschwemmt

## 4 „Built on solid Seas“

Aufgrund meiner Schlüsse aus den vorherigen Kapiteln erklärt sich relativ grob, wie ein Computerspiel in diesem Rahmen aussehen muss. Für den Spielenden-zentrierten Aspekt in „Built on solid Seas“ ist es extrem wichtig, dass eine Charaktererstellung und somit ein Rollenspiel vorliegt, oder die Spielenden direkt (durch die „vierte Wand“) angesprochen werden. Ich entschied mich hier für Ersteres. Auch gegebene Umstände der schulpolitischen Situation stellen viele Anforderungen an ein Spiel. Je nach Bundesland, Schulform und verwendeten Unterrichtsmaterialien ist es unterschiedlich, welche Aufgaben gestellt werden sollen, und welche Kapitel wie aneinander gereiht werden sollten. Daher ist es von enormer Wichtigkeit, dass die Aufgaben und die Dialoge veränderbar sind. Darüber hinaus muss es Elemente haben, die alle Typen von Lernenden motivieren. Aus den am häufigsten gespielten Spielen geht hervor, dass die Welt nicht statisch sein soll, sondern wie bei Minecraft[25] die Möglichkeit existieren muss, die Welt zu bearbeiten und zu etwas Eigenem zu formen. Dann werden auch Herausforderungen gefordert, für deren Bewältigung Belohnungen existieren müssen. Ein Ego-Shooter<sup>56</sup> wie ihn viele Schüler\*innen in ihrer Freizeit gerne spielen, lässt sich nicht im Bereich der Lernspiele erstellen, da man durch die Jugendschutzmaßnahmen gehindert sein könnte, mit den Schulen zusammen zu arbeiten oder die Spiele an die Zielgruppe zu liefern.

Ein Spielprinzip, das in der Langzeitmotivation sehr erfolgreich ist, ist auch das Prinzip des Tower-Defense Spiels. Hier werden Türme gebaut, die angreifende Elemente unschädlich machen sollen, bevor sie die eigene Basis erreichen. Dieses Prinzip habe ich anstelle des Ego-Shooter Prinzips eingebaut. Auch bietet dieses Spielprinzip einige actionreiche Szenen, aber ebenso viele entspannte Szenen, in denen viel nachgedacht werden kann und wird. In letztere Szenen habe ich die Aufgaben eingebaut, da die Spielenden für das Rechnen eine ruhige Situation brauchen.

Auch die finanzielle Situation der Zielgruppe ist eine ganz Besondere. Es muss möglich sein, die Zielgruppe unabhängig von ihrer finanziellen Lage mit dem Spiel versorgen zu können, da es sonst passieren kann, dass Schulklassen nur teilweise, oder gerade daher gar nicht das Spiel spielen. Dieses hat dann wiederum Auswirkungen auf die Langzeitmotivation, da die Manipulierbarkeit der Welt einen sozialen Charakter der Motivation unterstützt, also hier das Prahlen mit -, und den Stolz auf die eigenen Erfolge. Das nächste Kapitel soll behandeln, wie die Spiel-Elemente das Spiel zu dem machen was es ist, ein Spiel, also nach der Definition von Amy Jo Kim (CEO von Shufflebrain) „a structured Experience with rules and goals that’s fun“ Hier befinden sich die Aspekte des DPE Frameworks, die nicht unter den Bereich Learning fallen, und mehr motivieren sollen, zu lernen.

<sup>56</sup>Schießspiel aus der Perspektive des Protagonisten

## 4.1 Storytelling

Der Aspekt des Storytellings kommt in vielen Spielen, die die Zielgruppe spielt, auf den ersten Blick zu kurz. Einfache kurze Geschichten oder keine fest implementierte Geschichte in Minecraft[25], Call of Duty[39], League of Legends[15], World of Tanks[40] usw. lassen vermuten, dass Storytelling für Videospiele mit dieser Zielgruppe irrelevant geworden ist. Dem ist jedoch nicht immer so, zumindest nicht vollständig. Zwar haben die meisten Spiele keinen konsequenten Handlungs-Strang, bieten jedoch viele Möglichkeiten, eine Story darzustellen. Wie Brian M. Winn in seinem DPE-Framework bereits anerkennt („all games have a player’s story, which at the very least reflects the story of the gameplay challenges encountered by the player and how the player addressed them“[41, S. 8]), gibt es in jedem Spiel eine Geschichte, zumindest die der behandelten Herausforderungen, die die Spielenden bewältigt haben. So kommt es in diesem Spiel ganz besonders darauf an, die Story nicht von der Seite der Designenden zu definieren, sondern, wie auch bei Minecraft[25], den Spielenden das entwickeln der Story eher selbst zu überlassen.

Der Punkt des Storytellings ist in vielen anderen Lernspielen relativ wenig Spielendenbasiert. Ob es eine fest einprogrammierte Story gibt, ist zwar für die Zielgruppe irrelevant, hier wird jedoch in einigen Lernspielen noch viel Wert darauf gelegt. Storytelling im Medium der Schulbücher findet sich kaum. Mir sind wenige Schulbücher bekannt, die Storytelling-Aspekte überhaupt aufnehmen.

### 4.1.1 Character Setting and narrative

In „Built on solid Seas“ möchte ich eine realistische Möglichkeit haben, wie sich die Begrenzung der Welt, dadurch dass sie nicht zufallsgeneriert ist, erklären lässt. Eine räumliche Begrenzung lässt sich nur in geschlossenen Räumen oder auf Inseln erklären. In meinem Fall entschied ich mich für eine Insel, und da Aufbauspiele ein großes Potential bei Spielenden haben (siehe Minecraft[25]), benötige ich noch eine Problemstellung und eine Quelle für Ressourcen, die ich beide in dem Plastik der Meere sehe. So kann ich auch, wie im SGDA-Framework genannt, einen weiteren Purpose<sup>57</sup> des Spiels hineinbringen, indem ich über eine Problematik berichte, die sich in den Ozeanen der realen Welt gerade in gigantischem Ausmaß ereignet.

Des weiteren benötige ich Charaktere die möglichst vielfältig sind, damit sich die Spielenden in eine Person hineinversetzen können. Eine Empfehlung aus der Interdisziplinären Zusammenarbeit mit der Frauenberatungsstelle „Frauen helfen Frauen“ in Siegen war, möglichst vielfältige Charaktere, Menschen mit unterschiedlichen Kleidungsstilen und Körperformen sowie unterschiedlichen Fähigkeiten und Beziehungen darzustellen.

Daher habe ich eine Auswahl von 12 Menschen erstellt, die zwar nicht jede Gesellschaftsgruppe repräsentieren oder repräsentieren können, aber dennoch eine möglichst vielfältige Auswahl schaffen sollen.

Auf der Insel leben neben dem Hauptcharakter noch eine protestierende Jugendliche namens Marja, deren Eltern in „Der Firma“ abseits der Insel arbeiten. Aufgrund ihrer Familiengeschichte ist sie ein wenig paranoid. Dann existiert noch eine Familie aus den Müttern Anja und Mina sowie dem Sohn Fritz. Anja betreibt die Werkstatt und Mina kümmert sich um die Anbindung der Insel an den Weltmarkt. Die nächste Familie besteht aus den Eltern Torben und Jessica und dem androgynen Kind Jay, Torben entspricht einem etwas fülligeren Mann, der in der Werkstatt bei Mina hilft. Jessica ist eine Lehrperson,

---

<sup>57</sup>Zweck

die mit Kopftuch in der Schule unterrichtet, und Jay ist ein Kind, das die gesellschaftlichen Strukturen der Insel versteht. Dann gibt es noch den Bürgermeister Maurice, der früher mit Mina verheiratet war, und durch den Druck von allen Seiten oft die Regierung der kleinen Insel nicht beherrscht und daran verzweifelt. Als letztes gibt es dann noch die Familie Meier, bestehend aus Mutter Angie und den Söhnen Maik und Lucia, Angie kümmert sich um ihre Familie und ist als alleinerziehende Mutter damit ausgelastet, da Lucia als körperlich durch Krankheit geschwächte Person einen sehr hohen Drang nach Sport entwickelt hat, und Maik als Designer auch über den Weltmarkt ein bisschen mehr Unruhe in die sonst so ruhige Insel bringt.

Die hier genannte „Firma“ soll als Großkonzern für Plastikverpackung dargestellt werden, und damit als Mitschuldige für die Verschmutzung der Meere. Hier wird keine real existierende Firma genannt. Eine rechtliche Absicherung mit einem entsprechenden Statement findet sich im Startbildschirm des Spiels.

#### 4.1.2 Storytelling

Die Geschichte, die sich in „Built on solid Seas“ darstellen lässt, besteht aus 4 Kapiteln, die der Thematik entsprechend aufgeteilt werden.

Anfänglich sollen die Spielenden vor die alltägliche Situation der Insel gestellt und von Marja mehr und mehr auf die Problematik der „Firma“ aufmerksam gemacht werden, um dann mit Marja die Idee zu entwickeln, die Insel zu übernehmen, und gegen das Verhalten der „Firma“ vorzugehen.

Im nächsten Kapitel sollen die Spielenden dann schon die Möglichkeiten haben diese Dinge zu tun, und sich der Bewältigung dieser Aufgabe zu widmen, indem sie Rohstoffe sammeln und ihre Insel ausbauen.

Das darauffolgende Kapitel soll mehr und mehr die Auseinandersetzung mit der „Firma“ zeigen, und die Spielenden dazu motivieren, ihre Insel größer auszubauen, und ihre Weltmarktstellung zu steigern. Im letzten Kapitel jedoch erscheint die Problematik des Überstehens der Katastrophe eines in die Strömung geratenen Plastik-Kontinents. Die Beschaffung eines Bauplans vom Grund des Meeres wird zur Priorität. Die Bewältigung dieses Bauprojekts einer gewaltigen Plastik-Kuppel über der Insel stellt dann den finalen Kampf mit der „Firma“ dar, und das Spiel endet in der geglückten Beendigung dieses Bauprojekts und der Überschwemmung der Kuppel mit Plastik.

#### 4.1.3 Story

Die Geschichte von „Built on solid Seas“ im Sinne von Story<sup>58</sup> soll eine der Selbstermächtigung und des Umweltschutzes sein. Wie diese sich jedoch im Einzelnen entwickelt ist letztendlich individuell, und kann nur grob gelenkt werden. Ich hoffe, dass ich sie anhand der Umstände klar genug eingegrenzt habe, und, was besonders im Punkt der Selbstdarstellung wichtig ist, dass ich genügend Freiheit gelassen habe, sodass die Spielenden sich ausleben und ihre Story entwickeln können.

## 4.2 Gameplay

Mit dem Bereich Gameplay wird die Art bezeichnet, wie Spielende auf das Spiel einwirken können. Von spielender Seite soll dies aus Affekt, Emotion, oder aus der Ästhetik

---

<sup>58</sup>Die Geschichte, die die Spielenden erleben

heraus funktionieren. Die Menschen sollen motiviert werden, das Spiel zu spielen, und diese Emotion durch die gegebenen Mechaniken ausleben können. Die Mechaniken, die Spiele bieten, um Affekt, Emotion und Ästhetik zu bewirken, finden sich in keinem Schulbuch, und stellen den zentralen Vorteil von Videospiele gegenüber Schulbüchern dar. Viele der Mechaniken eines Computerspiels werden allerdings in den weniger beliebten Lernspielen kaum beachtet. Hier liegt offenbar ein zentraler Punkt, in dem Lernspiele eindeutig der herkömmlichen Spielbranche hinterher hinken und sie als weniger spielswert erscheinen lassen.

In diesem Punkt zeige ich die Umsetzung der verschiedenen Aspekte der Frameworks MDA, MDE und der dritten Ebene des DPE Frameworks auf.

#### 4.2.1 Mechaniken

Die Mechaniken, die „Built on solid Seas“ bietet, sind die Möglichkeiten, Sachen zu sammeln und zu erschaffen. Über einen Spielteil im Stil eines klassischen Tower-Defense-Spiels werden Plastikressourcen gesammelt, die für selbiges Spiel, und den Aufbau der Insel genutzt werden sollen. Hierfür können die Spielenden Aufgaben von den anderen nicht-spielbaren Charakteren (NPCs) bekommen, für die sie teilweise Berechnungen anstellen müssen. Des weiteren soll es immer Möglichkeiten geben, Sachen nachzuschauen, Hilfestellungen zu bekommen, und auf Hilfsmittel verwiesen zu werden. (siehe „Instructional“ des Frameworks „The I’s have it“) Amy Jo Kim (CEO der Firma Shufflebrain) teilt den Großteil von Mechaniken in ihrem Auftritt zum Google Tech-Talk auf. Die wichtigsten Elemente einer Mechanik, die die Spielenden in ihren Bann ziehen sollen, sind nach ihrer Aussage

- Collecting - Das Sammeln von Gegenständen und Auszeichnungen
- Points - Das Sammeln von Punkten, die für Handlungen benötigt werden können, auch Levels fallen hier drunter
- Feedback - Eine direkte Antwort vom System, ob die Aufgabe richtig gelöst ist
- Exchange - Soziale Interaktion zwischen Spieler und System/Spieler
- Customization - Ausdruck der Person selbst in Manipulation der eigenen System-Elemente (hier Charakter-Eigenschaften)

(vgl. [18, min. 8:25 - 19:25]) Diese Elemente hoffe ich mit „Built on solid Seas“ ausreichend abgedeckt zu haben. Schließlich fehlen diese Lernspielen viel zu oft.

#### 4.2.2 Dynamiken

Beim Spielen von „Built on solid Seas“ sollen sich immer wieder bestimmte Handlungsabläufe entwickeln, die zur Gewohnheit werden, aber nicht monoton werden, sogenannte Dynamiken. Diese lassen die Spielenden in der Motivation versinken, und durch eingestreute Neuerungen ihre Erfolge ausleben, oder sich das Spiel einfacher machen. So können die Spielenden ihre Plastikressourcen für spezielle Fertigkeiten des Hauptcharakters (schnellere Laufgeschwindigkeit, automatische Winkelmessung) einsetzen, jedoch auch die Insel größer bauen und neue Türme bzw. Wände im Towerdefense-Spielfeld bauen. Hier können sie ihre eigenen Fähigkeiten einschätzen und den Flow-Chanel individuell ein wenig

anpassen, indem sie sich mehr oder weniger Zeit zwischen den Aufgaben nehmen. Im Laufe des Spiels werden, um die Spielenden nicht zu überfordern, nur Stück für Stück immer mehr Möglichkeiten für die Spielenden aufgezählt. Diese „Increasing Complexity“ findet sich auch in dem Framework „The I’s have it“

### 4.2.3 Affekt, Emotion, Ästhetik

Mit Affekt, Emotion oder Ästhetik werden verschiedene Aspekte beschrieben, wie das Spiel die Spielenden motivieren soll. Affekt zielt hier mehr auf das Erreichen der Aufmerksamkeit ab, Emotion soll schauen ob die Spielenden noch Spaß haben, Ästhetik unterteilt die Aspekte des Spiels in Teilaspekte, die auf unterschiedliche Weise überzeugen sollen.

Um den Aspekt des Affekts (siehe DPE) für sich zu gewinnen, muss ein Spiel Gründe liefern, warum spielende Personen es spielen wollen, statt anderes zu tun. Solche Gründe können Teil einer Ästhetik des Spiels sein, zielen jedoch auch oft auf ein Belohnungs-System ab, das durch die gewonnenen Rohstoffe, die neuen Fertigkeiten, die größer-werdende Insel sowie der intrinsischen Motivation, etwas für den Unterricht zu tun, gegeben sein soll.

Die Emotion (Siehe MDE) der Spielenden soll durchgehend ausreichend gefordert und motiviert sein. Durch das Erstellen einer eigenen Welt begleitet die Spielenden ein konstantes Reward-System<sup>59</sup>, das hoffentlich ausreichend ist, um die Spielenden im Flow-Gefühl zu halten. Wenn sie jedoch mehr Fragen stellen, soll das Spiel auch einige einfachere Aufgaben stellen, um mehr Übung in den Ablauf zu bringen. Gerade der Einstieg in das Spiel fällt vielen oft schwer. Mathematik stellt viele schnell vor die Situation, dass die Fähigkeiten minimal und die Herausforderungen maximal erscheinen. Würde sich dies mit Senken der Herausforderungen in den Flow-Channel bringen lassen, wäre dieses Problem einfach zu beheben. Jedoch führt dies oft zu einer Apathie. Es muss also hier anfangs viel motiviert werden. Viele kleine Erfolge müssen ein positives Feedback hervorbringen.

Diese Art, auf die Spielenden genau einzugehen, und die Herausforderungen an die Fertigkeiten anzupassen, findet sich auch im Ansatz von „Informed Teaching“ im Framework „The I’s have it“.

Die Ästhetik (Siehe MDA) soll im Falle von „Built on solid Seas“ besonders auf Challenge und Expression liegen. Challenge im Sinne von, dass das Spiel fordert, und Expression, dadurch dass die Insel nach den Wünschen der Spielenden geformt wird. Die Expression kommt aufgrund der zeitlichen Einschränkung etwas kurz. So ließe sich die neu erschaffene Insel vielerorts noch intensiver gestalten, um die eigene Insel individueller zu formen. Dennoch fehlt ohnehin eine Anbindung an einen Server und daher oftmals die soziale Interaktion, die nötig ist um daraus besonders stark Motivation zu ziehen. Ein wichtiger Punkt der in späteren Projekten Beachtung finden sollte, da nach Amy Jo Kims „social Engagement Verbs“ auch ein wichtiger Typ von Spielenden nicht adressiert wird, nämlich der der collaborating-Gamers<sup>60</sup>, die viel Interaktion mit anderen Spielenden benötigen.

