

BACHELORARBEIT

Effekte eines adaptiven Schwierigkeitsgrades auf Flow in Videospielen

vorgelegt am 28. Februar 2025

Felix Kock

Matrikelnummer: 

Erstprüfer: Prof. Dr. Eike Langbehn
Zweitprüfer: Simon Dewert

**HOCHSCHULE FÜR ANGEWANDTE
WISSENSCHAFTEN HAMBURG**

Department Medientechnik
Finkenau 35
20081 Hamburg

Abstrakt

In dieser Arbeit werden die Effekte eines adaptiven Schwierigkeitsgrades auf Flow in Videospielen untersucht. Die adaptive Schwierigkeit wird in Form von verschiedenen Gegnerkompositionen, die dynamisch während der Spielsitzung ermittelt werden, gegen starr vorhersagbare Gegnerkompositionen gestellt. Dem Spiel in beiden Versionen liegen Umfragebögen bei, welche die Merkmale beziehungsweise Bausteine des Flowzustands abfragen. Durch das Gegenüberstellen der Ergebnisse kann dann ermittelt werden, inwieweit der adaptive Schwierigkeitsgrad Einfluss auf den Flowzustand hat. Dies hat ergeben, dass ein messbarer Unterschied zwischen den Versionen vorhanden ist.

Abstract

In this paper the effects of an adaptive difficulty level in regards to flow in video games were investigated. This adaptive difficulty was realized in the form of dynamically constructed waves of differing enemy types, as opposed to fixed wave compositions. Both versions of the game, adaptive and static, were be played and the participants filled out a questionnaire afterwards. By comparing the scored points of the two game versions, their respective effect on the flowstate was analyzed. The results hinted at a measurable difference between the two versions.

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Ziele	1
1.3	Methode	2
2	Grundlagen	3
2.1	Motivation	3
2.2	Engagement, Vertiefung und Immersion in Videospielen	6
2.3	Flow	8
2.4	Spieltypen	11
2.5	Schwierigkeit in Spielen	11
3	Das Spiel	14
3.1	User Interface	14
3.2	Spielmechaniken	16
3.3	Gegnertypen	18
3.4	Schwierigkeitskurve/Wavespawner	19
3.5	Technische Implementation	20
4	Ergebnisse der Umfrage	27
4.1	Ablauf	27
4.2	Spielbezogene Fragen	27
4.3	Spielmusterbezogene Fragen	32
4.4	Demographische Daten	35
4.5	Freitextfeld	36
4.6	Auswertung der Ergebnisse	36
5	Ausblick und Fazit	38
A	Anhang - Umfrageergebnisse	43

Abbildungsverzeichnis

1	Der Flowzustand als Verhältnis von Herausforderungen und Fähigkeiten (eigene Darstellung nach [48])	8
2	Spielansicht (<i>Quelle: Eigene Darstellung</i>)	14
3	Steuerungsübersicht (<i>Quelle: Eigene Darstellung</i>)	15
4	Healthbar (<i>Quelle: Eigene Darstellung</i>)	15
5	Game Over Screen (<i>Quelle: Eigene Darstellung</i>)	15
6	Der Spielcharakter (<i>Quelle: Eigene Darstellung</i>)	16
7	Normaler Angriff (<i>Quelle: Eigene Darstellung</i>)	16
8	Spieler mit Projektil (<i>Quelle: Eigene Darstellung</i>)	17
9	Parieren (<i>Quelle: Eigene Darstellung</i>)	17
10	Spielcharakter im Dash (<i>Quelle: Eigene Darstellung</i>)	18
11	Simpler Gegner (<i>Quelle: Eigene Darstellung</i>)	18
12	Fernkampfgegner (<i>Quelle: Eigene Darstellung</i>)	19
13	Schwerer Gegner (<i>Quelle: Eigene Darstellung</i>)	19
14	Diagramm zur Veranschaulichung des Programms(<i>Quelle: Eigene Darstellung</i>)	20
15	Hast du Version A oder B gespielt? (<i>Quelle: Eigene Darstellung</i>)	28
16	Die Schwierigkeitskurve über die Wellen hinweg war angemessen (<i>Quelle: Eigene Darstellung</i>)	28
17	Du konntest dich auf das Spiel konzentrieren (<i>Quelle: Eigene Darstellung</i>)	29
18	Dir war klar, was zu tun ist (<i>Quelle: Eigene Darstellung</i>)	29
19	Das Spiel hat dir ausreichend Feedback über den Spielverlauf geboten (<i>Quelle: Eigene Darstellung</i>)	30
20	Du warst in das Spiel vertieft (<i>Quelle: Eigene Darstellung</i>)	30
21	Die Reaktion des Spiels auf deinen Input war entsprechend schnell und nach deiner Erwartung (<i>Quelle: Eigene Darstellung</i>)	31
22	Du konntest in das Spiel eintauchen (<i>Quelle: Eigene Darstellung</i>)	31
23	Du konntest dich im Spiel verlieren (<i>Quelle: Eigene Darstellung</i>)	32
24	Wieviel Erfahrung hast du mit Videospielen? (<i>Quelle: Eigene Darstellung</i>)	32
25	Achievements und dergleichen sind dir in Videospielen wichtig (<i>Quelle: Eigene Darstellung</i>)	33
26	Die Möglichkeit zu Erforschen ist dir in Videospielen wichtig (<i>Quelle: Eigene Darstellung</i>)	33
27	Du bevorzugst Multiplayerspiele gegenüber Singleplayerspielen (<i>Quelle: Eigene Darstellung</i>)	34
28	Es ist wichtig für dich in Videospielen möglichst “gut” zu sein (<i>Quelle: Eigene Darstellung</i>)	34
29	Wie oft spielst du Videospiele? (<i>Quelle: Eigene Darstellung</i>)	35
30	Wie alt bist du? (<i>Quelle: Eigene Darstellung</i>)	35
31	Wie identifizierst du dich? (<i>Quelle: Eigene Darstellung</i>)	36
32	Gibt es sonst noch Anmerkungen? (<i>Quelle: Eigene Darstellung</i>) .	36

Abkürzungsverzeichnis

CET Cognitive Evaluation Theory. 5

OIT Organismic Integration Theory. 4

PLOC Perceived Locus of Causality. 3

SCI-Modell Sensory-, Challenge-Based-, Imaginative-Immersion-Modell. 7

SDT Self-Determination Theory. 4, 5

1 Einleitung

Zunächst werden Motivation, Ziele und die Methode dieser Arbeit präsentiert. Anschließend folgen die Grundlagen, Spiel- und Umfragenaufbau sowie die Evaluation der Ergebnisse.

1.1 Motivation

Videospiele sind eine etablierte Freizeitbeschäftigung, deren Popularität weiter anwächst [36] [49]. Wegen des stetig wachsenden Zugangs zu Aktivitäten und Spielen ist dies heute mehr denn je der Fall [43]. Sie haben eine Entwicklung von vergleichsweise simplen interaktiven Spielen bis hin zu riesigen, fortlaufenden interaktiven Welten durchlaufen [17]. Die Urnatur von Videospielen ist die einer reinen Freizeitaktivität. Zwar haben sie auch eine soziale Komponente im Zuge von Multiplayerspielen oder eine künstlerische Ebene angenommen, dennoch steht im Zentrum jedes Spiels die Freude am Spielen. Der primäre Reiz mag für jedes Individuum anders sein, jedoch sind Engagement und Immersion ausschlaggebend für die empfundene Freude und dementsprechend für die Entscheidung, ein Videospiel zu spielen. Hier ist allerdings zu unterscheiden, um welche Art von spielender Person es sich handelt: Periodisch Spielende sind eher extrinsisch motiviert, durch Freundeskreis oder ähnliches, während regulär Spielende eher intrinsisch motiviert sind. [49]

Es gibt viele Gründe, Videospiele zu spielen. Insbesondere am Anfang ihrer Entwicklung waren diese Gründe oft hedonischer Natur, dienten also dem Spaß und der Freude [57]. Jedoch gibt es zunehmend Spiele, die eine eudaimonische Erfahrung bieten, also eine ernstere, emotionale Erfahrung darstellen und somit die spielende Person anziehen [28]. In Kombination mit den vielfältigen Genres und Sub-Genres von Videospielen [17] stellen sie ein facettenreiches Medium dar.

Jedoch gibt es oft unterschiedlichste Gegnertypen, die in ihrem Verhalten ganz eigene Herausforderungen darstellen. Ob diese Verhaltensweisen oder eher die Komposition der Gegner einen nennenswerten Einfluss auf das Erreichen des Flowzustands haben, versucht diese Arbeit zu ermitteln.

1.2 Ziele

In dieser Arbeit wird ermittelt, ob der von Csikszentmihalyi beschriebene Zustand des Flows durch adaptive Schwierigkeit zustandegebracht werden kann. Dies geschieht mithilfe eines Spiels, bei dem zwei Versionen gespielt werden. Danach soll ein Fragebogen ausgefüllt werden, um die Merkmale des Flowzustands nach Csikszentmihalyi zu ermitteln und auszuwerten, ob diese Veränderung einen Einfluss auf den Zustand hat. Der Unterschied der beiden Versionen liegt in der veränderten Gegnerkomposition der Wellen, womit etwaige Änderungen bzw. Erkenntnisse eindeutig auf diese Anpassung zurückzuführen sind, und nicht durch zu viele veränderte Variablen beeinflusst werden. Dementsprechend wird am Ende der Arbeit, durch Gegenüberstellen der Ergebnisse, untersucht inwieweit der adaptive Schwierigkeitsgrad Einfluss auf den Flowzustand der spielenden Personen hat.

1.3 Methode

Im Rahmen dieser Arbeit wurde ein Videospiel in zwei Versionen entwickelt. Dieses Spiel ist ein Top-Down Arena Wave Beat 'Em Up. Die Versionen unterscheiden sich in der Komposition der Gegnerwellen ab der vierten Welle. In der 'statischen' Version gibt es zwischen den Spielsitzungen keine Unterschiede. Die 'adaptive' Version sammelt die Daten des vom Spielcharakter erlittenen Schaden und baut die späteren Wellen basierend darauf zusammen.

Die Befragten füllen nach spielen der jeweiligen Version einen Fragebogen zu dieser aus. Basierend auf dem Vergleich der Antworten wird evaluiert inwieweit die Adaptivität einen Einfluss auf den Flowzustand der Spielenden hatte.

2 Grundlagen

In diesem Kapitel werden einige Grundlagen, die in das Design des Spiels, den Aufbau des Experiments und der Umfrage mit einfließen, erläutert.

2.1 Motivation

'To be motivated means to be moved to do something.'

Motiviert zu sein, bedeutet, dazu bewegt zu sein, etwas zu tun. [53] Die Quelle dieser Bewegung, also die Quelle der Motivation, ist eine immer wiederkehrende Forschungsfrage in der Psychologie [53]. Es gibt mehrere Modelle der Motivationsforschung, hierbei sind die Konzepte von extrinsischer und intrinsischer Motivation von Relevanz:

- Self-Determination Theory [53]
- Organismic Integration Theory [53]
- Theory of Flow Experience [26]
- Attribution Theory [59][37][58]

Zuerst werden die zugrundeliegenden Konzepte von intrinsischer und extrinsischer Motivation erforscht, sowie den sogenannten Perceived Locus of Causality und Regulatory Styles. Und schließlich die verschiedenen Motivationstheorien genauer betrachtet.

Ryan und Deci [53] definieren und verwenden mehrere Begriffe im Zusammenhang der Motivationsforschung. Diese sind:

PLOC oder *Perceived Locus of Causality* ist ein von Heider [38] geprägter und von deCharms [29] erweiterter Begriff, welcher sich auf die empfundene Quelle der Motivation bezieht. 'Empfunden' wird deshalb verwendet, weil die Art der Motivation im Zusammenhang mit der Quelle Einfluss auf die Motivation hat. Bei extrinsischen Quellen der Motivation ist der subjektive Standpunkt zu ebenjener Quelle von Bedeutung, da die Nähe beziehungsweise die eigene Position relativ zum Motivator für mehr oder weniger Internalisierung der Motivation verantwortlich ist [52].

Amotivation ist die Abwesenheit von Motivation und ist charakterisiert durch das Ausführen keiner Tätigkeit, oder das Ausführen ohne Absicht [53]. Sie ist zu betrachten als ein stumpfes Ausführen der erforderlichen Aktionen, jedoch von extrinsischer Motivation zu trennen, da entweder gar keine Tätigkeit ausgeführt wird, oder ein willenloses Handeln vorliegt [53]. Die Amotivation stammt aus der Empfindung, dass die jeweilige Aktion keinen Wert für das Individuum hat [55], einer empfundenen Kompetenzlosigkeit [18] oder der Erwartung, dass die jeweilige Tätigkeit nicht das gewünschte Ergebnis erzielt [56].

Extrinsische Motivation stammt aus einer externen Quelle. Hierbei ist der Ausdruck des regulatorischen Stils von Interesse, da laut Ryan et al. [53] ein Unterschied, basierend auf der Position des Individuums zum extrinsischen Motivator existiert. Im Rahmen der *Organismic Integration Theory* werden diese noch genauer erläutert.

Intrinsische Motivation stammt aus dem *Inneren*. Ein jeweiliges Subjekt ist also von sich aus motiviert, etwas zu tun. Weil sie dementsprechend höchst individuell ist, kann nicht generalisiert werden, was als intrinsische Motivation gilt, da der Antrieb etwas zu tun von Mensch zu Mensch unterschiedlich ist und sich die Tätigkeiten zu denen sie motiviert sind unterscheiden. Sie repräsentiert die höchste Form der Autonomie und die prototypische Instanz der Selbst-Bestimmung [53].

Self-Determination Theory ist eine von Ryan und Deci erstellter Ansatz zur Quelle von menschlicher Motivation [30]. Self-Determination Theory basiert auf empirischen Methoden und betont die evolutionsbedingten inneren Ressourcen für Persönlichkeitsentwicklung und Selbstregulierung des Verhaltens [54]. Zentral in dieser Theorie stehen nachfolgende Bedürfnisse, welche vorhanden sein müssen, um optimale Funktion und Wachstum zu erlauben:

- Kompetenz: Das Verlangen den Ausgang zu beeinflussen und Versiertheit zu erfahren [53].
- Angehörigkeit: Das Verlangen mit anderen Individuen zu interagieren und positiven Austausch zu erfahren [53].
- Autonomie: Das Verlangen, dass eigene Leben zu kontrollieren und auf eine Art und Weise zu handeln, die dem eigenen Wertesystem entspricht [53].

Organismic Integration Theory oder *Organismic Integration Theory* ist eine von Ryan und Deci definierte Subtheorie zur obigen SDT und differenziert den Begriff der extrinsischen Motivation. Sie stellt ein Continuum von extrinsischen Motivationsquellen in Verbindung mit dem Regulierungsstil dar, welche, je nach Grad der Internalisierung, einen entsprechenden Einfluss auf die Autonomie der Regulierung haben [53]. Diese Regulierungsstile sind als die Art der Expression der Motivation zu verstehen. So sind extrinsische Motivatoren nicht für jedes Individuum äquivalent, sondern davon abhängig, inwieweit sich beispielsweise ein Individuum mit der präsentierten Motivation identifiziert, oder sie als 'Notwendigkeit' einer höhergestellten intrinsischen Motivation betrachtet. Nach Ryan und Deci [53] werden vier Regulierungsstile im Rahmen der extrinsischen Motivation differenziert, sowie je einer für Amotivation und intrinsische Motivation.

