

Bearbeitungsbeginn: 01.03.2022

Vorgelegt am: 30.06.2022

Thesis

zur Erlangung des Grades

Bachelor of Science

im Studiengang Medieninformatik

an der Fakultät Digitale Medien

Huchler Maurice und Bodmer Tobias

Matrikelnummer: 260374 / 260528

**Entwicklung eines kooperativen Rogue-Like-Game auf Basis von
FUDGE**

Erstbetreuer: Prof. Jirka Dell'Oro-Friedl

Zweitbetreuer: Prof. Christoph Müller

Inhaltsverzeichnis

1	Abbildungsverzeichnis	1
2	Abstract.....	3
3	Einleitung	3
3.1	Ideenfindung	4
3.2	Genre	5
3.3	Zielgruppe	5
4	Konzept.....	6
4.1	Wahl des Settings [<i>Gemeinsam</i>].....	6
4.2	Story [<i>Maurice Huchler</i>]	7
4.3	Gegenstände [<i>Gemeinsam</i>].....	8
4.4	Weltaufbau [<i>Gemeinsam</i>].....	9
4.4.1	Händlerraum [<i>Gemeinsam</i>].....	9
4.4.2	Schatzraum [<i>Gemeinsam</i>]	9
4.4.3	Herausforderungsraum [<i>Gemeinsam</i>].....	10
4.4.4	Bossraum [<i>Gemeinsam</i>]	10
4.5	Grafischer Aufbau [<i>Gemeinsam</i>].....	11
4.6	Kooperative Elemente [<i>Gemeinsam</i>]	11
4.6.1	Herausforderungen	11
5	Umsetzung.....	14
5.1	Verwendete Technologien	14
5.1.1	FUDGE	14
5.1.2	TypeScript.....	14
5.1.3	GitHub.....	14
5.1.4	Heroku [<i>Tobias Bodmer</i>].....	15
5.1.5	Aseprite [<i>Maurice Huchler</i>].....	15
5.2	Das Spiel.....	16
5.2.1	Spieldurchlauf [<i>Gemeinsam</i>].....	16
5.2.2	Grundlegende Spielmechaniken	16
5.2.3	Interaktionen zwischen Avatar und Avatar [<i>Gemeinsam</i>].....	17
5.2.4	Weltgenerierung [<i>Maurice Huchler</i>]	17
5.2.5	Verhalten der Gegner [<i>Maurice Huchler</i>]	18
5.2.6	Gegnertypen [<i>Gemeinsam</i>].....	21
5.2.7	Boss Mechaniken	23
5.2.8	Attributsystem [<i>Gemeinsam</i>].....	29

5.2.9	Bufs [<i>Gemeinsam</i>]	31
5.2.10	Items [<i>Gemeinsam</i>]	34
5.2.11	User-Interface [<i>Tobias Bodmer</i>]	38
5.2.12	Netzwerk Übertragung [<i>Tobias Bodmer</i>]	41
5.3	Probleme	42
5.3.1	Asynchronität	42
5.3.2	Verzögerte Bewegung [<i>Tobias Bodmer</i>]	42
5.3.3	RTC Verbindungen [<i>Tobias Bodmer</i>]	44
6	Testphase	45
6.1	Allgemeines	45
6.2	Kooperation	46
6.3	Steuerung	47
6.4	Design	48
6.5	Spaßfaktor	49
6.6	User Interfaces	50
6.7	Eigene Meinung	51
6.8	Zusammenfassung der Testphase	53
7	Ausblick	54
8	Fazit	56
9	Literaturverzeichnis	57
10	Eidesstattliche Erklärung Huchler Maurice	58
11	Eidesstattliche Erklärung Bodmer Tobias	59

1 Abbildungsverzeichnis

Abbildung 1 „common“ – Maurice Huchler	8
Abbildung 2 „rare“ – Maurice Huchler.....	8
Abbildung 3 "epic" – Maurice Huchler	8
Abbildung 4 "legendary" – Maurice Huchler	8
Abbildung 5 Thors Hammer – Tobias Bodmer	12
Abbildung 6 Thors Hammer – Tobias Bodmer	12
Abbildung 7 coherence.....	18
Abbildung 8 separation.....	18
Abbildung 9 alignment.....	19
Abbildung 10	19
Abbildung 11	19
Abbildung 12 - Maurice Huchler	21
Abbildung 13 – Maurice Huchler	21
Abbildung 14 – Maurice Huchler	22
Abbildung 15 – Maurice Huchler	22
Abbildung 16 – Maurice Huchler	22
Abbildung 17 Concept Art - Maurice Huchler	23
Abbildung 18 Summoner Pixel Art - Maurice Huchler	23
Abbildung 19 Summoner Verhaltensmuster	24
Abbildung 20 Waterball - Maurice Huchler	24
Abbildung 21 Summoner Summon - Maurice Huchler	25
Abbildung 22 Big Boom Pixel Art - Maurice Huchler	26
Abbildung 23 Big Boom Concept Art - Maurice Huchler	26
Abbildung 24 Big Boom Verhaltensmuster	27
Abbildung 25 Bleed - Maurice Huchler	31
Abbildung 26 Heal - Maurice Huchler	31
Abbildung 27 Ice - Maurice Huchler	31
Abbildung 28 Exhausted - Maurice Huchler	32
Abbildung 29 Furious - Maurice Huchler	32
Abbildung 30 Immune - Maurice Huchler	33
Abbildung 31 Poison - Maurice Huchler	33
Abbildung 32 Ice Bucket - Tobias Bodmer	34
Abbildung 33 Toxic Relationship - Maurice Huchler	34

Abbildung 34 Projectiles Up - Tobias Bodmer	35
Abbildung 35 Ein Projektil - Tobias Bodmer	35
Abbildung 36 Zwei Projektile - Tobias Bodmer	35
Abbildung 37 Drei Projektile - Tobias Bodmer	35
Abbildung 38 Shroom Down - Tobias Bodmer	36
Abbildung 39 Shroom Up - Tobias Bodmer	36
Abbildung 40 Met - Maurice Huchler	36
Abbildung 41 Homecoming - Tobias Bodmer	36
Abbildung 42 Sharpening Stone - Tobias Bodmer	37
Abbildung 43 Speed Up - Maurice Huchler	37
Abbildung 44 Startbildschirm	38
Abbildung 45 Credits	39
Abbildung 46 Verbindungsbildschirm	39
Abbildung 47 Head Up Display - Tobias Bodmer	40
Abbildung 48 Head Up Display - Tobias Bodmer	40
Abbildung 49 Bossraum Icon - Tobias Bodmer	41
Abbildung 50 Herausforderungsraum Icon - Tobias Bodmer	41
Abbildung 51 Handelsraum Icon - Tobias Bodmer	41
Abbildung 52 Schatzraum Icon - Tobias Bodmer	41
Abbildung 54	45
Abbildung 54	45
Abbildung 55	46
Abbildung 56	46
Abbildung 58	47
Abbildung 58	47
Abbildung 59	48
Abbildung 60	49
Abbildung 62	50
Abbildung 62	50
Abbildung 63	51

2 Abstract

In Ära übernimmt der Spieler:in einen gestorbenen Wikinger, der sich seinen Platz in Walhalla verdienen muss. Mit einer Walküre an seiner Seite kämpft er sich durch Ebenen an Gegnern und löst dabei kooperative Aufgaben. Ziel dieser Bachelorarbeit war es einen spielbaren Prototyp mithilfe der Spiele-Engine FUDGE zu erstellen. Bei dem Prototyp handelt sich um einen 2D kooperativen Action-Rogue-Like, der dank FudgeNet über das Internet gespielt werden kann. Aus den Rogue-Like Prinzipien wurden die wichtigsten Elemente herausgearbeitet und darauf aufbauend eigene Kooperative-Elemente konzipiert, um ein besseres Gemeinschaftsgefühl beim Spielen zu fördern.

3 Einleitung

Diese Abschlussarbeit umfasst Inhalte von Zwei Autoren. Beide sind große Fans von Rogue-Likes und spielen diese regelmäßig in unserer Freizeit. Da Beide gerne mit anderen Mitspieler:innen zusammenspielen, besteht ein großes Interesse an „Multiplayer“ und kooperativen Spielen. Leider findet man selten Rogue-Like-Spiele, welche es ermöglichen, zusammen durch die generierten Welten zu wandern und das Gefühl, jeden Durchlauf etwas Neues zu entdecken, vereinen.

Die Spiele-Engine „FUDGE“ entwickelt von Prof. Jirka Del’Oro-Friedl bekam im vorgehenden Semester dieser Abschlussarbeit durch Falco Böhnke eine Netzwerkkomponente eingebaut. Hierdurch lassen sich, mithilfe eines Servers, diverse Nachrichten zwischen Host und Client versenden.

Auf Grundlage dessen war die Idee geboren, ein kooperatives Rogue-Like Game für FUDGE zu entwickeln.

3.1 Ideenfindung

Bei der Ideenfindung wurden verschiedenste Rogue-Like-Games genauer betrachtet und deren Kernelemente gesammelt. So konnten Gemeinsamkeiten herausgearbeitet werden wie beispielsweise prozedural generierte Welten, überwiegende 2D Ansicht mit „Topdown“ Perspektive, Spezialfähigkeiten oder durchlaufbeeinflussende Gegenstände. Daraufhin wurde festgelegt was davon übernommen und was ersetzt werden sollte.

Anhand der Gemeinsamkeiten und der Überlegungen konnte die MoSCoW-Priorisierung durchgeführt werden bei der Anforderung in unbedingt erforderlich, sollte umgesetzt werden, kann umgesetzt werden und wird nicht umgesetzt kategorisiert werden. Diese sieht angewendet wie folgt aus:

„Must Haves“	„Should Haves“	„Could Haves“	„Won't Haves“
Kooperative Elemente Lobby mit Character Auswahl Character Movement Synchronität Welt Generierung Items mit Auswirkungen Funktionierender Game-Loop	Pixelarts Sprites HUD-Design Damage Indicator Buffs GUI-Design Sounds	Animations Lore mehrere Spielfiguren	

3.2 Genre

Ära ist ein 2D kooperativer Action-Rogue-Like und ordnet sich dem Genre Coop Rogue-Like-Game unter. Dieses Genre umfasst Spiele, die dem Original Spiel Rogue nahekommen. Kernelemente für Rogue-Like-Game sind: Kein Gameloop übergreifender Fortschritt, Procedural generierte Welt, zufällige Gegenstände in jedem Gameloop. (vgl. Mitchell Loewen 2020)

3.3 Zielgruppe

Ära spricht Rogue-Like Fans jeden Alters an, die ein Interesse an Wikingern und gleichgesinnte Freund:innen haben. Es eignet sich aber auch für Spieler:innen die Action sowie Zusammenarbeit schätzen und dabei Augenmerk auf Abwechslung, interessante Herausforderungen, schwere Bosskämpfe und Wiederspielwert legen.

4 Konzept

4.1 Wahl des Settings *[Gemeinsam]*

Um ein vielversprechendes Setting auszusuchen, wurde eine große Auswahl von Settings zusammengestellt. Themen wie Fantasy-Mittelalter, Vampire, Religion, Steampunk, Unterwasser-Welten und Wikinger. Um herauszufinden, ob vielversprechenden Inhalt hinter solch einem Thema steckt, wurde anhand eines Brainstorms Ideen gesammelt. Signifikant waren vor allem wie die einzelnen Gegner aussehen könnten und mit welcher Angriffsart, Nah- oder Fernkampf, primär und sekundär gekämpft werden soll. Große Menge an mögliche Gegnertypen und Waffen spiegelte das Wikinger-Setting wider. Auch wie im nächsten Abschnitt beschrieben beinhaltet die nordische Mythologie eine gute Voraussetzung, um eine Geschichte aufzubauen, die den kooperativen Aspekt unterstützt. Auch bietet dieses Thema Möglichkeiten in Richtung Sagen und abwechslungsreicher Welten, wie Asgard, Midgard oder auch Muspelheim.

4.2 Story *[Maurice Huchler]*

Die Erzählung der Geschichte beginnt mit dem Tod des Avatars. Dieser fällt in einer Schlacht als tapferer Krieger. Das Leben endet in der nordischen Mythologie nicht mit dem Tod. Nach dem Tod gibt es zwei Möglichkeiten: Entweder der/die Krieger:in wird ausgewählt und ist würdig nach Walhalla zu gelangen oder der/die Krieger:in wird nach Folkvang, „Feld des Volkes“ geschickt (vgl. David Schiffer 2022).