## 4.3 User Experience

Mit User Experience wird bezeichnet, wie die Spielenden mit dem Spiel interagieren können. Hier greift ganz besonders das Framework „The I’s have it“ [5] ein, und definiert, wie die Spielenden auf das Spiel einwirken (Engagement) und mit ihm interagieren (Interactivity). Das User Interface beschreibt hier die Darstellung und Steuerung, die die

<sup>59</sup>Belohnungs-System

<sup>60</sup>zusammen arbeitende Spielende

Designenden entwickeln, damit die Spielenden mit dem Spiel interagieren können.

#### 4.3.1 User Interface

Die Spielenden sind in „Built on solid Seas“ grafisch immer darüber informiert, wie viele Punkte sie bereits gesammelt haben, und wie viel Plastik ihnen zur Verfügung steht. Das Plastik können sie entweder verwenden, um Ziele zu erreichen, indem sie die Insel vergrößern, oder um für die Plastiksammlung neue Geräte anzuschaffen.

Die Plastiksammlung funktioniert in etwa wie ein normales Tower-Defense Spiel. An einer Ecke eines Rechtecks fließt Wasser in das Spielfeld und in diesem Wasser fließen Objekte. Verschiedene Türme sammeln diese Objekte auf, andere pumpen das Wasser ab. Gelingt es einem, das Feld vor der Überschwemmung zu bewahren, übersteht man die Runde und kann das gewonnene Plastik wiederverwerten, die Türme und Pumpen aufwerten und reparieren. Für Reparaturen und Aufwertungen müssen die dafür nötigen Aufgaben abgeschlossen werden, und per Zufall fordern einige Reparaturen oder Aufwertungen noch zusätzliche Aufgaben.

In dem Spiel gibt es die Möglichkeit, sich im dreidimensionalen Raum als ein selbst erstellter Charakter frei zu bewegen, sodass die Spielenden besser in das Spiel eintauchen können. Dann gibt es die verschiedenen Charaktere in dem Spiel, die die Aufgaben geben, für deren Berechnung das Zeichenbrett des Hauptcharakters nötig ist. Dort werden verschiedene Längen und Winkel der Geometrie entweder angezeigt, als messbar markiert oder zur Berechnung gegeben. Wenn alle Berechnungen eingegeben sind, kann ein Klick auf „bestätigen“ im späteren Spielverlauf die richtigen Antworten ausblenden, lässt aber die falsch angegebenen Antworten weiter angezeigt. Diese Funktion lässt sich mit einer bestimmten „Fähigkeit“ aktivieren. Anfangs werden die Spielenden gezwungen sein, ihre Antworten bei dem zuständigen NPC abzugeben, und dann auf eine Bestätigung oder Ablehnung zu stoßen. Die Fähigkeiten sind Elemente, die man in der spiel-internen Schule für Spielpunkte erwerben kann. Diese beinhalten Boni wie, dass Winkel nicht mehr gemessen werden müssen, dass der Charakter schneller laufen kann oder dass bestimmte Antworten automatisch überprüft werden, aber auch die Möglichkeiten, bestimmte Gebäude zu bauen oder aufzubessern.

#### 4.3.2 Interactivity

Die Spielenden könnten unterschiedlichst mit Spielen interagieren. Bei Spielen dieser Art, die darauf aus sind, Lernmaterial zu sein, ist es jedoch essenziell, dass die Steuerung keine Kostenfaktoren einbringt. Auch wenn es möglich wäre, und auch Gewiss Vor- und Nachteile bezüglich Intuitivität bietet, ein Lernspiel auf Tablets, oder mit Gamepad bzw. Gesten- und Sprachsteuerung anzubieten, ist eine Steuerung mit Maus und Tastatur die einfachste und günstigste Möglichkeit. Die Steuerung soll an die Steuerung der von der Zielgruppe häufig gespielten Spiele erinnern, und bietet damit die klassischen Möglichkeiten, mit der Maus die Kamera zu steuern, sich zu bewegen („W“, „A“, „S“, „D“ und Pfeiltasten auf der Tastatur), sowie mit der Leertaste zu springen. Die Interaktion mit Objekten im Raum erfolgt über Linksklick, und das Inventar lässt sich mit gedrückter Strg-Taste mit dem Scrollrad manipulieren. All diese Tasten finden sich in den meisten Rollenspielen und scheinen sich seit dem ersten großen Spiel aus der Ego-Perspektive (Wolfenstein3D[34]) bis zu dem aktuell bei der Zielgruppe beliebtesten Computerspiel (Minecraft[25]) zu einem Standard der Computerspiele entwickelt zu haben, der von Spieler\*innen grundsätzlich

erwartet wird.

Hiermit behandle ich auch die Ebene „Interactivity“ von dem Framework „The I’s have it“

### 4.3.3 Engagement

Die Spielenden sollen durch die eigens erstellten Charaktere in dem dreidimensionalen Raum ein Gefühl der Identifizierung mit den Hauptcharakteren entwickeln. Dadurch, dass die Spielenden in der Ästhetik eine Expression, also der Selbstdarstellung, haben, ist es von enormer Wichtigkeit, hier die Identität in den Mittelpunkt zu stellen. Es muss also ein Spiel erstellt werden, das den Spielenden die Möglichkeit bietet, sich auch gerade in den eigenen Avatar hineinzusetzen. Hierfür benötigte ich einen Charakter-Erstellungs-Menü. Durch die Studien der Thematiken „Gender“ und dem von Judith Butler geprägten Begriff der „Gender Performativity“ kam ich hier zu dem Schluss, dass es nur ein Ausrichten nach dem Stand der Forschung ist, das Geschlecht der Charaktere nicht in das Spiel einfließen zu lassen. Butler sagt hier „Gender is not a fact, the various acts of Gender creates the idea of Gender, and without those Acts, there would be no Gender.“[29] Es ist also wissenschaftlich korrekter, das Ausleben des Geschlechts den Spielenden zu überlassen und ihnen keine binären (oder nicht-binären) Möglichkeiten zu bieten. So ist man auch als Spiele-entwickelnde Person auf der sicheren Seite, wenn bei Spielenden das Geschlecht im Sinne von Gender im Konflikt mit dem Körper steht, und man erzeugt keine Geschlechts-Stereotype. Auch muss gesonderter Wert darauf gelegt sein, dass Menschen ihre spezielle Hautfarbe frei wählen können, und Kleidung angeboten wird, die kulturelle Konflikte vermeidet. Mit dem so erstellten Abbild ihrer Identität sollen die Spielenden nun in die Welt eintauchen. Hierfür ist die dreidimensionale Welt von entscheidendem Vorteil, ebenso wie die Wahl der Perspektive. So können Spielende die Kamera drehen und mit dem Scrollrad der Maus rein-zoomen, bis die Perspektive zu einer Ego-Perspektive wechselt. In diesem Punkt zeigen sich auch die Punkte „Immersion“ und „Identity“ von dem Framework „The I’s have it“

## 4.4 Zusammenfassung

Spiele können Büchern gegenüber oft Vorteile bieten. Sie können motivieren, auf die Spielenden reagieren, und bieten eine Abstraktion der Erfolge, die für die Selbstdarstellung von großer Wichtigkeit ist.

- Um eine breite Zielgruppe in dem Kontext einer 9. oder 10. Klasse zu erreichen, bedarf es einer möglichst frei definierten Story, die den Spielenden viel Freiraum lässt, sich auszuleben.
- Um die Zielgruppe zu motivieren muss das Gameplay stimmen. Ziel ist es hier, im Flow-Channel zu bleiben, und immer wieder zu motivieren (vgl. [41, p. 10]).
  - um im Flow-Channel zu bleiben müssen Mechaniken oder künstliche Intelligenzen greifen, die die Schwierigkeit nach Bedarf erhöhen oder senken.
  - um die Spielenden zu motivieren, müssen intrinsische und extrinsische Motivationen ausgereizt werden, der Fokus sollte jedoch auf den Intrinsischen liegen.
    - \* Intrinsische Motivation wie Lernen, Macht, Autonomie und Spaß sind Faktoren, die von innen heraus die spielenden motivieren (vgl. [11, p. 84]).

- \* Extrinsische Motivation wie Punkte, Levels, Aufgaben und Preise motivieren die Spielenden von außen, wenn die Spielenden nicht intrinsisch motiviert sind (vgl. [16, p. 4]).
- Auch das Kennen der Zielgruppe ist essenziell, möchte die Zielgruppe eher gegeneinander oder miteinander spielen, möchten sie eher sich ausdrücken oder forschen? Nur so werden die Motivationsziele „Challenge“, „Fellowship“, „Discovery“ usw. vorbestimmt (vgl. [31, p. 2]).
- Je nach Wahl der Arten der Spiele stellt sich die Frage: „Motiviere ich oft genug?“ „Gibt es schnelles Feedback?“ und „Sind die Ziele klar im Blick der Spielenden?“ (vgl. [26, p. 2])
- Essenziell ist dann nur noch, ob sich die ganze Zielgruppe in die Charaktere hineinversetzen kann, um in das Spiel einzutauchen, und es, wenn das User Interface funktioniert, gelingt, dass jede Person, egal wie und wo geboren, sich einen Charakter erstellen kann, der ihr entspricht (vgl. [5, p. 107]).

Werden alle diese Aspekte beachtet, dürfte dem Erfolg der Anwendung nichts mehr im Weg stehen. Wie dies in „Built on solid Seas“ in der Praxis umgesetzt wird, behandle ich im nächsten Kapitel.

## 5 Entwicklung von „Built on solid Seas“

Da diese Anforderungen relativ groß sind, stand auch fest, dass hierfür eine Game-engine<sup>61</sup> nötig war. Hier bot sich direkt eine erfahrene open-source<sup>62</sup> Engine an. Godot[21] ist eine solche Engine, die für dieses Projekt aufgrund des open-source Aspekts und der Erfahrung im Erschaffen professioneller Titel die Voraussetzungen erfüllte. Godot besitzt eine eigene Programmiersprache (GDScript), die etwa wie Python[32] funktioniert, basiert intern auf C++[35], ist Plattform-unabhängig und bietet sowohl in 2D als auch in 3D viele Möglichkeiten, ohne viel Programmierung viele herkömmliche Spielelemente zusammen zu stellen. Auch die Performance der exportierten Spiele ist relativ stark, und mit weniger als 34 megabyte Größe als gesamtes Programm ist Godot eine sehr leistungsstarke, aber auch ressourcenschonende Alternative zu herkömmlichen Engines, wie Unity3D oder der Cry-Engine. Das Spiel entstand aufgrund der aktuell laufenden Entwicklungen an Godot in vielen verschiedenen Versionen des Programms, begonnen bei dem letzten Alphatest der Version 2.0 über verschiedene Betatest-Versionen bis hin zur Version 2.0.2 Diese Art, ein Spiel zu programmieren und zwischendurch die Engine mehrmals zu wechseln, weil eine aktuellere Version erschienen ist, bringt sicherlich Risiken für die Kompatibilität und für Fehler mit sich, wurde in diesem Fall allerdings nicht zu einem Problem. Im Folgenden gehe ich auf die Szenenstrukturen, die verschiedenen globalen Klassen und einzelne Spielelemente genauer ein.

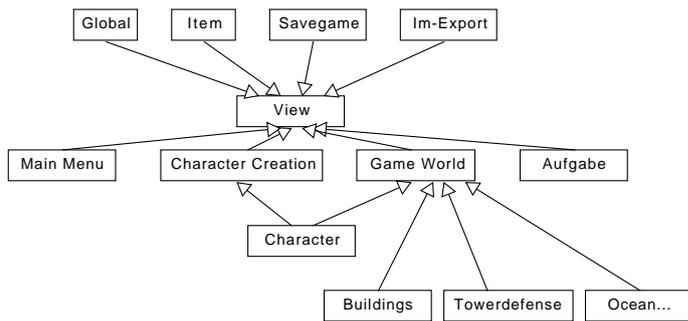


Abbildung 11: Die Struktur der Klassen in „Built on solid Seas“

## 5.1 Szenenstrukturen

Das Spiel teilt sich auf in 4 verschiedene Szenen, die jeweils aus kleineren Szenen bestehen können. Hier gibt es:

- das Hauptmenü: Diese Szene sehen die Spielenden als erstes.
- die Charaktererstellung: Diese wird beim neuen Spielstart einmal gestartet.
- die Spielwelt: Hier läuft der Großteil des Spiels ab.
- der Aufgabenbildschirm: Hier sollen die Spielenden ihre Aufgaben lösen können.

Diese vier Szenen werden jeweils geladen sobald sie benötigt werden. Dafür steht noch eine weitere Szene bereit, ebenso wie es eine Szene gibt, die für das Erstellen von Dialogen genutzt werden kann. Für den Teil des Programms, den die Spielenden sehen, existieren jedoch nur diese vier Szenen.

### 5.1.1 Hauptmenü

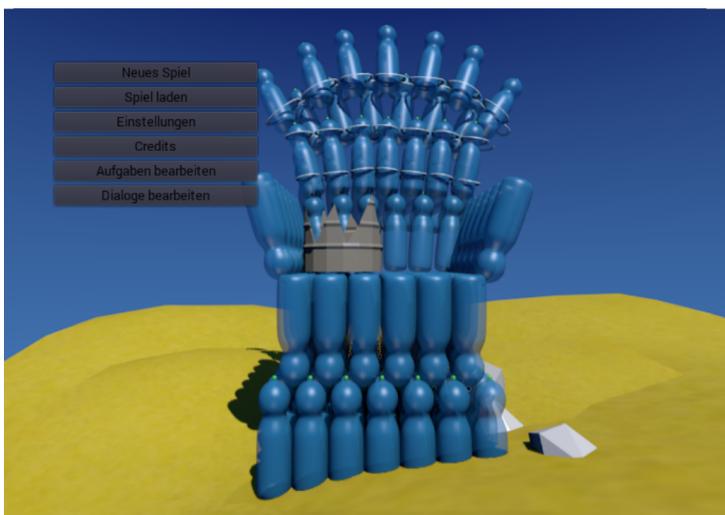


Abbildung 12: Das Hauptmenü von „Built on solid Seas“

<sup>61</sup>ein Programm, das in der Regel zum einfacheren entwickeln von bzw. immer zum Starten von Spielen, die für die entsprechende Engine entwickelt wurden, genutzt wird.

<sup>62</sup>Quelloffen

Im Hauptmenü können die Spielenden die Spielinformationen lesen (wer das Spiel erstellt hat, in welchem Kontext etc.), Einstellungen an der Darstellung vornehmen (Auflösung, Post-Processing und Schattenberechnung), ein Spiel laden oder ein neues Spiel starten. In der Version, die die Lehrpersonen spielen können, gibt es hier zusätzlich auch die Möglichkeiten, Dialoge und Aufgaben zu bearbeiten. Hier befinden sich ausschließlich 2D-Elemente.

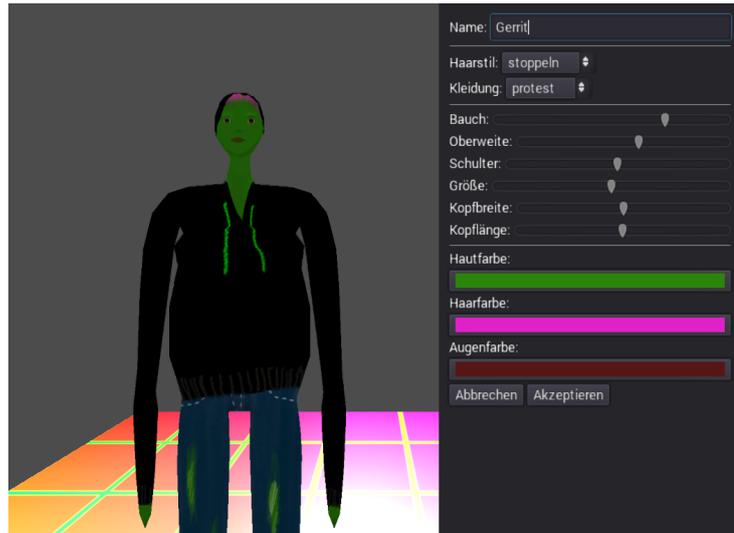


Abbildung 13: Die Charakter-Erstellung in „Built on solid Seas“

### 5.1.2 Charaktererstellung

In der Charaktererstellung können die Spielenden das Aussehen ihres Charakters ändern, einen Namen einstellen, die Kleidung ändern und den Charakter betrachten. Anders als in klassischen Rollenspielen wird aufgrund des Standes der Forschung im Bereich der Geschlechter (Queertheory und Genderforschung) darauf verzichtet, eine Einstellung für das Geschlecht der Spielenden einzubauen, da das Geschlecht performativ ist. Es äußert sich also darin, wie sich die Person verhält, nicht jedoch darin, wie die Person gebaut ist (Judith Butler). Hier finden sich sowohl 2D-Elemente für die Bearbeitung der Einstellungen, als auch 3D-Elemente für die Darstellung des Charakters. Durch die Schieberegler und andere Formen der Einstellungen des Charakters können die Spielenden sich deutlich effektiver mit der Figur identifizieren, wie Leonard A. Anetta in einer Studie in 2006-2008 herausfand, in der er 2 Klassen gegenüberstellte, eine die nur zwischen 2 Charakteren wählen konnte, und eine, die zwischen 100 wählen konnte. Die Klasse mit der größeren Auswahl hatte mehr Freude an dem Spiel als die, die nur zwischen einem männlichen und einem weiblichen Charakter wählen konnte ([5, p. 107-108]). Durch die Ergebnisse der Befragung der Schüler\*innen ist zu erkennen, dass diese Meinung geteilt wird, da die Charaktererstellung durchgehend als gelungen empfunden wurde.