- *External Regulation* sind Verhaltensweisen, die oft an eine externe Belohnung oder Nachfrage gekoppelt sind. Der empfundene Locus ist extern.
- *Introjected Regulation* ist teilweise internalisierte Motivationen, die an das Vermeiden von Schuld und Schuldbewusstsein oder die Befriedigung

von Ego, beziehungsweise Stolz, gekoppelt sind. Auch wenn ein intrinsischer Teil vorhanden ist, ist der Locus extern.

- *Identified Regulation* ist eine deutlich autonomere Form als die oben genannten Regulierungsstile. Diese erfolgt, wenn sich mit dem motivierenden Faktor identifiziert oder dieser als wichtig erachtet wird. Dadurch bewegt sich der empfundene Locus ins Interne.
- *Integrated Regulation* ist die autonomste Form der extrinsischen Motivation. Sie ähnelt der intrinsischen Motivation am meisten, ist jedoch von dieser zu trennen, da trennbare Ziele verfolgt werden, anstatt dass die Tätigkeit autotelisch ist.

Cognitive Evaluation Theory ist eine weitere Subtheorie der SDT und fokussiert die intrinsische Motivation des Menschen. Ihr primärer Fokus liegt auf der Analyse des Zwischenspiels zwischen internen und externen Faktoren, insbesondere dem Abnehmen intrinsischer Motivation bei zunehmend externen Einflüssen, wie Strafe oder Belohnung, bezüglich der Tätigkeit [51].

Attribution Theory ist eine von Weiner [58] und von Harvey und Martinko [37] aufgestellte, sowie von Seligman [56] erweiterte Theorie, welche das Verhalten in Bezug auf das Zuschreiben von Attributen beschreibt. Grundlage hierfür sind die folgenden Dimensionen:

- Intern bis Extern beschreiben den *Perceived Locus of Causality*. Intern attribuiert das Selbst' und extern das 'Andere' ('Ich bin Schuld' gegen 'Andere sind Schuld') [37].
- Stabil bis Variabel beschreibt die Überzeugtheit des Individuums in Bezug auf das betroffene Problem. Stabilität besagt in diesem Zusammenhang eine empfundene Unveränderlichkeit, während Variabel einen änderbaren Zustand impliziert ('Ich bin nicht gut in Videospielen' gegen 'Ich habe diese Runde nicht gut gespielt') [37].

Das Attributieren selbst ist jedoch nicht unbedingt eine akkurate Darstellung der Zustände. Menschen, welche eher zu Attributierungsfehlern neigen, haben oft einen voreingenommenen *Attributierungsstil*:

Passiert zum Beispiel ein Unfall, bei dem der Hergang offensichtlich ist, so wird dies auch entsprechend zugeschrieben. Wenn ebenjener Hergang nun mehrdeutig ist, so würde ein intern attributierender Mensch sich selbst die Schuld zuweisen und ein extern Attributierender dem Anderen.

Das individuelle Muster eines jeweiligen Menschen hat hierbei Einfluss auf die Motivation: Ein Extern, Spezifisch und Stabil attributierender Mensch würde Motivation verlieren, da der PLOC extern liegt und als unumstößlich empfunden wird.

Hierzu definieren Harvey und Martinko [37] spezifische Attributierungsstile:

- Optimistisch: Negative Situationen werden als extern und Positive als intern attribuiert [37].
- Pessimistisch: Negative Situationen werden als intern und Positive als extern attribuiert [37].

- Feindselig: Negative Situationen werden als extern und stabil attribuiert [37].

Weiterhin werden bestimmte Attributierungsstile mit bestimmten Motivationszuständen assoziiert:

- 'Erlernte Hilfslosigkeit' oder der Glaube, dass Versagen unvermeidbar ist, wird mit dem pessimistischen Stil assoziiert [37].
- 'Aggression', entweder instrumentell oder feindselig, ist sowohl mit dem pessimistischen und dem feindseligen Stil assoziiert. Allerdings wird feindselige, also impulsivere, Aggression eher mit dem feindseligen Attributierungsstil assoziiert [37].
- 'Ermächtigung' ist mit dem optimistischen Attributierungsstil assoziiert [37].

2.2 Engagement, Vertiefung und Immersion in Videospielen

Sobald Motivation vorhanden ist, wird mit einem Medium interagiert. Übergreifend sind nach Brown et al. [21] drei Interaktionsstadien zu definieren. Diese sind in entsprechender Reihenfolge *Engagement*, *Engrossment* und *Immersion*, was als Engagement, Vertiefung und Eintauchen beziehungsweise Immersion übersetzt werden kann.

Engagement ist die geringste Form von Interaktion mit einem Spiel. Um diese zu erreichen, müssen Spielende Zeit, Mühe und Aufmerksamkeit investieren. Ebenso spielt der Zugang eine Rolle, von Brown et al. [21] definiert als Zugang zu der Spielsteuerung. Die Bedienung muss dafür zugänglich genug sein, damit die spielende Person diese meistern kann. Am Wichtigsten hierbei sind jedoch die Kernelemente der Spielmechaniken. [21]. Im ersten Moment bezieht sich das Engagement auf die Präferenzen der spielenden Person. Wenn beispielsweise das Genre nicht dem üblichen Geschmack entspricht, dann wird die Person sich nicht freiwillig damit auseinandersetzen. [21]. Weiterhin muss die spielende Person Zeit investieren, um das Spiel zu erlernen, sowie Mühe oder Energie, welche sich ebenfalls auf das Erlernen der Spielmechaniken bezieht [21]. Außerdem steht der Aufwand in Abhängigkeit von dem Spiel und der spielenden Person, hat diese Person bereits Erfahrung mit dieser Art von Spiel, so gibt es weniger Aufwand. Durch die investierten Mühen haben Spielende oft das Gefühl ein bestimmtes Ergebnis oder eine Belohnung zu erwarten [21]. Aufmerksamkeit ist beschrieben als die Bereitschaft sich zu konzentrieren. Während es sich initial zwar um eine Investition von Seiten der spielenden Person handelt, so muss das Spiel auch etwas bieten, das es wert ist, ihm Aufmerksamkeit zu schenken [21].

Vertiefung ist die fortschreitende Verwicklung der spielenden Person und die dazugehörige Barriere wird von Brown et al. als *game construction* bezeichnet [21]. Diese wird als die Kombination einzelner Spielelemente, auf eine Art und Weise, welche direkt die Emotionen der Spielenden ansprechen, bezeichnet. Durch die bereits erfolgten Investitionen sind Spielende hier

emotional an das Spiel gebunden und es kann zu einem Gefühl von emotionaler Ausgelaugtheit [21] kommen, wenn die spielende Person aufhört zu spielen.

Immersion oder *totale Immersion*, wie sie von Brown et al. [21] beschrieben wird, ist *Präsenz*, also das konzentrierte geistige Dabeisein einer Person. Teilnehmende der von Brown et al. [21] durchgeführten Studie haben ein Gefühl der Abgeschnittenheit der Realität bis zu einem Grad beschrieben, ab dem nurnoch das Spiel wichtig war [21]. Zusätzlich definieren Ermi und Mäyrä [31] drei Unterklassen von Immersion, innerhalb des sogenannten SCI-Modell:

- **Sensory** oder sensorische Immersion ist die audiovisuelle Präsentation beziehungsweise Ausführung von Spielen. Auch Menschen, die keine besondere Affinität oder Vorerfahrung mit Videospielen haben, können diese oft erkennen [31].
- **Imaginative-based** oder fantasievolle Immersion ist abhängig von der erzählten Geschichte, den Charakteren und der gebauten Welt. Diese Art von Immersion findet sich außerdem in ihrer 'Reinform' in Büchern wieder, wo die erzählte Geschichte und somit die Fantasie des Konsumenten am meisten angesprochen werden [31].
- **Challenge-based** oder herausforderungsbasierte Immersion ist die, die besonders in Videospielen oft eine tragende Rolle spielt: Jene Art von Immersion, welche zum Tragen kommt, wenn eine Balance zwischen Herausforderung des Spiels und den Fähigkeiten der spielenden Person erreicht wird [31].

Dieses Zusammenspiel von Herausforderungs- und Fähigkeitsgrad der herausforderungsbasierten Immersion ist ein Motiv, welches sich stark im Konzept des Flows, geprägt von Csikszentmihalyi [26] wiederfindet. Im Rahmen dieser Arbeit wird ein besonderer Fokus hierauf gesetzt. Es ist jedoch zu vermerken, dass konkurrierende Ansichten bezüglich der Überschneidung beziehungsweise Äquivalenz zwischen Flowzustand und Immersion existieren, so behaupten Michailidis et al. [45] es würde weitere Forschung benötigt werden um diese wirklich zu differenzieren, während Jennett et al. [42] behaupten, es gäbe eine klare Differenzierung.

2.3 Flow

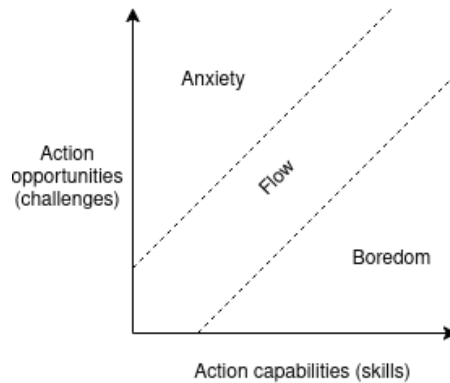


Abbildung 1: Der Flowzustand als Verhältnis von Herausforderungen und Fähigkeiten (eigene Darstellung nach [48])

Der Flowzustand ist ein Zustand der Konzentration des Bewusstseins [27]. Hierbei steht das jeweilige Individuum vor einer zu bewältigenden Aufgabe, welche die vorhandenen Fähigkeiten des Individuums fordert, aber nicht übersteigt. Innerhalb dieses Zustands ist die Person als ganzes, mit Gedanken, Gefühlen und den Sinnen, auf die Aufgabe fokussiert [27]. Insbesondere im Bereich der Videospiele ist dieser oft anzutreffen [47]. Das Erreichen dieses Zustandes steht in Abhängigkeit von der Fähigkeit der Spielenden und der Schwierigkeit des Videospieles.

Im Kontext von Videospiele wird dieser von Csikszentmihalyi beschriebene Zustand also beim Spielen des Spiels erreicht. Erreichen ist in diesem Kontext ein irreführender Begriff, da der Zustand nicht willkürlich zustandegebracht werden kann, sondern an mehrere Faktoren gebunden ist. Die beiden Faktoren, die gegeben sein müssen um den Flowzustand zu erreichen sind folgende [48]:

- Empfundene Herausforderungen, oder Möglichkeiten zur Aktion, die die Fähigkeiten einer Person fordern, aber nicht über- oder unterfordern [48].
- Klare, erreichbare Ziele und sofortiges Feedback über den Fortschritt dieser Aktion [48].

Sind diese Grundlagen gegeben, so ist es möglich, dass das jeweilige Individuum in den Flowzustand übergeht. Diese subjektive Erfahrung wird durch das nahtlose Ineinandergreifen von Momenten, dem Fortschritt und der Aktion beschrieben [48]. Diese subjektive Erfahrung ist der Inhalt des Bewusstseins ('*Consciousness*'). Dieses wiederum ist definiert als das System, welches Informationen verarbeitet und speichert, durch die vorige Selektion durch die Aufmerksamkeit ('*Attention*') [48]. Dadurch betritt die Information die Wahrnehmung ('*Awareness*'), welche als Subsystem des Bewusstseins Prozesse wie 'Denken', 'Wollen' und 'Fühlen' umfasst. Schließlich landen abgespeicherte Informationen in der Erinnerung ('*Memory*') [48]. In diesem Zustand gibt es acht Faktoren, welche oft im Zusammenhang mit dem Flowzustand auftreten. Diese Faktoren sind nicht als Checkliste zu verstehen,

deren Existenz im Ausführen der Tätigkeit gegeben sein muss, sondern als zusammenspielende Elemente, welche zusammen den Zustand definieren, wenngleich nicht jedes Element gegeben sein muss.

Die oben genannten Elemente, welche als Bausteine der *Flow Experience* gelten, sind [24]:

- *There are clear goals every step of the way.*

Es gibt klare Ziele an jeder Etappe des Weges bedeutet, dass die Zielsetzung nicht schwammig und unklar sein sollte. Im Beispielraum der Videospiele wäre dies unter Anderem das Erreichen eines bestimmten Levels [24].

- *There is immediate feedback to one's actions.*

Es gibt sofortige Rückmeldung über die ausgeführte Tätigkeit bedeutet, dass entsprechendes Feedback über die ausgeführte Tätigkeit geliefert wird, wie zum Beispiel eine visuelle Rückmeldung beim Besiegen eines Gegners [24].

- *There is a balance between challenges and skills.*

Es existiert eine Balance zwischen Herausforderung und Fähigkeit bedeutet, dass die Herausforderungen entsprechend dem Fähigkeitslevel der Spielenden angepasst sind. Videospiele implementieren dies oft durch eine Schwierigkeitskurve in Form von Levels der Spielenden gegen die Level der Gegner [24].

- *Action and awareness are merged.*

Aktion und Achtsamkeit sind vereint bedeutet, dass das Bewusstsein auf die aktuell ausgeführte Tätigkeit ausgerichtet ist. Bei Videospielen bedeutet dies oft die Ausrichtung des Blicks auf den Monitor und eventuelle akustische Signale. Bei manchen Konsolen zählt das haptische Feedback über den Controller dazu [24].

- *Distractions are excluded from consciousness.*

Ablenkungen sind vom Bewusstsein ausgeschlossen bedeutet, dass das Bewusstsein voll auf die ausgeführte Tätigkeit gerichtet ist und sich in einer Art Tunnelblick befindet [24].

- *There is no worry of failure.*

Es gibt keine Versagensangst bedeutet, ähnlich wie die beiden Punkte davor, einen vollen Fokus auf die ausgeführte Tätigkeit, ohne Zweifel an den ausgeführten Aktionen, da das Bewusstsein zu sehr mit der Tätigkeit beschäftigt ist, um zweifeln oder hinterfragen zu können [24].

- *Self-consciousness disappears.*

Das Selbstbewusstsein verschwindet bedeutet, dass im Zuge des erreichten Konzentrationszustandes das Selbstbewusstsein zur Seite weicht, da das Individuum zur handelnden Instanz geworden ist und anderweitige 'Ablenkungen', wie Körperbefinden, das emotionale Befinden und etwaige andere Gedanken quasi verschwinden [24].

- *The sense of time becomes distorted.*

Das Zeitgefühl wird verzerrt bedeutet, dass die Zeitwahrnehmung des Individuums sich staucht. Nach dem Verlassen des Flowzustands ist für das Individuum mehr Zeit vergangen, als dieses Individuum vermuten würde [39] [24].