In der Geschichte von Ära wird sich auf die Reise nach Walhalla fokussiert. Bevor einem der Einzug in die Hallen gewährt wird, steht ein weiter Weg bevor. Hinzu kommt noch, dass dieser bewacht und übersät mit Aufgaben und Monstern ist. Für einen einzelnen Wikinger ist dies kaum zu schaffen. Hierfür wurde in der alten nordischen Mythologie schon Abhilfe geschaffen. „Die Walküren, „die Wählerinnen der Gefallenen“ sind wunderschöne Kriegerinnen, die Odin in vielerlei Hinsicht unterstützen“ (vgl. David Schiffer 2022). Dies ermöglicht es den/der zweiten Spieler:in als direkte Hilfe einzubauen. Dieser spielt als Walküre, die den Avatar des ersten Spielers ausgewählt hat, um nach Walhalla zu gelangen. Als gefallener Krieger und Walküre ziehen diese los, um sich nach Walhalla durzuschlagen. Doch was kann erwartet werden?

Hierbei wird von der nordischen Mythologie abgewichen. Handlungen sind frei erfunden und dienen nur zum Zwecke der Unterhaltung. Der Weg, der sich ausgedacht wurde, beinhaltet insgesamt drei Ebenen. Die Beiden Avatare sind darauf angewiesen zusammen nach Walhalla anzukommen. Scheitert einer der Beiden, sind sie dazu bestimmt den Weg immer und immer wieder durchzustehen. Die nordischen Götter wissen von diesem Vorhaben und Stellen den Antretenden immer wieder Herausforderungen, die sie meistern müssen.

4.3 Gegenstände *[Gemeinsam]*

Innerhalb der Welt werden diverse Gegenstände gefunden, die den eigenen Avatar stärken. Diese Gegenstände besitzen unterschiedliche Seltenheitsgrade.



Abbildung 1 „common“ – Maurice Huchler

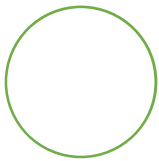


Abbildung 2 „rare“ – Maurice Huchler

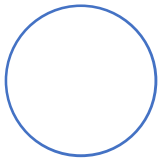


Abbildung 3 "epic" – Maurice Huchler

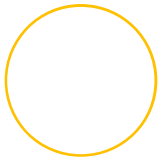


Abbildung 4 "legendary" – Maurice Huchler

Anhand dieser Seltenheit werden die Gegenstände ausgewählt.

4.4 Weltaufbau *[Gemeinsam]*

Ein Level besteht aus einer Ebene, eine Ebene wiederum besteht aus mehreren Räumen. Die Räume sind anhand eines Rasters angeordnet. Ein Raum kann dadurch maximal vier Nachbarn besitzen. Die Anzahl der Räume ist vordefiniert und für jede Ebene gleich.

4.4.1 Händlerraum *[Gemeinsam]*

Im Händlerraum kann, wie der Name schon verrät, gehandelt werden. Es wurde absichtlich auf ein standardisiertes Bezahlungssystem verzichtet, um zusätzliche Währungen zu vermeiden. Bezahlt wird nicht mit Münzen, sondern mit Gegenständen, die der Avatar besitzt. Ein Gegenstand kann gegen einen gleichwertigen eingetauscht werden. Ein Gegenstand ist gleichwertig, wenn er dieselbe Seltenheit besitzt wie der Gegenstand. Ein höherwertiger Gegenstand, also eine Seltenheitsstufe oberhalb, kann ebenfalls erworben werden. Hierfür müssen allerdings drei geringerwertige Gegenstände eingetauscht werden.

4.4.2 Schatzraum *[Gemeinsam]*

Der Schatzraum wird pro Spieldurchlauf zufällig generiert. Innerhalb dieses Raumes befinden sich keine Gegner. Stattdessen sind zwei zufällige Gegenstände enthalten. Eine Dopplung ist nicht ausgeschlossen. Die Spieler:innen müssen selbst abwägen, ob sie den Gegenstand mitnehmen möchten oder nicht. Je nach Avatar und Gegenstand machen manche Items mehr und manche weniger Sinn.

4.4.3 Herausforderungsraum *[Gemeinsam]*

Das Ziel dieses Raumes ist es, die Herausforderung zu schaffen, die Spieler:innen gestellt werden. Eine Herausforderung ist abgeschlossen, wenn die Kriterien erreicht sind. Die Kriterien richten sich nach der Aufgabe, die innerhalb des Raumes gestellt wird. Welche Herausforderung angetroffen wird, bestimmt der Zufall, die Ebene oder der Durchlauf.

4.4.4 Bossraum *[Gemeinsam]*

Das Finale jeder Ebene wird durch einen Bosskampf abgeschlossen. Auf jeder Ebene befindet sich ein anderer Boss, der zu der jeweiligen Ebene passt. Nach bestehen des Kampfes wird den Spielern jeweils ein Item für ihre Avatare gewährt. Zusätzlich öffnet sich der Zugang für die nächste Ebene und erlaubt es den Spielern in die nächste Ebene zu gelangen.

4.5 Grafischer Aufbau *[Gemeinsam]*

Der Grafische Aufbau des Spiels wurde sehr stark an dem zugehörigen Setting und der Story orientiert. Das bedeutet, dass Gegner so gestaltet wurden, dass diese in den Ort, an dem sie leben, hineinpassen. Am Beispiel der Sumpflandschaft leben dort Kreaturen, die sich in der Nähe von Sümpfen aufhalten: Spinnen, Zecken aber auch Skelette, die von Personen stammen, die dort ertrunken oder gestorben sind. Allgemein wurde darauf geachtet ein stimmiges grafisches Design einzuhalten, um für mehr Immersion zu sorgen.

4.6 Kooperative Elemente *[Gemeinsam]*

Hierbei handelt es sich um Spielelemente, welche gezielt auf eine Zusammenarbeit der beiden Spieler:innen ausgerichtet sind. Dabei wird versucht komplementäre Eigenschaften umzusetzen. Als konkretes Beispiel aus der Realität, kann man sich Hammer und Nagel vorstellen. Allein sind die beiden Gegenstände nutzlos. Nur zusammen ergeben sie einen Sinn.

Angewandt auf unser Spiel gibt es Items und Gegner, bei denen man auf die Hilfe des anderen Mitspielers angewiesen ist. Genauer ist in den einzelnen Gegenständen beschrieben.

4.6.1 Herausforderungen

Grundlegende Idee der Herausforderungen ist es die Zusammenarbeit der beiden Spieler:innen zu stärken. Eine besonders gute Zusammenarbeit wird belohnt anhand eines zusätzlichen Gegenstandes für beide Spieler:innen. Bei nicht bestehen ist das Spiel zu Ende. Die Herausforderungen sind somit ein High-Risk High-Reward-Modell. Dadurch soll die Kommunikation zwischen den beiden Spieler:innen angehoben werden.

4.6.1.1 Thors Herausforderung



Abbildung 5
Thors Hammer –
Tobias Bodmer

Mjölhnir, Thors Hammer lässt sich nicht von jedem anheben, er wählt seinen Besitzer. Mjölhnir spielt bei Thors Herausforderung eine Schlüsselrolle, da die Avatare entwaffnet wurden und sich einer Horde Gegnern gegenübersehen.

Thors Herausforderung ist eine der Aufgaben, denen die Spieler:innen sich gegenübersehen, wenn sie einen der Herausforderungsräume betreten.

Die Herausforderung liegt dabei das Mjölhnir immer abwechselnd geworfen werden muss. Der/Die Spieler:in, der den Hammer wirft, kann ihn danach nichtmehr aufsammeln. Hierdurch entsteht eine Art „Ping-Pong“-Spiel zwischen beiden Spielern:innen. Dies zwingt die Spieler:innen dazu miteinander zu kommunizieren und mitzuteilen wohin sie den Hammer als nächstes werfen werden.



Abbildung 6
Thors Hammer –
Tobias Bodmer

4.6.1.2 Logis Herausforderung

Die Herausforderung von Logi, dem Feuerriesen, stellt ebenfalls die beiden Spieler:innen auf die Probe. Diese kann nur in der dritten Ebene, dem feurigen Abgrund, angetroffen werden. Hierbei überlässt Logi einem der Avatare die Macht des Feuers. Dadurch wird dessen Atem in lodernde Flammen verwandelt. Jedoch verursacht dieser keinen Schaden an den Magma Kreaturen, die den Spielern gegenüberstehen. Der andere Avatar bekommt Bombenfässer, welche er frei im Raum platzieren kann.

Die Herausforderung liegt hierbei darin, die Bombenfässer so zu platzieren, dass der andere Avatar sie anzünden kann. Im selben Moment muss beachtet werden, weitgenug weg zu sein, um keinen Schaden zu nehmen, jedoch die Gegner so nah heranzulocken, dass diese Schaden bekommen.

4.6.1.2.1 Freyas Herausforderung

Freya, Göttin der Liebe, stellt eine weniger kriegerische Herausforderung. Sie gibt den Spielern die Macht der Anziehung. Mit dieser Macht können die Spieler:innen sich anziehen und abstoßen, wie mit einem Magneten. Doch sie nimmt nicht nur die Waffen der Avatare, sondern auch die Möglichkeit euch zu bewegen, vorübergehend an sich.

Um die Herausforderung zu bestehen, müssen die Spieler:innen überleben, in dem sie sich im Raum abstoßen und anziehen, um den Gegner und deren Angriffe zu entgehen. Ist dies geschafft bekommen die Spieler:innen ihre Bewegungsfreiheit und Waffen wieder zurück.

5 Umsetzung

5.1 Verwendete Technologien

5.1.1 FUDGE

Der Prototyp wurde mithilfe von FUDGE, dem Furtwangen University Didactic Game Editor entwickelt. FUDGE verwendet die Programmiersprache TypeScript. Da FUDGE im Browser läuft kann es auf fast jedem Gerät verwendet werden.

5.1.2 TypeScript

Als Programmiersprache kam TypeScript zum Einsatz, nicht nur da FUDGE TypeScript verwendet, es bietet auch Programmierkonstrukte an die über die Möglichkeiten von JavaScript hinaus gehen. Klassen, Namespaces und Interfaces schaffen einen besseren Überblick und schaffen einen Rahmen innerhalb des Codes. (vgl. TypeScript 2022)

5.1.3 GitHub

GitHub ist ein Dienst, der zur Versionsverwaltung genutzt wird. Assets, Programmcode sowie Konzepte können dort gespeichert werden. Das Projekt ist zentral gespeichert und es können Abbilder erstellt werden. Dadurch kann gleichzeitig an einem Projekt gearbeitet werden und ebenfalls bei Problemen zurückgesetzt werden. Eine weitere Funktion die GitHub bietet ist GitHub Pages. Mithilfe von GitHub Pages war es möglich direkten Zugang zum Spiel zu gewähren und es direkt von dieser Seite abzurufen. (vgl. Company GitHub 2022)

Der Prototyp wird über GitHub verwaltet, um simultan an einem Projekt arbeiten zu können, sowie um die Vorteile der Versionsverwaltung zu nutzen und die Applikation direkt über GitHub Pages zur Verfügung zu stellen.

5.1.4 Heroku *[Tobias Bodmer]*

Heroku ist eine Cloud Applikationsplattform, auf der man Applikationen zu Verfügung stellen kann. Die Applikation kann dann über eine URL erreicht werden.

Für das Networking des Prototypens wurde ein zentraler Server benötigt, um die Spieler:innen miteinander zu verbinden. Über Heroku wurde die FudgeNet Server Implementation bereitgestellt. Dafür wurde die FudgeNet Server Implementation um eine von Heroku benötigte Datei ergänzt und über Git auf Heroku gepusht. (vgl. Salesforce 2022)

5.1.5 Aseprite *[Maurice Huchler]*

Aseprite ist ein Bildbearbeitungsprogramm, welches den Fokus stark auf Pixelgrafiken legt. Es erlaubt die Erstellung eigener Pixel-Animationen sowie die automatische Generation von „spritesheets“ (vgl. Aseprite 2022)

Dadurch war es möglich die Animationen für die jeweiligen Game-Assets zu erstellen die für das Spiel benötigt waren.

5.2 Das Spiel

5.2.1 Spieldurchlauf *[Gemeinsam]*

Als Spieldurchlauf oder auch Game-Loop bezeichnet man die Ereigniskette eines Videospiels. Wird das Spiel gestartet so beginnt die Ereigniskette und auch der Spieldurchlauf beginnt. Stirbt der Avatar, so wird in Rogue-Like-Games der Spieldurchlauf beendet und beginnt neu. Die Ereigniskette reagiert hierbei auf Eingaben des/der Spieler:in so wie ebenfalls auf bestimmte Events. Der Spieldurchlauf wird bei Rogue-Like-Games auch Run genannt. Diese Runs unterscheiden sich in Rogue-Like-Games in den Positionen der Gegner, den gefundenen Gegenständen und der generierten Welt. (Ulrich Wimmermoth 2020)

5.2.2 Grundlegende Spielmechaniken

5.2.2.1 Bewegung & Angriff *[Tobias Bodmer]*

Die Tasten W-A-S-D sind für die Bewegung des Avatars zuständig. Um die Richtung der Angriffe vorzugeben, wird der Mauszeiger verwendet. Angriffe erfolgen also immer in Richtung Mauszeiger.