### 5.1.3 Spielwelt

Die Spielwelt besteht aus einer Insel mit verschiedenen Häusern, in denen einige Charaktere ihre Wege laufen. Mit den Charakteren kann man sich unterhalten. Die Dialoge können dazu dienen, Aufgaben und Gegenstände zu bekommen, Aufgaben und Gegenstände abzugeben, Aufgaben und Gegenstände zu tauschen oder Informationen zu bekommen. Je nach Funktion wirken sich die Antwortmöglichkeiten anders aus.

Geben Spielende erledigte Aufgaben ab, überprüft „Built on solid Seas“ ob sie erfolgreich erledigt wurden. Wurden sie dies, erhält man Punkte und in einigen Fällen passieren spezielle Ereignisse in der Spielwelt (Sachen werden freigeschaltet, man erhält Belohnungsgegenstände usw.). Wurden die Aufgaben nicht korrekt gelöst, wird ein anderer Teil des Dialogs angezeigt, der darstellt, dass man offenbar noch einige Fehler gemacht hat. Darüber hinaus erhält man weniger Punkte, wenn man die Aufgabe nicht beim ersten mal

löst, und „Built on solid Seas“ ordnet die Fähigkeiten der Spielenden in diesem Bereich niedriger ein.

Wenn man weit genug fortgeschritten ist, wird ein Tor in einem alten Fabrikgelände sichtbar und es wird der Towerdefense Aspekt freigeschaltet. Ab diesem Punkt können die Spielenden Plastik als Ressource sammeln, um die Insel zu vergrößern. Hier werden mit Klicks auf den Meeresboden die Elemente einer Tilemap (3D-Zellen in einer Tabellenstruktur) ausgetauscht und die Änderungen abgespeichert. Diese Tilemap bewirkt, dass „Built on solid Seas“ ein Design hat, das Minecraft ähnlich sieht, und es sich relativ einfach gestaltet, die Insel zu bearbeiten, ohne riesige Mengen an Dateien zu speichern.

Zu der Spielweltszene gehören ebenso noch 2D-Elemente für die Dialoge und Pop-up-Fenster, die die Story erklären sollen, oder Menüs darstellen sollen. Dann gibt es noch Anzeige-Elemente um Punkte und Gegenstände darzustellen, Häuser und die Pfade, die die ebenfalls enthaltenden Charaktere entlanglaufen. Für die Charaktere wird hier kein aufwändiger Pfad-Findungs-Algorithmus<sup>63</sup> verwendet, da die Zeit fehlte.

Klickt man in dem eigenen Haus auf eine Staffelei, während man eine Aufgabe aktiv im Inventar hat, wird der Aufgabenbildschirm geöffnet und diese Aufgabe geladen.



Abbildung 14: Die Spielwelt in „Built on solid Seas“

#### 5.1.4 Aufgabenbildschirm

Im Aufgabenbildschirm wird eine geometrische Figur gezeichnet, zu der einige Längen, Winkel, Flächen oder Volumen berechnet werden sollen. Diese Ergebnisse sollen die Spielenden in die verschiedenen Textfelder eintragen, und auf „Bestätigen“ oder „Abbrechen“ klicken. Am rechten Bildschirmrand wird noch eine Beschreibung der Aufgabe in Textform gegeben (hier wird bbcode verwendet, es könnten also auch Links und Bilder eingefügt werden). Bisher wird bei Bestätigen die Aufgabe auf ihre Richtigkeit geprüft, diese gespeichert und zurück zur Spielwelt manövriert („Abbrechen“ überspringt die Überprüfung), und erst beim Abgeben der Aufgabe erhält man im Dialog eine Antwort, ob die Aufgabe so richtig gelöst ist oder nicht.

Ein Charakter in der Spielwelt mit dem Namen Jessica hat die Möglichkeit, zu analysieren, um welche Art von Aufgaben es sich handelt. Dieser Charakter gibt einem für den Notfall ein paar Hinweise oder verweist auf das Buch, das diese Schule ohnehin verwendet

<sup>63</sup>Pfad-Findungs-Algorithmus

(letzteres ist optional, da es aber nur 4-5 Bücher für die Schulen in NRW gibt, ließe sich dies ohne weiteres anpassen bzw. austauschen).



Abbildung 15: Die Aufgabenbearbeitung von „Built on solid Seas“

### 5.1.5 Dialoggenerator

Dieser Bildschirm kann von den Spielenden nie gesehen werden und dient den Lehrenden dazu, die Dialoge zu bearbeiten und neue Dialoge dem Spiel hinzuzufügen. Hier gibt es viele verschiedene 2D-Elemente, die dazu dienen, Einfluss auf die NPC-Dialoge im Spiel zu nehmen. Dialoge und Aufgaben werden automatisch in „Built on solid Seas“ geladen, und werden somit direkt nach ihrer Erstellung aktiv. Dadurch erlaubt das Spiel das Austauschen sämtlicher Dialoge und Aufgaben mit einfachen Aktionen. Die genaue Funktion des Dialoggenerators wird in dem entsprechenden Spielelement genauer beschrieben.

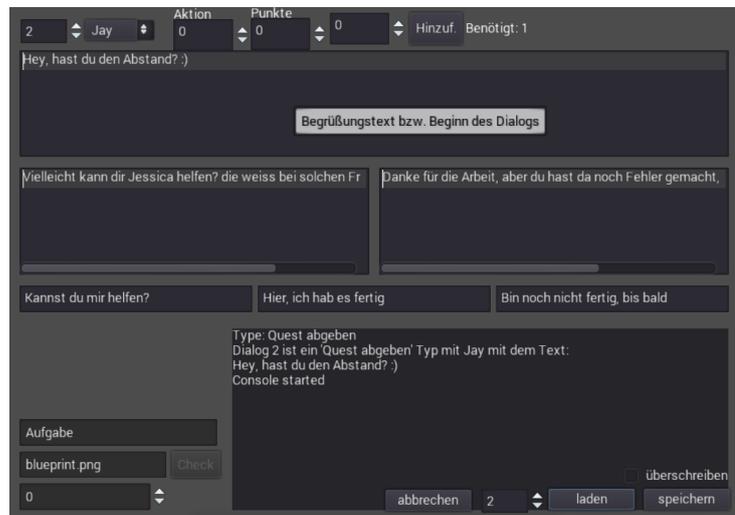


Abbildung 16: Die Dialoggenerierung in „Built on solid Seas“

## 5.2 Globale Klassen

Die globalen Klassen werden unabhängig von den Szenen geladen und können globale Parameter und Funktionen speichern. In Godot werden sie als Autoload-Klassen<sup>64</sup> mit jeder Szene geladen, bzw. bleiben geladen, wenn man die Szene wechselt.

<sup>64</sup>Automatisch vorgeladene Klassen

### 5.2.1 Import-Export

Die Import-Export Klasse bietet Funktionen, um mit der Dateiverwaltung zu interagieren. Hier werden Aufgaben, Dialoge, Speicherstände usw. gespeichert und geladen. Mit der jeweiligen Importfunktion werden sie aus dem Speicher in ein Array geladen, das an die Klasse zurückgegeben wird, die die Information angefordert hat. Mit den Exportfunktionen werden Dateien mit einem oder mehreren Arrays gefüllt, und in eine sinnvolle Struktur gebracht. Hier wird einfach gesprochen alles, was nicht Textur ist, geladen und gespeichert.

Die Dateiformate sind jeweils einzeln entstanden und funktionieren nach dem Prinzip eines einfachen endlichen Automaten mit Zuständen. Es wird zeilenweise in menschenleslicher Textform gespeichert, und wenn ein Array fertig gespeichert ist, wird ein neuer Zustand mit einem Schlüsselwort markiert.

Wenn die Import-Funktionen diese Dateien nun laden wollen, lesen sie eine Zeile, überprüfen ob diese ein Zustandsmarker ist, wenn ja, springen sie in einen entsprechenden Zustand, wenn nicht, wird diese (und eventuell noch ein paar weitere) je nach aktuellem Zustand in einen bestimmten Platz des Lese-Arrays geschrieben.

```
func exportInventory( items ):
»   var index=0
»   while(file.file_exists(str("Saves/info/inventory",index,".info"))):
»       index+=1
»       var filename=str("Saves/info/inventory",index,".info")
»       file.open(filename,file.WRITE)
»       for item in items:
»           file.store_line(str(item.aufgabe))
»           file.store_line(item.icon.get_path())
»           file.store_line(item.name)
»       file.close()
»       return filename
```

Abbildung 17: Der Export am Beispiel des Inventars

```
func importInventory(filename):
»   file.open(filename,file.READ)
»   var info=[]
»   while(!file.eof_reached()):
»       var tmp=[]
»       tmp.push_back(int(file.get_line()))
»       tmp.push_back(file.get_line())
»       tmp.push_back(file.get_line())
»       info.push_back(tmp)
»   file.close()
»   info.resize(info.size()-1)
»   print(info)
»
»   return info
```

Abbildung 18: Der Import am Beispiel des Inventars

Diese Speichermethode ist nicht effizient und lässt zu, dass Spielende die Speicherstände selber lesen und bearbeiten. Da aber einerseits die Dateien in der Regel pro Datei keinen Kilobyte überschreiten, ist dies nach aktuellem Stand der Technik irrelevant und die Manipulierbarkeit der Speicherstände ist auch aktuell nur zu Debugging-Zwecken gegeben. Sie ließe sich mit 2 Zeilen Code in eine verschlüsselte Variante umschreiben.

### 5.2.2 Savegame

```

func loadGame(number):
» var metadata=get_node("/root/import_export").importMetadata(number)
» get_node("/root/global").PlayerName=metadata[0]
» Points=metadata[1]
» Resources=metadata[2]
» doneDialogs=metadata[3]
» actionsdone=metadata[4]
» leveltype=metadata[5]
» ItemsLoading=get_node("/root/import_export").importInventory(metadata[6])
» print("ITEMSTOLOAD: ",ItemsLoading)
» aufgabenheader=get_node("/root/import_export").importAufgabenheader(metadata[7])
» TDPlane=get_node("/root/import_export").importTDState(metadata[8])
» var tmp=get_node("/root/import_export").importWorldState(metadata[9])
» changedWorldcells=tmp[0]
» changedWorldcellto=tmp[1]
» calculateNextDialogs()
» get_node("/root/global").updateGui()

```

Abbildung 19: Das Laden des Spiels zeigt die komplexe Speicherstruktur.

Die Klasse Savegame speichert und lädt den aktuellen Stand von „Built on solid Seas“ zur Laufzeit. Sie arbeitet eng mit der Import-Export Klasse zusammen und hilft, die benötigten Dateien zu sortieren, zu bearbeiten, aktuelle

Dialoge und Aufgaben auszugeben oder Änderungen abzuspeichern. Da es so eng mit den Aufgaben und dem Inventar verwoben ist, ist hier auch eine Funktion eingebaut, die dem Charakter Jessica hilft, die aktuelle Aufgabe der Spielenden zu analysieren und Hinweise zu geben. Auch finden sich hier Algorithmen, die entscheiden, welcher Dialog bei welchem Charakter als nächstes angezeigt werden soll, sowie Algorithmen, die die Leistung der Spielenden bewerten und über eine passende nächste Aufgabe entscheiden. (der Ansatz Informed Teaching aus A. Anettas Framework „The I’s have it“ [5, p. 109])

### 5.2.3 Item

Die Klasse „Item“ speichert und verwaltet die Informationen über einen Gegenstand, den die Spielenden im Spiel besitzen können. Hier reichen derzeit ein Icon, ein Name und meistens auch eine Aufgabe aus. Aus diesem Grund wird die Item-Klasse meistens eher normal aus den Spielressourcen geladen und benutzt, statt sie als globale Variablen/Funktionenverwaltung zu verwenden. Sie ist allerdings auch nötig im Hinblick auf die Savegame Klasse, da das Inventar abgespeichert und geladen werden muss, und diese Klasse für die Items noch die Icons lädt, die sonst nur als Dateiname gespeichert sind.

### 5.2.4 Globals

Die Klasse „Globals“ verwaltet die Szenenübergänge und einige wenige Informationen, die auch das Hauptmenü betreffen. Hier finden sich die Einstellungen zu der Darstellung der Welt, diverse Funktionen für die verschiedenen Events und Belohnungen im Spiel, sowie die setScene-Funktion, die den Ladebildschirm öffnet und diesen anweist, die neue Szene zu laden.

## 5.3 Spielelemente

In diesem Kapitel möchte ich noch einmal genauer auf verschiedene Spielelemente oder Aspekte von „Built on solid Seas“ eingehen und deren Funktionsweise etwas genauer beleuchten. Anfangen möchte ich hier bei den Dialogen, Aufgabenerstellung und deren Bearbeitung bzw. Erstellung. Dann möchte ich genauer beschreiben, wie die Spezialereignisse auf das Spiel einwirken, wie sie ausgelöst werden und wie genau der Towerdefense-Aspekt funktioniert. Danach unterscheide ich die verschiedenen Motivationsaspekte des Spiels

und wie die Spielenden bewertet werden, sodass das Spiel sie im Flow-Channel halten soll. Zu guter Letzt beschreibe ich, welche Aspekte von Shading ich für die Darstellung besonders behandelt habe.

### 5.3.1 Dialoge- und Aufgabenerstellung und ihre Bearbeitung



Abbildung 20: Die Aufgabenerstellung in „Built on solid Seas“

Die Dialoge und Aufgabenstellungen sind ein wichtiger Punkt von „Built on solid Seas“. Hierunter ist zu verstehen, dass diese von der Savegame Klasse automatisch behandelt werden müssen, und vollkommen austauschbar sein müssen, damit das Spiel in jede Klasse jeder Schule getragen werden kann und dort genutzt werden kann. Für jede Aufgabe, die von den NPCs<sup>65</sup> gestellt werden soll, benötigt „Built on solid Seas“ je einen Dialog für das Erhalten der Aufgabe und das Abgeben der Aufgabe.

In der Klasse der Aufgaben von „Built on solid Seas“ ist definiert, welche Geometrien (Punkte, Linien, Kreise, Ellipsen und Kreis/ Ellipsenabschnitte) gezeichnet werden sollen. Dann ist definiert, wo genau welche Längen, Flächen, Winkel und Volumina gegeben sind, und welche gesucht sind, inklusive ihres Ergebnisses. Außerdem gibt es noch Informationen über die Art, das Kapitel und die Schwierigkeit des Spiels. Zusätzlich gibt es noch eine Möglichkeit, ein internes Geodreieck freizuschalten, das auf 20 Pixel pro Längeneinheit und 5 Grad pro Winkleinheit der Ursprungsform des Bildes (720 \* 480 pixel) definiert ist, weil das Bild je nach Auflösung und Form des Monitors schließlich mit skalieren würde, und Winkel bzw. Längen nicht vom Bildschirm ablesbar wären. Möchte man den Schüler\*innen also diese Möglichkeit geben, Längen und Winkel zu messen und zu schätzen, kann man diese Möglichkeit für alle gleichermaßen freischalten. Danach gibt es nur noch eine Möglichkeit, einen Text anzeigen zu lassen, der die Aufgabe genau beschreibt. Für die einfache Erstellung eines solchen Texts ist ein Textgenerierungs- Button vorhanden, der das Kapitel und die Aufgabenart abliest und eventuell Aufgabentextelemente vorschlägt. Zu guter Letzt gibt es noch ein Textfeld und einen Knopf, anhand dessen Bilddateien in den Hintergrund geladen werden können.

<sup>65</sup>Nicht spielbare Charaktere, englisch non-playable Characters

Wie man anhand dieser Schreibweise schon erkennen kann, hat die Szene für die Aufgaben 2 Modi. Startet man sie aus der Spielwelt heraus, so startet sie im Modus des Spielens, zeigt Längen und Winkel als Textelemente an, versteckt den Großteil der GUI und schaltet die Funktion der „Bestätigen“- und „Abbrechen“-Knöpfe auf die für das Spielen notwendigen Funktionen. Startet man sie jedoch von außerhalb der Spielwelt, zeigt sie alle nötigen Elemente zum Bearbeiten einer Aufgabe an. Hier werden Elemente freigeschaltet, die einstellen, was man gerade genau bearbeiten möchte, welche vorher erstellten Aufgaben man gerne laden würde und welche Art von Aufgabe mit welchem Aufgabentext werden soll. Dann können die Lehrpersonen den Modus der GUI<sup>66</sup> aus „Punkte Bearbeiten“, „Linien ziehen“, „Kreise zeichnen“, „Winkel hinzufügen“, „Warten“, „Aufgabe“ und „Labels“ einstellen. Die ersten 4 Elemente sind relativ einfach. Im „Punkte Bearbeiten“-Modus werden Punkte mit einem Linksklick erstellt oder verschoben (je nachdem ob man einen existierenden Punkt anklickt oder nicht). Diese Punkte sind intern als Vektoren mit 2 Elementen gespeichert.