- *The activity becomes autotelic.*

Die Tätigkeit wird autotelisch bedeutet, dass die Tätigkeit selbstbelohnend wird. Es wird kein übergeordnetes Ziel verfolgt und wenn doch, dann weicht dieses zur Seite, da die Aktivität an sich als belohnend wahrgenommen wird. Im Kontext von Videospielen bedeutet dies oft, dass das Besiegen eines besonders schweren Gegners, das Erreichen eines bestimmten Levels oder Erlangen eines bestimmten Gegenstandes an sich belohnend ist, ohne dass extrinsische Faktoren nötig sind [24].

Wie bereits erwähnt, sind diese Faktoren nicht als Checkliste zu verstehen, aber sie bieten ein Referenzframework, anhand dessen versucht werden kann, den Flowzustand zu messen.

Weiterhin beobachtet Csikszentmihalyi folgende Eigenschaften, die oft von Teilnehmenden an Studien bezüglich des Flowzustands beschrieben werden [27]:

- Die Aufgabe beziehungsweise das Ziel ist machbar oder erreichbar [27].
- Die Teilnehmenden sind in der Lage, sich darauf zu konzentrieren, was sie tun [27].
- Die Aufgabe hat klare Ziele [27].
- Die Aufgabe beziehungsweise Tätigkeit liefert unmittelbar Rückmeldung [27].
- Die Teilnehmenden sind in der Tätigkeit involviert und das Alltagsleben rückt in den Hintergrund [27].
- Die Teilnehmenden üben Kontrolle darüber aus, was sie tun [27].
- Die Sorge um das - beziehungsweise die bewusste Beschäftigung mit dem Selbst verschwindet, tritt danach paradoxerweise jedoch stärker hervor [27]. ,
- Die Zeitwahrnehmung ist verzerrt [27].

Diese beobachteten Eigenschaften werden zur Formulierung eines Fragebogens herangezogen, da diese Sammlung von Eigenschaften und Empfindungen für ein Individuum eher bewertbar sind, als eine Abfrage, 'ob das Selbstbewusstsein im Rahmen des Spiels verschwunden ist'. Eine umfangreichere Studie könnte dies wahrscheinlich umsetzen. Aber dadurch, dass die befragten Personen jedoch für sich den Fragebogen ausfüllen, ohne eine Möglichkeit zum Dialog mit einer studententreibenden Person zu haben, könnte dies schnell zu Verwirrung und einer Verfälschung der Ergebnisse führen.

2.4 Spieltypen

In der Definition nach Bartle existieren vier grundlegende Spielertypen [19]. Mit Spieltypen ist in diesem Zusammenhang gemeint, dass die jeweiligen Typen aus einer anderen Primärmotivation heraus (Video-)spiele spielen. Der Grund für das Spielen, und somit die Faktoren an die die jeweilige Befriedigung beziehungsweise der Spaß gebunden sind, ändert sich also. Diese vier Typen sind [19]:

- *Achiever* sind Personen, welchen der Punktestand und das erreichte Level im jeweiligen Spiel am Wichtigsten sind.
- *Explorer* sind Personen, welchen die Erforschung des Spiels, sowohl im Raum der Spielwelt, als auch auf der Metaebene am Wichtigsten sind.
- *Socializer* sind Personen, welchen die sozialen Interaktionen des Spiels am Wichtigsten ist.
- *Killer* sind Personen, welchen das Messen der eigenen Fähigkeiten beziehungsweise Stärke im Spiel am Wichtigsten ist.

Diese vier Spieltypen sind nicht exklusiv zu betrachten, sondern besitzen durchaus Überschneidungen. So bringt Bartle [19] das Beispiel des *Achievers*, welcher in seiner Primärmotivation die Sekundärmotivationen des *socializings* als Austausch über die Möglichkeiten des *achievements* sieht. Das *exploring* dient nur dem Finden neuer Punkt- oder Levelquellen und das *killing* wird als Notwendigkeit gesehen, um an Punkte zu gelangen [19].

2.5 Schwierigkeit in Spielen

Schwierigkeit in Spielen, oder eher die Skalierung dieser, ist ein wichtiges Element im Game Design. Ohne steigende Schwierigkeit verfällt das Interesse der spielenden Personen. Ist die Schwierigkeit zu hoch, dann sind die Spielenden überfordert und verliert sich im oberen linken Quadranten der Flowkurve (Abb. 1). Ist die Schwierigkeit zu niedrig oder leicht, so langweilen sich die spielenden Personen. Um eine sinnvolle Schwierigkeitskurve zu erstellen, kann man sich an den drei Phasen der *Skill Aquisition* orientieren, die Fitts [32] und Anderson [16] für den echtweltlichen Kontext aufgestellt haben, und laut Bostan [20] auch für den Raum der Videospiele anwendbar sind:

Skill Aquisition ist das Erlernen einer Fähigkeit, welche, laut Fitts [32] und Anderson [16], im echtweltlichen Kontext in drei Phasen erfolgt, welche auch auf das Spielen von Videospiele zu übertragen sind [20]:

- Die **kognitive Phase** ist die Phase, in welcher die lernende Person primär durch Anleitung, Ausprobieren oder das Kopieren bekannter Muster lernt [20].
- Die **assoziative Phase** ist die Phase, in welcher die lernende Person sich von langsamen und bewussten Entscheidungen entfernt und die Fähigkeit flüssiger und fehlerfreier ausgeführt wird [20].

- Die **autonome Phase** ist der Punkt an dem die Fähigkeit bei der lernenden Person ins Unterbewusste gesickert ist und so schnell und ohne Zweifel oder Überlegung ausgeführt werden kann [20].

Bostan et al. haben dieses Lernmuster auf das Spielen übertragen und ihnen zufolge sollten diese Lernmuster sich in den Schwierigkeitskurven von Videospielen wiederfinden [20]. Dieser Ansatz wird auch hier verfolgt.

Die Form der Schwierigkeitskurve ist ebenso wichtig wie der Grad der Schwierigkeit nach der die Schwierigkeit im Spielverlauf ansteigt. Steigt die Kurve nur linear, so kann es sich für *Casual* Spielende anfangs frustrierend anfühlen, da die Schwierigkeit für sie zu schnell steigt, während erfahrenere Spielende Frust im späteren Teil des Spiels erfahren, da sie nicht adäquat gefordert werden [44].

Dementsprechend ist die Wahl einer Schwierigkeits*kurve*, anstatt einer Schwierigkeits*geraden* die bessere Wahl. Diese erlaubt erfahrenen Spielenden die ersten Teile des Spiels entsprechend schnell zu durchlaufen, während sie dann in späteren Teilen entsprechend gefordert werden. *Casual* Spielenden hingegen wird die Möglichkeit und Zeit geboten, sich an das Spiel zu gewöhnen, ohne zu schnell abgehängt zu werden.

Die Art der Schwierigkeit zielt im Allgemeinen auf eine *mechanische Schwierigkeit* nach Jagoda [40](siehe unten) ab. Viele Videospiele bieten entweder gar keine Konfigurationsmöglichkeit des Schwierigkeitsgrades, oder eine nach festen Sparten aufgeteilte Konfigurationsmöglichkeit, wie 'Leicht', 'Normal' und 'Schwer' an [40]. Um diese Schwierigkeit zu fabrizieren, können jedoch verschiedenste Methoden verwendet werden, Jagoda [40] definiert hier drei zu differenzierende Schwierigkeitsarten:

- *Mechanical Difficulty* oder *mechanische Schwierigkeit* ist die durch die interaktive und teilnehmende Natur von Videospielen geprägte Schwierigkeit, welche Fähigkeits- und Leistungs basiert ist [40].
- *Interpretive Difficulty* oder *Interpretative Schwierigkeit* ist eine der 'klassischeren' Künsten, wie Dichtkunst und Kunstwerken, nahegelegte Schwierigkeit. Diese Schwierigkeit äußert sich in der Interpretation des Werks: So kann ein Gedicht gelesen und als solches verstanden werden, jedoch fehlt beispielsweise ohne den entsprechenden historischen Kontext etwas, um das Gedicht, beziehungsweise die Aussage dessen, zu begreifen [40]. Erwähnenswert sind hierbei vier Subtypen der interpretativen Schwierigkeit: *Contingent*, *modal*, *tactical* und *ontological difficulties*[40]. Die Unterschiede verlaufen folgendermaßen: Als 'kontingente' Schwierigkeit bezeichnet man die Situation, wenn ein Wort oder Symbol unbekannt ist, als 'modale' Schwierigkeit, wenn ein Text verstanden aber nicht eingeordnet werden kann, sodass beispielsweise ein bestimmter Unterton verloren geht, als 'taktische' Schwierigkeit, wenn absichtlich stilistische Hindernisse eingebaut werden, beispielsweise um aus politischen Gründen zu beschränken, wer einen Text verstehen soll, und als 'ontologische' Schwierigkeit, wenn grundlegende Konzepte, ver- oder entfremdet werden [40].

- *Affective Difficulty* oder *affektive Schwierigkeit* ist ein aus dem emotionalen Raum der Videospiele stammende Schwierigkeit. Sie spannt sich über die Bandbreite des menschlichen emotionalen und reaktionalen Spektrums - von Freude und Neugier bis hin zu Ärger und sogenanntem 'Rage-Quitting' [40].

In dieser Arbeit wird der Fokus auf der mechanischen Schwierigkeit liegen, da diese am leichtesten modulierbar ist und deren Messbarkeit weniger subjektiven Einflüssen als die affektive oder interpretative unterliegt.

3 Das Spiel

Das Spiel, in dessen Rahmen dieses Experiment durchgeführt wird, entspricht einem simplen Top-Down Arena Wave Beat 'Em Up.

Für die Betrachtung des Flowzustands gibt es zwei Versionen: Die Statische und die Adaptive.

In beiden stehen den Spielenden eine Reihe grundlegender und vergleichsweise simpler Fähigkeiten an die Hand und sie müssen sich durch die immer größer werdenden Gegnerwellen kämpfen. In beiden Versionen wächst der Wellenwert exponentiell, die Komposition unterscheidet sich jedoch.

Während in der statischen Version die Gegner in einem 3:2:1 (Simpel, Fernkampf und Schwer) Verhältnis gespawnt werden, besteht die Komposition in der adaptiven Version aus derselben Grundlage, aber wenn die Spielenden Schaden erleiden, werden die im Hintergrund implementierten Gewichte angepasst. Mit zunehmendem Schaden und zunehmender Gegnermenge werden die Wellen also zunehmend mit Gegnern ausgestattet, welche den Spielenden, basierend auf den Informationen der aktuellen Sitzung, die meisten Schwierigkeiten machen.

Das Spiel findet auf einer gleichbleibenden Map statt, mit gleichbleibenden Spawnpunkten für die Gegner in den vier Ecken der Map.

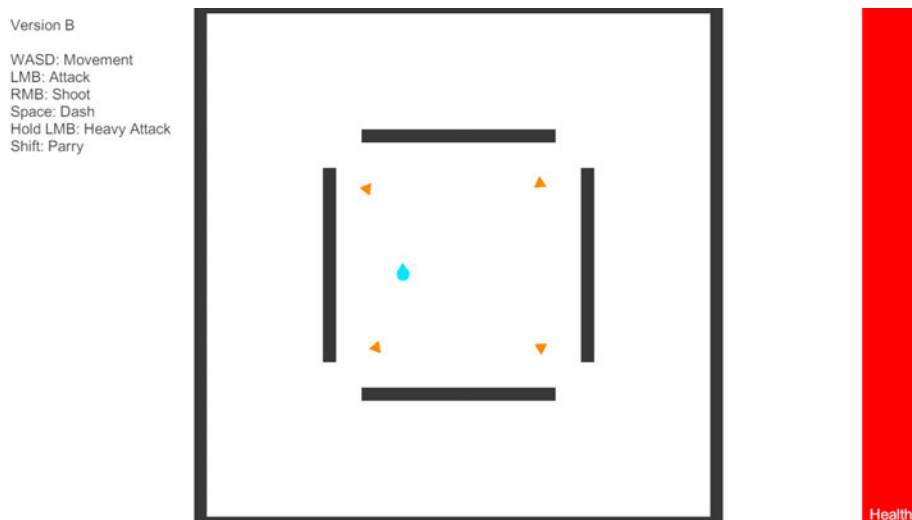


Abbildung 2: Spielansicht (*Quelle: Eigene Darstellung*)

3.1 User Interface

Das Interface des Spiels (Abb. 2) hat im Wesentlichen drei Komponenten:

Die Steuerungsübersicht (Abb. 3) befindet sich im oberen linken Bereich und bietet eine kurze Übersicht über die verfügbaren Fähigkeiten der Spielenden, inklusive der jeweiligen Tasten.

WASD: Movement
LMB: Attack
RMB: Shoot
Space: Dash
Hold LMB: Heavy Attack
Shift: Parry

Abbildung 3: Steuerungsübersicht (*Quelle: Eigene Darstellung*)

Die Healthbar (Abb. 4) der Spielenden befindet sich am rechten Bildschirmrand und zeigt die verbleibenden Lebenspunkte des Spielcharakters auf einem Slider an.



Abbildung 4: Healthbar (*Quelle: Eigene Darstellung*)

Die Game-Over-Ansicht ist ein Overlay, welches nach dem Ableben des Spielcharakters eingeblendet wird. Im mittleren Bereich des Bildschirms befindet sich der Schriftzug 'Game Over', sowie eine Anzeige welche Welle erreicht wurde. Zusätzlich gibt es einen Button mit dem Aufdruck 'Restart', um dem Spieler die Möglichkeit zu geben das Spiel neu zu starten (Abb. 5).



Abbildung 5: Game Over Screen (*Quelle: Eigene Darstellung*)

Außerdem gibt es zu Beginn des Spiels einen Countdown, welcher von fünf herunterzählt, um den Spielenden Orientierungsmöglichkeit in der Spielwelt zu geben. Jedes Weiterschalten des Countdowns ist desweiteren akustisch begleitet [4].

3.2 Spielmechaniken

Orientierung

Abbildung 6: Der
Spielcharakter
(Quelle: Eigene
Darstellung)



Der Spielcharakter (Abb. 6) hat eine Spitze welche die Orientierung innerhalb der Spielwelt indiziert. Diese orientiert sich an der Position des Mauszeigers, ähnlich wie in vielen bekannten Top-Down-Spielen. Der Vorteil gegenüber einer Rotationssteuerung, z.B. über Q und E, ist die schnelle und präzise Umorientierung, welche sich gerade im Zusammenhang mit dem Dash sowie den Angriffsmöglichkeiten für die Kernmechaniken des Spiels besser eignen, welche im Nachfolgenden noch erläutert werden. Spiele in denen eine ähnliche Mechanik vorkommt sind *Enter the Gungeon* [50] und *Hotline Miami* [33].

Bewegung

Die Bewegung wird durch eine WASD-Steuerung realisiert, da diese im Allgemeinen geläufig ist und sich für viele Spielende am natürlichsten anfühlen wird [9]. Die visuelle Repräsentation dieses Inputs äußert sich in der Bewegung der Spielenden. Spiele in denen eine ähnliche Mechanik vorkommt sind *Enter the Gungeon* [50] und *Hotline Miami* [33].