5.2.2.2 Fähigkeiten *[Gemeinsam]*

Fähigkeiten können mit der „Leertaste“ ausgeführt werden. Diese besitzen eine festgelegte Abklingzeit. Erst nach ablaufen dieser Zeit kann die Fähigkeit erneut eingesetzt werden.

5.2.2.3 Interaktionen *[Tobias Bodmer]*

Spielinteraktionen wie durch Türen gehen oder mit NPCs sprechen werden mit „E“ ausgelöst.

5.2.2.4 Diverses *[Tobias Bodmer]*

Mithilfe der Escape -Taste lässt sich das Spiel pausieren. Ebenfalls werden dadurch die Einstellungen, z. B. für die Lautstärke, sichtbar.

5.2.3 Interaktionen zwischen Avatar und Avatar *[Gemeinsam]*

Zur Stärkung des kooperativen Gefühls, dass man voneinander abhängig ist, wurden bestimmte Spielelemente eingebaut. Das sogenannte „friendly fire“ erlaubt es gegenseitig Schaden zu machen. Dadurch wird erhofft, dass die beiden Spieler:innen mehr kommunizieren und besser aufeinander achten. Kurz gesagt ein stärkeres Gefühl von Gemeinsamkeit entsteht.

5.2.4 Weltgenerierung *[Maurice Huchler]*

Die prozedurale Generierung die Ära enthält, ist die vereinfachte Form der Generierung. Als Inspiration wurde die Weltgenerierung von dem Spiel „Binding of Isaac“ benutzt (vgl. Boris 2022). Es werden vordefinierte Räume erstellt und diese nach Zufallsprinzip aneinandergereiht.

Die einzelnen Ebenen des Spiels sind prozedural generiert. Dies bedeutet, dass sie vor jedem Durchlauf automatisch und zufällig erstellt werden. Bei der Generierung wird beachtet wie viele „Normal Rooms“ erstellt werden. Des Weiteren ist garantiert, dass es immer einen „Boss Room“ und einen „Merchant Room“ geben muss. Anhand eines prozentualen Werts können zufällig „Treasure Rooms“ und „Challenge Rooms“ erscheinen. Dadurch ergibt sich der Vorteil, dass die Level einen hohen Wiederspielwert haben.

Vorgehensweise der Generierung:

Zu Beginn wird ein viereckiger Raum erstellt, der als „Start-Room“ definiert wird. Ausgehend von diesem wird an einer zufällig ausgewählten Seite ein weiterer Raum angehängen. Dadurch sind die Räume aneinanderhängend und bilden ein Schlauch-System. Bei der Generierung wird darauf geachtet, dass keine Überlagerungen entstehen.

Aufbauend auf die erstellten Räume wird nun an den Letzen Raum, der erstellt wurde, der „Boss Room“ angehängt. Ebenfalls wird an den ersten Raum der „Merchant Room“ angehängt. Anschließend wird bei jedem „Normal Room“ durch Zufall entweder ein „Treasure Room“ oder „Challenge Rooms“ angehängt. Um die Erstellung einer vollständigen Ebene zu garantieren, sind Sicherheitsmechanismen eingebaut. Wird zum Beispiel der Bossraum nicht platziert, wird ein „failed“-Status ausgerufen, der dazu führt, dass die komplette Generierung erneut ausgeführt wird.

5.2.5 Verhalten der Gegner [Maurice Huchler]

Eines der spannendsten Spielelemente, vieler Spiele, sind die Gegner. Ihr Design und Verhalten entscheidet über den Spielspaß, den die Spieler:innen wahrnehmen. Aufgrund dessen wurde besonderen Wert darauf gelegt, die Gegnern anhand der Spielwelt zu designen.

Die Gegner-Typen basieren alle auf der gleichen künstlichen Intelligenz. Die Richtung der Gegner wird durch einen Vektor bestimmt. Aufgrund dessen Verhalten sich alle Gegner gleich. Zusätzlich ist es möglich, dass mehrere Gegner auf der gleichen Position stehen. Dadurch überlappen ihre Körper und es ist nichtmehr sichtbar, wie viele Gegner sich auf der Stelle befinden. Eine Lösung ist, dass der Nachbar beobachtet wird und das eigene Verhalten entsprechend anpasst. Dies lässt sich Mithilfe des schon bestehenden Schwarmverhalten simulieren. Der Algorithmus am Ende nur einen einzelnen Richtungsvektor zurück, der besagt in welche Richtung die Einheit bewegt wird. Jede Einheit hat ihren eigenen Sichtradius der bei der Berechnung beachtet wird. Mithilfe von Gewichtungen werden mehrere Vektoren zusammengerechnet und zu einem einzelnen Vektor vereint. Die drei Grundlegenden Vektoren bei der Berechnung von Schwarmverhalten sind: „Coherence“, „Separation“ und „Alignment“ (vlg.Ben Eater 2022).

„Coherence“ beschreibt den Ort, an dem sich alle Einheiten treffen wollen, die wahrgenommen werden. Anhand von allen Positionen der Einheiten wird eine Durchschnittsposition ermittelt auf die alle Einheiten zulaufen (vgl. Ben Eater 2022).

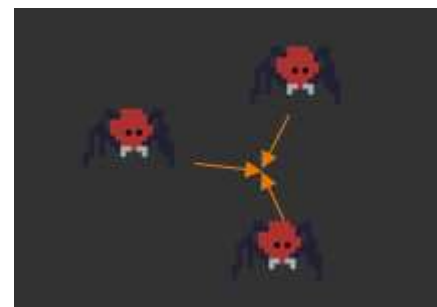


Abbildung 7 coherence

„Separation“ beschreibt den Abstand zwischen den Einheiten. Geraten diese zu nah aneinander laufen Sie voneinander weg. Falls mehre Einheiten zu nahe sind wird der Mittelwert genommen (vgl. Ben Eater 2022).

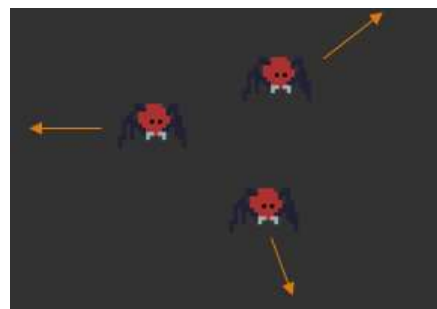


Abbildung 8 separation

„Alignent“ beschreibt die Richtung in die die Einheit sich bewegt. Jede Einheit versucht sich der Richtung seiner Nachbarn anzugleichen. Dies wird nicht durch eine spontane Richtungsänderung verursacht sondern durch annäherndes Verhalten. (vgl. Ben Eater 2022)

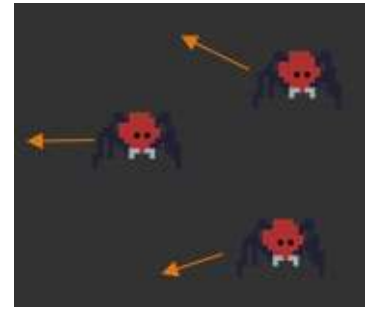


Abbildung 9 alignment

Anhand verschiedener Gewichtungen der einzelnen Vektoren lassen sich diverse Punktwolken aus Gegnern kreieren. Durch diesen Mechanismus wirken die Gegnertypen, wie Spinnen, viel lebendiger und haben ein natürlicheres Verhalten.



Abbildung 11

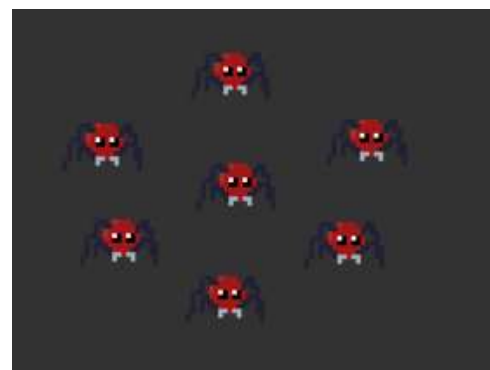


Abbildung 10

Um schlussendlich zu erreichen, dass Gegner unterschiedliche Verhaltensweisen besitzen, die abhängig von den Avataren sind, muss das System noch erweitert werden. Hierfür wurden noch weitere Vektoren in die Gewichtung mitaufgenommen, die im Folgenden beschrieben werden.

5.2.5.1 „Dumb-Enemies“

Diese Generation ist sehr simpel aufgebaut und hat nur ein Ziel: den Avatar. Kommt ein Avatar zu nahe berechnet diese künstliche Intelligenz den direkten Weg zum Ziel.

5.2.5.2 „Dash-Enemies“

Die „Dash-Enemies“ erweitern die Funktionen der „Dumb-Enemies“. Sind sie dem Avatar nahe genug, führen sie die Fähigkeit: „Dash“ aus. Durch ihre Fähigkeit werden sie ruckartig nach vorne katapultiert und simulieren einen Sprungangriff von z.B. Spinnen.

5.2.5.3 „Smash-Enemies“

Diese Gegner zeigen dasselbe Grundverhalten wie die „Dumb-Enemies“. Kommen sie nah genug an ihr Ziel heran führen sie eine Schlag-Animation aus.

5.2.5.4 „Circle-Enemies“

Ab einem bestimmten Abstand zum Ziel wird bei diesem Gegner der Richtungsvektor um 90 Grad gedreht, was dazu führt, dass die Gegner um das Ziel herumlaufen.

5.2.5.5 „Shoot-Enemies“

Hier ist das Verhalten des „Dumb-Enemies“, umgedreht. Kommt der Avatar zu nahe läuft der Gegner von ihm weg. Hat er eine bestimmte Distanz zwischen dem Avatar und sich selbst, schießt er einen Pfeil in Richtung Avatar.

5.2.6 Gegnertypen [Gemeinsam]

Die Gegner in Ära sind vielseitig einsetzbar. Die Infrastruktur ist so aufgebaut, dass jeder Gegner mit jeder künstlichen Intelligenz verknüpft werden kann. Es wird eine andere Hülle über das Verhalten gelegt. Innerhalb des Prototyps wurden die Gegner an eine feste KI gebunden. Die folgenden Kombinationen sind im Prototyp vorzufinden:

5.2.6.1 Fledermaus

Künstliche Intelligenz: „Cirlice-Enemies“



Abbildung 12 - Maurice Huchler

Leben	10
Schaden	1,0
Geschwindigkeit	2,5
Größe	1,0
Rückstoß	0,1
Rüstung	0,0
Abklingzeit	1,0
Genauigkeit	-

5.2.6.2 Kleine Zecke

Künstliche Intelligenz: „Dumb-Enemies“



Abbildung 13 – Maurice Huchler

Leben	10
Schaden	1,0
Geschwindigkeit	2,5
Größe	1,0
Rückstoß	0,1
Rüstung	0,0
Abklingzeit	1,0
Genauigkeit	-

5.2.6.3 Rote Zecke

Künstliche Intelligenz: „Dash-Enemies“



Leben	15
Schaden	1,0
Geschwindigkeit	1,5
Größe	1,0
Rückstoß	0,1
Rüstung	0,0
Abklingzeit	1,0
Genauigkeit	-

Abbildung 14 – Maurice Huchler

5.2.6.4 Skelett

Künstliche Intelligenz: „Shoot-Enemies“



Leben	20
Schaden	1,0
Geschwindigkeit	4,0
Größe	1,0
Rückstoß	0,1
Rüstung	0,0
Abklingzeit	1,0
Genauigkeit	80,0

Abbildung 15 – Maurice Huchler

5.2.6.5 Oger

Künstliche Intelligenz: „Smash-Enemies“



Leben	40
Schaden	5,0
Geschwindigkeit	1,0
Größe	1,0
Rückstoß	0,1
Rüstung	25,0
Abklingzeit	1,0
Genauigkeit	-

Abbildung 16 – Maurice Huchler

5.2.7 Boss Mechaniken

5.2.7.1 „Summoner“ [Maurice Huchler]

Der „Summoner“ ist der Endboss der ersten Ebene. Bei seinem Design wurde darauf geachtet, möglichst viele natürliche Elemente einzubauen. In der Geschichte herrscht der „Summoner“ über die versumpften Lande. Sein Aussehen baut deshalb auf Körperteile von anderen Tieren, die dort ihr Ende gefunden haben.