In dem „Linien ziehen“-Modus werden Linien zwischen 2 Punkten gezeichnet. Hier klickt man nacheinander auf 2 Punkte und erstellt so ein Element, das an ein Array<sup>67</sup> für die Linien angehängt wird. Hier wird ein Vektor mit 3 Elementen genutzt, das Erste bezeichnet den Key<sup>68</sup> für den ersten Punkt, das Zweite den Key für den Zweiten Punkt, und das Dritte die Länge der Linie in Zentimetern.

In dem „Kreise zeichnen“-Modus kann die Person erst anhand eines Parameters in einem Textfeld entscheiden, wie gestaucht der gezeichnete Kreis sein soll (um z.B. Zylinder zu zeichnen ist das nötig), und dann klickt die Person auf einen Punkt um einen Kreis um den Punkt bis zu der Position des Cursors mit den entsprechenden Parametern zu erstellen. Hier wird intern ein Vektor mit 3 Elementen gespeichert, der Key für den Punkt, der Stauchungsfaktor und der Radius.

In dem „Winkel zeichnen“-Modus werden Kreisbögen erstellt. Hierfür klickt die Person zuerst auf den Punkt, um den dieser Bogen entstehen soll, dann nacheinander entgegen des Uhrzeigersinnes auf die verschiedenen Punkte von wo bis wo der Kreisbogen entstehen soll. Danach skaliert die Person mit gedrücktem Mauszeiger, wie weit der Kreisbogen sein soll. So lassen sich auch die Kreisbögen zeichnen und die Winkel darstellen. Intern werden hier 5 Werte gespeichert: Der Key des Mittelpunkts, der Radius, der Key des Startpunkts, der Key des Endpunkts und der Winkel. Hierfür benötigte ich einen Vektor mit 2 Elementen und einen Vektor mit 3 Elementen, da diese Vektoren eigentlich eher für grafische Anwendungen waren, und es daher leider keine Vektoren mit mehr als 3 Elementen gibt. Diese werden wieder normal in Arrays gespeichert.

Sobald die Elemente erstellt sind, werden sie auf dem Bildschirm gezeichnet. Die Längen bzw. Winkel werden als Textfelder mitgeliefert, und können dadurch nach-skaliert werden, wenn die automatische Skalierung (1 Zentimeter als 20 Pixel berechnet, die Winkel nach einer nicht existierenden Stauchung oder Streckung des Bildschirms berechnet) nicht ausreicht. Mit einem Rechtsklick werden die Elemente auch wieder gelöscht, und alle Elemente, die auf diese referenzieren, werden ebenfalls gelöscht. Die übrigen Keys werden angepasst, da sie intern auf die Position im Array zeigen.

Im „Warten“-Modus passiert an sich nichts mit Links/Rechtsklick, außer dass die Personen hier ungestört die Textfelder auswählen können um Längen und Ergebnisse einzugeben. Der „Aufgabe“ und der „Label“ Modus wird wieder etwas stärker spezifiziert. Hier kann

---

<sup>66</sup>grafisches Nutzer\*innen-Interface

<sup>67</sup>Liste von Werten im Speicher, im Fall von GDScript mit Dynamischer Länge

<sup>68</sup>Bezeichner, direkte Referenz auf ein explizites Element

man jeweils wählen, was für Aufgaben oder Labels<sup>69</sup> man erstellen möchte (Längen, Winkel, Flächen, Volumen). Dann werden für Winkel, Flächen und Volumen jeweils Punkte benutzt. Man kann also auf die Punkte klicken um ein entsprechendes Label oder eine entsprechende Aufgabe an dieser Position mit der entsprechenden Einheit erstellen. Dann gibt man die Ergebnisse ein, und intern wird das gespeichert als Vektor mit den Werten Aufgabenart, Key für den Punkt und Ergebnis. Mit Rechtsklick werden die Elemente wieder gelöscht (oder bei Längen und Winkeln nur versteckt). Bei Längen werden hier für den Key keine Keys von Punkten verwendet, sondern die von Linien. Die Labels und Aufgaben befinden sich schließlich hier in der Mitte zwischen den Punkten.

Dann gibt es noch die Dialoge. Dialoge teilen sich, wie in den Szenenstrukturen bereits erwähnt, bei mir auf in die Typen „Quest erhalten“, „Quatschen“, „Quest abgeben“ und „Tauschen“. Die Dialoge haben einen Dialogtext, drei Antwortmöglichkeiten (bestätigend, nachfragend und beendigend), bis zu zwei Reaktionen auf die Antworten (auf „bestätigend“ bei allen außer bei „Quest erhalten“, sonst immer auf „nachfragend“) und bei „Quest erhaltend“ bzw. „Tauschen“ ein Item das man erhalten kann, sowie bei „Quest abgeben“ und „Tauschen“ ein Item, das man abgeben kann.

Die Dialoge haben ebenfalls noch interne Informationen. Hier gibt es eine Liste von Dialognummern, die abgeschlossen sein müssen, Punktzahlen, die erreicht sein müssen und Ereignisse, die passiert sein müssen, damit der Dialog freigeschaltet wird, und den Namen der Person, mit der man den Dialog hat.

Für den aktuellen Dialog mit einer Person wird für jede Person jeder Dialog der Person überprüft. Wenn der Dialog stattfinden kann (wenn benötigte Dialoge angegeben sind, sind diese alle abgeschlossen? Wenn nicht, ist die Punktzahl des Dialogs erreicht und ist das nötige Ereignis schon passiert? Wenn ja, dann kann der Dialog stattfinden), wird er in eine Liste eingefügt. Danach wird überprüft, ob ein „Quest abgeben“-Dialog darunter ist. Diese Dialoge hätten höchste Priorität. Danach kommen Handelsdialoge und „Quest erhalten“-Dialoge und zu guter letzt normale „Quatschen“-Dialoge. Ist der nächste Dialog ein „Quest erhalten“-Dialog wird noch einmal geschaut, welche Aufgabe von den Dialogen die Aufgabe ist, die den Fähigkeiten der Person am ehesten entspricht. Die am ehesten passenden werden dann per Zufall ausgelost.

Auch einen Dialog kann man mit „Built on solid Seas“ erstellen. Hierfür gibt es ein entsprechendes Fenster, in dem die verschiedenen Texte eingegeben werden können, in dem Gegenstände eingestellt werden (die GUI verändert sich nach Typ und lässt nur das zu, was der Typ zur Verfügung hat) und die Bedingungen für die Ausführung des Dialogs beschrieben werden. Zur Erleichterung werden viele Informationen der Einstellungen, die man vornimmt, in eine Spiel-interne Textausgabe geschrieben, sodass die Spielenden genau wissen, was sie bearbeiten.

### 5.3.2 Spezialereignisse

Die Spezialereignisse sind oftmals Popups, die erscheinen, wenn eine bestimmte Quest erledigt ist, und sich die Geschichte weiter verändert. Sie sind festgeschrieben in der Globals-Klasse als mögliche Belohnungen für Quests und erklären die Änderungen in der Spielwelt sowie die freigespielten Mechaniken. Diese Spezialereignisse, wie dass z.B. ein Kapitel beendet wurde, schalten neue Dialoge frei und die Story von „Built on solid Seas“ handelt sich entlang dieser Ereignisse.

---

<sup>69</sup>Bezeichner

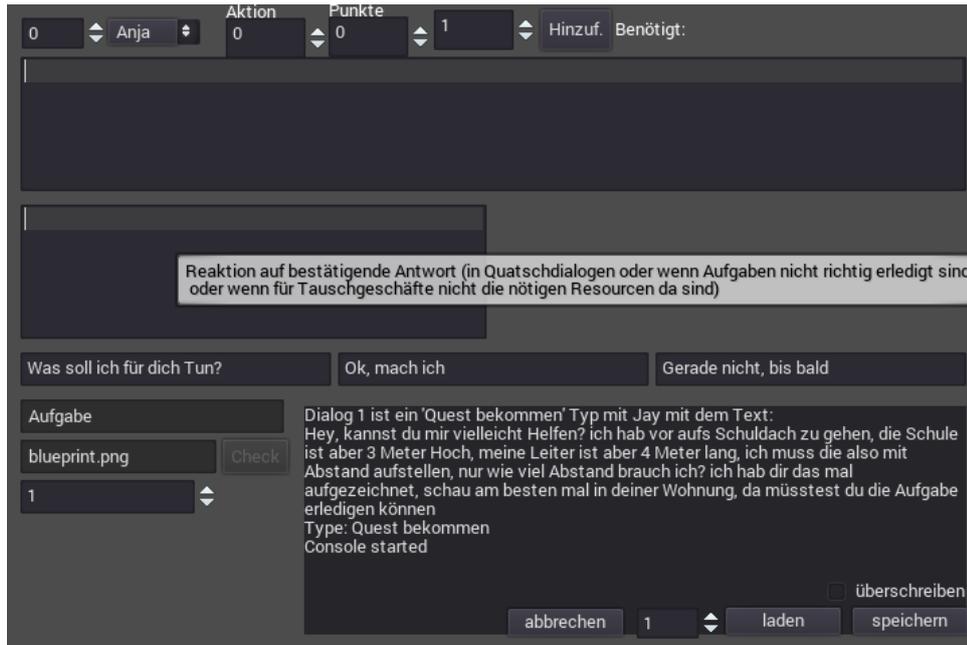


Abbildung 21: Das Fenster zum Bearbeiten von Dialogen

### 5.3.3 Towerdefense

```

func generatePrices():
» Prices.push_back([100])
» Prices.push_back([100,"Hälfte1"])
» Prices.push_back([100,"Chap1done"])
» Prices.push_back([100,"Door"])
» Prices.push_back([100,"TowerLv2"])
» Prices.push_back([100,"Tools1"])
» Prices.push_back([100,"Tools2"])

func reward(name):
» if(name=="Door"):
» » get_node("/root/savegame").actionsdone[1]=true
» elif(name=="Hälfte1"):
» » get_node("/root/savegame").actionsdone[2]=true
» » if(get_tree().get_root().has_node("Welt")):
» » » get_tree().get_root().get_node("Welt").updatetowervisibility()
» » » get_tree().get_root().get_node("Welt/GUI/PopupMenu5").popup()
» » get_node("/root/savegame").Resources+=200
» » updateGui()
» elif(name=="Chap1done"):
» » get_node("/root/savegame").actionsdone[3]=true
» elif(name=="TowerLv2"):

```

Abbildung 22: So sind die Ereignisse in der Globals Klasse definiert und aufgerufen

Der Towerdefense-Aspekt von „Built on solid Seas“ findet auf einem alten Fabrikgelände auf der Insel statt. Sobald man eine kleine Menge Aufgaben zur geschichtlichen Einleitung erledigt hat, wird hier ein Spielelement freigeschaltet. Dieses Spielelement funktioniert so, dass ein Level gestartet wird, indem an der Steuerung des Tors ein Interface mit einem Klick geöffnet wird. Dann wird das Level aus einer Liste ausgewählt, und mit dem

Klick auf „ok“ wird es gestartet. Level bestehen aus zeitlich eingeteilten Elementen, die die Menge des Wassers, das ab diesem Zeitpunkt auf das Fabrikgelände fließt, und die Menge des darin enthaltenen Plastikmülls regeln. Das Wasser fließt in 19 Kacheln über das Gelände und schwemmt das Plastik mit sich. Erreicht das Plastik die letzte Kachel (gegenüber dem Eingang) so ist das Level nicht bestanden und es wird ohne Gewinne beendet. Gelingt es den Spielenden jedoch, den Wasserstand gering genug zu halten, und/oder das Plastik

rechtzeitig aus dem Wasser zu fischen, so wird das Level als bestanden gewertet und die Spielenden erhalten je nach Schwierigkeit des Levels (Menge des Wassers und Plastiks) unterschiedlich viele Ressourcen, mit denen sie die Insel vergrößern können, oder Wände auf das Fabrikgelände bauen können. Damit das Wasser abfließen kann, und das Plastik gesammelt werden kann, wird den Spielenden die Möglichkeit gegeben, auf den Boden des Geländes Abflüsse zu bauen, bzw. auf Wände Sammelstationen. Diese erhalten die Spielenden, indem sie Aufgaben lösen. Wände können auf trockenen Boden gebaut werden. Diese kosten einige Ressourcen, benötigen aber noch keine erledigten Aufgaben.

Bei der Umsetzung dieses Spielelements stellten sich viele technische Herausforderungen. Es musste eine Wassersimulation selbstgeschrieben werden, die Pfadfindung des Plastiks musste geschrieben werden, die drehbaren Kescher der Sammelstationen mussten genauer animiert werden, um nicht mit Wänden zu kollidieren und es mussten Möglichkeiten gefunden werden, wie sicher gestellt ist, dass Wasser immer bis zum Ende des Spielfelds fließen kann. Der Ablauf des Levels an sich war relativ einfach zu programmieren, hier orientierte ich mich ebenfalls an einem endlichen Zustandsautomaten.

Die Simulation des Wasserflusses war die denkbar größte Herausforderung. Ich habe mich in diesem Spiel aus Performance-Gründen darauf beschränkt, dass das eingegrenzte Feld des Wassers aus 19x19 Kacheln besteht. Jede Kachel hat 4 Strahlen von ihrem Mittelpunkt in die verschiedenen Richtungen, die Darstellung in der realen Welt, und ein Kollisionsobjekt. Insgesamt (mit zwei Nodes<sup>70</sup> als Parent-Nodes<sup>71</sup>) besteht eine Kachel also aus 8 Knoten, das gesamte Feld also aus 2888 Knoten. Diese Knoten sind damit 4/5 der Knoten der Spielwelt und wirken sich für die Verhältnisse des wenig-fordernden Spiels massiv auf die Performance aus. Würde ich das Spiel aus Performance Gründen noch einmal überarbeiten, so würde ich genau hier ansetzen.

Diese Kacheln werden nun pro Frame<sup>72</sup> mit Wasser gefüllt, oder entleeren sich (die Fläche die den Wasserspiegel repräsentiert, steigt oder sinkt). Dies besteht aus 3 Schritten. Es wird geprüft, ob sich das Spiel in einem Zustand befindet, in dem Wasser in das Level fließen soll. In diesem Fall wird der Wasserstand der Kachel (0,0) entsprechend des Wertes im Level angehoben. Dann wird in der Wasser-Animation-Klasse geprüft, ob es Wassersimulation geben muss (ein active-Flag<sup>73</sup> entscheidet dies). Anschließend wird das active-Flag erst mal auf false gesetzt, für den Fall dass keine Wasseranimation stattfindet, und dann wird von jeder Kachel überprüft, ob in Richtung Norden sowie in Richtung Osten jeweils ein Nachbar existiert und wie viel Unterschied zu der aktuellen Kachel besteht. Wenn der Unterschied größer als ein bestimmter Wert ist, „fließt“ Wasser in die Richtung des Nachbarn. Es wird also die benachbarte Kachel um ein Drittel des Unterschieds angehoben und die aktuelle Kachel um ein Drittel des Unterschieds gesenkt, was darin resultiert, dass der Unterschied zwischen den Kacheln nur noch ein Drittel des vorherigen Unterschieds ist. Dadurch, dass ich nur die nördlichen und östlichen Nachbarn überprüfe, spare ich die Hälfte der nötigen Arbeitslast. Auf der anderen Seite mit der Überprüfung jeder Kante zwischen zwei Kacheln von der angrenzenden südlichen, oder der angrenzenden westlichen Kachel aus, stelle ich sicher, dass ich keine unnatürlichen Eigenschaften des Wasserflusses provoziere. Hier handelt es sich also um eine Funktion mit quadratischem Aufwand. In früheren Versionen verwendete ich hier Funktionen mit exponentiellem Aufwand, die rekursiv arbeiteten und den Computer überlasteten. Sobald ein Wasserfluss ausgeführt

---

<sup>70</sup>Knoten, Objekte in der Godot-Engine

<sup>71</sup>Hierarchisch über einem Knoten stehende Knoten

<sup>72</sup>Bild, das der Computer berechnet

<sup>73</sup>Binäre Variable die einen Zustand markiert

wurde, wird die active-Flag wieder auf true gesetzt, damit die Animation im kommenden Schritt fortgeführt wird. Danach wird überprüft, ob über einem Abfluss in der Kachel Wasser enthalten ist. In diesem Fall wird aus dieser ein entsprechender Wert abgezogen. In dieser Funktion wird ebenfalls noch ein Flag überprüft, das anzeigt, ob die Nachbarn neu berechnet werden sollen (was pro Kachel bedeutet, dass ein Raycast<sup>74</sup> in jede Richtung durchgeführt wird, und wenn es kollidiert, werden keine Nachbarn eingetragen, kollidiert es nicht, wird ein Nachbar in die Richtung eingetragen).

```

if(active):
    » active=false
    » undermin=true
    » for i in range(19):
    » » for j in range(19):
    » » » var neighbours = get_node(str("Cube_",i,"_",j)).hasneighbours
    » » » var amount = 0
    » » » if neighbours[0]:
    » » » » var dif=heights[i][j+1]-heights[i][j]
    » » » » difs[i][j]=difs[i][j]+dif
    » » » » difs[i][j+1]=difs[i][j+1]-dif
    » » » » amount+=2
    » » » » if neighbours[1]:
    » » » » » var dif=heights[i+1][j]-heights[i][j]
    » » » » » difs[i][j]=difs[i][j]+dif
    » » » » » difs[i+1][j]=difs[i+1][j]-dif
    » » » » » amount+=2
    » » » » if neighbours[2]:
    » » » » » amount+=2
    » » » » if neighbours[3]:
    » » » » » amount+=2
    » » » » if(abs(difs[i][j])>0.05):
    » » » » » get_node(str("Cube_",i,"_",j)).drain(-difs[i][j]/amount)
    » » » » » difs[i][j]=0
    » » » » » active=true
    » » » » if(undermin and heights[i][j]>0.5):
    » » » » » undermin=false
    » for drainpoint in drainpoints:
    » » if(heights[drainpoint.x][drainpoint.y]>0.05):
    » » » get_node(str("Cube_",drainpoint.x,"_",drainpoint.y)).drain(drainpoint.z)
for bottle in bottles:
    » if(heights[path[floor(bottle.x)].x][path[floor(bottle.x)].y]>0.5):
    » » bottlesettopos(bottles.find(bottle),bottle.x+delta/2,delta)
  
```

Abbildung 23: Der Code für die Wassersimulation

Eine Schleife wird durchgeführt, die für jede Flasche in dem Feld diese anhand des gegebenen Pfades bewegt.