Normaler Angriff

Abbildung 7:
Normaler Angriff
(Quelle: Eigene
Darstellung)



Durch das Betätigen der linken Maustaste wird der sogenannte 'Normale Angriff' ausgeführt, welcher das visuelle Element, zu sehen in (Abb. 7), 90 Grad vor dem Spielcharakter schwingen lässt und mit einem entsprechenden Ton akustisches Feedback liefert [15]. Dieser ist schneller gehalten als die 'normalen Angriffe' der Gegner um den Spielenden einen Vorteil in den wachsenden Gegnermengen zu bieten. Im Hintergrund gibt es eine rechteckige Hitbox, welche einen leicht größeren Bereich abdeckt, als das visuelle, 90 Grad schwingende Element vermuten lässt, um Frustration bei den spielenden Personen vorzubeugen. Dies geschieht im Sinne der sogenannten 'Coyote Time' [7], einem Effekt welcher in Jump 'n' Run Spielen verwendet wird. Hier sind die Spielenden in der Lage zu springen, auch wenn

der Spielcharakter die Plattform bereits verlassen hat. Dies führt zu weniger Frust, da es weniger wahrscheinlich ist, dass die Spielenden den Sprung nicht schaffen, oder in diesem Fall der Angriff nicht verbindet. Die besagte Hitbox überprüft beim Start des Angriffs, um eine möglichst responsive Interaktion zu bieten. Spiele in denen eine ähnliche Mechanik vorkommt sind *Curse of the Dead Gods* [35] und *Hotline Miami* [33].

Fernkampfangriff

Das Betätigen und Gedrückt-halten der rechten Maustaste löst einen Fernkampfangriff (Abb. 8) aus, in welchem das Projektil vor dem Spielcharakter instanziiert wird und ein entsprechender Ton abgespielt wird [3]. Dieser Fernkampfangriff ähnelt dem der Gegner sehr stark, wird durch die Spielenden allerdings zerstört und fügt diesem keinen Schaden zu. Darüber hinaus zerstören sich kollidierende Projektile jeglicher Art und die Fähigkeit kann somit zur Defensive verwendet werden, indem gegnerische Projektile 'abgeschossen' werden. Spiele in denen eine ähnliche Mechanik vorkommt sind *Enter the Gungeon* [50], *Risk of Rain 2* [34] und *Hotline Miami* [33].

Abbildung 8: Spieler mit Projektil
(Quelle: Eigene Darstellung)



Schwerer Angriff

Der 'Schwere Angriff' wird durch das Betätigen und Betätigt-halten der linken Maustaste aufgeladen. Beim Beenden der Betätigung, insofern die nötige, jedoch kurz gehaltene, Aufladezeit erreicht wurde, wird ein stärkerer, jedoch langsamerer Angriff gestartet, welcher sich die akustischen [15] und visuellen (Abb. 7) Elemente mit dem normalen Angriff teilt. Während der Aufladezeit sind die Bewegungsgeschwindigkeit und Dashreichweite halbiert, jedoch bietet der Angriff die Möglichkeit, Gegnergruppen auf sich zukommen zu lassen und sie somit in größerer Menge treffen zu können. Wie der 'normale Angriff' auch, hat die Aktion eine größere Hitbox als visuell zu vermuten. Wie auch beim 'normalen Angriff', wird zu Beginn der Animation auf getroffene Gegner überprüft. Spiele in denen eine ähnliche Mechanik wiederzufinden ist, sind *Risk of Rain 2* [34], *Wizard of Legend* [23], *Curse of the Dead Gods* [35], *Cult of the Lamb* [46] und *Hollow Knight* [22].

Abbildung 9: Parieren
(Quelle: Eigene Darstellung)



Parieren

Durch Betätigen der linken SHIFT-Taste wird das Parieren aktiviert (Abb. 9). Das Parieren wirft während der aktiven Zeit alle Angriffe zurück, sowohl Nah- als auch Fernkampfattacken. Die Nahkampfattacken werden direkt auf die Quelle des Schadens zurückgeworfen, während parierte Projektile vor dem Spielcharakter, wie bei der Fernkampfattacke des Spielcharakters, zurückgeworfen werden, anstatt dass es sich wie zum Beispiel ein Spiegel (mit Eingangswinkel gleich Ausgangswinkel) verhielte. Dies bietet den Spielenden die Möglichkeit, das Parieren als offensives Werkzeug, statt nur als einfaches Blocken zu verwenden. Das visuelle Element, welches den parierenden Zustand indiziert, ist eine sich über die Zeit des Parierens hinweg verkleinernde Raute, damit die

Spielenden abschätzen können, wann dieser Zustand endet. Zusätzlich wird ein Audioclip abgespielt, um den Spielenden Rückmeldung über das Aktivieren zu liefern [14]. Das Parieren, wenngleich das visuelle Element unidirektional ist, ist omnidirektional. Spiele mit vergleichbaren Mechaniken wären *Curse of the Dead Gods* [35], *Risk of Rain 2* [34] und *Hollow Knight* [22].

Abbildung 10: Spielcharakter im Dash (Quelle: Eigene Darstellung)



Dash/Sprint Durch das Betätigen der Leertaste 'dasht' der Spielcharakter (Abb. 10). Hierbei bewegt sich der Spielcharakter in kurzer Zeit über eine beträchtliche Distanz. Dies kann sowohl offensiv, als auch defensiv verwendet werden, um beispielsweise mit einem geladenen 'Schweren Angriff' in eine Gegnergruppe zu sprinten, oder um sich vor einem Schwall von Projektilen zu retten. Beim Dash werden Partikel gespalnt, welche einem, zusätzlich zur tatsächlichen Bewegung, visuelles Feedback über die getätigte Aktion bieten. Außerdem wird ein kurzer Ton abgespielt, um den Spielenden die Verwendung der Fähigkeit zurückzumelden [11]. Spiele die eine vergleichbare

Mechanik enthalten sind *Hollow Knight* [22], *Curse of the Dead Gods* [35], *Enter the Gungeon* [50] und *Wizard of Legend* [23].

Knockback Während das 'Knockback' keine eigentliche Fähigkeit des Spielcharakters ist, im dem Sinne, dass sie durch Tastendruck aktiviert würde, ist sie dennoch ein beträchtlicher Teil der Angriffe. Jeder erlittene Schaden löst einen proportionalen Rückstoß im getroffenen Gegner aus. Weiterhin sind die Gegnertypen mit verschiedenen KnockBackModulatoren ausgestattet, welche dies noch anpassen, in der nächsten Sektion wird dieses genauer erläutert. Spiele die eine vergleichbare Mechanik enthalten sind *Hollow Knight* [22], *Curse of the Dead Gods* [35], *Enter the Gungeon* [50] und *Wizard of Legend* [23].

3.3 Gegnertypen

Simpler Gegner Der 'simple' Gegner (Abb. 11) ist die Grundform der Gegner und orange gekennzeichnet. Dieser Gegner verfügt über einen 'normalen' Angriff, analog zu dem des Spielcharakters, mit ein paar kleinen Anpassungen: Die Hitbox entspricht dem visuellen Indikator und diese überprüft erst getroffene 'Gegner', spricht den Spielcharakter, wenn die Animation zur Hälfte durchgelaufen ist. Dies dient dem Zweck, den Spielenden genügend visuellen Vorlauf zu geben, um das Parieren effektiv einsetzen zu können.

Die Bewegungsgeschwindigkeit ist um 40% höher, als die des Spielcharakters, um ein 'stumpfes' kiten vonseiten der Spielenden zu vermeiden, da dieser in Folge seines Werkzeugsatzes, insbesondere des Dashes, einen mechanischen Vorteil besitzt. Dieser Gegnertyp erleidet 100%, des Rückstoßes beziehungsweise einen unmodulierten Knockback. Alle Gegner teilen sich außerdem das akustische Signal, wenn sie erfolgreich vom Spielcharakter getroffen wurden [2]. Beim 'Besiegen' dieses Gegnertyps werden orangene Partikel instanziiert, um ein Feedback über das Ableben eines solchen Gegnertyps zu geben.

Abbildung 11: Simpler Gegner (Quelle: Eigene Darstellung)



Abbildung 12: Fernkampfgegner (*Quelle: Eigene Darstellung*)



Fernkampfgegner Der 'Fernkampfgegner' (Abb. 12) ist farblich lila codiert und verfügt über einen Fernkampfangriff, jedoch keinen Nahkampfangriff. Die dabei verwendeten Projektile sind leicht kleiner als die der Spielenden und weniger schnell, haben jedoch denselben Cooldown. Im Gegensatz zu dem Projektil der Spielenden fügen sie anderen Gegnern keinen Schaden zu, und verschwinden bei Kontakt mit ebenjenen, dem Spielcharakter jedoch schon. Diese Art von Gegner hat weniger Lebenspunkte und Bewegungspunkte als die 'simplen', da sie einen Vorteil in der Reichweite besitzen. Weil sie durch die Reichweite, und somit höhere Distanz zu den Spielenden, einen Vorteil gegenüber den anderen Gegnertypen haben, erleiden sie 20% mehr Knockback, als der 'simple' Gegner. Genau wie die 'simplen' Gegner hat dieser Gegnertyp auch eine Ablebensanimation, bei der Partikel entsprechend der Farbcodierung 'explodieren'.

Schwerer Gegner Der 'schwere Gegner' (Abb. 13) ist rot codiert und verfügt, ähnlich dem simplen Gegner, über einen Nahkampfangriff. Im Gegensatz zum Simplen ist dieser jedoch langsamer und verursacht mehr Schaden, ähnlich dem 'schweren Angriff' des Spielcharakters. Diese Gegnerart bewegt sich langsamer als der Spielcharakter und hat mehr Leben als die anderen Gegnertypen, um ihm zu ermöglichen, seine Angriffe auszuführen, ohne ihn jedoch zu stark zu machen. Auch in diesem Sinne ist es, dass der 'schwere Gegner' nur 80% des Knockbacks durch Spielcharakterangriffe erleidet. Wie bei den anderen Gegnertypen auch, wird beim Besiegen dieses Gegnertyps eine rote Partikelexplosion instanziiert, um den Spielenden Feedback über die veränderte Gegnermenge zu geben.

Abbildung 13: Schwerer Gegner (*Quelle: Eigene Darstellung*)



3.4 Schwierigkeitskurve/Wavespawner

Zusätzlich zu den oben genannten Spielelementen läuft im Hintergrund der sogenannte Wavespawner. Dieser Wavespawner hat, wegen des Ziels dieser Arbeit, zwei Versionen, auf deren Unterschiede im Weiteren noch eingegangen wird. Im Wesentlichen ist der Wavespawner dafür verantwortlich, die Gegner innerhalb von Wellen in die Map zu spawnen. Die ersten drei Wellen sind in beiden Versionen gleich, es werden erst jeweils vier simple Gegner, dann vier Fernkämpfer und vier schwere Gegner instanziiert. Dies dient dem Heranführen der Spielenden an die verschiedenen Gegnertypen, sowie dazu, die gegebenen Fähigkeiten isoliert erforschen zu können, damit das nötige Verständnis geschaffen wird, die späteren, vielschichtigeren Gruppenkompositionen bestehen zu können. Hierbei schreiben sich die jeweils gespawnten Gegner in eine Liste und löschen sich heraus, sobald sie besiegt wurden. Ist diese Liste leer, sprich, sind alle Gegner besiegt, wird die nächste Welle gestartet. Sobald die Spielenden sich durch diese einfach aufgebauten

Wellen gekämpft hat, greift das Wellenpunktesystem des Wavespawners, entsprechend des Schwierigkeitskurvenansatzes aus Sektion 2.5

$$wavePoints = wavePointModulator^{waveNumber} \quad (1)$$

Dieser Exponentialfunktion folgend werden sogenannte 'Wellenpunkte' berechnet. Diese Wellenpunkte stellen die 'Gesamtpunktzahl' der jeweilig gespawnten Gegnermengen dar. Die simplen Gegner haben eine Punktzahl von eins, Fernkämpfer von zwei und Schwere Gegner von drei. Zuerst werden die Punkte der schweren Gegner berechnet und abgezogen, dann die der Fernkämpfer und schließlich wird mit simplen Gegnern 'aufgefüllt'. Die jeweiligen Punktzahlen werden mit prozentualen Gewichtungen versehen, um ein ungefähres Equilibrium der Punkteverhältnisse zu gewährleisten.

Normal Der 'normale' Wavespawner verhält mit starren Gewichtungen und die Wellen sind vorhersagbar, da die Berechnung der Punktzahlen von keinem Faktor als denen, die bereits zum Start des Spiels festgelegt sind, abhängig ist.

Adaptiv Der 'adaptive' Wavespawner hat, wie der Name schon impliziert, eine adaptive Komponente. Diese äußert sich insofern, als dass bei erlittenem Schaden die Schadensquelle, also ein spezifischer Gegnertyp, zur jeweiligen Gewichtung hinzuzählt. In der Auswertung der Punktzahl und Punkteverteilung, mit den jeweiligen Gewichtungen wird diese Gesamtgewichtspunktzahl ermittelt, mit den jeweiligen Gewichtspunktzahlen zu einem Prozentualwert verrechnet und mit den bereits beschriebenen Gegnertyppunktzahlen verrechnet:

$$gegnerZahl = wavePoints * (gegnerGewicht / gesamtGewicht) \quad (2)$$

Das weitere Spawnen der Wellen erfolgt wie im 'normalen' Wavespawner.

3.5 Technische Implementation

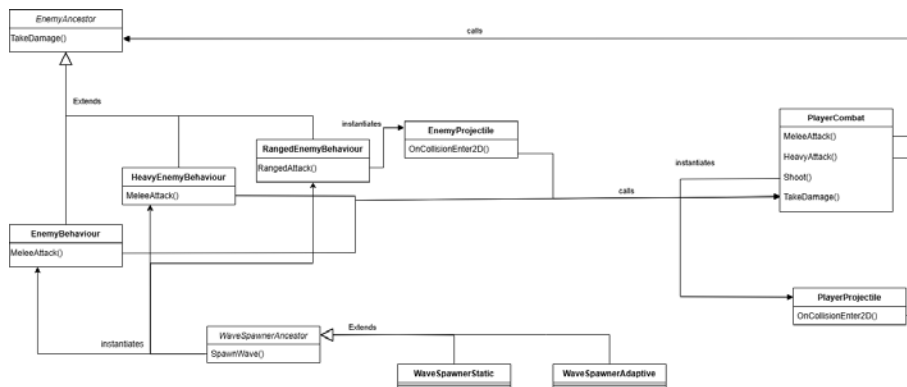


Abbildung 14: Diagramm zur Veranschaulichung des Programms(Quelle: Eigene Darstellung)

Das Spiel ist mit der GameEngine Unity Version 2022.3.9f1 in C# programmiert. Sowohl die Gegner als auch der Spielcharakter sind als einzelne GameObjects modelliert, deren Verhalten mit Skripten gesteuert wird (Abb. 14).

Der Spielcharakter besitzt zwei Skripte: *PlayerBehaviour* für die Bewegung und Orientierung in der Welt und *PlayerCombat* für die Fähigkeiten. Die Unterteilung dient der Übersichtlichkeit.