Sein Kopf besteht aus dem toten Schädel eines Tieres. Um seinen Körper hat der „Summoner“ einen langen dunklen Mantel gelegt, der mit Federn schmückt ist. Als Waffe hält dieser einen alten verwurzelten



Abbildung 17 Concept Art - Maurice Huchler

Holzstab in der Hand. Er besitzt keine Füße und fliegt durch die Ebenen.

Da der Schauplatz innerhalb eines Sumpfes befindet, war Wasser ein naheliegendes Element, um seine Angriffe zu unterstützen. Um seine Gegner zu treffen, feuert der „Summoner“ eine Wasserkugel in Richtung Spieler:in ab. Doch nicht nur das Aussehen wurde nach seinem „Wohnort“ entwickelt, auch seine Angriffe sind anhand dessen ausgerichtet.



Abbildung 18 Summoner Pixel Art - Maurice Huchler

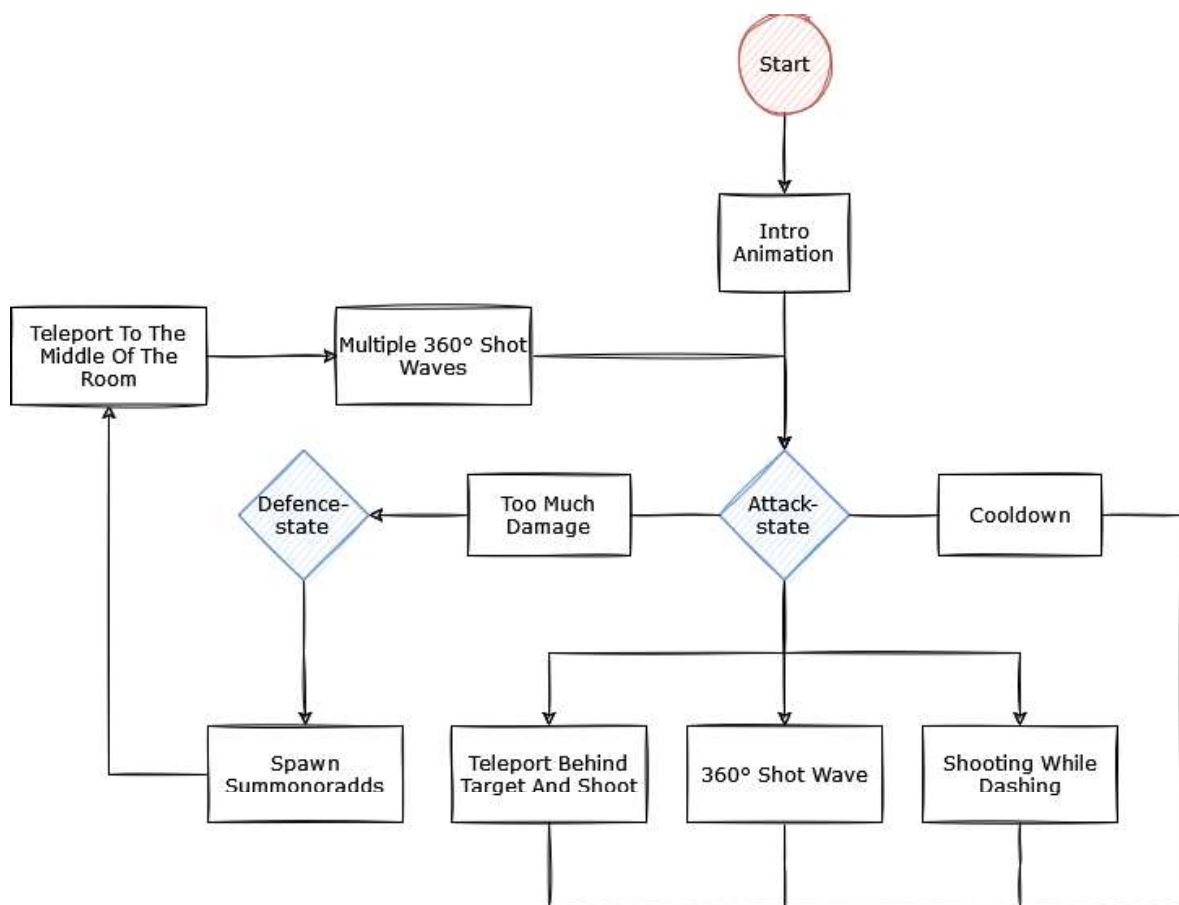


Abbildung 19 Summoner Verhaltensmuster

Der “Summoner” besitzt zwei Arten von Verhaltensweisen: Angriff und Verteidigung.

In der Angriffs-Phase ist er sehr wendig und ständig in Bewegung. Er besitzt drei Angriffe welche zufällig ausgewählt werden:

Teleportation:

Er kann sich an jeden vorhergesehen Ort teleportieren.

360°-Angriff:

Bei dem 360°-Angriff teleportiert der „Summoner“ sich in die Mitte des Raumes, wird immun gegenüber Angriffen, und löst dreimal in geregelten Abständen eine Welle an Wasser aus, die nach außen hin größer wird.



Abbildung 20
Waterball -
Maurice Huchler

„DashNSlash“.

Durch die Fähigkeit „Dash“ wird er temporär schneller und wischt dadurch anhand eines Halbkreises um sein Ziel herum. Dabei feuert er mehrere Wassergeschoße in Richtung seines Ziels.

Als Schutzmechanismus wurde die Verteidigungsphase eingebaut. Diese wird ausgelöst, wenn eine bestimmte Höhe an Schaden, durch die Spieler:innen, erreicht ist. Während seiner kompletten Vorgehensweise ist er unverwundbar. Durch die Teleportation gelangt er an eine bestimmte Stelle innerhalb des Boss Raumes und beschwört mithilfe seines Stabes Kreaturen aus der ersten Ebene. Hierbei wurde eine Mechanik eingebaut, damit die anderen Spieler:innen aufeinander aufpassen müssen. Die Gegner, welche beschwört wurden, haben von Beginn an den gleichen Avatar

im Visier und laufen ihm hinterher. Der Avatar, auf den sie fokussiert sind, kann den Kreaturen jedoch keinen Schaden machen. Nur der andere Spieler:in kann Schaden an Gegner verursachen, die ihm nicht folgen. Die beiden Spieler:innen müssen ihre Avatare so manövrieren, dass sie die Kreaturen des jeweils anderen Avatars ausschalten können, ohne sich selbst in die Quere zu kommen.



Abbildung 21
Summoner Summon -
Maurice Huchler

5.2.7.2 „Big Boom“ [Tobias Bodmer]

Endboss der zweiten Ebene ist Felsbrocken „Big Boom“. Das Design von „Big Boom“ orientiert sich an der bergigen Landschaft von Ebene zwei, den kühlen Klippen. Er ist komplett aus Stein und besitzt sehr viel Kraft, mit der er große Felsbrocken bewegen kann. Er hat sehr starke Arme, die seinen ganzen Körper stützen. Die Beine selbst sind zu schwach, um seinen Körper allein zu tragen. Obwohl er ein Stein ist, ist er sehr temperamentvoll. Sein Gemütszustand ist leicht zu brechen, was sich in seinen Attacken widerspiegelt.



Abbildung 23 Big Boom Concept Art - Maurice Huchler

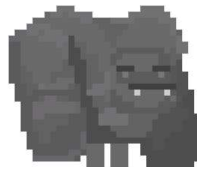


Abbildung 22 Big Boom Pixel Art - Maurice Huchler

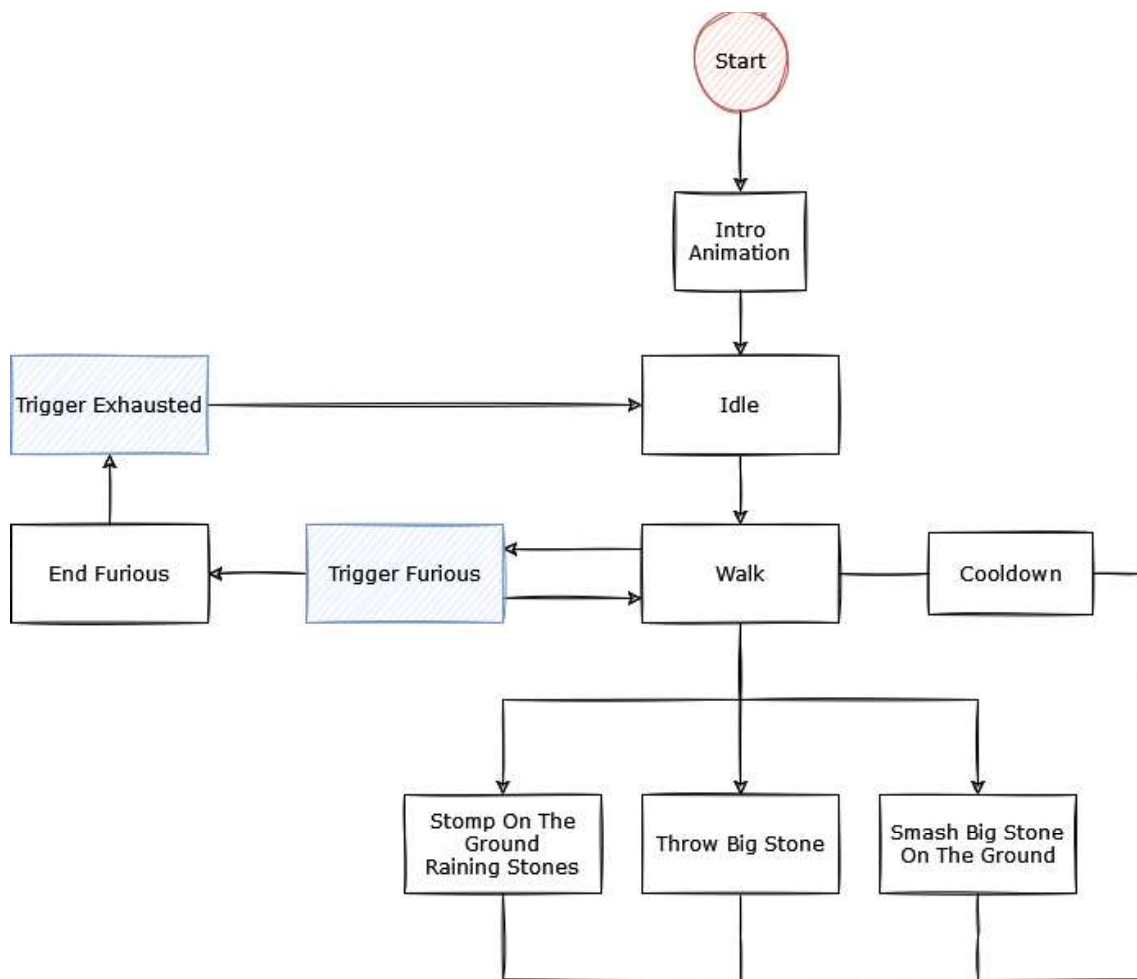


Abbildung 24 Big Boom Verhaltensmuster

Big Booms Verwundbarkeit, Schaden und Verhalten ist abhängig von seinem aktuellen Kampfmodus. Er besitzt drei unterschiedliche Modi.

Der erste Modus ist der „normale Modus“. Im normalen Modus besitzt er seine vordefinierten Werte. Sein Schaden und seine Rüstung haben keine bestimmten Änderungen. Innerhalb dieses Modus kann er zwischen drei Angriffen wählen, die er ausführen kann.

Im „wütenden Modus“ ist sein Schaden erhöht und seine Verwundbarkeit stark reduziert er bekommt keine neuen Angriffe dazu. Der „wütende Modus“ wird ausgelöst, wenn Big Boom zu schnell zu viel Schaden bekommt oder wenn eine gewisse Zeitspanne vergeht. Hier gilt er als sehr gefährlich, die Spieler:innen sollten, so gut es geht aus dem Weg gehen.

Wütend sein kann ganz schön anstrengend sein. Aus diesem Grund wechselt er anschließend in den „erschöpften Modus“. Hier ist er sehr verwundbar und bewegt sich eine gewisse Zeit lang überhaupt nichtmehr. Er schnauft durch und macht sich bereit für eine erneute Runde. Hat Big Boom sich genug ausgeruht geht er wieder in den „normalen Modus“ über.

„Stomp On The Ground – Raining Stones“:

Big Boom stampft mit voller Kraft immer wieder auf den Boden und lässt so Steine vom Himmel regnen. Diesen fallenden Steinen müssen die Spieler:innen ausweichen. Aufgrund der implementierten Schatten kann man die Flugbahn auch in der 2D Welt gut erkennen.

„Throw Big Stone“:

Bei diesem Angriff weicht Big Boom erst schnell ein Stück zurück, nimmt dann einen großen Stein und wirft diesen auf das momentan ausgewählte Ziel.