Damit kommen wir zur zweitgrößten Herausforderung. Diese besteht aus dem Pfadfinden. Wie oben schon erwähnt hat jede Kachel eine Anzahl an Nachbarn. Für die Pfadfindung verwendete ich also einen klassischen Algorithmus, ich beginne also bei der ersten Kachel, trage hier den Wert 0 in eine Tabelle ein, weil ich diese Kachel von dem Anfang des Feldes aus direkt erreiche, und trage dann in jedes andere Feld einen Wert ein, der größer als die Anzahl der Felder ist. Dann beginne ich, und trage in jedes Feld das Minimum des eigenen Wertes und des inkrementierten Wertes jedes Nachbarn ein. So erhalte ich eine Tabelle, die darstellt, wie viele Felder minimal zwischen ihm und dem Anfang liegen. Ist dies fertiggestellt, wird von dem letzten Feld beginnend aus überprüft, aus welcher Richtung das vorherige Feld, also das, dessen Wert genau eins kleiner ist, kommt. Darüber

<sup>74</sup>Das Werfen eines Strahls in eine Richtung

baut sich ein Stack<sup>75</sup> auf, dessen Punkte den minimalen Pfad zwischen Anfang und Ende beschreiben.

Will man eine Mauer an einer bestimmten Stelle bauen, wird eine Funktion überprüft, die analog funktioniert, und lediglich zurückgibt, ob das letzte Feld erreichbar ist oder nicht. Wenn ja, darf die Mauer gesetzt werden. Hier ist mir jedoch ein wichtiger gedanklicher Fehler im Laufe der Programmierphase passiert, und zwar wird die Physik-Engine von Godot nach jedem Frame neu aufgebaut, die „Process“-Funktion, die unter anderem das Setzen neuer Mauern möglich machen sollte, wird jedoch durchgehend zwischen den Frames aufgerufen. Bewege ich also ein Objekt in der „Process“-Funktion, kollidiert währenddessen kein Strahl mit dem Objekt an der neuen Position, erst wenn das Frame beendet ist, werden die Objekte tatsächlich bewegt. Also musste ich die Nachbarn manuell bearbeiten, und konnte nicht eine Wand setzen, die Nachbarn nachberechnen lassen, überprüfen ob das letzte Feld erreicht wird, und dann entscheiden, ob die Wand stehen bleiben darf oder nicht.

Daher ergibt sich aus Performance- Gründen die Frage, ob die Berechnung der Nachbarn anhand der Strahlen überhaupt nötig ist, und ich hier nicht 60-70 Prozent der Knoten der Spielwelt einsparen könnte. Auch diese Frage wird sich aus Zeitgründen, und weil das Spiel auch so ganz gut funktioniert, erst nach der Bachelorarbeit klären lassen.

### 5.3.4 Motivationselemente

„Built on solid Seas“ hat nun verschiedene Motivations-Elemente, die sicherstellen sollen, dass die Spielenden das Spiel spielen wollen. Hier drunter fallen klassische Mechaniken wie die Punkte und das Sammeln von Ressourcen und Gegenständen, die den klassischen Mechaniken der Gamification<sup>76</sup> nahe liegen, die oft grob als „PBL“ („Points, Badges and Leaderboards“<sup>77</sup>) zusammengefasst werden.

Punkte und Abzeichen werden hier mit den Punkten des Spiels und den Gegenständen dargestellt, die Ranglisten ergeben sich durch soziale Interaktion mit den anderen Spielenden.

Amy Jo Kim erwähnte in ihrer Präsentation zu casual connect 2011 im Hinblick auf unterschiedliche Spielendtypen, dass weiblich sozialisierte Menschen dazu tendieren, zusammenzuarbeiten, männlich sozialisierte auf der anderen Seite lieber gegeneinander antreten wollen („females in General like to collaborate [...], males tend to like to compete“ [17, min. 4:17 - 4:30]). Sie spielt damit auf die Spielendtypen von Richard Bartle an. Dieser ordnet die oben genannten Punkte dem Spielendtyp der „Achiever“<sup>78</sup> zu („Achievers regard points-gathering and rising in levels as their main goal, and all is ultimately subserviant to this.“ [6]). Da er seine Spielenden in 4 Typen aufteilt (Killers<sup>79</sup>, Achievers<sup>80</sup>, Socializers<sup>81</sup> und Explorers<sup>82</sup> [6]) und da ich mit den oben genannten offenbar nur die „Achievers“ also eher die männlich sozialisierten Spielenden anspreche, es mir aber in einem Spiel für eine Schule nicht leisten kann, etwa die Hälfte der Zielgruppe zurückzulassen, benötige ich

---

<sup>75</sup>Stapelstruktur in der Programmierung

<sup>76</sup>spielbarmachung von ernsten Themen

<sup>77</sup>Punkte, Abzeichen und Ranglisten

<sup>78</sup>Erobernden

<sup>79</sup>tötende

<sup>80</sup>erobernde

<sup>81</sup>sozialisierende

<sup>82</sup>erforschende

auch andere Formen der Motivation.

Um den Spielentyp des Explorers anzusprechen, ist es wichtig, die Möglichkeit zu geben, die Insel zu vergrößern und neue Spezialereignisse zu entdecken, was wiederum den „Achiever“-Typ teilweise motiviert. Auch eine durchdachte soziale Interaktion mit den NPCs und die Charaktererstellung sind wichtig um die „Socializer“ zu erreichen. Durch die soziale Interaktion gewinnen die „Quatschen“- und „Handeln“- Dialoge an Wichtigkeit. Ebenso sind die speziellen Eigenschaften und Geschichten der NPCs, die man im Laufe des Spiels erforschen kann Teil des Konzepts für die Motivation der „Socializers“, was wieder den „Explorer“-Typ motivieren kann. Darüber hinaus erreichen die Level in dem Towerdefense Aspekt, die nebenbei wieder den „Achiever“-Typ motivieren, die Motivation der „Killers“.

Dennoch ist die Motivation von „Socializers“ im späteren Verlauf des Spiels nicht optimal, da die Zusammenarbeit von Spielenden über Online-Handel oder die Möglichkeit, die Insel der anderen Spielenden zu besuchen, und die Eigene zu Dekorieren, fehlen. Dies ist leider bedingt dadurch, dass eine Online-Anbindung des Spiels Serverkapazitäten und Zeit benötigt, die ich zum aktuellen Zeitpunkt nicht habe. Dennoch ist es wichtig dies anzumerken, um zukünftige Spiele oder Versionen des Spiels zu optimieren. Einen Ausgleich schafft allerdings derzeit durch die enge Klassen-Zusammenarbeit die Möglichkeit, den Mitspielenden (anderen Schüler\*innen) die eigene Insel und die individuelle Entwicklung dieser an dem eigenen Computer zu zeigen.

### 5.3.5 Einschätzung der Spielenden

Um dem Aspekt des Informed Teaching [5, p. 109] gerecht zu werden, war es nötig, erst einmal herauszufinden, in welchen Aufgaben die Spielenden wie gut sind. Hier wird bei jeder abgegebenen Aufgabe je nach Anzahl der Versuche ein aktueller Wert (anfangs 0,5) mit einem Faktor multipliziert, der mal größer, mal kleiner als 1 ist. Das Ergebnis wird wiederum eingespeichert und stellt als Wert dar, wie gut der/die Spielende in dieser Art der Aufgaben ist. Wenn nun den Spielenden eine Aufgabe gestellt wird, so wird geschaut, wie groß der Abstand der Schwierigkeit der Aufgabe (zwischen 0 und 1) zu dem aktuellen Wert bei diesem Aufgabentyp liegt. Die Aufgaben mit dem geringsten Abstand sind die, die für diese Spielenden am besten geeignet sind. Aus dieser Gruppe wird nun per Zufall eine gewählt und diese den Spielenden gestellt. Dadurch hoffe ich, dass ich sowohl die Schüler\*innen erreiche, die diese Aufgaben gut schaffen, als auch die, die etwas länger brauchen.

Da einige Schüler\*innen nun etwas länger brauchen, wird die Aufgabe, die die Kapitel beendet, an der Punktzahl der Spielenden festgemacht. Sobald eine bestimmte Punktzahl erreicht ist, wird die Möglichkeit freigeschaltet, eine Aufgabe zu lösen, die das nächste Kapitel freischaltet. Das erste Kapitel (Satz des Pythagoras) wird beispielsweise mit 2500 Punkten freigeschaltet. Schnelle Schüler\*innen schaffen dies in 25 Aufgaben und können sich direkt auf die nächsten Aufgaben stürzen. Brauchen sie länger, haben sie allerdings noch Möglichkeiten, weitere Aufgaben zu lösen, bis sie diese Punktzahl erreichen. Auch dadurch erhoffe ich mir, die Spielenden unabhängig ihrer Leistung in dem sogenannten „Flow-Channel“ [10] zu halten.

### 5.3.6 Shading

Im Laufe der Entwicklung stellte sich heraus, dass die Wände flach werden würden. Daraufhin beschloss ich dieses Element zu einem Stil-Element zu machen. Daher wurden bei-

spielsweise die Charaktere in dem Videospiel ebenso flach und warfen für mich die Frage auf, ob dieses Design-Element für Spielende mit Magersucht eine Gefahr darstellen kann. Dies sind Fragen, die sich Spiele-Entwickler dringend stellen sollten, da Wissenschaftler\*innen der Freien Universität Berlin herausfanden, dass die Darstellung insbesondere von Frauen gerade in diesem Alter in alltäglichen Medien die Entwicklung von Magersucht begünstigen kann (Siehe [13, p. 34], [14, p. 22,23])

Daher setzte ich hier, weil ich selber kein Experte in diesem Fachgebiet war, auf die Mithilfe von Experten. Die Frauenberatungsstelle von dem Verein „Frauen helfen Frauen“ in Siegen beschäftigt sich unter anderem mit diesem Thema und konnte mir in einem Telefoninterview versichern, dass dieses Stilmittel die Entwicklung von Magersucht nicht unterstützt, solange es eine angemessene Anzahl an Variation in der Darstellung der Charaktere geben würde. Da diese ohnehin gegeben war beschäftigte ich mich daraufhin mit dem Erstellen eines Shaders<sup>83</sup>, der die 2D Charaktere nicht komplett flach erscheinen ließ. Hierzu war eine Manipulation der Normalen nötig. Ich musste eine entsprechende Textur und einen Shader entwerfen, die gemeinsam die Normalen an den Rändern nach außen biegen.

Hierfür nutzte ich eine klassische Normalmap<sup>84</sup>. Von den Farbwerten wird erst jeweils 0,5 abgezogen, um sowohl den positiven als auch den negativen Aspekt der Veränderung der Normalen hereinzubringen. Dann wurde der Rot-Anteil der Textur in x-Richtung addiert, der Grün-Anteil in y, der Blau-Anteil in z. Eine weitere Normalmap bearbeitet den Brust-Umfang und wird erst aktiviert wenn der Oberkörper eine gewisse Breite erreicht, damit es realistisch erscheint, dass der jeweilige Charakter Brüste gebildet hat.

Dann musste ich noch die Textur des Charakters bestimmen. Hier mussten die Augenfarbe, Hautfarbe, Haarfarbe, Kleidungsstil, Frisur, Gesichtsausdruck und (für Lippensynchronisation, die zeitlich nicht mehr geglückt ist) die Einstellung des Mundes einen Teil dazu beitragen. Hier habe ich einfach die verschiedenen Texturen mit Alphawerten nacheinander überlagert. Zu Grunde liegt eine Fläche mit der Hautfarbe.

Darauf wird eine Fläche mit den Augen gelegt, bei dieser Textur wird jedoch der Rot-Anteil durch die Augenfarbe ersetzt.

Als nächstes überlagert dies der Gesichtsausdruck, da die Augenlider sehr stark die Emotion des Charakters vermitteln können (falls diese aus dem Dialogtext erlesen werden können). Aktuell wird das noch nicht behandelt. Es gibt jedoch die Möglichkeit, mit einem Wert den Gesichtsausdruck des Charakters zu verändern. Zugrunde liegt hier eine Textur mit allen Gesichtsausdrücken, von der immer genau ein Abschnitt sichtbar und an der richtigen Stelle ist.

Dann wird dies überlagert mit der Textur der Lippen, die nach einer ähnlichen Technik auswählt, welche Lippenposition vorhanden ist. Hier gibt es Frames für die Vokale und einen Ausgangs-Frame. Aktuell werden diese Werte jedoch nicht berührt. Eine Möglichkeit wäre es, den Questtext Buchstabe für Buchstabe erscheinen zu lassen, jeden Vokal mit dem passenden Frame anzeigen zu lassen, und bei Konsonanten oder Satz/Leerzeichen das Ausgangs-Frame wieder aufzurufen.

Dieses wird nun erst von den Haaren und dann von der Kleidung überlagert, was dazu führt, dass Zöpfe nicht über die Kleidung hinaus gehen können, aber auf der anderen Seite Mützen und Kopftücher möglich sind. Die Haare werden wieder vorher eingefärbt und die Kleidung bleibt derzeit unverändert.

---

<sup>83</sup>Programme, die aus den 3D Objekten das Bild generieren

<sup>84</sup>Bilddatei, die anhand der UV Koordinaten die Normalen verändert

## 6 „Built on solid Seas“ im Test

Auf die Entwicklung folgend war es nötig zu erfahren, wie das Spiel von der Zielgruppe angenommen wird, welche Aspekte gut sind und welche einer Verbesserung bedürfen.

### 6.1 Erste Befragung der Schüler\*innen

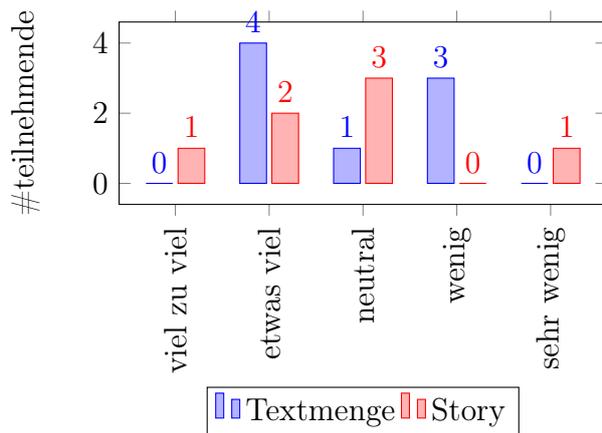
Die Befragung der Schüler\*innen stellte sich aufgrund der Osterferien als relativ kurzfristiges und schwieriges Unterfangen dar. Dennoch war es möglich, das Spiel rechtzeitig in eine spielbare Phase zu bringen und eine Umfrage zu beginnen. Anfangs stellten sich einige Schwierigkeiten heraus, da es zu einem betriebssystemweiten Ausfall der Sichtbarkeit des Cursors kam. Dieser Fehler ist mir aufgrund des Betriebssystem-Unterschieds zur Programmierphase nicht aufgefallen, was die ersten Bewertungen mit beeinflusste. Die Umfrage ergab einige wichtige Punkte. Bis auf eine Person, die sich stark an dem Bug gestört hat, kam bei allen Schüler\*innen die Charaktererstellung durchgehend gut an, die Art der Aufgabenstellung wurde eher als gut empfunden und das Spiel beginnt für viele eher gut bis mittelmäßig, steigert sich jedoch mit der Langzeitmotivation, was mitunter an der Komplexität des Spiels liegen kann. Auch die Menge an erklärendem Text wurde eher als zu wenig bewertet und die Story des Spiels scheint zu viel Text zu beinhalten. Dies stützt meine Hauptthese, dass Spielende dieser Zielgruppe sich nicht auf eine ausgeklügelte Story freuen, sondern sich viel mehr auf die gegebenen Mechaniken und Regeln stürzen, diese kurz und präzise formuliert haben wollen, und dann nicht von Story-Elementen aufgehalten werden wollen.

Es stellte sich sogar heraus dass die Spielenden es als Lernwerkzeug unabhängig von den schulisch gegebenen Lernwerkzeugen gerne hin und wieder nutzen würden, so sie denn die Möglichkeit dazu haben.