PlayerBehaviour greift den Unity-Inputmanager im wiederkehrend aufgerufenen Update-Event ab. Hier werden die Horizontal- und Vertikal-Achse des Mauszeigers ausgelesen, sowie die Tasten 'W', 'A', 'S' und 'D', welche jeweils die Vertikal-Positive, Horizontal-Negative, Vertikal-Negative und Horizontal-Positive Bewegungsachse repräsentieren. Weiterhin wird der Tastenanschlag der Leertaste überprüft, welche den Dash triggert. Die tatsächliche Bewegung wird in der, ebenfalls wiederkehrend aufgerufenen, FixedUpdate-Funktion umgesetzt, da diese mit derselben Frequenz wie das Physics-System aufgerufen wird und besagtes Physics-System nach dem FixedUpdate-Aufruf berechnet wird [13]. Hier werden die abgegriffenen Werte in ihren jeweiligen Funktion in das entsprechende Spielverhalten umgewandelt:

- Die Funktion *Movement* wandelt die Bewegungsachsensdaten mithilfe der Movement-Variable, sowie dem Faktor *Time.FixedDeltaTime* in einen Bewegungsvektor um, welcher mit der Integrierten Funktion *MovePosition* den Spielcharakter entsprechend bewegt.
- Die *Orientierung* richtet den Spielcharakter zum Mauszeiger aus. Dies geschieht über die Berechnung:

$$\text{atan2}(\text{lookDirection.y}, \text{lookDirection.x}) * \frac{360}{\pi * 2} - 90 \quad (3)$$

Hierbei werden über *ScreenToWorld* die Mauszeigerkoordinaten des Fensters in Weltkoordinaten umberechnet und durch Subtraktion von der Spielcharakterposition wird der Vektor *lookDirection* berechnet, welcher die gewünschte Blickrichtung repräsentiert. Über *Atan2* [12] wird der Winkel berechnet und mit der integrierten Funktion *setRotation* entsprechend gesetzt. Die Rotation zieht eine veränderte Ausrichtung des sogenannten *AttackPoint* nach sich, welcher als Referenzpunkt für die verschiedenen Fähigkeiten genutzt wird.

- Über den Tastenanschlag der Leertaste, in Unity über die Funktion *Input.GetKeyDown('Space')* abgefangen, wird die Funktion *Dash* aufgerufen. Dieser, ähnlich wie *Movement*, greift die Bewegungsachsen ab und berechnet eine Kraft, die über *MovePosition* angewandt wird. Der Unterschied hierbei ist jedoch, dass der Wert *dashSpeed* verrechnet wird, anstatt des Movement-Wertes, und der Dash nach Abklingen der Zeiteinheit *dashTime* stoppt, und wieder mit *moveSpeed* als Wert gearbeitet wird. Im spielmechanischen Teil wird der Dash so zur Ausnahme, da es die einzige Fähigkeit des Charakters ist, die nicht anhand des Mauszeigers orientiert wird.

PlayerCombat Im Skript **PlayerCombat** werden, wie in **PlayerBehaviour**, über den **InputManager** Tastenanschläge abgegriffen. Jedoch werden die meisten Inputs in die oben beschriebenen Fähigkeiten über das Animationssystem realisiert, anstatt dass Funktionen in **FixedUpdate** oder **Update** aufgerufen werden. Die einzige Ausnahme hierzu bildet das Projektil, welches als sogenanntes **Prefab**, ein vorkonfiguriertes **GameObject**, instanziiert wird.

- Der normale Angriff wird durch den Anschlag der linken Maustaste, abgegriffen durch **Input.GetButtonDown**, ausgelöst und ruft die Funktion **TryMeleeAttack** auf. Diese verhält sich folgendermaßen: Erstens wird der Status der boolean Animationsvariable **attacking** geprüft, welche, wenn sie auf **TRUE** steht, einen aktuell aktiven Angriff indiziert. Ist sie **FALSE**, so wird aktuell kein Angriff ausgeführt und der zweite Teil greift, indem eben diese Variable auf **TRUE** gesetzt wird. Dies löst den Übergang des Animationsautomaten aus und spielt somit die Animation ab. Dort gibt es zwei Eventtrigger: Zuerst, am Anfang der Animation, löst die Funktion **MeleeAttack** aus, welcher die in den Spielmechaniken beschriebene rechteckige Hitbox auslöst. Alle sich in dieser Hitbox befindlichen Objekte, die den Tag 'Enemy' besitzen, werden in ein Array geschrieben, welches in einer for-Schleife durchlaufen wird. Hier wird die **TakeDamage**-Funktion aus dem **EnemyAncestor**-Skript aufgerufen, von welchen alle Gegner erben. Der nächste EventTrigger, am Ende der Animation, ruft die Funktion **EndMeleeAttack** auf, welche die Animationsvariable **attacking** zurück auf **FALSE** setzt.
- Der schwere Angriff wird durch das Gedrückthalten und anschließendes Loslassen der linken Maustasten, mit einem kleinen Mindestzeitraum dazwischen, ausgelöst. Dies wird realisiert durch die Events **Input.GetButton**, welche den Flankenzustand **HIGH** abgreift, und schließlich durch **Input.GetButtonUp**, welche die fallende Signalflanke abgreift. Insofern die **ChargeTime** verstrichen ist, welche durch das aufaddieren von **Time.deltaTime** berechnet wird, wird beim **ButtonUp**-Event die Zeit zurückgesetzt, sowie die Funktion **TryHeavyAttack** aufgerufen. **TryHeavyAttack** verhält sich ähnlich wie **TryMeleeAttack** und überprüft die Animationsvariable **heavyAttack** auf **TRUE** oder **FALSE**. Bei **TRUE** wird eine bereits laufende Attacke erkannt und abgebrochen, bei **FALSE** läuft die Funktion weiter und besagte Variable wird auf **TRUE** gesetzt und löst so die entsprechende Animation aus. In dieser Animation befinden sich, ebenfalls wie beim normalen Angriff, zwei Eventtrigger, je einer am Anfang und einer am Ende. Der Erste löst die Funktion **HeavyAttack** aus und instanziiert eine Hitbox mit denselben Dimensionen wie der normale Angriff, auch hier werden getroffene **GameObject**e mit dem Tag **Enemy** in ein Array geschrieben, welches durchlaufen wird und die **EnemyAncestor**-Funktion **TakeDamage** aufruft, hier jedoch mit dem Integer 20. Der Eventtrigger am Ende der Funktion setzt die Animationsvariable wieder zurück über **EndHeavyAttack**.
- Die Funktionalität des Fernkampfangriffs funktioniert im Gegensatz zum normalern und schweren Angriff nicht über das Animationssystem,

sondern über ein instanziiertes Prefab. Bei Betätigen oder Gedrückthalten der rechten Maustaste wird über `Input.GetButton('Fire2')` die Funktion `Shoot()` aufgerufen, in welcher der `currentRangedCooldown` überprüft wird. Ist dieser kleiner gleich 0, so wird er auf `rangedCooldown` hochgesetzt und dann ein `Projectile-Prefab` im `AttackPoint` instanziiert welches mit der Funktion `AddForce` nach vorne, vom Spielcharakter aus gesehen, beschleunigt wird. Dieses Objekt hat ein eigenes Skript, welches bei Kollision mit einem anderen Objekt auf dessen Tag prüft. Ist der Tag `Enemy`, so wird die `TakeDamage`-Funktion des `EnemyAncestor`-Skripts mit dem Wert 5 aufgerufen, ist der Tag ein anderer, also `'Player'` oder `'Environment'` so zerstört sich das `GameObject` selbst. Der `currentRangedCooldown` wird durch das subtrahieren von `Time.deltaTime` im `Update`-Event heruntergezählt.

- Parieren wird ebenfalls über den `InputManager` abgegriffen. Wenn die `'Shift'`-Taste angeschlagen wird, wird die Funktion `TryParry()` aufgerufen und überprüft die Variable `currentParryCooldown`. Wenn diese kleiner gleich 0 ist und die Animationsvariable `parrying` auf `FALSE`, ist, wird eben diese Animationsvariable auf `TRUE` gesetzt und somit die Parry-Animation gestartet. Diese Animation hat nur einen Eventtrigger, welcher am Ende der Animation liegt und sowohl die Animationsvariable auf `FALSE` zurücksetzt und den `currentParryCooldown` auf `parryCooldown` hochsetzt. `currentParryCooldown` wird in `Update` durch Subtrahieren von `Time.deltaTime` runtergezählt. Wenn die Variable `currentlyParrying` aktiv gesetzt ist, wird im `PlayerCombat` nicht offensichtlich etwas ausgelöst. Sie ist jedoch wichtig für die Funktion `TakeDamage`, da diese auf sie zurückgreift. Dies wird unter `TakeDamage()` weiter erläutert.
- `TakeDamage()` wird durch andere Skripte aufgerufen, sobald eine Kollision beziehungsweise ein Hit erkannt wird. Die jeweiligen Skripte übergeben mit dem Aufruf einen Integer in Höhe des Schadenswertes. Dieser bestimmt, wieviel von `currentHealth` subtrahiert wird. `TakeDamage` gibt in diesem Fall `TRUE` zurück und es wird ein Sound [8] ausgelöst, welcher den Spielenden zusätzliches akustisches Feedback über erlittenen Schaden liefert. Fällt der Wert von `currentHealth` unter null, so wird von `TakeDamage` das `GameOver`-Event getriggert und die Ansicht aus (Abb. 5) wird eingeblendet. Begleitend wird ein Sound abgespielt [1]. Befindet sich der Spielcharakter jedoch im Zustand des Parierens, was durch die Überprüfung `currentlyParrying = TRUE` abgegriffen wird, so gibt die `TakeDamage`-Funktion `FALSE` zurück. Dem aufrufenden Skript wird somit ein parierter Angriff signalisiert und es ruft die jeweils eigene `TakeDamage`-Funktion auf. Gegnerische Projektile bilden hier die Ausnahme, wenn diese sich in diesem Teil der Logik wiederfinden, dann instanziiieren sie eine Kopie am `AttackPoint` des Spielcharakters und zerstören sich daraufhin. Diese Instanziierung erfolgt mit einem kleinen Delay, da den Spielenden die Möglichkeit geboten werden soll, die abgefangen Projektile 'zurückzuschießen', würden diese sofort instanziiert werden, dann würden bei frontalen Treffern die Kollision beider Projektile triggern und sich gegenseitig zerstören.

Die Arena ist einfach aufgebaut: Mit vier äußeren Wänden, die das Ende der navigierbaren Spielwelt bilden, und vier Inneren, welche eine Aufteilung, sowie Deckung für die Spielenden bilden. Dadurch, dass die Spawnpunkte in den Ecken der Arena liegen, bietet diese Aufteilung den Spielenden die nötige Reorientierungszeit, nachdem eine neue Welle gespawnt ist. Ansonsten besteht die Gefahr, dass die spielenden Personen direkt von den Gegnern angesteuert und überhäuft werden. Der Aufbau sollte den Spielenden genügend Möglichkeiten zum Navigieren auf der Karte, sowie dem Ausnavigieren der größer werdenden Gegnermengen geben.

Die Gegner sind in einer Ancestorklasse namens **EnemyAncestor** implementiert, durch die die einzelnen Gegnertypen ihr Verhalten vererbt bekommen. Die Bewegung in der Spielwelt wird mithilfe eines Navmeshs und einer externen Library [5] realisiert. Die Spawnpunkte sind festgesetzt in den vier Ecken der Arena.

Simple Gegner sind die einfachste Form der Gegnertypen und bewegen sich auf den Spielcharakter zu, stoppen, kurz nachdem sie in Angriffsreichweite gekommen sind, und greifen dann an. Der Angriff ist an einen Cooldown gebunden und die Interaktion des Angriffs mit dem Spielcharakter ist, über das Animationssystem und Eventtrigger gelöst. Ist der Spielcharakter in Reichweite, so wird die Funktion **TryMeleeAttack** ausgeführt, in welcher besagter Cooldown abgefragt wird. Ist dieser kleiner gleich 0 so setzt das Skript die Animationsvariable **attacking**, welches wiederum die Angriffsanimation ausführt. Diese ist analog zu dem normalen und schweren Angriffs des Spielcharakters gestaltet, jedoch wird die Hitbox, welcher die Treffer einsammelt erst in der Mitte der Animation, welche auch langsamer ist als die des Spielcharakters, durch den Eventtrigger aktiviert. Auch wie beim **PlayerCombat**-Skript wird die gesammelte Liste, welche stets die Länge 1 haben sollte, da es nur ein Objekt mit dem Tag **Player** geben sollte, durchlaufen und von **PlayerCombat** die **TakeDamage**-Funktion mit dem Integer 10 aufgerufen. Hierbei läuft ein If-Statement auf den Rückgabewert, das bei **FALSE** die eigene **TakeDamage**-Funktion aufruft. Die erste Aktion dieser ist, dass von der variable **currentHealth** der jeweilige Wert abgezogen wird. Falls dieser Wert dann unter 0 fällt, zerstört sich das **GameObject**. Die Zweite ist, insofern das **GameObject** nicht zerstört wurde, das **knockBackForce** und **currentKnockbackTime** aktualisiert werden. Während **currentKnockbackTime** durch **knockBackTime**, einen konstanten Wert, definiert wird, ist **knockBackForce** abhängig vom erlittenen Schaden und vom Wert **knockBackModulator**. diese werden miteinander multipliziert und für die Dauer von **currentKnockbackTime** als Kraft auf den Gegner angewandt. Die Richtung wird vom Bewegungsvektor vorgegeben und ist entgegengesetzt des berechneten Pfades.

Schwere Gegner verhalten sich zum größten Teil ähnlich wie die simplen Gegner. Die Unterschiede liegen primär in der Bewegungsgeschwindigkeit, Lebenspunkten, Angriffsstärke und -tempo, sowie ihrem Knockbackmodulator. Wie der Name schon vermuten lässt, bewegen sie sich langsamer, und greifen auch langsamer an, auch über zwei Eventtrigger in der Animation, wobei diese

prinzipiell gleich, jedoch langgezogener sind. Wenn der schwere Gegner sich in Angriffsreichweite befindet, dann wird die Funktion **TryHeavyAttack** aufgerufen, welche die Animationsvariable **attacking** auf **TRUE** setzt und somit den Übergang der State Machine in die Angriffsanimation triggert. Hier wird im Zenith der Animation die Funktion *HeavyAttack* aufgerufen, welche die Hitbox triggert, dies geschieht mit demselben Ablauf wie beim simplen Gegner. Der von ihnen zugefügte Schaden ist höher, der zweite Eventtrigger, welche die Attacke beendet, setzt die Animationsvariable zurück, sodass der Gegner wieder in der Idle-Animation ist. Der schwere Gegner besitzt die meisten Lebenspunkte unter den Gegnertypen und wenn ihm Schaden zugefügt wird, erleidet er weniger Rückstoß.

Die Fernkampf-Gegner sind die einzige Gegnerart, welche aus der Ferne angreifen können. Sie verfügen über keinen Nahkampfangriff und über keine Animation im eigentlichen Sinne. Ähnlich wie die anderen Gegnertypen jedoch bewegen sie sich auf eine bestimmte Distanz an den Spielcharakter heran und greifen ihn dann an. Dieser Angriff ist ein Projektil-Prefab, welches im **AttackPoint** gespawnt wird, ähnlich wie bei der Projektilattacke des Spielcharakters, und bei der Instanziierung nach vorne weg, vom Gegner aus gesehen, beschleunigt wird. Diese Projektile sind langsamer als beim Spielcharakter, verfügen jedoch über denselben Schadenswert. Auch dieses Projektil ist mit einem Skript ausgestattet, welches dem des Spielcharakters ähnelt: Eine Initialkraft wird angewendet und im Kollisionsevent wird auf den Tag des getroffenen Objektes überprüft. Bei **Enemy** zerstört sich das Objekt selbst, um 'Friendly Fire' zu verhindern, bei **Player** ruft es die **TakeDamage**-Funktion von **PlayerCombat** auf und bei Kollision mit allem weiteren, also anderen Projektilen und Mauern, zerstört es sich schlicht selbst.