„Smash Big Stone On The Ground“:

Kommt Big Boom nah genug heran führt er den Angriff aus. Mit einem großen Felsen schlägt er zu und stampft dabei alles in den Boden was ihm in die Quere kommt. Wird der Spieler:in getroffen steckt man bis zum Kopf im Boden fest und kann sich nichtmehr bewegen. Nur durch die Hilfe des anderen Spielers ist es möglich wieder befreit zu werden.

5.2.8 Attributsystem *[Gemeinsam]*

In Ära verwenden alle Einheiten das gleiche Attributsystem. Hierfür wurde sich an Attributsystem von Binding of Isaac orientiert (vgl. Perkins Miller 2022).

5.2.8.1 Lebenspunkte

Die Lebenspunkte geben Auskunft darüber wie viel Schaden eine Einheit noch nehmen kann, bevor sie stirbt. Dafür gibt es ein Max und ein aktuelles Leben. Max Leben gibt das maximale Leben an, das durch Items erhöht werden kann und immer das Maximum des aktuellen Lebens widerspiegelt. Das aktuelle Leben zeigt die Lebenskraft an, die durch Schaden gesenkt wird und durch Heilung erhöht werden kann.

5.2.8.2 Rückstoßkraft

Ein Angriff macht nicht nur Schaden, sondern schlägt auch getroffene Einheiten zurück. Je größer die Rückstoßkraft desto weiter werden Einheiten weggeschleudert.

5.2.8.3 Angreifbar

Angreifbar gibt Auskunft darüber, ob die Einheit aktuell immun gegen Angriffe ist. Ist dieser Wert „true“ kann die Einheit zu dieser Zeit keinen Schaden erleiden.

5.2.8.4 Rüstung

Rüstung nimmt Einfluss auf den erlittenen Schaden. Je nach Höhe der Rüstung kann der Schaden enorm reduziert werden oder komplett an die Lebenspunkte der Einheit übergeben werden.

5.2.8.5 Schnelligkeit

Die Schnelligkeit bezieht sich in Ära auf das Lauftempo. Sie kann direkt durch Gegenstände beeinflusst werden oder indirekt durch das Attribut Größe, wie genaue Funktionsweise wird im Unterpunkt Größe behandelt.

5.2.8.6 Schadenspunkte

Der Schaden ist abhängig von der Waffe, die eine Einheit besitzt und deren Schadenspunkte. Die Waffe hat einen Grundschatenswert, der durch die Schadenspunkte der Einheit skaliert wird.

5.2.8.7 Abklingzeitverringern

Die Angriffsrate wird von der Abklingzeitverringern beeinflusst. Je höher die Abklingzeitverringern desto schneller kann nacheinander angegriffen werden.

5.2.8.8 Größe

Größe beeinflusst mehrere Faktoren. Einerseits nimmt es Einfluss auf die Größe der Einheit, andererseits nimmt es Einfluss auf die max. und aktuellen Lebenspunkte. Schadenspunkte, Rückstoßkraft und Schnelligkeit skalieren ebenfalls mit der Größe. Alle Einheiten werden stärker je größer die Einheit ist und schwächer je kleiner die Einheit Ausschließlich beim Lauftempo verhält sich diese Mechanik gegensätzlich.

5.2.8.9 Genauigkeit

Die Richtung der Angriffe in Ära sind abhängig von der Mausposition und der Genauigkeit. Ist diese gering, weicht die Richtung des Angriffs stark ab. Ist die Genauigkeit hingegen hoch, trifft der Schuss immer ins Schwarze.

5.2.9 Buffs [Gemeinsam]

Buffs sind meist temporäre Status Effekte, die eine gewisse Zeit anhalten können. In Ära gibt es die nachfolgenden aufgeführten Buffs, die einen Effekt auf die Statuswerte haben. Um eine direkte Rückmeldung an den Spieler:in zu geben, sind sie ebenfalls visuell dargestellt.

5.2.9.1 „Bleed“



Abbildung 25 Bleed - Maurice Huchler

Bluten senkt über Zeit das Leben der Einheit, es kann sogar dazu führen, dass die Einheit verblutet. Signal für „Bluten“ sind Blutspritzer, die aus der Einheit herausspritzen.

5.2.9.2 „Heal“



Abbildung 26 Heal - Maurice Huchler

Der Heilungsbuff erhöht über Zeit das Leben der Einheit. Der Buff kennzeichnet sich durch grüne Partikel aus, die an der Einheit nach oben strömen.

5.2.9.3 „Ice“



Abbildung 27 Ice - Maurice Huchler

Der „Ice“-Buff hat Auswirkungen auf das Lauftempo der Einheit. Das Lauftempo ist hierbei für eine Zeitspanne eingeschränkt. Visuell erkennbar ist der „Ice“-Buff an Eiskristallen und Schneeflocken.

5.2.9.4 „Exhausted“



Abbildung 28 Exhausted - Maurice Huchler

Ist eine Einheit erschöpft, so ist sie vorübergehend stark verwundbar. Sie verliert ihren Rüstungswert und erhält den gesamten eingehenden Schaden. Signal für Erschöpfung ist ein Schweißtropfen, der sich an der Einheit zeigt.

5.2.9.5 „Furious“



Abbildung 29 Furious - Maurice Huchler

Raserei erhöht temporär den Rüstungswert der Einheit und macht sie schneller. Zusätzlich halbiert es die Abklingzeit, so dass die Einheit eine erhöhte Angriffsrate bekommt. Schwebende Steine, die sich um die Einheit bewegen weisen auf Raserei hin.

5.2.9.6 „Immune“



Abbildung 30 Immune - Maurice Huchler

Der Immun-Buff sorgt dafür, dass die Einheit, die diesen Buff aktiv hat, keinen Schaden nehmen kann. Die Einheit ist in einer großen gelblichen Kugel eingeschlossen, die den Schaden von ihm ablenkt.

5.2.9.7 „Poison“



Abbildung 31 Poison - Maurice Huchler

Ist eine Einheit vergiftet, so steigen an ihr lila Blasen auf, die am höchsten Punkt platzen. Vergiftung zieht über einen Zeitraum Leben ab, dieser Zeitraum kann entweder eine gewisse Zeit anhalten oder bis die Einheit den Wert von 20 Prozent ihres maximalen Lebens unterschritten hat.

5.2.10 Items [Gemeinsam]

Items oder auch Gegenstände nehmen Einfluss auf die Attribute oder Umgebung des Spielers.

5.2.10.1 „Ice Bucket“



Abbildung 32 Ice Bucket - Tobias Bodmer

„Cools not only you but also your cooldown down.“

Der „Ice Bucket“ ist in Ära ein Internal Item das die Abklingzeit von Fähigkeiten und Angriffen verringert. Dadurch entsteht ein leichte Schadenserhöhung, ohne den Schaden direkt anzuheben. Außerdem eröffnet es dem Spieler:in mehr Möglichkeiten, schneller auf Dinge zu reagieren und Fähigkeiten öfter einzusetzen.

5.2.10.2 „Toxic Relationship“



Abbildung 33 Toxic Relationship - Maurice Huchler

„Everyone is toxic today.“

Dieses Item, ist ein zweischneidiges Schwert. Einerseits verleiht es Angriffen die Fähigkeit, Gegner zu vergiften, jedoch wie der Name schon besagt, wirkt sich dies auch auf den eigenen Avatar aus. Sobald das Item aufgenommen wird, ist der Avatar vergiftet und bekommt so lange Giftschaden, bis sein Leben auf einem geringen Level angekommen ist.

5.2.10.3 „Projectiles Up“



Abbildung 34 Projectiles Up - Tobias Bodmer

„It's dangerous to go alone, take this.“

„Projectiles Up“ erhöht die Anzahl der Projektile. Das bedeutet pro eingesammeltes Item erhöht sich die Anzahl der Geschosse um eins. Je nach Anzahl der Geschosse verändert sich auch ihre Flugbahn.

Ein Projektil

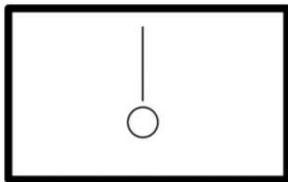


Abbildung 35 Ein Projektil - Tobias Bodmer

Zwei Projektile

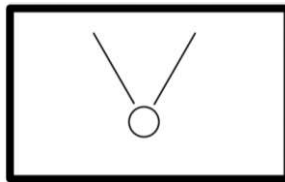


Abbildung 36 Zwei Projektile - Tobias Bodmer

Drei Projektile

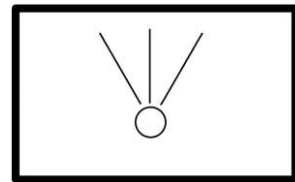


Abbildung 37 Drei Projektile - Tobias Bodmer

Die veränderte Flugbahn macht das Item zu einem Segen und Fluch zugleich. Der Spieler:in bekommt einen zusätzlichen Schuss, allerdings fliegt dieser nichtmehr in gezielte Richtung. Erst bei einem erneuten aufsammeln und insgesamt drei Projektilen entfaltet das Item seine volle Wirkung.

5.2.10.4 „Shroom Up & Down“



Abbildung 38
Shroom Down -
Tobias Bodmer



Abbildung 39
Shroom Up - Tobias
Bodmer

„Shroom Up & Down“ sind zwei unterschiedliche Items, die aber gut zusammen erklärt werden können. Sie nehmen Einfluss auf die Größe des Avatars aber nicht nur auf dessen Erscheinungsbild, sondern auch auf die Hit Box. So ist es einfacher oder schwerer gegnerischen Angriffen auszuweichen. Jedoch hat es nicht nur Vorteile klein zu sein, da sich auch der Knock back entsprechend anpasst.

5.2.10.5 „Met“



Abbildung 40 Met - Maurice Huchler

„Mhhh what a wonderful mug of Met of Gods. One sip of this delicious drink ups your health“

Das Item „Met“ erhöht das maximale Leben des Avatars um einen gewissen Wert.

5.2.10.6 „Homecoming“



Abbildung 41 Homecoming - Tobias Bodmer

„Homecoming“ verändert das Flugverhalten von Geschossen, sie versuchen nun nachhause zukommen. Das Zuhause ist hierbei der Gegner, den der Spieler:in versucht zu erwischen. Das Flugverhalten ist vergleichbar mit Tracking Missiles.

5.2.10.7 „Sharpening Stone“



Abbildung 42 Sharpening Stone - Tobias Bodmer

„A divine whetstone, sharpening your weapons with it will make them incredibly sharp. With such a weapon, even bones can be cut without effort.“

Sharpening Stone erhöht den ausgeteilten Schaden.

5.2.10.8 „Speed Up“



Abbildung 43 Speed Up - Maurice Huchler

Dieser Gegenstand ermöglicht es den Avataren sich schneller fortzubewegen.

5.2.10.9 „Slowy Slow“

Slowy Slow hüllt die Waffe des Avatars in eine Substanz die Feinde verlangsamt und sie einschränkt. Diese Eigenschaft der Substanz hält jedoch nicht für immer.

5.2.10.10 „Armor Up“

Armor Up erhöht die Rüstung des Avatars und verringert damit den erlittenen Schaden.

5.2.11 User-Interface *[Tobias Bodmer]*

5.2.11.1 Startbildschirm *[Tobias Bodmer]*



Abbildung 44 Startbildschirm

Der Startbildschirm ist das Erste was die Spieler:innen zu sehen bekommen. Hier erscheinen drei Buttons: Der Start-Button startet das Spiel und ruft die Verbindungsoberfläche auf. Der Option-Button ruft den Pause Bildschirm auf, der auch während des Spielens mit der Escape Taste erreichbar ist. Hier sollten unter anderem Einstellungen für den Sound vorgenommen werden. Diese Funktion ist nach aktuellem Stand noch nicht vorhanden. Unter dem letzten Button finden sich die Credits.