#### 6.1.1 Ergebnis

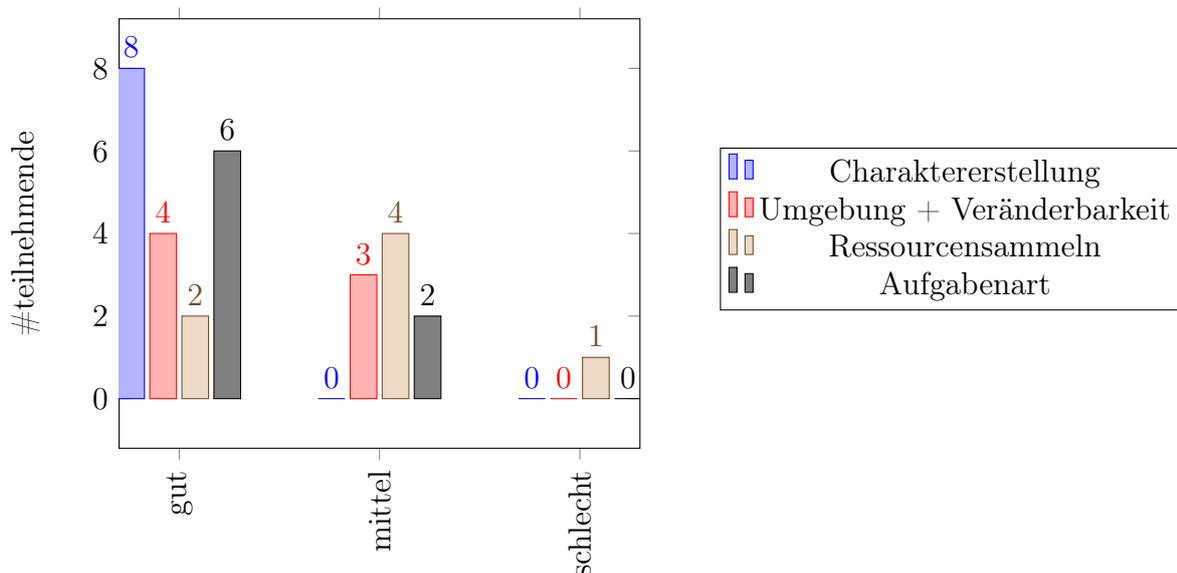
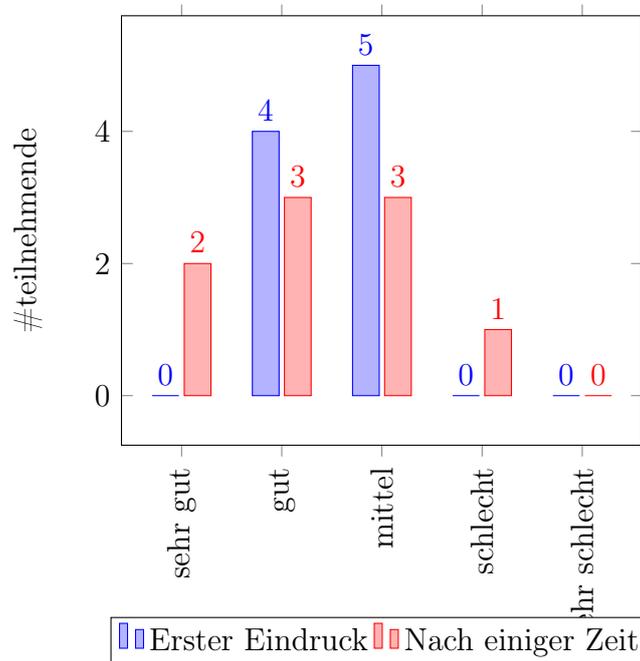
Die Spielenden haben das Spiel während einer Schulstunde in Zweiergruppen gespielt, und pro Gruppe einen Fragebogen ausgefüllt. Das Spiel begann bei den Spielenden relativ mittelmäßig Spaß zu machen, da anfangs Erklärungen fehlten, wie man an diesem Diagramm erkennen kann (vgl. mit den Umfrage-Ergebnissen im Anhang):

(neutral war in der ersten Umfrage zur Textmenge nicht wählbar, es wurde allerdings eindeutig von einer befragten Person zwischen den Feldern angekreuzt, was ich als neutral wertete), die Motivation steigerte sich jedoch sobald die Erklärungen dann da waren:



Besonders hervorzuheben ist die besondere Freude über die Charaktererstellung, die grundsätzlich positiv angenommen wurde.

Mein abschließendes Fazit zu dieser Umfrage ist also, dass das Spiel etwas weniger Text und etwas mehr Erklärung benötigt, es aber dennoch deutlich beliebter ist, als die echten Lernspiele (nach Lehrplan NRW), die den Schüler\*innen bisher bekannt waren. Es ist also auf einem guten Weg und könnte sich im Vergleich zu anderen Lernspielen und Unterrichtsmaterialien größerer Beliebtheit erfreuen.



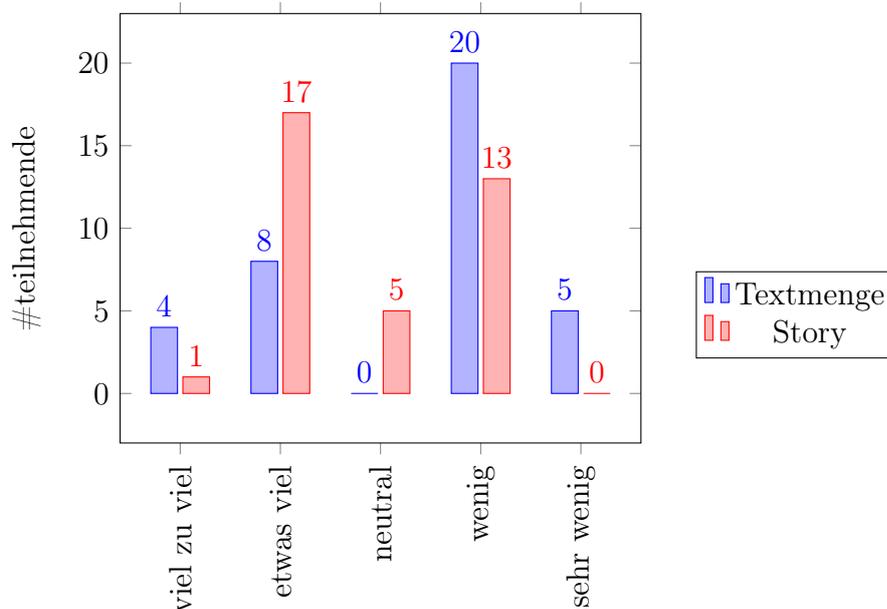
Viel mehr lässt sich aus diesen Daten nicht lesen, da die Umfrage als eine Umfrage mit langer Testphase zuhause geplant war, sie allerdings mit einer 30minütigen Testphase in der Schule durchgeführt wurde.

## 6.2 Zweite Befragung der Schüler\*innen

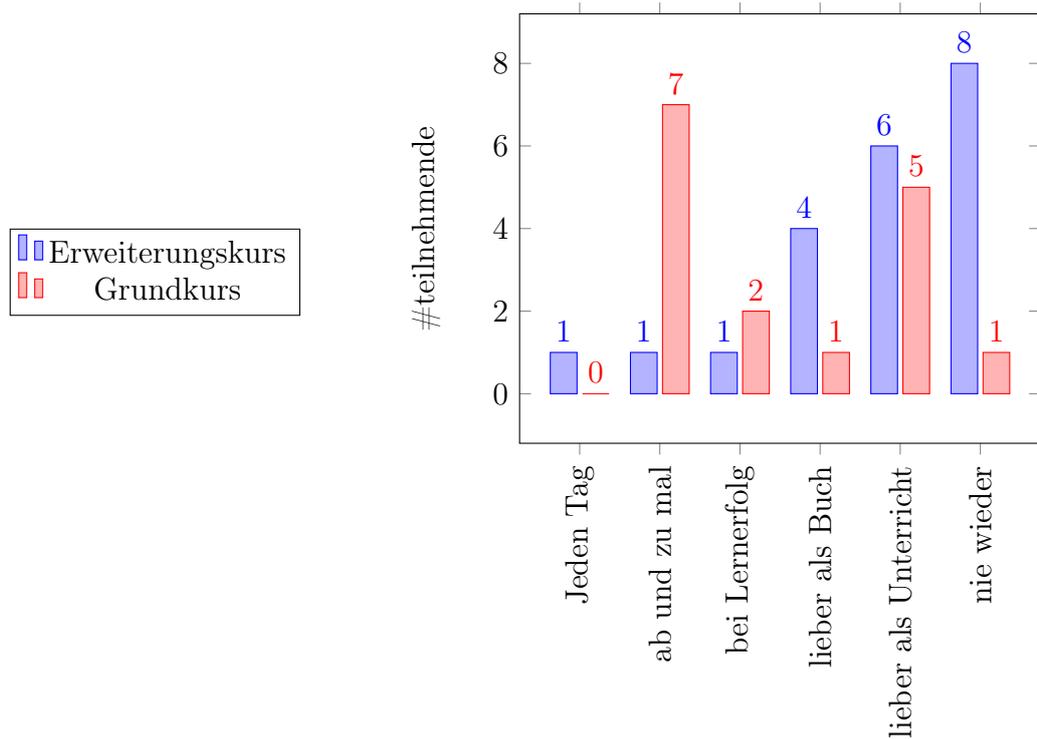
Die zweite Umfrage fand etwas später statt. Einige Fehler wurden in der Zwischenzeit behoben und einige Features ausgebaut. Auch in dieser Umfrage war geplant, eine längere Testphase der Spielenden an dem heimischen PC durchzuführen. Dies war jedoch aufgrund technischer Schwierigkeiten nicht möglich, und so fand eine zweite kurze Testphase in der Schule statt, die allerdings ein etwas neueres Produkt testete.

### 6.2.1 Ergebnis

Auch in dieser Befragung kam es wieder zu Fehlern des Programms, die allerdings deutlich geringer waren. Hier wurden aus einem Erweiterungskurs und einem Grundkurs Umfrageergebnisse zu den gleichen Fragen gesammelt. Immer noch scheinen Erklärungen zu fehlen, und der Textanteil scheint immer noch etwas viel zu sein (vgl. mit den Umfrageergebnissen im Anhang):

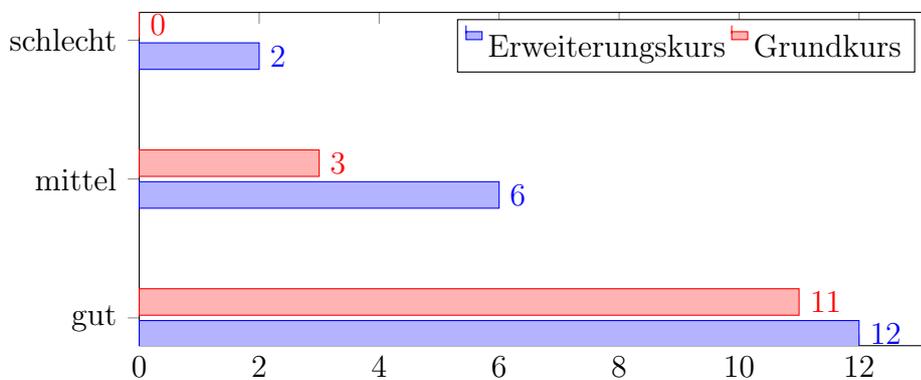


Die Kurz- und Mittelfristmotivation verhalten sich in dieser Umfrage etwa gleich, die Langfristmotivation scheint hingegen von dem Kurs stark abhängig zu sein, der Erweiterungskurs gab hier eine sehr negative, der Grundkurs jedoch eine sehr positive Bewertung, die Spielenden nannten, dass sie das Spiel so gern spielen wollen würden:



Doch was durch alle Kurse bisher als sehr positiv angesehen wird, und was als einziges wirklich von allen Spielenden erlebt wurde ist die Charaktererstellung, die sogar von 19 der 37 Spielenden explizit in dem Textfeld für positive Elemente des Spiels genannt wurde:

Charaktererstellung



Aus diesen Ergebnissen lese ich, dass die Spielenden mit den Elementen, die sie gesehen haben, durchaus zufrieden sind, der Einstieg aber noch nicht so flüssig verläuft. Einige Fehlerbehebungen und ein Tutorial-level sollten helfen, das Spiel besser zu verstehen, diese können jedoch nicht mehr in der Zeit der Bachelorarbeit realisiert werden.

### 6.3 Befragung der Lehrer\*innen

Im Anschluss führte ich ein Interview mit einer Lehrperson, die als Lehrer bereits 23 Jahre aktiv ist. Hier wollte ich in Erfahrung bringen

- wie einfach es ist, von der Euregio Gesamtschule Rheine auf andere Schulen in Deutschland zu schließen was die Aufgaben angeht,

- wie wichtig überhaupt der Inhalt im Vergleich zur Motivation bei solchen Spielen ist und wie wichtig stumpfes Wiederholen im Vergleich zu gezieltem Informed Teaching[5, S. 109] sein kann und im Anschluss an einen Spieltest
- ob der Funktionsumfang da ist, ob es verständlich war und welche Features noch benötigt werden.

### 6.3.1 Ergebnis

Als Ergebnis dieses Interviews lässt sich sagen, dass der Vergleich zwischen den Bundesländern und auch zwischen den Kursen nach Angabe dieser Lehrperson nicht so groß zu sein scheint, eine Austauschbarkeit der Aufgaben kann allerdings von entscheidendem Vorteil zu sein. Hier sollte jedoch die Möglichkeit bestehen, ein Aufgabenpaket online zu speichern, da nicht jede Schule eine Lehrperson hat, die die Muße und die Kenntnisse hat, eine solche Aufgabenmenge zu erstellen. Auch sollte stärker darauf geachtet werden, dass Schüler\*innen unterschiedliche Fähigkeiten haben und in der Aufgabenstellung teilweise mehr Hilfe benötigen.

Aufgabenwiederholungen haben nach Aussage des Lehrers in einer Phase vor der Prüfung große Wichtigkeit, müssen allerdings auch mit der Möglichkeit gestellt werden, sie zu überspringen.(vgl. Anhang)

## 7 Ergebnisse

Um das Thema abzurunden möchte ich erst auf die verschiedenen Frameworks eingehen, und darstellen, welche wichtigen Informationen sie geben können, um ein gutes Lernspiel zu erstellen, sodass es für das Szenario entsprechend funktioniert. Unterschiedliche Frameworks legen hier den Fokus auf verschiedene Elemente, zusammen erschaffen sie allerdings einen groben Überblick.

### 7.1 Frameworks

Die Frameworks sind eng miteinander verwoben, da sie alle versuchen, die gleichen Fragen zu beantworten. Beginnend mit dem ältesten, MDA[31], zeigt sich die Relevanz von Mechaniken, oftmals missverstanden als „Points, Badges and Leaderboards“, die jedoch deutlich mehr sein können und nach Bartles Spielenden-Typen [6] je nach Ästhetik verschiedene Arten von Spielenden-Typen motivieren und erreichen können. Darauf aufbauend zeigt sich mit DPE[41], dass dies nicht alles ist, was ein gutes Spiel ausmacht. Es gehört gerade bei Lernmaterialien ein ganz wichtiger Aspekt des Inhalts, der in P-III[11] besser behandelt wird, hinzu. Hier bedarf es einer funktionierenden interdisziplinären Zusammenarbeit, die nicht nur daraus besteht, um den Content ein Spiel zu erschaffen (wie Tim 7 oder Addy wirkt), sondern von Grund auf, spielendenbasiert, wie bei „the I’s have it“[5], ein Erlebnis für die Spielenden zu schreiben, das sie tief in die Welt eintauchen lässt. Hier ist es wichtig, dass, wenn ein eigener Charakter existiert, dieser einer ist, mit dem sich alle Spielenden identifizieren können, oder die Spielenden „durch die vierte Wand“<sup>85</sup> angesprochen werden, und somit sie selber zu dem Hauptcharakter werden. Die Story und die Charaktere müssen die Identifizierung mit den Charakteren stärken, und dafür

<sup>85</sup>„Die vierte Wand durchbrechen“ steht für „die Zuschauenden direkt ansprechen und involvieren“ und ist eine bekannte Technik in Theater, Film und Spiel

sorgen, dass die Spielenden von dem Spiel „gefesselt“ werden, ihre Emotion wie im Framework MDE[16, S. 7] überprüft wird, und sie sich zur Bearbeitung der Aufgaben im Flow-Channel[10] aufhalten. Diese Konzentration soll jedoch nicht in sinnfreier Spielerei verloren gehen, sondern wie in SGDA[26] immer dem Purpose<sup>86</sup> des Spiels dienen.

## 7.2 Vergleich mit Büchern

Büchern gelingt es ganz wunderbar, sich auf die Lernmaterialien zu beschränken, in jedem Punkt dem Zweck zu folgen, doch machen sie genau die Fehler, die ihre Autor\*innen auch in Videospielen zu machen scheinen: Sie haben kaum nennenswerte Mechaniken. Einige Lernbücher haben Story-Elemente und Charaktere, oder bieten kleine optionale Punktesysteme, aber dennoch sind die Motivations-Elemente (vgl. [11, p. 84]) und das informed Teaching (vgl. [5, p. 109]) mit den dynamischen Reaktionen auf Spielenden-Aktionen die Hauptvorteile eines Computerspiels.