Navmesh und Wegfindung Die gegnerische Wegfindung wurde mithilfe des A*-Projekts von Aron Granberg [5] realisiert. Das Framework bietet die Möglichkeit simple, jedoch erweiterbare, Wegfindungsalgorithmen mit einzubinden. Im Rahmen dieses Projektes sind folgende Komponenten relevant:

- Die **Pathfinder**-Komponente wird verwendet um das NavMesh zu erzeugen. Ein Layer welches hinderndes Terrain einschließt, wird übergeben und basierend darauf wird ein Navmesh erzeugt.
- Die **Seeker**-Komponente, welche jedem Gegnertypen angehängt ist, dient, als suchendes Element, welches basierend auf dem gesetzten Ziel des Spielcharakters, einen Weg erzeugt. Die jeweiligen Movement-Funktionen der einzelnen Gegner bewegen diese mittels **MovePosition** in die entsprechende Richtung. Diese Richtung ist durch den jeweils nächsten Wegpunkt angegeben, welche bei Erreichen gelöscht und sich danach in die Richtung des darauffolgenden Wegpunktes begeben wird.

Wavespawner Die **Wavespawner**-Skripte erben von **WavespawnerAncestor**, in welchem die grundlegende Funktionalität des Wellensystems implementiert ist. In diesem werden die drei Gegnertypen referenziert, sowie die Gewichte

und Wellenpunkte gehalten. Das Skript hängt an einem leeren **GameObject** namens **GameManager**, welches vier weitere leere **GameObjects** als Kinder hat, welche als die vier **SpawnPoints** dienen. In besagtem Skript gibt es die Funktionen **SpawnWave** und **GameOver**. Das **GameOver**-Event wird nicht in den Erben angepasst, da es lediglich den **GameOver**-Screen inklusive Restart-Button und erreichter Welle darstellt. Die Funktion **SpawnWave** ist hier mit einem Switch-Statement ausgestattet, welches die ersten drei Wellen erst je nur vier Simple, dann vier Fernkampf und schließlich vier schwere Gegner spawnnt. Darüber hinaus wird die Funktion **SpawnWave** in den Erben angepasst, welches in den nächsten beiden Punkten erläutert wird:

WaveSpawnerStatic ist die Version, deren Verhalten am ehesten aus Videospielen bekannt ist: Mit der Funktion aus Gleichung 2 werden Welle für Welle die Punkte berechnet und diese Punktzahl zusammen mit den Gegnertypgewichten **3 : 2 : 1 Simple:Fernkampf:Schwer** berechnet. Diese werden in eine Gegneranzahl der jeweiligen Gegner umberechnet und schließlich wird eine doppelt verschachtelte Schleife durchlaufen. Die übergreifende Schleife geht durch die Kinder des **GameObjects** durch, welche die einzelnen Spawnpunkte sind und instanziiert die jeweiligen Anzahlen der Gegnertypen in den jeweiligen Spawnpunkten.

WaveSpawnerAdaptive ist strukturell gleich aufgebaut, jedoch unterscheidet sich die Berechnung der Gegnermengen stark. Sie startet mit einer **1 : 1 : 1** Gewichtung, allerdings werden diese Gewichte erhöht, wenn der Spielcharakter Schaden aus einer der jeweiligen Schadensquellen, normaler, schwerer und Fernkampfgegner erhält. Die nächste gespawnte Welle greift somit auf die veränderten Gewichte zu und spawnnt eine Gegnerkomposition basierend auf dem erlittenen Schaden, also mit Gewichtung zu den Gegnertypen, welche dem Spielcharakter am meisten Schaden zugefügt haben. Diese Gewichte werden nicht zurückgesetzt und setzen sich in der jeweiligen Session fort. Der Unterschied in der Initialisierung der Gewichte, verglichen mit der statischen Version, ist dem Trefferweise-Hochzählen geschuldet. Würden die Gewichte äquivalent initialisiert, so würde die adaptive Version erst nach zwei oder mehr Treffern durch schwere Gegner entsprechend mehr schwere Gegner spawnnen, obwohl dieses Verhalten bereits ab einem Treffer gewünscht ist.

Sounds Die Sounds sind universell über die in Unity integrierte Klasse *AudioSource* [6] gesteuert. Die in Sektion 3.2 beschriebenen Geräuschquellen, werden jeweils über den folgenden Funktionsaufruf abgespielt, mit dem jeweils abzuspielenden Sound an Stelle von **audioClip** eingefügt:

```
AudioSource.PlayClipAtPoint(audioClip, transform.position, 1f);
```

4 Ergebnisse der Umfrage

Im Folgenden werden die Umfrage, die Fragen vor ihrem jeweiligen Hintergrund sowie die Ergebnisse betrachtet.

4.1 Ablauf

Die statische und adaptive Version wurden über die Onlineplattform itch.io [10] bereitgestellt, Version A ist die adaptive, Version B die statische. Außerdem wurde dort die Umfrage verlinkt. Die Teilnehmenden wurden gebeten, eine Version zu spielen und danach den Fragebogen auszufüllen. Anschließend sollten sie die andere Version spielen und ebenfalls den Fragebogen ausfüllen. Der Fragebogen wurde anonym ausgefüllt. Der Fragebogen wurde 18 mal anonym ausgefüllt. Die Natur der Umfrage sah zwei ausgefüllte Bögen pro teilnehmender Person vor. Die Ergebnisse im demografischen Bereich lassen jedoch vermuten, dass nicht alle Teilnehmenden beide Bögen ausgefüllt haben. Fast alle Fragen wurden auf einer Skala von eins bis fünf beantwortet, mit den Extremen definiert als 'Trifft nicht zu' bis 'Trifft voll zu' und 'Wenig' bis 'Viel'. Die einzigen Ausnahmen bilden das Freitextfeld bei den Anmerkungen sowie die Alterspektren bei 'Wie alt bist du?', die Geschlechtsauswahl bei 'Wie identifizierst du dich?' und die Frequenz bei 'Wie oft spielst du Videospiele?'.

4.2 Spielbezogene Fragen

In diesem Teil werden fast alle Fragen als Aussage formuliert und mit einer Skala von eins bis fünf, von 'trifft nicht zu' bis 'trifft voll zu', beantwortet. Die einzige Ausnahme bildet die erste Frage, welche sich auf die gespielte Version bezieht. Anhand dieser werden die jeweiligen Ergebnismengen getrennt und dann gegenübergestellt betrachtet.

Die einzelnen Fragen basieren auf den von Mihaly Csikszentmihalyi et al. beobachteten Faktoren in Bezug auf die 'Phänomenologie des Vergnügens' [25]. Die beobachteten Faktoren werden in derselben Arbeit noch abgewandelt und gekürzt auf die intersubjektive Beziehung des Eltern-Kind-Verhältnisses übertragen. Weiterhin wird das Zusammenspiel von Purpose (Zweck), Resolution (Entschluss) und Harmony (Harmonie) betont, die in einem übergreifenden und philosophischen Ansatz als sinngebende Faktoren des Lebens bezeichnet werden [25]. Diese Arbeit bezieht sich jedoch auf die Anwendbarkeit der definierten Faktoren auf das Spielen eines Videospieles.

Hast du Version A oder B gespielt? Wie bereits erwähnt, dient diese Frage lediglich der Differenzierung zwischen den Versionen. 'A' entspricht der adaptiven, während 'B' die statische Version repräsentiert.

Die Verteilung hierbei liegt bei 50% pro Antwortmöglichkeit (Abb. 15), was die vorgesehene Methodik repräsentieren würde. Wie jedoch aus den demografischen Daten ersichtlich sein wird, liegt dies leider nicht vor.

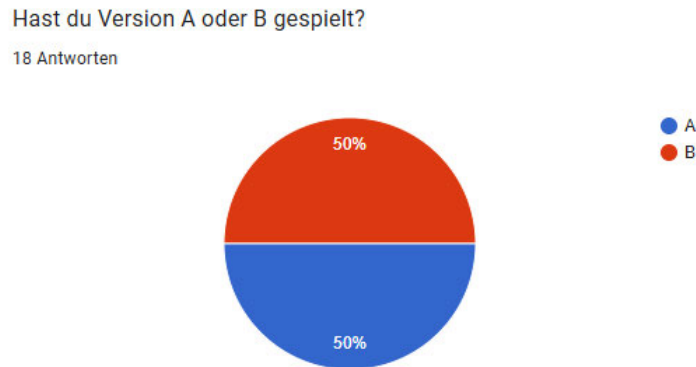


Abbildung 15: Hast du Version A oder B gespielt? (*Quelle: Eigene Darstellung*)

Die Schwierigkeitskurve über die Wellen hinweg war angemessen bezieht sich auf die Balance zwischen Herausforderung und Fähigkeit und fragt die Eigenschaft *Die Aufgabe beziehungsweise das Ziel ist machbar oder erreichbar* [27] aus Sektion 2.3 ab.

Die Durchschnitte für die adaptive Version und die statische Version liegen bei je $4, \bar{2}$ (Abb. 16a) und 3 (Abb. 16b).

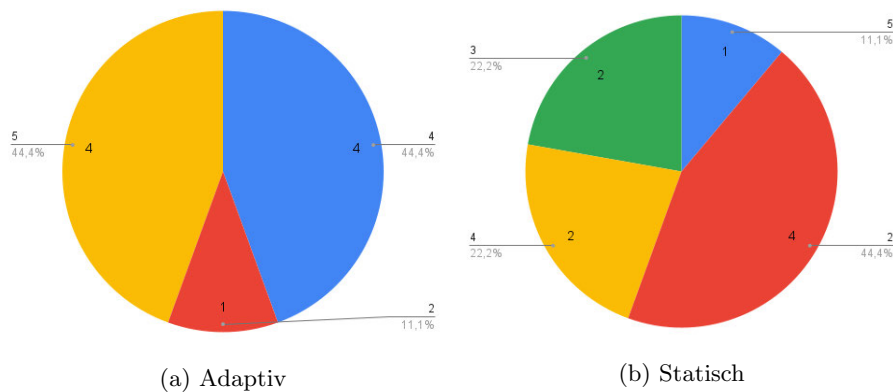


Abbildung 16: Die Schwierigkeitskurve über die Wellen hinweg war angemessen (*Quelle: Eigene Darstellung*)

Du konntest dich auf das Spiel konzentrieren fragt ab, inwieweit etwaige Ablenkungen die spielende Person beeinflusst haben, und entspricht *Die Teilnehmenden sind in der Lage sich darauf zu konzentrieren, was sie tun* [27] aus Sektion 2.3.

Die Durchschnitte für die adaptive Version und die statische Version liegen bei je $4, \bar{7}$ (Abb. 17a) und $3, \bar{7}$ (Abb. 17b).

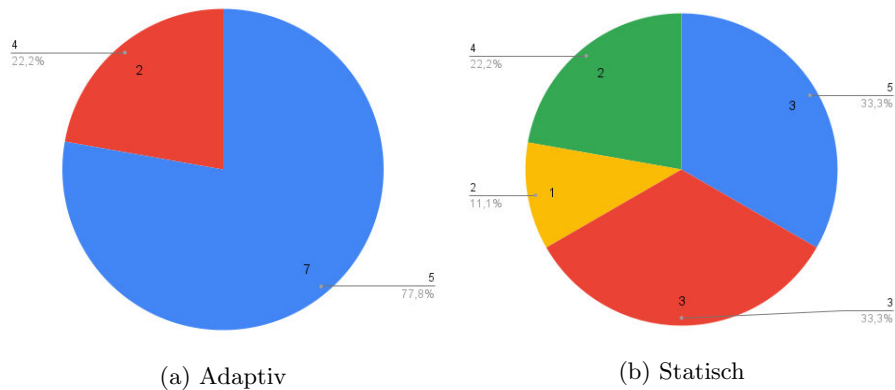


Abbildung 17: Du konntest dich auf das Spiel konzentrieren (*Quelle: Eigene Darstellung*)

Dir war klar, was zu tun ist zielt darauf ab *Die Aufgabe hat klare Ziele* [27] aus Sektion 2.3 zu überprüfen.

Die Durchschnitte für die adaptive Version und die statische Version liegen bei je 5 und $4\frac{1}{3}$ (Abb. 18). Alle neun Antworten zur adaptiven Version wurden mit 5 bewertet. Auf die grafische Darstellung wurde verzichtet.

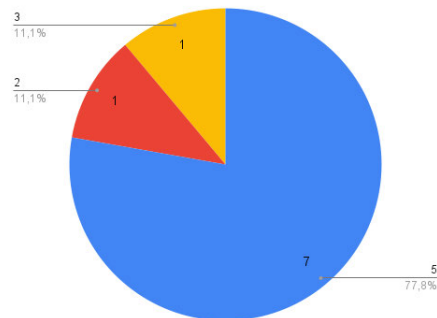


Abbildung 18: Dir war klar, was zu tun ist (*Quelle: Eigene Darstellung*)

Das Spiel hat dir ausreichend Feedback über den Spielverlauf geboten entspricht *Die Aufgabe beziehungsweise Tätigkeit liefert unmittelbare Rückmeldung* [27] aus Sektion 2.3.

Die Durchschnitte für die adaptive Version und die statische Version liegen bei je $4\frac{1}{5}$ (Abb. 19a) und $3\frac{1}{8}$ (Abb. 19b).

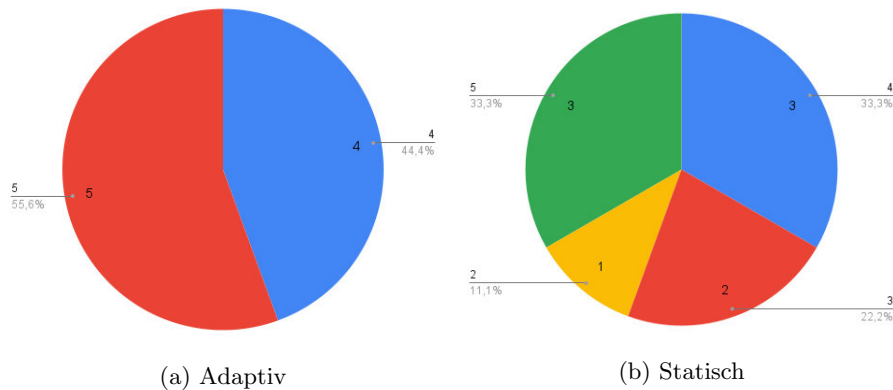


Abbildung 19: Das Spiel hat dir ausreichend Feedback über den Spielverlauf geboten (*Quelle: Eigene Darstellung*)

Du warst in das Spiel vertieft zielt darauf ab, abzufragen, inwieweit die spielende Person in das Spiel versunken war und entspricht *Die Teilnehmenden sind in der Tätigkeit involviert und das Alltagsleben rückt in den Hintergrund* [27] aus Sektion 2.3. Es wurde sich gegen die Referenzierung des Alltagslebens oder das 'Vergessen der Sorgen' in der Umfrage entschieden, da die Frage in so einem vergleichsweise kleinen Rahmen sonst zu geladen gewesen wäre. Die Durchschnitte für die adaptive Version und die statische Version liegen bei je $4, \bar{5}$ (Abb. 20a) und $3, \bar{5}$ (Abb. 20b).