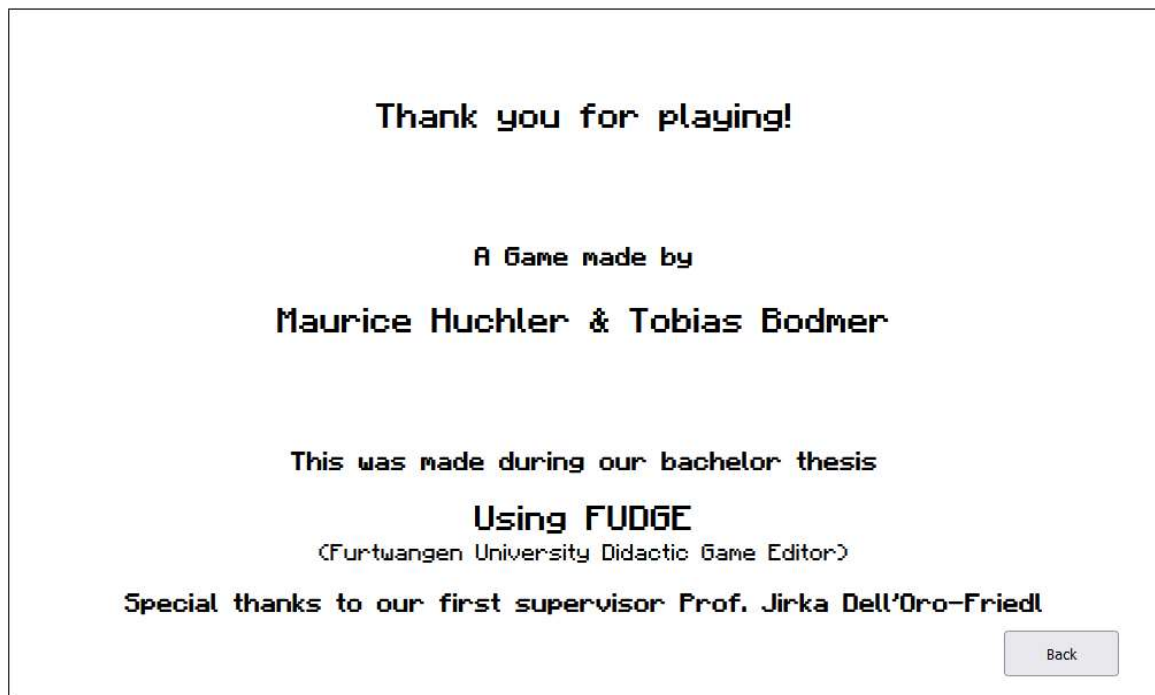


Abbildung 45 Credits

5.2.11.2 Verbindungsoberfläche [Tobias Bodmer]

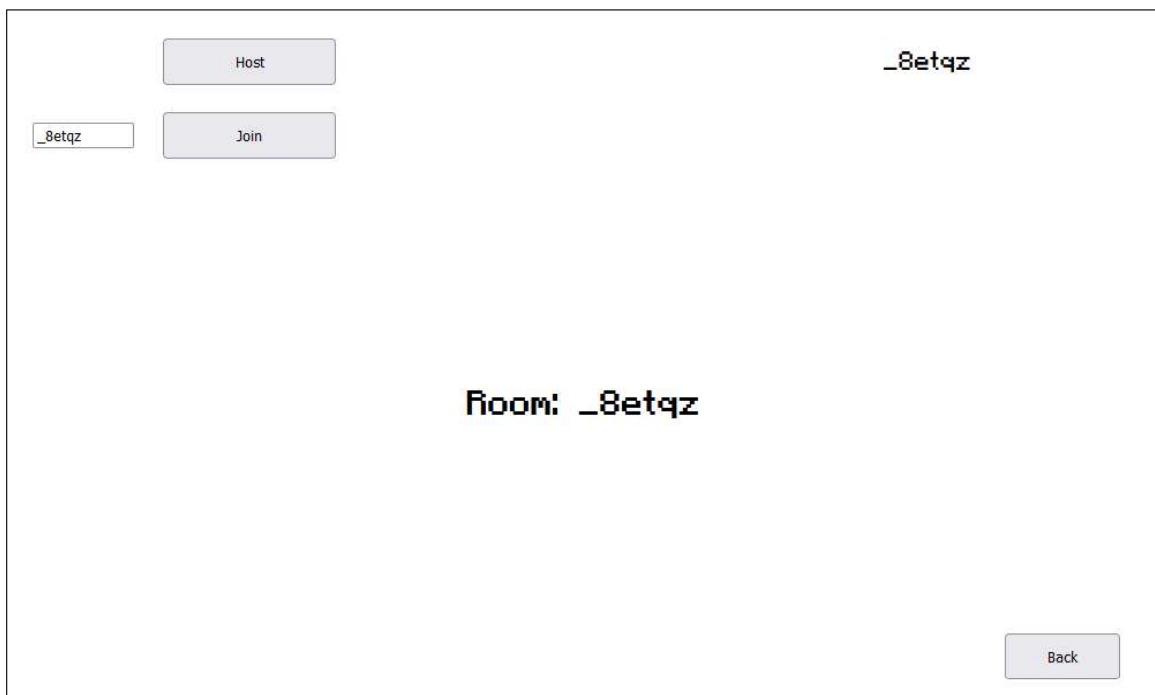


Abbildung 46 Verbindungsbildschirm

Die Verbindungsoberfläche oder auch der Host-Screen ermöglicht es den Spieler:innen sich miteinander zu verbinden. Über den Host-Button kann einer der Spieler:innen einen Raum erstellen und sieht anschließend mittig die ID des

Raums. Der Andere kann im Textfeld neben dem Join-Button die ID des Raumes angeben und diesem Raum dann per Druck auf den Join-Button beitreten. Zusätzlich gibt es noch rechts eine Liste aller aktuellen Räume und einen Back-Button, um zurück auf den Startbildschirm zu kommen.

5.2.11.3 Head Up Display *[Tobias Bodmer]*

Im Head Up Display sind alle wichtigen Informationen für den Spieler:in visualisiert, um sie so immer übersichtlich im Blick zu haben.

5.2.11.3.1 Avatar Information *[Tobias Bodmer]*



Abbildung 47 Head Up Display - Tobias Bodmer

Innerhalb der Avatar Information werden für jeden Avatar einzeln die aktuellen Lebenspunkte abgebildet. Unterhalb des Lebensbalken wird das Inventar des Avatars in Icons visualisiert.

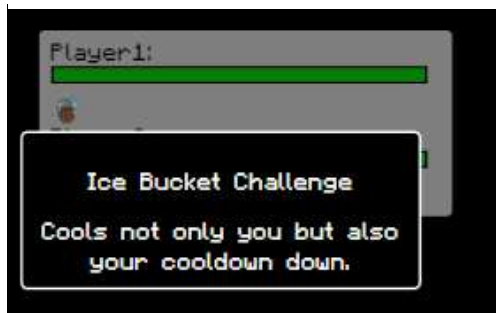


Abbildung 48 Head Up Display - Tobias Bodmer

Schwebt man mit der Maus über ein Item bekommt man über ein Pop-Up die Informationen über das Item. Angezeigt werden zum einen der Name und zum anderen die Beschreibung des jeweiligen Items. Dopplungen sind dabei nicht vorgesehen.

5.2.11.4 Minimap [Tobias Bodmer]

Die Minimap befindet sich in der rechten oberen Ecke des Displays. Verschiedene Icons zeigen hierbei die Raumtypen an.

Bossraum



*Abbildung 49
Bossraum Icon -
Tobias Bodmer*

Herausforderungsraum



*Abbildung 50
Herausforderungsraum Icon -
Tobias Bodmer*

Händlerraum



*Abbildung 51
Handelsraum Icon -
Tobias Bodmer*

Schatzraum



*Abbildung 52
Schatzraum Icon -
Tobias Bodmer*

Ein Pointer zeigt dabei in welchem Raum sich die Spieler:innen im Moment befinden.

5.2.11.5 Damage UI [Tobias Bodmer]

Die Damage UI spawnt bei jedem Treffer eine Zahl die den Gewirkten Schaden visualisiert. Dabei kommen Sprites zum Einsatz die nach einem Cooldown wieder entfernt werden.

5.2.12 Netzwerk Übertragung [Tobias Bodmer]

Der kooperative Aspekt wurde mit FudgeNet realisiert. FudgeNet ist die Netzwerkkomponente in Fudge und ermöglicht es Verbindungen zwischen Server und Client herzustellen auch Session Management ist möglich.

Wichtige Informationen wie Gegnerpositionen, Kollisionen und ähnliches werden vom Host verwaltet und an den Client gesendet, der diese wiederum verarbeitet und sein Spiel updatet. Damit wird sichergestellt das der Client und der Host synchron bleiben.

5.3 Probleme

In diesem Abschnitt werden Probleme, die während der Umsetzung entstanden sind und mögliche oder eingesetzte Lösungen aufgeführt.

5.3.1 Asynchronität

Asynchronität ist ein gravierendes Problem bei Koop oder auch Multiplayer Spielen. Zwei Geräte, die identische Berechnung ausführen, können dennoch asynchrone Ergebnisse errechnen. Daher ist es sinnvoll ein Gerät auszuwählen, das die Berechnungen durchführt und dem anderen Gerät seine Ergebnisse mitteilt. Das zweite Gerät nimmt diese Ergebnisse und führt diese bei sich aus. Es bekommt somit ein Abbild geschickt, welches es umsetzt.

Doch Asynchronität kann nicht nur wie oben beschrieben durch geteilte Berechnungen, sondern auch durch Erhöhung der zu synchronisierenden Objekten und Variablen passieren. Kommt die Client-Server Verbindung an ihre Grenzen, durch zu viele Anfragen auf dem Server, kann durch eine direkte Kommunikation zwischen den Clients eine schnellere Verbindung aufgebaut werden. Diese Kommunikation nenn man RTC-Verbindungen.

5.3.2 Verzögerte Bewegung *[Tobias Bodmer]*

Die Bewegung des Client Avatars ist eine wichtige Berechnung, die dem Client nicht oder nicht ganz überlassen werden sollte, da der Host für Berechnungen wie Kollision zuständig ist und dafür auch die genaue Position des Client Avatars benötigt.

Jedoch sieht man sich einem Problem entgegengestellt sollte man dem Client nur noch den Input entgegennehmen und diese dann dem Host senden lassen, um dort die Position des Client Avatars zu berechnen, welches wiederum and den Client sendet. Es kommt zu verzögerter Bewegung, welches von der Stärke der Verbindung zwischen Host und Client beeinflusst wird.

5.3.2.1 Client-Vorhersage *[Tobias Bodmer]*

Die Lösung für dieses Problem sind Client-Vorhersagen was auch „Client-Prediction“ genannt wird. Dabei wird dem Client das Recht seinen Avatar zu bewegen zurückgegeben, jedoch überwacht der Host die Berechnungen.

Der Client sendet seinen Input und den dazugehörigen Tick an den Host, welcher diese Informationen wiederum in einer Queue abarbeitet und seine errechneten States zurücksendet. Der Client überprüft dann seinen State mit dem des Servers. Sollte es dabei zu Asynchronität kommen, die außerhalb einer vorgegebenen Toleranz ist, muss der Client sich korrigieren. Dies macht er indem er den Avatar auf den letzten bekannten Server State zurücksetzt und die Inputs, die in der Zwischenzeit passiert sind, erneut abarbeitet. So wird sichergestellt, dass die Asynchronität auf eine vorgegebene Toleranz reduziert wird (Gabriel Gambetta 2022).

5.3.3 RTC Verbindungen *[Tobias Bodmer]*

Beim Spielen über zwei Geräte, die nicht in einem Netzwerk sind, kommt es zu Verbindungsproblemen durch die Peer-Verbindung, da diese nicht hergestellt werden kann. Jedoch wurden zwei mögliche Lösungen für dieses Problem, während dem Testen gefunden, welche nachfolgen aufgeführt werden. Die Erste ist einfach und schnell überprüft, die zweite dauert etwas länger und ist nur für eine spezielle Gruppe geeignet.

5.3.3.1 Firewall deaktivieren *[Tobias Bodmer]*

Die erste und einfachste Möglichkeit ist es die Windows Firewall zu deaktivieren. Sollten eine Antiviren Software installiert sein, muss auch diese deaktiviert werden, da diese auch das Verbinden verbieten können. Ist dies gemacht kann versucht werden sich erneut zu verbinden. Sollte weiterhin keine Verbindung geschaffen werden könnte das an installierter Antiviren Software liegen oder daran das der verwendete Browser eine integrierte Firewall besitzt. Jedoch bedeutet das in den meisten Fällen, dass diese Option nicht ausreichend ist. Eine weitere Möglichkeit gibt es für Fritz Box Nutzer.

5.3.3.2 Fritz Box Möglichkeiten *[Tobias Bodmer]*

Innerhalb des Fritz Box Dashboards gibt es die Möglichkeit Geräten die Portfreigaben zu ermöglichen. Hier fügt man das gewünschte Gerät hinzu und wählt bei IPv4 und IPv6 die Option „Dieses Gerät komplett für den Internetzugriff über ... freigeben (Exposed Host)“ aus. Diese Option macht die IPv4 und 6 sichtbar nach aus und sollte daher nicht dauerhaft aktiviert sein, da sie Eindringlingen eine Sicherheitslücke im Netzwerk eröffnet.

6 Testphase

Innerhalb einer Testphase soll der Prototyp getestet werden. Zu diesem Zweck sollen die eingebauten Elemente überprüft und potenzielle verbessert werden. Hierfür wurde ein Fragebogen zusammengestellt, der den Tester:innen vorgelegt und von diesen nach mehreren Testrunden ausgefüllt wurde.