## 7.3 Vergleich mit anderen Lernspielen

Andere Lernspiele, wie schon vorher erwähnt, scheinen zentrale Fehler zu machen: Sie beschränken sich sehr schnell darauf, entweder hauptsächlich Mechaniken zu liefern und als Spiel zu funktionieren, oder sich auf die Lehrpläne zu fokussieren und somit als Lernsoftware zu agieren. Die Herausforderung, vor denen Lernspiele stehen, ist offenbar, gleichermaßen zu motivieren, wie auch Aufgaben nach Kernlehrplänen der Bundesländer zu fordern. Auch schaffen es Lernspiele, die sich an Jahrgängen orientieren, weniger, sich auf dem Markt zu behaupten, da zwar möglicherweise in dieser Zeit gelernt wird, kurz darauf jedoch das Produkt verkauft wird, obwohl die gezielte Wiederholung alter Elemente bei Bedarf ein wichtiger Vorteil eines Videospieles sein kann.

## 7.4 Zusammenfassung

Abschließend lässt sich sagen, dass es einen großen Bedarf an guten Lernspielen gibt, da Lernspiele im Gegensatz zu Schulbüchern individuell motivieren können, die meisten Lernspiele es jedoch nicht schaffen, diesen Punkt zu finden, in dem der Lehrplan nicht vernachlässigt und zum Lernen trotzdem ausreichend motiviert wird. Mein Konzept eines modularen open-source Lernspiels ist ein Ansatz, diese Lücke zwischen Spiel und Lernen zu schließen, und somit ein Versuch einer Gamification der Klassen 9 und 10 Mathematik.

## 7.5 Ausblick

Mit Blick auf die Zukunft lässt sich sagen, dass „Built on solid Seas“ mehr Elemente eines klassischen Spiels übernehmen sollte, um den Einstieg einfacher zu gestalten. Des Weiteren sollte die Bedienung sowohl für Spielende ohne Erfahrung mit Computerspielen als auch für jene Spielende mit mehr Mathematikfähigkeiten attraktiver gestaltet werden. Auch wäre eine online-Anbindung mit Klassenzimmern und schuleigenen Aufgabenpaketen wichtig. Dennoch zeigt sich, dass der Ansatz von „Built on solid Seas“ funktionieren kann und Lernspiele viel mehr auf die Implementierung von Mechaniken achten sollten. Im Vergleich mit Büchern wird es über kurz oder lang dazu kommen, dass Lernspiele wie

---

<sup>86</sup>Zweck

„Built on solid Seas“ mit Motivationsmechaniken (vgl. [11, p. 84]) und „Informed Teaching“ [5, p. 109] besser die Schüler\*innen erreichen können, und ihnen die Thematiken schneller näher bringen können.

## Literatur

- [1] phase 6 GmbH. *phase 6*. 2007.
- [2] FEO Media AB. *Quizduell*. 2013.
- [3] Stöer Media Brands AG. *SpielAffe.de*. 2004.
- [4] Markus Hohenwarter et al. *GeoGebra*. 2006.
- [5] Leonard A. Annetta. „The “I’s” Have It: A Framework for Serious Educational Game Design“. In: *Review of General Psychology* (2010).
- [6] Richard A. Bartle. *Hearts, Clubs, Diamonds, Spades: Players who suit MUDs*. 1996. URL: <http://mud.co.uk/richard/hcdfs.htm>.
- [7] Bethesda. *Fallout 4*. 2015.
- [8] Breakthrough. *ICED - I Can End Deportation*. 2008.
- [9] EA Canada. *FIFA*. 1993.
- [10] M. Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Perennial Modern Classics. Harper & Row, 1990. ISBN: 9780060162535. URL: <https://books.google.de/books?id=V9KrQgAACAAJ>.
- [11] Vero Vanden Abeele Bob De Schutter Luc Geurts und Stef Desmet Jeroen Wauters Jelle Husson Lieven Van den Audenaeren Frederik Van Broeckhoven Jan-Henk Anema und David Geerts. „P-III: A Player-Centered, Iterative, Interdisciplinary and Integrated Framework for Serious Game Design and Development“. In: *Communications in Computer and Information Science* (2012).
- [12] Blizzard Entertainment. *World of Warcraft*. 2005.
- [13] *Essstörungen*. TRIAS, 2004.
- [14] *Essstörungen*. Hogrefe-Verlag, 2004.
- [15] Riot Games. *League of Legends*. 2009.
- [16] Leyland Pitt Karen Robson Kirk Plangger Jan H. Kietzmann und Ian McCarthy. „Is it all a game? Understanding the principles of gamification“. In: *Business Horizons* (2015).
- [17] Amy Jo Kim. *Creating Compelling Experiences*. 2011. URL: <https://www.youtube.com/watch?v=F4YP-hGZTuA>.
- [18] Amy Jo Kim. *Putting the Fun in Functional Applying Game Mechanics to Functional Software*. 2009. URL: <https://www.youtube.com/watch?v=ihUt-163gZI>.
- [19] Heureka Klett. *Tim 7 und die Erinnerungen eines Vergessenen*. 2000.
- [20] Ministerium für Schule Jugend und Kinder des Landes Nordrhein-Westfalen. „Mathematik“. In: *Kernlehrplan für die Gesamtschule – Sekundarstufe I in Nordrhein-Westfalen*. 2004.
- [21] Juan Linietsky und Ariel Manzur. *GODOT Engine*. 2008-.
- [22] little loud. *SWEATSHOP*. 2011.
- [23] Maxis. *The Sims*. 2015.
- [24] Ludger Meier. *Euregio Gesamtschule Rheine*. 1989-.

- [25] Mojang. *Minecraft*. 2015.
- [26] Konstantin Mitgutsch und Narda Alvarado. „Purposeful by Design? A Serious Game Design Assessment Framework“. In: *Association for Computing Machinery* (2012).
- [27] Nintendo. *Dr. Kawashimas Gehirn-Jogging - Wie fit ist Ihr Gehirn?* 2006.
- [28] Bildungsministerium NRW. *Zuordnung der Fächer zu den Zulassungswegen*. 2016. URL: <https://www.schulministerium.nrw.de/docs/Schulsystem/Medien/Lernmittel/Kontext/Zuordnung/index.html>.
- [29] „Performing feminisms: feminist critical theory and theatre“. In: Johns Hopkins University Press, 1990. Kap. Performative Acts and Gender Constitution: An Essay in Phenomenology and Feminist Theory.
- [30] Iris Bellinghausen und Ralph Möllers. *Terzio Verlag*. 1997-2011.
- [31] Robin Hunicke Marc LeBlanc und Robert Zubek. *MDA: A Formal Approach to Game Design and Game Research*. Techn. Ber. Department of electrical engineering und computer science, Northwestern University, 2004.
- [32] Guido van Rossum. *Python*. 1989.
- [33] FAKT Software. *Crazy Machines II*. 2007.
- [34] id Software. *Wolfenstein 3D*. 1992.
- [35] Bjarne Stroustrup. *C++*. 2011.
- [36] Terzio. *S.O.S. Tiefsee*. 2001.
- [37] Valve. *Counter-Strike*. 2000.
- [38] Coktel Vision. *ADDY Deutsch Kl. 5+6*. 2002.
- [39] Infinity Ward. *Call of Duty®*. 2003.
- [40] Wargaming.net. *World of Tanks*. 2010.
- [41] Brian M. Winn. „The Design, Play, and Experience Framework“. In: *Communication Arts & Sciences Building* (2009).
- [42] *Zahlen und Größen 9*. Cornelsen, 2013.

# Appendices

<sup>87</sup> Umfrage vor der Entwicklung

---

<sup>87</sup> Anhang

## Umfrage zu Computerspielen

Sehr geehrte Damen und Herren,

vielen Dank dafür, dass Sie sich bereit erklärt haben an unserer Umfrage teilzunehmen. Der vorliegende Fragebogen besteht aus maximal 8 Fragen und Ihre Teilnahme wird ca. 5 Minuten dauern. Im Vorfeld der eigentlichen Befragung möchten wir Ihnen zunächst grundlegende Informationen über den verfolgten Zweck der Umfrage geben.

An der Universität Siegen wird in der naturwissenschaftlich technischen Fakultät im Bereich wissensbasierte Systeme, bei Prof. Fathi und Marjan Khobreh, unter anderem im Bereich der Gamification, also einem Bereich des spielbar Machens von ernsten Themen, geforscht, wie sich beispielsweise das Lernen für die Schule mit Videospiele unterstützen lässt. Hierfür ist es nötig, dass ein ausreichend breiter Umfang an Befragten ihre Meinungen schildern, um bei zukünftigen Videospiele in diesem Bereich potentiell stärker motivierende Produkte erstellen zu können. Die zunehmende Digitalisierung des Alltags, und die stärkere Gamification anderer Bereiche des Lebens, die immer stärker ihre Aufmerksamkeit auf sich ziehen, begründet, warum gerade jetzt die Gamification und die dafür nötige Forschung vorangetrieben werden muss. Das erlangte Wissen soll im folgenden helfen zu erkennen welche Computerspiele der Zielgruppe gefallen, und welche ihnen nicht gefallen.

Die nachfolgende Umfrage dient aus dem Kontext einer Bachelorarbeit zu diesem Thema dazu, einen Eindruck zu bekommen, welche Computerspiele und/oder Lernspiele Schüler\*innen gerne spielen. Auf diesen Ergebnissen soll daraufhin ein Computerspiel entstehen.

## Umfrage zu Computerspielen

Spielst du Computerspiele?

- Ja
- Nein

Wenn Ja:

Welche Computerspiele spielst du?

---

---

---

Welche Genres spielst du am liebsten?

- RPG (Rollenspiele)
- Aufbauspiele
- Strategiespiele
- andere: \_\_\_\_\_
- was läberst du?

Hast du schonmal ein Lernspiel gespielt?

- Ja
- Nein

Wenn ja, welche Lernspiele hast du gespielt?

- \_\_\_\_\_  Hat Spaß gemacht  War ertragbar  War langweilig  
\_\_\_\_\_  Hat Spaß gemacht  War ertragbar  War langweilig  
\_\_\_\_\_  Hat Spaß gemacht  War ertragbar  War langweilig

Glaubst du dass dir ein Lernspiel je Spaß machen kann?

- Ja
- Nein

Willst du das Spiel probespielen während ich es entwickle?

- Ja (Emailadresse: \_\_\_\_\_ )
- Nein

Dein Kommentar zu der Aktion:

---

---

---

---

Umfrageergebnisse vor Entwicklung

Sheet1

Spielt Computerspiele	Ja	Nein	Spiel1	Spiel2	Spiel3
	x		LOL	Minecraft	CS
	x		Minecraft		
		x			
	x		Minecraft		
	x		COD	LOL	
		x			
		x			
		x			
	x		Fifa	CS	Minecraft
		x	Spielaffe		
		x	Spielaffe		
	x		COD	Assassins Cre	Fifa
	x		COD	CS	WOW
		x			
		x			
		x			
		x			
		x			
	x		CS	COD	Clash of Clans
	x		Destiny	WOW	Castle Clash
	x		LOL	Minecraft	LKW-Simulator
	x		WOW	COD	Aion
	x		CS	Fallout	WOT
	x		Fifa	MW	
		x			
		x			
	x		Minecraft	F1	
	x		LOL	Fifa	
	x		Minecraft		
	x		Aion	COD	
	x		CS	WOT	
	x		Minecraft	WOT	CS
	x		WOT	Supertux	Panzer
	x		Minecraft	Spielaffe	Supertux
	x		COD	Minecraft	Southpark & GTA
		x			
		x			

Am häufigsten gespielte Spiele: Minecraft COD CS LOL Fifa  
Ihre Häufigkeit 10 8 7 4 4

Am häufigsten gespielte Lernspiele: Vokabelspiel Phase6 Gehirnjogging Crazy Machine Tim7  
Ihre Häufigkeit 4 3 3 3 3  
Positive Bewertung 1 1 3 1 0  
Neutrale Bewertung 2 1 0 0 3  
Negative Bewertung 1 1 0 0 0

Sheet1

Genre1	Genre2	Genre3	Spielt Lernspiele		Spiel1
			Ja	Nein	
RPG	Aufbau			x	
Aufbau				x	
Aufbau			x		Mathespiel
RPG	Strategie	Shooter	x		Mathepirat
Aufbau			-	-	
Aufbau			-	-	
Aufbau			-	-	
				x	
				x	
			x		Clever.de
			-	-	
RPG	Strategie			x	
RPG	Strategie	Shooter		x	
RPG			x		
			x		Tim7
			x		Tim7
			x		Tim7
			x		
				x	Babbel
Shooter				x	
Shooter			x		Crazy Machine
RPG	Strategie	Simulation	x		Skoyo
RPG	Shooter		x		Crazy Machine
RPG	Strategie	Shooter	x		
Shooter				x	
			x		Gehirnjogging
			x		Gehirnjogging
			x		Phase6
Strategie			x		Gehirnjogging
			x		Phase6
RPG	Strategie	Shooter	x		Crazy Machine
Strategie			x		Vokabelspiel
Strategie	Minecraft		x		Vokabelspiel
			x		Vokabelspiel
			x		Phase6
Aufbau	Strategie	Shooter	x		Quizduell
			x		knobeln
			x		Förder-Pyrami
WOT	Spielaffe	WOW			
	4	3	3		

Sheet1

Spaß

Spiel2

Spaß

Spiel3

Spaß

Glaubt an Spaß in Lernspielen

0  
0

-

0  
0  
0

+

+

0

es

+

+

Vokabelspiel +  
0

+

-

es

0

-

0

+

+

+

0



## Schüler\*innen-Tests Umfrageergebnisse vor Entwicklung

## Umfrage zum Spiel (Qualitätssicherung)

Sehr geehrte Damen und Herren,

vielen Dank dafür, dass Sie sich bereit erklärt haben an unserer Umfrage teilzunehmen. Der vorliegende Fragebogen besteht aus maximal 20 Fragen und Ihre Teilnahme wird ca. 20 Minuten dauern. Im Vorfeld der eigentlichen Befragung möchten wir Ihnen zunächst grundlegende Informationen über den verfolgten Zweck der Umfrage geben.

An der Universität Siegen wird in der naturwissenschaftlich technischen Fakultät im Bereich wissensbasierte Systeme, bei Prof. Fathi und Marjan Khobreh, unter anderem im Bereich der Gamification, also einem Bereich des spielbar Machens von ernsten Themen, geforscht, wie sich beispielsweise das Lernen für die Schule mit Videospiele unterstützen lässt. Hierfür ist es nötig, dass ein ausreichend breiter Umfang an Befragten ihre Erfahrungen anhand des Prototypen schildern, um bei zukünftigen Videospiele in diesem Bereich potentiell stärker motivierende Produkte erstellen zu können.

Die zunehmende Digitalisierung des Alltags, und die stärkere Gamification anderer Bereiche des Lebens, die immer stärker ihre Aufmerksamkeit auf sich ziehen, begründet, warum gerade jetzt die Gamification und die dafür nötige Forschung vorangetrieben werden muss.

Das erlangte Wissen soll im folgenden helfen zu erkennen welche Aspekte dieses Spiels nun stärker, oder weniger stark motivieren.

Die nachfolgende Umfrage dient aus dem Kontext einer Bachelorarbeit zu diesem Thema dazu, bereits aus diesem Projekt gewonnene Erkenntnisse zu validieren und hilfreiche Anregungen aus der Praxis zu erhalten. Der Fragebogen ist dabei aus zwei Teilbereichen aufgebaut; der eine ist vor, der andere nach dem Test auszufüllen. Neben Fragen zur kurz-, mittel- und längerfristigen Motivation und der Gesamteinschätzung der Testenden ist auch wichtig zu erfassen, wie das Spiel auf bisher unterschiedlich benotete Menschen wirkt, und wie lange die Motivation anhält, da man im Optimalfall das Zeitgefühl vergisst

### Bevor du spielst:

Das Spiel ist in einer sehr frühen Testphase, es können Fehler auftreten. Wenn dir welche auffallen, schreib mir eine Email an [gerrit@familiegrosskopf.de](mailto:gerrit@familiegrosskopf.de), schreib mich auf

<http://gerrit.familiegrosskopf.de> an, oder schreib mir auf Telegram ( den Link zu der Gruppe findest du auf <http://gerrit.familiegrosskopf.de> im internen Bereich mit dem Passwort „euregio“)

Und hol dir gern einen Taschenrechner, du wirst ihn brauchen.

### Nun noch ein paar Fragen bevor du anfängst:

Schau auf die Uhr, schreib dir die Uhrzeit auf, das ist später wichtig

Wie gut glaubt die Schule, dass du in Mathe bist:

- E-kurs
- G-kurs

Wie gut bist du in Mathe? (deine Einschätzung)

- Sehr gut
- Gut
- Mittelmäßig
- Schlecht
- Oh nein, Mathe O.o

Glaubst du, dass dir ein Lernspiel Spaß machen kann?

- Ja
- Vielleicht Ja
- Wird sich zeigen
- Eher nicht
- Nein

Anmerkungen:

---

---

---

**Nach dem Spiel:**

Wie lange hast du gespielt?

\_\_\_\_\_ (dafür war die Uhrzeit)

Wie gut war dein erster Eindruck?