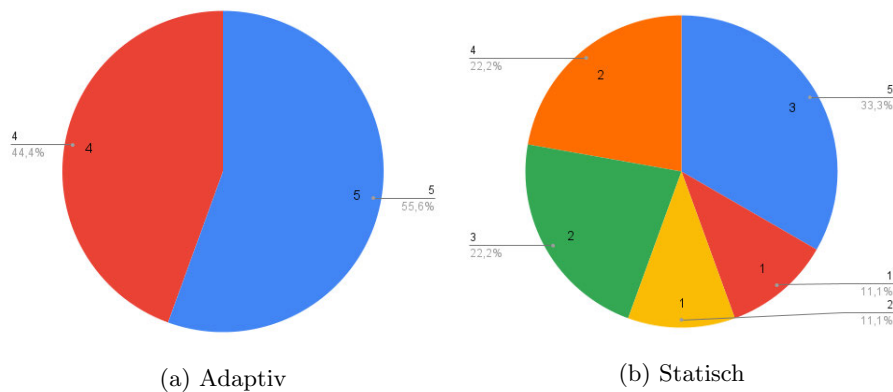


Abbildung 20: Du warst in das Spiel vertieft (*Quelle: Eigene Darstellung*)

Die Reaktion des Spiels auf deinen Input war entsprechend schnell und nach deiner Erwartung fragt ab, ob die sofortige Rückmeldung gegeben war und ein Gefühl von Kontrolle gegeben war. Diese Frage entspricht *Die Teilnehmenden üben Kontrolle darüber aus, was sie tun* [27] aus Sektion 2.3.

Die Durchschnitte für die adaptive Version und die statische Version liegen bei je $4, \bar{6}$ (Abb. 21a) und $4, \bar{2}$ (Abb. 21b).

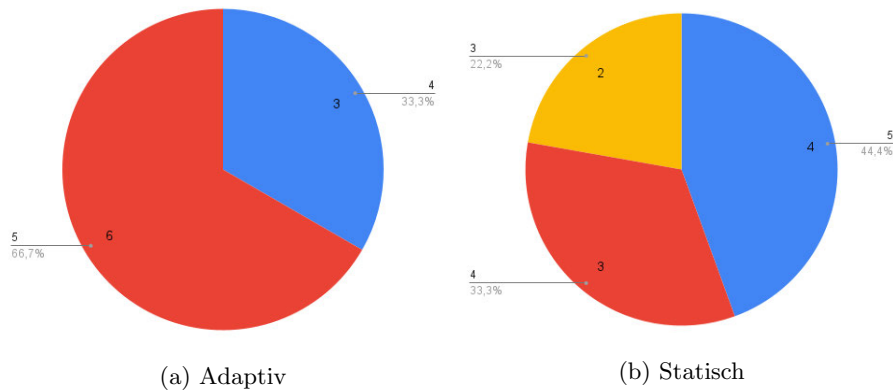


Abbildung 21: Die Reaktion des Spiels auf deinen Input war entsprechend schnell und nach deiner Erwartung (*Quelle: Eigene Darstellung*)

Du konntest in das Spiel eintauchen soll *Die Sorge über das Selbst verschwindet, tritt danach paradoxerweise jedoch stärker hervor* [27] aus Sektion 2.3 abfragen. Auch hier wurde der Wortlaut stark vereinfacht um eine Geladenheit der Frage zu minimieren. Die Durchschnitte für die adaptive Version und die statische Version liegen bei je $4, \bar{4}$ (Abb. 22a) und $3, \bar{6}$ (Abb. 22b).

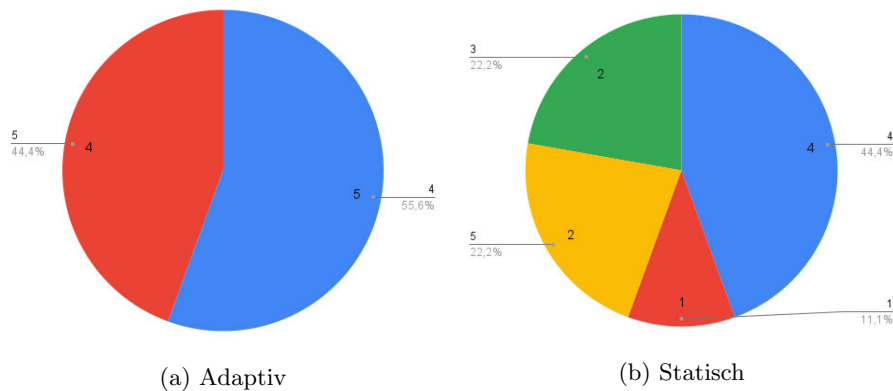


Abbildung 22: Du konntest in das Spiel eintauchen (*Quelle: Eigene Darstellung*)

Du konntest dich im Spiel verlieren entspricht *Die Zeitwahrnehmung ist verzerrt* [27] aus Sektion 2.3 und wurde stark moduliert, da eine Überprüfung der Zeitwahrnehmung der Teilnehmenden durch Aufbau und Durchführung der Umfrage nicht in einem kontrollierten Umfeld gewährleistet werden kann. Die Durchschnitte für die adaptive Version und die statische Version liegen bei je $4, \bar{4}$ (Abb. 23a) und $3, \bar{3}$ (Abb. 23b).

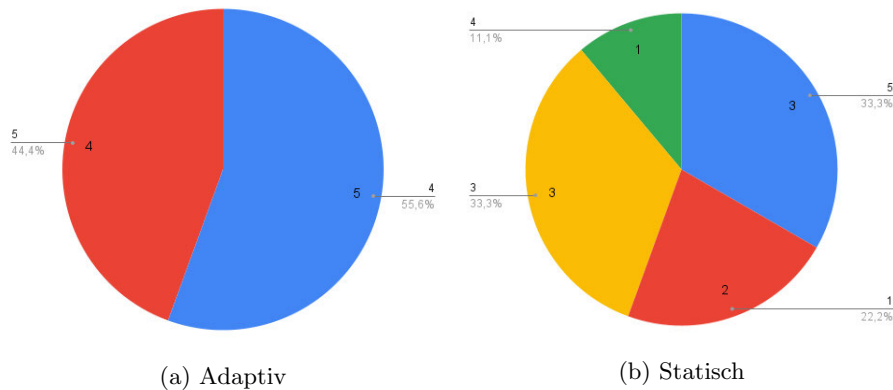


Abbildung 23: Du könntest dich im Spiel verlieren (*Quelle: Eigene Darstellung*)

4.3 Spielmusterbezogene Fragen

Diese Fragen beziehen sich auf die Spielmuster der jeweiligen Person, also ob sie öfter oder seltener spielt, wieviel Vorerfahrung besteht und was ihr in Videospielen wichtig ist.

Wieviel Erfahrung hast du mit Videospielen? ist relevant für den Flowzustand, da eine Person mit mehr Erfahrung entsprechend schneller in den Zustand hineinfindet, da die Engagementhürde [21] zum Erlernen der Mechaniken und zum Zurechtfinden in der Spielwelt geringer ist. Der Durchschnitt liegt bei 4,05 (Abb. 24), mit über 60% der Teilnehmenden, die sich als 'sehr erfahren' einstufen würden.

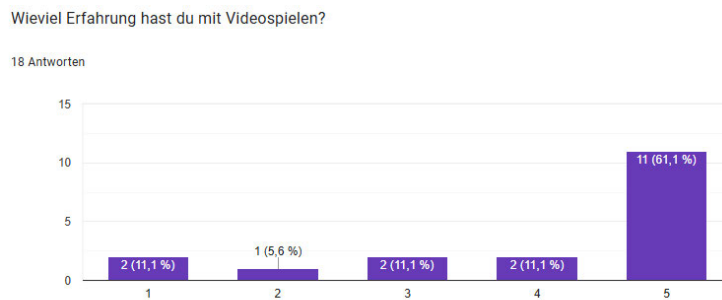


Abbildung 24: Wieviel Erfahrung hast du mit Videospielen? (*Quelle: Eigene Darstellung*)

Achievements und dergleichen sind dir in Videospielen wichtig entspricht einer Einordnung in den Spielertyp *Achiever* nach Bartle [19]. Der Durchschnitt liegt bei 3 (Abb. 25).



Abbildung 25: Achievements und dergleichen sind dir in Videospielen wichtig
(Quelle: Eigene Darstellung)

Die Möglichkeit zu Erforschen ist dir in Videospielen wichtig entspricht einer Einordnung in den Spielertyp *Explorer* nach Bartle [19]. Der Durchschnitt liegt bei 3,5 (Abb. 26).



Abbildung 26: Die Möglichkeit zu Erforschen ist dir in Videospielen wichtig
(Quelle: Eigene Darstellung)

Du bevorzugst Multiplayerspiele gegenüber Singleplayerspielen entspricht einer Einordnung in den Spielertyp *Socializer* nach Bartle [19]. Der Durchschnitt liegt bei 3,16 (Abb. 27).

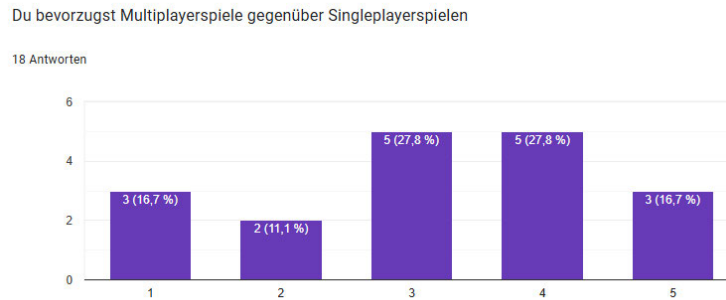


Abbildung 27: Du bevorzugst Multiplayerspiele gegenüber Singleplayerspielen
(Quelle: Eigene Darstellung)

Es ist wichtig für dich in Videospielen möglichst “gut” zu sein entspricht einer Einordnung in den Spielertyp *Killer* nach Bartle [19]. Der Durchschnitt liegt bei $3, \bar{4}$ (Abb. 28).

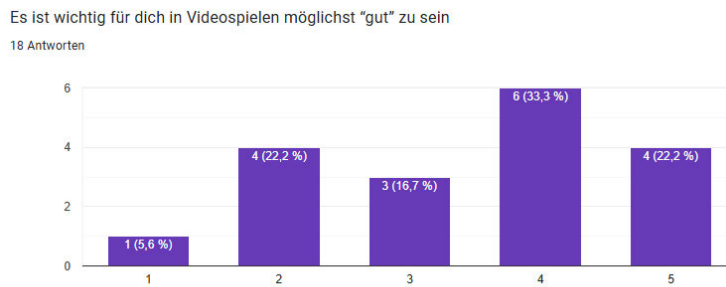


Abbildung 28: Es ist wichtig für dich in Videospielen möglichst “gut” zu sein
(Quelle: Eigene Darstellung)

Wie oft spielst du Videospiele? ist relevant für die Einordnung im Rahmen der Engagementhürden, ähnlich wie *‘Wieviel Erfahrung hast du mit Videospielen?’* und liefert auch Rückschlüsse auf die Art der spielenden Person, also ob es sich um eine periodisch spielende, und demnach eher extrinsisch motivierte, oder eine regulär spielende, also eher intrinsisch motivierte, Person nach Reid [49] handelt. Hierbei geben zwei Drittel der Befragten an, mindestens alle zwei bis drei Tage zu spielen, und wären so den regulär spielenden zuzuordnen (Abb. 29).

Wie oft spielst du Videospiele?

18 Antworten

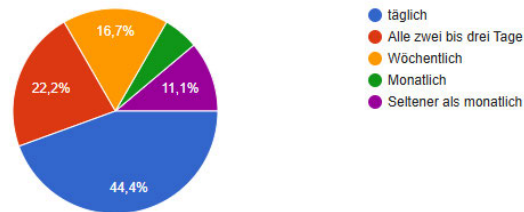


Abbildung 29: Wie oft spielst du Videospiele? (*Quelle: Eigene Darstellung*)

4.4 Demographische Daten

Dieser Teil fragt grundlegende demografische Informationen ab, um die teilnehmende Gruppe besser einordnen zu können.

Wie alt bist du? Die Teilnehmenden befinden sich mit einer starken Mehrheit von 77,8% im Alter von 25-30, mit je 11,1% im Alter von 18-25 und Alter von 30-40 und repräsentieren somit nur eine jüngere Gruppe (Abb. 30).

Wie alt bist du?

18 Antworten

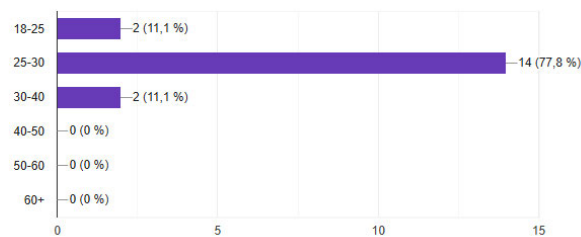


Abbildung 30: Wie alt bist du? (*Quelle: Eigene Darstellung*)

Wie identifizierst du dich? Die teilnehmende Gruppe bestand zu 50% aus männlichen Probanden, mit 38,9% weiblichen Probandinnen und 11,1% Menschen, die sich als divers einordnen (Abb. 31).

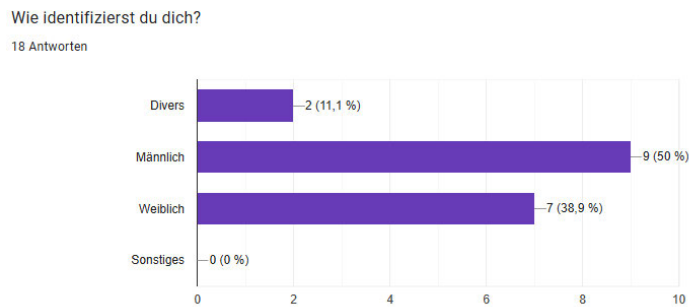


Abbildung 31: Wie identifizierst du dich? (*Quelle: Eigene Darstellung*)

4.5 Freitextfeld

Hier sollte den Befragten die Möglichkeit gegeben werden Anmerkungen, die nicht vom Fragebogen aufgefangen werden, zu hinterlassen.

Gibt es sonst noch Anmerkungen? Insgesamt wurden vier Anmerkungen hinterlassen (Abb. 32).