Die Fragensauswahl umfasst die wichtigsten Elemente, die für ein erfolgreiches Spielprinzip relevant sind. Die Elementauswahl bestand aus folgenden Kategorien: Kooperation, Steuerung, Design, Spaßfaktor, User Interfaces und eigene Meinung.

6.1 Allgemeines

Bevor die Tester:innen Fragen zum Prototyp beantworten sollten, konnten sie Angaben zu ihrer Spielorientierung machen. Diese Fragen sollen zeigen ob auch Rogue-Like Neulinge Spaß am Prototypen äußern.



Abbildung 54

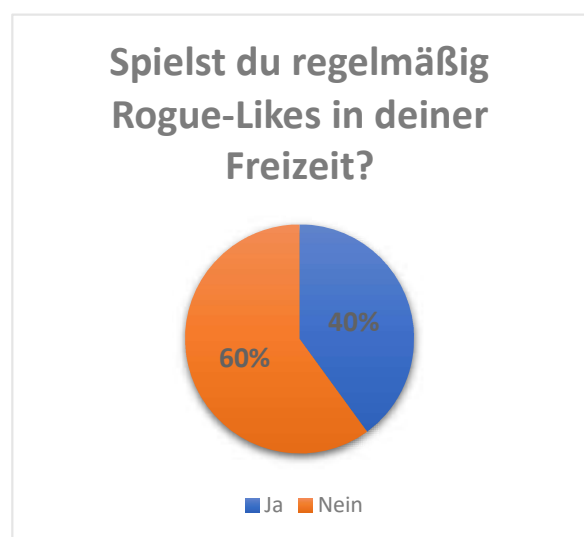


Abbildung 54

Wie in den Diagrammen zu sehen ist spielt die Mehrheit der Tester:innen zwar Videospiele in ihrer Freizeit, jedoch spielen die wenigsten von ihnen Rogue-Likes regelmäßig.

6.2 Kooperation

Bei diesem Abschnitt der Befragung ging es um die in den Prototypen bereits realisierten Koop-Elemente. Im Vordergrund stand der Kerngedanke der Zusammenarbeit und ob die umgesetzten Spielelemente das gewünschte Gefühl ausgelöst hat. Auch sollte herausgefunden werden, was das bisher interessanteste Element ist.



Abbildung 56

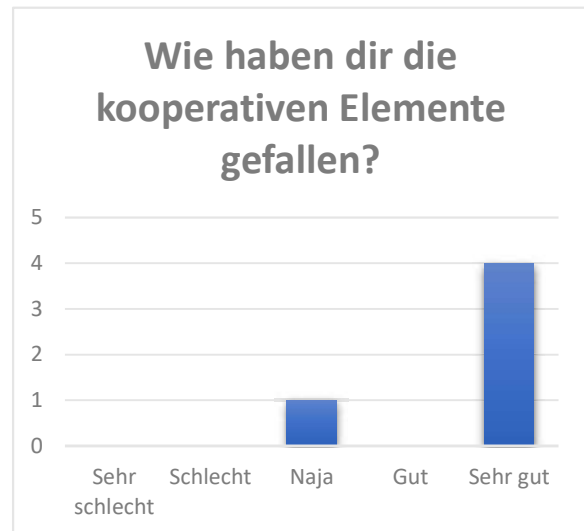


Abbildung 55

In der Befragung zeigt sich, dass nicht alle das Gefühl hatten, dass sie ihren Mitspieler:in benötigen. Das könnte daran liegen, dass nicht alle Tester:innen die gleichen Gegenstände bekommen haben und manche bereits zu Anfang ziemlich stark waren. Ebenfalls können die eingebauten Elemente noch nicht stark genug ausgereift sein, um den gewünschten Effekt zu erzielen. Im Gegensatz dazu konnten jedoch die kooperativen Elemente überzeugen. Um herauszufinden was das überzeugendste Elemente ist wurde die Frage gestellt, welche von der Mehrheit mit: „Thors Hammer“ beantwortet wurde. Manche schrieben die Summoneradds, die ebenfalls Zusammenarbeit erfordern.

6.3 Steuerung

Die Steuerung ist in schnellen Spielen wie Rogue-Likes besonders wichtig. Sie muss intuitiv sein und dem Spieler:in das Gefühl geben, volle Kontrolle über den Avatar zu haben. Vermittelt die Steuerung diese Punkte nicht können sich Tode unfair anfühlen und den Spaß am Spiel minimieren.

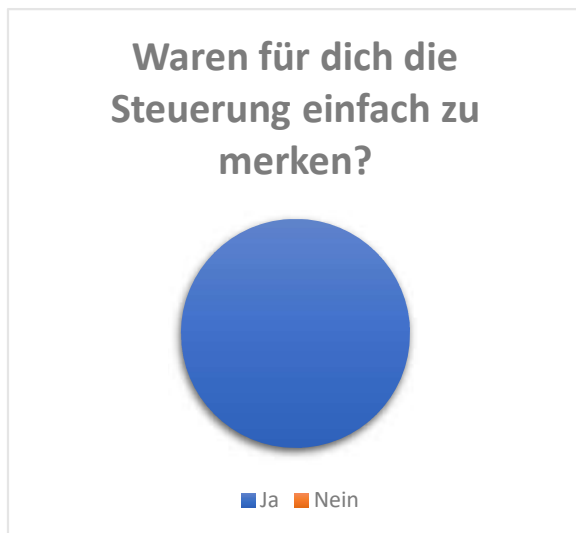


Abbildung 58



Abbildung 58

Die Antworten zeigen allerdings, dass die Steuerung den Testern:innen gut von der Hand ging und sich dabei auch gut angefühlt hat. Lediglich die Leertaste hat sich nicht allen Tester:innen direkt als Fähigkeitstaste ergeben.

6.4 Design

Im Design Abschnitt sollte vor allem herausgefunden werden, ob die Umgebung und Gegner sowie die Gegenstände stimmig sind und dem Spieler:in das Gefühl geben, sich wirklich in einer nordischen Zwischenwelt befinden, die den Weg nach Walhalla zeigt.

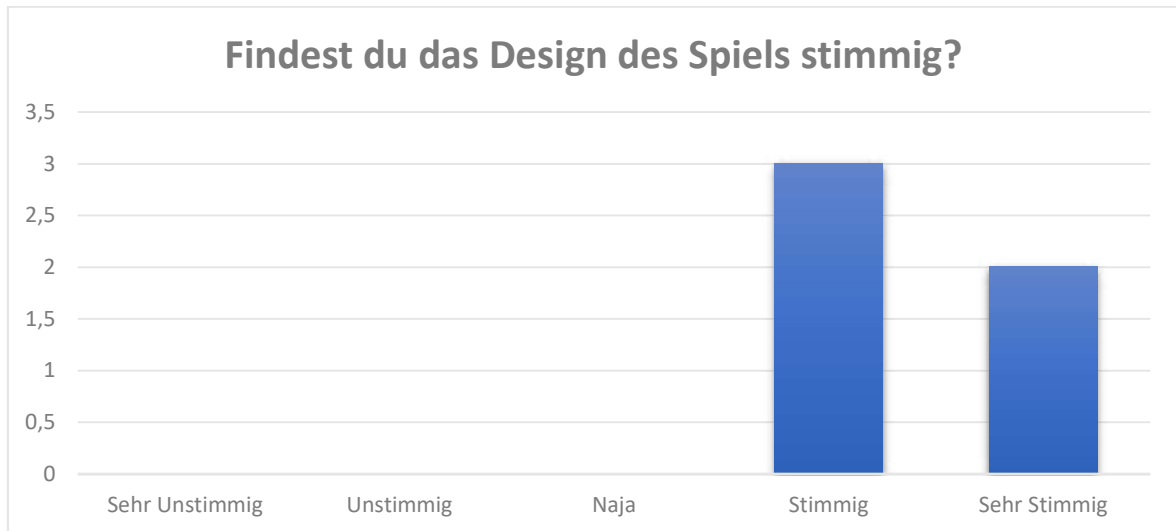


Abbildung 59

Die Tester:innen tendieren beim Design in eine sehr klare Richtung. Sie haben das Gefühl, dass das Spiel sehr stimmig ist und einen die Welt überzeugen kann. Keiner fand das Design oder auch die Welt mit den Gegenständen und Gegner unstimmig.

6.5 Spaßfaktor

Viele Spiele können mit einem schönen Design oder einer guten Steuerung überzeugen, doch viel wichtiger ist, haben die Spieler:innen Spaß? Haben die Spieler:innen Spaß am Spielen, ist die Wahrscheinlichkeit des erneuten Spiels stark erhöht.

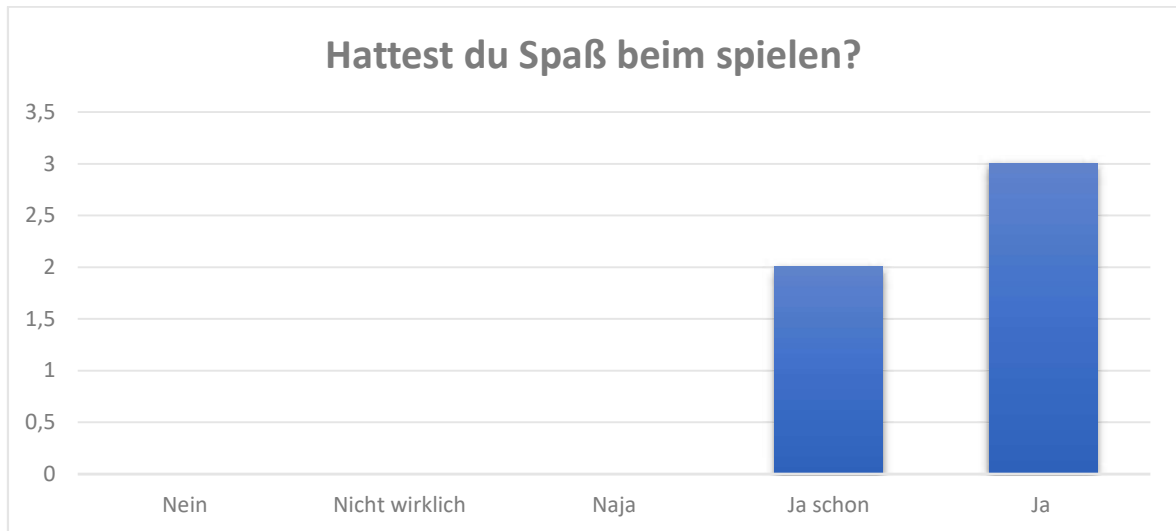


Abbildung 60

Auch hier haben die Tester:innen wieder eine sehr eindeutige Meinung: Keiner der Tester:innen empfand die Spielzeit als nicht Spaßig. Es hat allen gefallen und Spaß gemacht.

6.6 User Interfaces

User Interfaces geben Auskunft über die wichtigste Information, wie beispielsweise das Leben der Avatare, gefundenen Gegenstände oder Interaktionsmöglichkeiten. Sie geben ein Visuelles Feedback für Dinge, die dem Spieler:innen sonst nicht auffallen würden. Daher ist es wichtig das die UI alle wichtigen Informationen zeigt und dennoch übersichtlich bleibt.



Abbildung 62

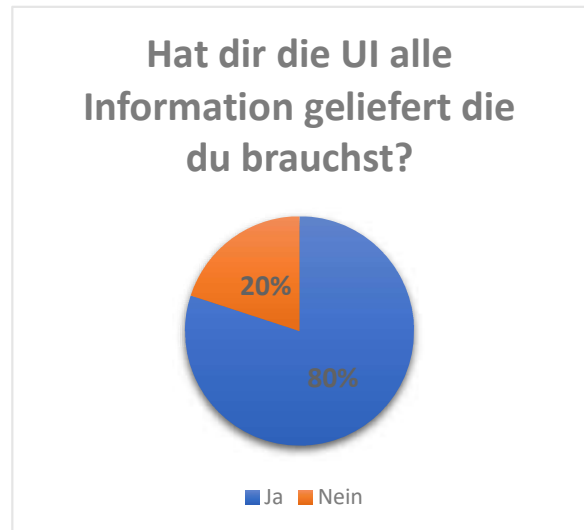


Abbildung 62

Hier gehen die Meinungen doch etwas auseinander. Die Übersichtlichkeit kann noch weiter ausgebaut werden und sollte überarbeitet werden. Jedoch sind die gelieferten Informationen ausreichend. 20 Prozent der Tester:innen haben eine oder mehrere Informationen gefehlt.

6.7 Eigene Meinung

In der eigenen Meinung wurde den Tester:innen die Möglichkeit gegeben auch einen Freitext zu verfassen, um mögliche Probleme oder Anmerkungen zu hinterlassen.