- Sehr gut
- Gut
- Mittelmäßig
- Schlecht
- Unglaublich schlecht

Wie gut hat sich das Spiel gehalten?

- Sehr gut
- Gut
- Mittelmäßig
- Schlecht
- Unglaublich schlecht

Wie gern würdest du dieses Spiel spielen?

- In meiner Freizeit, jeden Tag, mach mehr davon
- Ab und zu mal, ist ganz lustig
- Wenn ich was davon lernen kann, hin und wieder mal
- Lieber, als Aufgaben aus dem Buch zu machen
- Lieber, als Unterricht
- Am liebsten nie wieder

Was hat dir gefallen?

---

---

---

Was hat dir nicht gefallen?

---

---

---

War alles ausreichend erklärt?

- Viel zu viel Text überall!
- War etwas viel
- an einigen Punkten haben Erklärungen gefehlt:

---

---

an vielen Punkten haben Erklärungen gefehlt!:

---

---

anders: \_\_\_\_\_

Wie hat dir die Charaktererstellung gefallen?

- Gut
- Mittelmäßig
- Schlecht
- anders: \_\_\_\_\_

Wie hat dir die Umgebung und die Möglichkeit diese zu ändern gefallen?

- Gut
- Mittelmäßig
- Schlecht
- nicht bemerkt
- anders: \_\_\_\_\_

War die Story so in Ordnung?

- Viel zu viel Text!
- War etwas viel
- War ganz ok
- hätte ruhig mehr sein können
- die Story war noch sehr unklar
- anders: \_\_\_\_\_

Wie hat dir die Art, Ressourcen zu sammeln, gefallen?

- Gut
- Mittelmäßig
- Schlecht
- nicht bemerkt
- anders: \_\_\_\_\_

Wie hat dir die Art der Aufgaben gefallen?

- Gut
- Mittelmäßig
- Schlecht
- anders: \_\_\_\_\_

Glaubst du, dass du mit so einem Spiel etwas lernen könntest oder wolltest?

- Ja, auf jeden Fall
- Ja schon ein wenig
- Vielleicht
- Eher nicht
- gar nicht

Wenn du magst, speichere dein Spiel ab und schick mir den „Saves“-Ordner über einen der oben genannten Wege (gerne als Archiv komprimiert).

Mit freundlichen Grüßen,  
Gerrit Großkopf

## Umfrageergebnisse Alphatest













## Umfrageergebnisse Betatest

Sheet1

Nummer	Kurs	Selbsteinschätzung	Existiert Spaß in Lernspielen?
1	E-Kurs	sehr gut	wird sich zeigen
2	E-Kurs	mittelmäßig	vielleicht ja
3	E-Kurs	mittelmäßig	wird sich zeigen
4	G-Kurs	gut	wird sich zeigen
5	G-Kurs	gut	ja
6	E-Kurs	mittelmäßig	wird sich zeigen
7	G-Kurs	mittelmäßig	vielleicht ja
8	E-Kurs	mittelmäßig	wird sich zeigen
9	E-Kurs	mittelmäßig	wird sich zeigen
10	E-Kurs	gut	eher nicht
11	E-Kurs	mittelmäßig	wird sich zeigen
12	E-Kurs	gut	wird sich zeigen
13	E-Kurs	mittelmäßig	vielleicht ja
14	G-Kurs	schlecht	ja
15	G-Kurs	mittelmäßig	wird sich zeigen
16	E-Kurs	mittelmäßig	wird sich zeigen
17	G-Kurs	mittelmäßig	eher nicht
18	G-Kurs	mittelmäßig	wird sich zeigen
19	E-Kurs	gut	wird sich zeigen
20	G-Kurs	mittelmäßig	vielleicht ja
21	G-Kurs	mittelmäßig	wird sich zeigen
22	G-Kurs	schlecht	eher nicht
23	G-Kurs	gut	vielleicht ja
24	E-Kurs	sehr gut	wird sich zeigen
25	E-Kurs	mittelmäßig	wird sich zeigen
26	G-Kurs	mittelmäßig	wird sich zeigen
27	E-Kurs	mittelmäßig	wird sich zeigen
28	G-Kurs	mittelmäßig	wird sich zeigen
29	E-Kurs	gut	eher nicht
30	E-Kurs	sehr gut	wird sich zeigen
31	E-Kurs	mittelmäßig	vielleicht ja
32	G-Kurs	mittelmäßig	ja
33	E-Kurs	mittelmäßig	vielleicht ja
34	E-Kurs	mittelmäßig	wird sich zeigen
35	E-Kurs	mittelmäßig	ja
36	G-Kurs	oh nein, Mathe	ja
37	G-Kurs	mittelmäßig	vielleicht ja

Sheet1

erster Eindruck	Mittelfristmotivartion
mittelmäßig	unglaublich schlecht
mittelmäßig	mittelmäßig
mittelmäßig	mittelmäßig
gut	mittelmäßig
gut	mittelmäßig
schlecht	mittelmäßig
mittelmäßig	gut
mittelmäßig	mittelmäßig
schlecht	mittelmäßig
mittelmäßig	mittelmäßig
mittelmäßig	schlecht
mittelmäßig	mittelmäßig
mittelmäßig	schlecht
mittelmäßig	mittelmäßig
gut	gut
schlecht	mittelmäßig
mittelmäßig	schlecht
mittelmäßig	mittelmäßig
mittelmäßig	mittelmäßig
schlecht	mittelmäßig
schlecht	schlecht
sehr gut	mittelmäßig
mittelmäßig	schlecht
schlecht	schlecht
schlecht	schlecht
mittelmäßig	schlecht
schlecht	schlecht
mittelmäßig	mittelmäßig
schlecht	schlecht
schlecht	schlecht
gut	gut
gut	mittelmäßig
gut	mittelmäßig
unglaublich schlecht	mittelmäßig
unglaublich schlecht	schlecht
mittelmäßig	mittelmäßig
gut	mittelmäßig

Sheet1

Dauermotivation	positives Feedback
lieber, als Unterricht	
lieber, als Aufgaben aus dem Buch zu machen	Charaktererstellung
lieber, als Aufgaben aus dem Buch zu machen	Charaktererstellung
ab und zu mal, ist ganz lustig	Charaktererstellung
ab und zu mal, ist ganz lustig	Questmechanik
am liebsten nie wieder	
ab und zu mal, ist ganz lustig	Charaktererstellung
am liebsten nie wieder	Charaktererstellung
am liebsten nie wieder	
lieber, als Aufgaben aus dem Buch zu machen	Charaktererstellung
lieber, als Aufgaben aus dem Buch zu machen	Umgebung
Wenn ich was davon lernen kann, hin und wieder m	Charaktererstellung
am liebsten nie wieder	Charaktererstellung
ab und zu mal, ist ganz lustig	3D Welt
ab und zu mal, ist ganz lustig	Charaktererstellung
am liebsten nie wieder	
lieber, als Unterricht	Charaktererstellung
lieber, als Unterricht	
am liebsten nie wieder	
Wenn ich was davon lernen kann, hin und wieder m	Charaktererstellung
lieber, als Unterricht	Questmechaniken
ab und zu mal, ist ganz lustig	Charaktererstellung, Que
am liebsten nie wieder	
lieber, als Unterricht	Charaktererstellung
lieber, als Unterricht	Charaktererstellung
lieber, als Unterricht	
lieber, als Unterricht	
lieber, als Aufgaben aus dem Buch zu machen	Idee
am liebsten nie wieder	
lieber, als Unterricht	
in meiner Freizeit, jeden Tag, mach mehr davon	Charaktererstellung
ab und zu mal, ist ganz lustig	Charaktererstellung
ab und zu mal, ist ganz lustig	Charaktererstellung
lieber, als Unterricht	
am liebsten nie wieder	
lieber, als Unterricht	Charaktererstellung
Wenn ich was davon lernen kann, hin und wieder m	Charaktererstellung

Sheet1

negatives Feedback	Menge an Erklärungen
Aufgabenbeschreibung	an vielen Punkten hat Erklärung gefehlt
Steuerung	an einigen Punkten hat Erklärung gefehlt
Steuerung	war etwas Viel
Kammersteuerung	war etwas Viel
Erklärungen	an vielen Punkten hat Erklärung gefehlt
	an einigen Punkten hat Erklärung gefehlt
	an einigen Punkten hat Erklärung gefehlt
Aufgabenstellungen	war etwas Viel
Erklärungen, Grafik, Bugs	an einigen Punkten hat Erklärung gefehlt
Umgebung	an einigen Punkten hat Erklärung gefehlt
Bugs	an einigen Punkten hat Erklärung gefehlt
Grafik, Bugs	war etwas Viel
Aufgabenstellung	an einigen Punkten hat Erklärung gefehlt
Erklärungen	an einigen Punkten hat Erklärung gefehlt
Erklärungen	an einigen Punkten hat Erklärung gefehlt
Aufgabenstellung	an einigen Punkten hat Erklärung gefehlt
Bugs	an einigen Punkten hat Erklärung gefehlt
Grafik, zu viel Text, zu wenig Erklärungen	an einigen Punkten hat Erklärung gefehlt
Bugs	an einigen Punkten hat Erklärung gefehlt
begrenzte Welt	an einigen Punkten hat Erklärung gefehlt
Steuerung und Aufgaben	viel zu viel Text überall
Steuerung und Bugs	an vielen Punkten hat Erklärung gefehlt
Bugs, Erklärungen	war etwas Viel
Grafik und Bugs	war etwas Viel
Grafik und Bugs	an einigen Punkten hat Erklärung gefehlt
Erklärungen	an einigen Punkten hat Erklärung gefehlt
Erklärungen	an vielen Punkten hat Erklärung gefehlt
Erklärungen	an einigen Punkten hat Erklärung gefehlt
Schlechte Erklärung	war etwas Viel
Erklärung	an einigen Punkten hat Erklärung gefehlt
Bugs	war etwas Viel
Steuerung	viel zu viel Text überall
Steuerung	viel zu viel Text überall
Grafik, Figur, Erklärungen	an einigen Punkten hat Erklärung gefehlt
Figur, Grafik, Erklärungen	an einigen Punkten hat Erklärung gefehlt
Erklärung, begrenzte Welt	viel zu viel Text überall
Steuerung	an vielen Punkten hat Erklärung gefehlt

Sheet1

Charaktererstellung	Umgebung	Storymenge
anders	schlecht	die story war noch sehr unklar
mittelmäßig	mittelmäßig	die story war noch sehr unklar
gut	mittelmäßig	war ganz ok
gut	mittelmäßig	war ganz ok
gut	gut	hätte ruhig mehr sein können
mittelmäßig	schlecht	die story war noch sehr unklar
mittelmäßig	nicht bemerkt	die story war noch sehr unklar
gut	mittelmäßig	die story war noch sehr unklar
mittelmäßig	schlecht	die story war noch sehr unklar
gut	mittelmäßig	war ganz ok
mittelmäßig	anders	die story war noch sehr unklar
gut	mittelmäßig	war ganz ok
gut	schlecht	war ganz ok
gut	gut	war ganz ok
gut	mittelmäßig	die story war noch sehr unklar
gut	gut	war etwas viel
gut	gut	war ganz ok
gut	mittelmäßig	die story war noch sehr unklar
mittelmäßig	mittelmäßig	war ganz ok
mittelmäßig	anders	hätte ruhig mehr sein können
anders	schlecht	die story war noch sehr unklar
gut	gut	war ganz ok
mittelmäßig	nicht bemerkt	war ganz ok
gut	mittelmäßig	war ganz ok
gut	schlecht	hätte ruhig mehr sein können
gut	mittelmäßig	hätte ruhig mehr sein können
gut	mittelmäßig	war ganz ok
gut	mittelmäßig	hätte ruhig mehr sein können
mittelmäßig	mittelmäßig	war ganz ok
gut	mittelmäßig	die story war noch sehr unklar
gut	mittelmäßig	war ganz ok
gut	schlecht	die story war noch sehr unklar
schlecht	nicht bemerkt	war ganz ok
schlecht	schlecht	die story war noch sehr unklar
gut	anders	war ganz ok
gut	mittelmäßig	war ganz ok

Sheet1

Ressourcen sammeln	Aufgaben	Lernerfolg
nicht bemerkt	anders	gar nicht
mittelmäßig	mittelmäßig	ja, schon ein wenig
mittelmäßig	gut	ja auf jeden Fall
mittelmäßig	gut	ja, schon ein wenig
mittelmäßig	gut	ja auf jeden Fall
schlecht	anders	gar nicht
gut	gut	ja, schon ein wenig
mittelmäßig	mittelmäßig	eher nicht
schlecht	anders	gar nicht
mittelmäßig	mittelmäßig	eher nicht
anders	anders	vielleicht
mittelmäßig	mittelmäßig	vielleicht
mittelmäßig	schlecht	gar nicht
schlecht	mittelmäßig	eher nicht
gut	mittelmäßig	ja auf jeden Fall
mittelmäßig	mittelmäßig	eher nicht
gut	mittelmäßig	ja, schon ein wenig
schlecht	mittelmäßig	eher nicht
nicht bemerkt	mittelmäßig	eher nicht
anders	mittelmäßig	vielleicht
mittelmäßig	schlecht	eher nicht
anders	gut	vielleicht
mittelmäßig	mittelmäßig	eher nicht
mittelmäßig	mittelmäßig	ja, schon ein wenig
mittelmäßig	anders	ja, schon ein wenig
anders	schlecht	vielleicht
schlecht	mittelmäßig	vielleicht
mittelmäßig	mittelmäßig	ja, schon ein wenig
nicht bemerkt	schlecht	eher nicht
anders	mittelmäßig	gar nicht
gut	mittelmäßig	ja auf jeden Fall
mittelmäßig	schlecht	vielleicht
nicht bemerkt	anders	eher nicht
schlecht	schlecht	vielleicht
gut	gut	eher nicht
schlecht	schlecht	vielleicht

Interview mit Lehrer

## **Interview zu der Einsetzbarkeit von „Built on solid seas“ in Schulen.**

In diesem Interview mit Robert Großkopf, einem Lehrer mit 23 Jahren Berufserfahrung. Derzeit lehrt Herr Großkopf an der Euregio Gesamtschule Rheine unter anderem als Mathematiklehrer eines Grundkurses, der in die Tests involviert war:

### **Wie stark unterscheiden sich die Lehrpläne und ist es sinnvoll, zu versuchen, alle Bedingungen aller Lehrpläne in einem Spiel einzubauen?**

*Die Grundlagen sind in allen Bundesländern und auch allen Schultypen gleich. Die Bücher haben inzwischen durchweg Differenzierungsaufgaben und auch die Inhalte zwischen Kursen unterscheiden sich nicht allzu sehr. Deswegen ist ein Spiel, das zudem in den Aufgaben noch an der Schule selbst gestaltet werden kann, kein problem.*

### **Wie wichtig ist Inhalt bei lernbegleitenden Spielen?**

*Je höher die Motivation für eine Aufgabe ist, desto stärker ist das Engagement der SchülerInnen. Wiederholende Inhalte ("Päckchenaufgaben") haben in Übungsphasen natürlich ihren Platz. Solche Aufgaben sollten aber auch gegebenenfalls übersprungen werden können.*

### **Wie wichtig ist es, dynamisch auf die Schwächen und Stärken der Spielenden zu reagieren? Sind mehr Aufgaben sinnvoll oder gezielt erklärende Aufgaben?**

*Je größer der Sprachanteil in einer Aufgabe ist, desto eher haben schwächere SchülerInnen Probleme, diese Aufgabe mit mathematischen Mitteln zu bearbeiten. Schwächere SchülerInnen sollten mehr ähnliche Aufgaben gestellt bekommen und bei Textaufgaben ggf. eine erklärende Hilfe vorfinden. Stärkere SchülerInnen können dagegen aus Texten auf die entsprechenden Aufgaben schließen.*

### **- Spieltest**

### **Lassen sich mit dem gegebenen Spiel alle Geometrieaufgaben Klasse 9/10 bearbeiten?**

*Das müsste problemlos möglich sein. Wichtig wären Beispielaufgaben, die dann von den KollegInnen weiter entwickelt werden können.*

### **War die Bedienung verständlich?**

*Solange Beispielaufgaben vorliegen ist eine Änderung erst einmal unproblematisch. Die Erstellung von Figuren und Aufgaben ist so gestaltet, dass mit etwas Kenntnis von Geometrieprogrammen keine Probleme auftauchen. Der Button "Pause" erschließt sich nicht sofort. Hier wären zusätzliche Hinweise nötig (Hilfefunktion?).*

*Für die Erstellung von Dialogen müsste neben einer mit Aufgaben und Dialogen bereits ausgestalteten Version eine Version notwendig, die die Grundstruktur der Dialoge erklärt. Dafür wäre es vielleicht sinnvoll, nur Personen erscheinen zu lassen, denen auch Dialoge zugewiesen wurden.*

### **Welche Funktionen wünschen Sie sich noch?**

*Ein Forum, in dem sich Nutzer austauschen können. Hier könnten Schulen mit großem Potential an Aufgabenentwicklern Aufgaben anderen Schulen zur Verfügung stellen. Ansonsten könnte es schnell dazu kommen, dass an vielen Schulen das Programm zwar installiert wird, die LehrerInnen aber nicht tätig werden, um das Programm letztlich an die örtlichen Gegebenheiten anzupassen.*