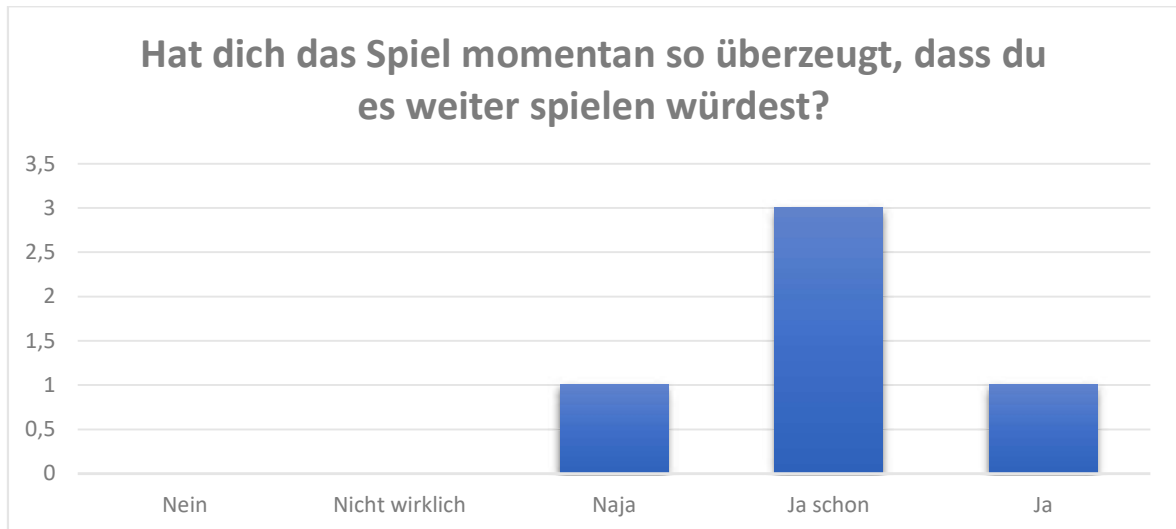


Abbildung 63

Das Spiel konnte die meisten Tester:innen überzeugen und würden auch das Spiel in einem Finalen Status gerne spielen. Nachfolgend werden noch die Freitexte der Tester:innen abgebildet.

„Der Basis-Gedanke des Spiels ist gut, das Layout mit der dynamischen Map und den Ebenen gefällt mir. Die Boss-Kämpfe sind cool. Gerne auch mehr Coop-Räume wie den mit Thors Hammer. Das man beim Einstieg in die normalen Räume so "überfallen" wird ist ggf. noch ein Balancing-Thema.“

„Es braucht an verschiedenen Ecken noch Feinschliff, funktioniert als Gesamtwerk aber gut.“

„Das Spiel hat auf jeden Fall großes Potenzial in Sachen Spielspaß und Design. Die Unterschiede der Schwierigkeitsstufen in den Ebenen könnten noch ein bisschen auffälliger sein aber die vorhandenen Gegner machen Spaß!“

„Bei der UI fehlt Anzeige für die Ability-Cooldown. Monsterrange etwas unklar aber sonst top.“

„An sich find ich das Spiel ganz lustig, fehlende Animationen und Sprites / UIs (Attack Cooldown; Dash Cooldown; Pfeile, welche zum Boss & Teammate zeigen, wenn sie außerhalb vom Screen sind; Steuerung anzeigen; Hinweise, wie kooperative Elemente funktionieren) machen das Game ohne Vorwissen etwas schwierig zu spielen, wenn dies jedoch alles gefixt wird, könnte ich mir vorstellen, es aktiver zu spielen“

Das erhaltene Feedback erweist sich als hilfreich bezüglich der zukünftigen Weiterentwicklung, da es den derzeitigen Stand, das Potenzial, sowie Verbesserungsmöglichkeiten aus Sicht der User:innen aufzeigt.

6.8 Zusammenfassung der Testphase

Zusammengefasst lässt sich sagen, dass die Testphase überwiegend positiv ausgefallen ist. Es gab eine geringe Anzahl an negativen Äußerungen, welche gut umsetzbar sind.

Das Konzept und die Idee hingegen funktionieren sehr gut und sind auf keine Kritik gestoßen. Dies zeigt, dass die Spiel-Idee, sowie das Setting und das Design Potenzial haben ein überzeugendes und spaßiges Spiel zu gestalten.

7 Ausblick

Der Prototyp beinhaltet viele der geplanten Features. Dennoch gibt es einiges zu tun, um ein vollwertiges Spiel zu entwickeln. Einige Konzepte haben es noch nicht in den Prototyp geschafft, auch wenn diese schon fertig auf dem Papier sind. Andere wichtige Elemente wurden vorerst außer Acht gelassen und als sekundär eingestuft. Des Weiteren sind während der Testphase mithilfe der Tester viele Verbesserungsmöglichkeiten aufgekommen.

Für die Zukunft kann noch an vielen Schrauben gedreht und Baustellen gearbeitet werden. Ein großer eigener Bereich ist Sound-Design. Hier können Sound-Effekte wie schlagen, laufen und ausweichen, eingebaut werden, so wie Hintergrundmusik und Umgebungsgeräusche. Das alles trägt zur Immersion bei und hat einen Einfluss auf die Spieler.

Der Anfang und die Rahmenbedingungen der Geschichte stehen. Was noch alles auf der Reise passiert, ist noch offen. Ebenso offen ist, wie die Geschichte erzählt wird und mit welchem Ausmaß. Ihre Ausgestaltung gibt dem Spiel einen eigenen Character.

Für den Prototyp wurde sich sehr stark auf die erste Ebene konzentriert. In der Geschichte gibt es noch zwei weitere Ebenen, die relevant sind. Diese zu gestalten und eigene Gegner auszudenken kann in der Zukunft angegangen werden. Einzigartige Gegner, die nur in dieser Ebene erscheinen mit eigenen Verhalten. Ebenso soll ihr aussehen, so wie bei Gegnern in der ersten Ebene an die Umgebung angepasst sein.

Es gibt kooperative Items innerhalb des Prototyps. Um die Kooperation zwischen den beiden Spielern zu stärken, ist ein Ziel mit großer Priorität die Umsetzung von neuen Items. Gleichgesetzt mit den Items sind neue Herausforderungen. Im Abschnitt Herausforderungen wurden bis auf Thors Hammer Herausforderungen beschrieben, welche im Spiel vorstellbar wären, aber noch nicht umgesetzt wurden.

Ein weiteres gutes Spielelement, welches gut vorstellbar wäre, sind "Area of Effects". Kurz: AOE. AOE's sind Flächen, in denen ein bestimmter Effekt angewandt wird. Dies äußert sich mit einem Buff, Schaden oder Heilung. Items oder Gegner, die solche AOE's mit sich tragen, eröffnen neue Möglichkeiten der Interaktion.

Bislang gibt es zwei Avatare zur Auswahl. Einen Fern- und Nahkampf. Neue Charaktere mit eigenen Fähigkeiten ist ebenfalls geplant. So soll eine neue Klasse: der Heiler, der nächste spielbarer Avatar sein.

Auch fehlen noch visuelle Feedbacks und Indikatoren wie beispielsweise eine Anzeige für die Abklingzeit der Ability, die anzeigt, wann die Abilitys wieder zur Verfügung stehen. Auch wurde in der Testphase erwähnt, dass eine Anzeige für den Boss und Mitspieler brauchbar wäre, wenn sich diese außerhalb des Sichtradius befinden. Eine sinnvolle Umsetzung wäre ein roter Pfeil am Rand des Bildschirms, der die ungefähre Position vom Boss anzeigt und ein blauer Pfeil für die Position des Mitspielers.

Der Gameloop ist noch nicht vollendet, es fehlt noch ein Death Screen und die Funktion, einen neuen Run zu starten, ohne das ganze Spiel Neustarten zu müssen. Es ist ebenfalls noch nicht möglich, einen Run zu speichern, um diesen später weiter spielen zu können.

8 Fazit

Abschließend können wir sagen, dass es uns definitiv gelungen ist, ein spielbarer Prototyp in FUDGE zu realisieren. Dieser konnte in einer Testphase überzeugen und zeigt dadurch, dass sich unsere Idee ausgezahlt hat.

Trotz der Probleme mit den RTC-Verbindungen, sehen wir den Prototypen als gelungen an. Es ist uns gelungen, die geplanten kooperativen Elemente einzubauen und so umzusetzen, wie wir sie uns vorgestellt haben. Dennoch muss man sagen, dass die Elemente nicht immer die gewünschte Wirkung haben, die wir uns vorgestellt hatten. Wir konnten eine Vielzahl von Spielmechaniken einbauen, welches sich ab und zu als Probleme herausstellte. Während der Prozess der Programmierung kam es zu Problemen, da wir uns zu viel vorgenommen hatten. Eine gute Aussortierung der wichtigen Features ist zu Beginn sehr wichtig. Dadurch kann sich darauf konzentriert werden und es werden weniger Ablenkungen geschaffen durch andere sekundäre Funktionen.

Mit den Verbesserungen, die im vorangegangenen Abschnitt behandelt wurden, lässt der Prototyp sich zu einem vollständigen Spiel erweitern, welches eine Vielzahl an Spieler ansprechen wird.

Die gesammelten Erkenntnisse nehmen wir als sehr positiv auf. Wir selbst bekamen dadurch einen tieferen Einblick in Netzwerke und wie Multiplayer spiele funktionieren. Wir sind stolz, ein eigenes Konzept und darauf aufbauend einen spielbaren Multiplayer-Prototyp mithilfe von FUDGE erstellt zu haben und freuen uns auf die weiteren Entwicklungen.

9 Literaturverzeichnis

Aesprite (2022). Aesprite Documentation. Online verfügbar unter

<https://www.aesprite.org/docs/> (abgerufen am 21.06.2022).

Ben Eater (2022). Boids algorithm. Online verfügbar unter <https://eater.net/boids>

(abgerufen am 25.06.2022).

Boris (2022). Dungeon Generation in Binding of Isaac. Online verfügbar unter

<https://www.boristhebrave.com/2020/09/12/dungeon-generation-in-binding-of-isaac/> (abgerufen am 17.06.2022).

Company GitHub (2022). Github Features. Online verfügbar unter

<https://github.com/git-guides> (abgerufen am 17.06.2022).

David Schiffer (2022). WALHALLA. Online verfügbar unter

<https://wikingersturm.com/blogs/nordische-mythologie/walhalla> (abgerufen am 17.06.2022).

Gabriel Gambetta (2022). Client-Side Prediction and Server Reconciliation. Online

verfügbar unter <https://www.gabrielgambetta.com/client-side-prediction-server-reconciliation.html> (abgerufen am 28.06.2022).

Mitchell Loewen (2020). What is a Roguelike? Online verfügbar unter

<https://www.cloudfallstudios.com/blog/2020/7/13/what-even-is-a-roguelike> (abgerufen am 21.06.2022).

Perkins Miller (2022). Binding of Isaac: Rebirth Wiki. Online verfügbar unter

<https://bindingofisaacrebirth.fandom.com/wiki/Attributes> (abgerufen am 23.06.2022).

Salesforce (2022). What is Heroku? Online verfügbar unter

<https://www.heroku.com/what> (abgerufen am 17.06.2022).

TypeScript (2022). What is TypeScript? Online verfügbar unter

<https://www.typescriptlang.org/> (abgerufen am 17.06.2022).

Ulrich Wimmermoth (2020). Nur noch eine Runde: 6 Roguelikes, die euch an den Bildschirm fesseln. Online verfügbar unter

<https://blog.de.playstation.com/2021/09/16/nur-noch-eine-runde-6-roguelikes-die-euch-an-den-bildschirm-fesseln/> (abgerufen am 17.06.2020).

10 Eidesstattliche Erklärung Huchler Maurice

Ich versichere, dass ich die vorstehende Arbeit selbständig verfasst und hierzu keine anderen als die angegebenen Hilfsmittel verwendet haben. Alle Stellen der Arbeit, die wörtlich oder sinngemäß aus fremden Quellen entnommen wurden, sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt oder an anderer Stelle veröffentlicht.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben kann.

Landau, 28.06.2022



Ort, Datum Name

11 Eidesstattliche Erklärung Bodmer Tobias

Ich versichere, dass ich die vorstehende Arbeit selbständig verfasst und hierzu keine anderen als die angegebenen Hilfsmittel verwendet haben. Alle Stellen der Arbeit, die wörtlich oder sinngemäß aus fremden Quellen entnommen wurden, sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt oder an anderer Stelle veröffentlicht.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben kann.

Balingen, 28.06.2022 *Tobias Bodmer*

Ort, Datum Name