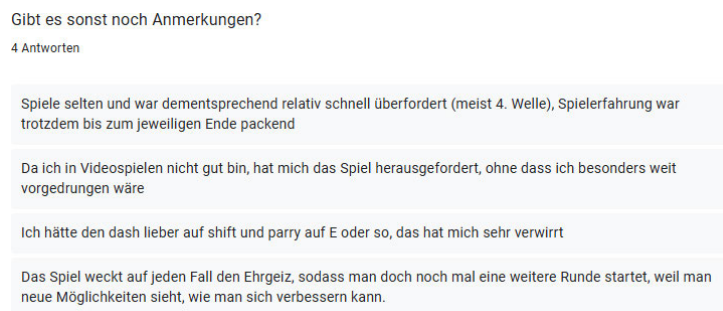


Abbildung 32: Gibt es sonst noch Anmerkungen? (*Quelle: Eigene Darstellung*)

4.6 Auswertung der Ergebnisse

Die acht spielbezogenen Fragen zeigen alle eine mehr oder weniger starke Tendenz, dass die Empfindungen, welche abgefragt wurden, bei der adaptiven Version stärker ausgeprägt sind, und demnach der Flowzustand nach Czikszentmihalyi [27] eher erreicht wurde, als bei der statischen Version. Dies bestätigt die Vermutung, dass ein adaptiver Schwierigkeitsgrad das Erreichen des Flowzustands unterstützt beziehungsweise begünstigt.

Im spielmusterbezogenen Teil zeigt sich, dass der Anteil der *Achievement*- und *Socializing*-orientierten Spielenden leicht geringer ausfällt, als jene, die mehr Wert auf den *Killing*- und *Exploring*-Teil von Spielen legen. Dies begünstigt natürlich das Ziel dieser Arbeit, allerdings ist die Abweichung von maximal 0,5 im Bewertungsschema vergleichsweise gering. Außerdem wurden bereits von Bartle [19] Überschneidungen der Spielertypen beobachtet.

Der hohe Anteil von erfahrenen und frequenten Spielenden ist ein relevanter Teil der Ergebnisse, da, wie bereits durch Brown et al. [21] definiert, das

Engagement eines Spielenden durch bestehende Vorerfahrung vereinfacht wird und somit schneller und leichter ein Flowzustand erreicht werden kann [27]. Im Folgenden werden die Anmerkungen betrachtet:

- 'Spiele selten und war dementsprechend relativ schnell überfordert (meist 4. Welle), Spielerfahrung war trotzdem bis zum jeweiligen Ende packend':
Hier zeigt sich die Relevanz der Abgestimmtheit von Herausforderung und Fähigkeiten. Die Fähigkeiten der teilnehmenden Person werden von der Herausforderung zu schnell überstiegen. Dementsprechend sollte unter Umständen die Schwierigkeitskurve angepasst und abgeflacht werden.
- 'Da ich in Videospielen nicht gut bin, hat mich das Spiel herausgefordert, ohne dass ich besonders weit vorgedrungen wäre':
Hier zeigt sich ein ähnliches Schema wie bei der ersten Anmerkung: Ein Abflachen der Schwierigkeitskurve wäre wohl förderlich.
- 'Ich hätte den dash lieber auf shift und parry auf E oder so, das hat mich sehr verwirrt':
Hier zeigt sich das Problem des nicht konfigurierbaren Steuerschemas. Ein möglicher Ansatz wäre die Konfiguration, da individuelle Präferenzen der Steuerungsbelegung stark durch die Vorerfahrung geprägt werden und starre Vorgaben wahrscheinlich oft verwirren beziehungsweise unintuitiv wirken.
- 'Das Spiel weckt auf jeden Fall den Ehrgeiz, sodass man doch noch mal eine weitere Runde startet, weil man neue Möglichkeiten sieht, wie man sich verbessern kann. ':
Die teilnehmende Person scheint entsprechend gefordert, aber nicht überfordert gewesen zu sein.

5 Ausblick und Fazit

Basierend auf den Ergebnissen im spielbezogenen Umfragenteil scheint der adaptive Schwierigkeitsgrad durchaus einen positiven Einfluss auf das Erreichen des Flowzustands zu haben. Hierbei ist jedoch zu beachten, dass die Umfrage nicht ganz im Sinne des vorgesehenen Ablaufs fortgeschritten ist, wie am Vergleich der demografischen Daten ersichtlich ist, da hier ein Equilibrium zwischen den Versionen herrschen sollte. Dem ist jedoch nicht so. Weiterhin ist das Abfragen der beobachteten Eigenschaften nicht zwangsläufig ein Indikator für das Erreichen des Flowzustands, weshalb sich für weitere Arbeiten in dieser Richtung eine Studie mit anschließender mündlicher Befragung der Teilnehmenden eher eignen würde. Während dieses Experiment versucht hat, nur einen Teil der Spielerfahrung zu modulieren und somit einen Flowzustand zu erzeugen, kann dieses System theoretisch auf weitere spielerische Aspekte erweitert werden. Ein großer Fokus lag auf der mechanischen Immersion sowie auf einer vergleichsweise kurzen Spielerfahrung, welche nach Bartle [19] primär Killer, weniger aber Explorer, Socializer und Achiever ansprechen würde, da viele der Mechaniken im Spiel, als Singleplayerspiel mit vergleichsweise geringem Scope, isoliert sind. Ein möglicher Ansatz, auch andere Spielertypen einzubinden, wäre ein Multiplayerspiel, welches sich lediglich in Unterschieden der verfügbaren Kommunikation (für Socializer), der Map bzw. den spielmechanischen Kombinationsmöglichkeiten (für Explorer) oder dem Sichtbarmachen eines Leaderboards (für Achiever) ändern könnte. Andere Möglichkeiten würden sich in der Art der Immersion nach Ermi und Mäyrä [31] finden. Es könnte ein Spiel mit zwei Versionen, deren Unterschied in der audiovisuellen Gestaltung liegt, entwickelt werden. Eine Version mit Pixelart, Lowpoly oder einer abstrakteren Darstellung könnte mit einer Version mit realistischerer Grafik verglichen werden. Dasselbe könnte für fantasievolle Immersion gestaltet werden: Beispielsweise, indem man dasselbe Spiel einmal mit und einmal ohne untergelegtes Narrativ anbietet. Hierbei müsste jedoch stark auf das Design geachtet werden, um das Ergebnis durch den Faktor des *Environmental Storytelling* [41] nicht zu verfälschen. Außerdem könnte man bei den anderen Schwierigkeitsformen nach Jagoda [40] ansetzen. Diese würden sich jedoch deutlich komplexer gestalten, da sie deutlich enger an das spielende Individuum gekoppelt sind. So ist die interpretative Schwierigkeit von der Vorerfahrung und jeweilig kontextbezogenem Wissen des Individuums abhängig, wenn die Studie beispielsweise auf die kontingente oder modale Schwierigkeit überprüfen würde. Ähnlich steht es mit der taktischen und ontologischen Schwierigkeit, welche zwar wahrscheinlich leichter zu erzeugen wären. Dabei wären aber ihre Aufnahme durch die spielende Person und damit verbunden der Flowzustand subjektiver und schwerer zu überprüfen. Die affektive Schwierigkeit wäre im Vergleich einfacher zu gestalten, indem beispielsweise eine Version 'fair' und die andere 'unfair' gestaltet wird. Dies würde natürlich eher auf das Austesten von Frust und negativen Gefühlen abzielen. Designentscheidungen hierfür wären das Hinzufügen oder Weglassen von vergebenden Spielmechaniken wie der bereits erwähnten *Coyote Time* [7]. Auf andere Emotionen könnte auch geprüft werden, hier würden sich story-basierte Spiele anbieten. Allerdings wären hier feinere und komplexere Narrative und Narrativentscheidungen nötig, um die Ergebnisse entsprechend gegenüberstellen zu können.

Literatur

- [1] 8-bit failure sound by aceofspadesproduc100.
<https://freesound.org/s/333785/>. Accessed: 02.02.2025, License: Attribution 4.0.
- [2] 8-bit hit by thedweebman. <https://freesound.org/s/277213/>.
Accessed: 02.02.2025, License: Attribution 4.0.
- [3] 8-bit 'pew' i by aceofspadesproduc100.
<https://freesound.org/s/333691/>. Accessed: 02.02.2025, License: Attribution 4.0.
- [4] 8-bit start-game by ksaplay. <https://freesound.org/s/758957/>.
Accessed: 02.02.2025, License: Creative Commons 0.
- [5] A*-pathfinding. <https://arongranberg.com/astar/download>.
Accessed: 23.01.2025.
- [6] Audiosource. <https://docs.unity3d.com/6000.0/Documentation/ScriptReference/AudioSource.html>. Accessed: 02.02.2025.
- [7] Coyote time: What games can teach us about forgiveness in learning.
<https://blogs.bsu.edu/teaching-innovation/2023/11/15/coyote-time-games-teach-forgiveness-learning/>. Accessed: 06.02.2025.
- [8] Fire 1 8 bit.wav by mrthenoronha. <https://freesound.org/s/508835/>.
Accessed: 02.02.2025, License: Attribution NonCommercial 4.0.
- [9] How wasd became the standard pc control scheme. <https://www.pcgamer.com/how-wasd-became-the-standard-pc-control-scheme/>.
Accessed: 02.02.2025.
- [10] Itch.io. <https://flow-in-games.itch.io/flow-in-games>. Accessed: 19.02.2025.
- [11] jump.wav by jeckkech. <https://freesound.org/s/391670/>. Accessed: 02.02.2025, License: Creative Commons 0.
- [12] Mathf. <https://docs.unity3d.com/6000.0/Documentation/ScriptReference/Mathf.html>. Accessed: 02.02.2025.
- [13] MonoBehaviour.fixedupdate. <https://docs.unity3d.com/6000.0/Documentation/ScriptReference/MonoBehaviour.FixedUpdate.html>.
Accessed: 23.01.2025.
- [14] Shining 8 bit by mrthenoronha. <https://freesound.org/s/399904/>.
Accessed: 02.02.2025, License: Attribution NonCommercial 4.0.
- [15] Wooden stick swing 1-8 by joao_janz.
<https://freesound.org/s/485249/>. Accessed: 02.02.2025, License: Creative Commons 0.
- [16] John R Anderson. Acquisition of cognitive skill. *Psychological review*, 89(4):369, 1982.

- [17] Dominic Arsenault. Video game genre, evolution and innovation. *Eludamos: Journal for computer game culture*, 3(2):149–176, 2009.
- [18] Albert Bandura et al. Social foundations of thought and action. *Englewood Cliffs, NJ*, 1986(23-28):2, 1986.
- [19] Richard Bartle. Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD research*, 1(1):19, 1996.
- [20] Barbaros Bostan and Sertaç Ögüt. Game challenges and difficulty levels: lessons learned from rpgs. In *International simulation and gaming association conference*, pages 1–11, 2009.
- [21] Emily Brown and Paul Cairns. A grounded investigation of game immersion. In *CHI'04 extended abstracts on Human factors in computing systems*, pages 1297–1300, 2004.
- [22] Team Cherry. Hollow Knight. https://store.steampowered.com/app/367520/Hollow_Knight/, 2017. Accessed: 12.02.2025.
- [23] Contingent99. Wizard of Legend. https://store.steampowered.com/app/445980/Wizard_of_Legend/, 2018. Accessed: 12.02.2025.
- [24] Mihaly Csikszentmihalyi. Flow and the psychology of discovery and invention. *HarperPerennial, New York*, 39:1–16, 1997.
- [25] Mihaly Csikszentmihalyi. *Flow: The psychology of happiness*. Random House, 2013.
- [26] Mihaly Csikszentmihalyi and Judith LeFevre. Optimal experience in work and leisure. *Journal of personality and social psychology*, 56(5):815, 1989.
- [27] Mihaly Csikszentmihalyi. *Flow: The psychology of optimal experience*. New York: Harper & Row, 1990.
- [28] Rowan Daneels, Nicholas D Bowman, Daniel Possler, and Elisa D Mekler. The ‘eudaimonic experience’: A scoping review of the concept in digital games research. *Media and Communication*, 9(2):178–190, 2021.
- [29] R DeCharms. Personal causation. new york: Academicpress, 1968.
- [30] Edward L Deci and Richard M Ryan. *Intrinsic motivation and self-determination in human behavior*. Springer Science & Business Media, 2013.
- [31] Laura Ermi and Frans Mäyrä. Fundamental components of the gameplay experience: Analysing immersion. In *Proceedings of DiGRA 2005 Conference: Changing Views: Worlds in Play*, 2005.
- [32] Paul M Fitts. Perceptual-motor skill learning. In *Categories of human learning*, pages 243–285. Elsevier, 1964.

- [33] Dennaton Games. Hotline Miami. https://store.steampowered.com/app/219150/Hotline_Miami/, 2012. Accessed: 12.02.2025.
- [34] Hopoo Games. Risk of Rain 2. https://store.steampowered.com/app/632360/Risk_of_Rain_2/, 2020. Accessed: 12.02.2025.
- [35] Passtech Games. Curse of the Dead Gods. https://store.steampowered.com/app/1123770/Curse_of_the_Dead_Gods/, 2021. Accessed: 12.02.2025.
- [36] Tilo Hartmann and Christoph Klimmt. Gender and computer games: Exploring females' dislikes. *Journal of computer-mediated communication*, 11(4):910–931, 2006.
- [37] Paul Harvey and Mark J Martinko. Attribution theory and motivation. *Organizational behavior, theory and design in health care*, pages 143–158, 2009.
- [38] Fritz Heider. The psychology of. *Interpersonal Relations*, 1958.
- [39] Soo-hyun Im and Sashank Varma. Distorted time perception during flow as revealed by an attention-demanding cognitive task. *Creativity Research Journal*, 30(3):295–304, 2018.
- [40] Patrick Jagoda. On difficulty in video games: Mechanics, interpretation, affect. *Critical inquiry*, 45(1):199–233, 2018.
- [41] Henry Jenkins. Game design as narrative architecture, 2004.
- [42] Charlene Jennett, Anna L Cox, Paul Cairns, Samira Dhoparee, Andrew Epps, Tim Tijs, and Alison Walton. Measuring and defining the experience of immersion in games. *International journal of human-computer studies*, 66(9):641–661, 2008.
- [43] Raphael Leroy. Immersion, flow and usability in video games. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7, 2021.
- [44] Mike Lopez. Gameplay design fundamentals: Gameplay progression. *Retrieved September*, 1:2007, 2006.
- [45] Lazaros Michailidis, Emili Balaguer-Ballester, and Xun He. Flow and immersion in video games: The aftermath of a conceptual challenge. *Frontiers in psychology*, 9:1682, 2018.
- [46] Massive Monster. Cult of the Lamb. https://store.steampowered.com/app/1313140/Cult_of_the_Lamb/, 2022. Accessed: 12.02.2025.
- [47] Fiona Fui-Hoon Nah, Brenda Eschenbrenner, Qing Zeng, Venkata Rajasekhar Telaprolu, and Sepandar Sepehr. Flow in gaming: literature synthesis and framework development. *International Journal of Information Systems and Management*, 1(1-2):83–124, 2014.

- [48] Jeanne Nakamura, Mihaly Csikszentmihalyi, et al. The concept of flow. *Handbook of positive psychology*, 89:105, 2002.
- [49] Gavin Reid. Motivation in video games: a literature review. *The computer games journal*, 1:70–81, 2012.
- [50] Dodge Roll. Enter the Gungeon.
https://store.steampowered.com/app/311690/Enter_the_Gungeon/, 2016. Accessed: 12.02.2025.
- [51] Richard M Ryan. Control and information in the intrapersonal sphere: An extension of cognitive evaluation theory. *Journal of personality and social psychology*, 43(3):450, 1982.
- [52] Richard M Ryan, James P Connell, and Edward L Deci. A motivational analysis of self-determination and self-regulation in education. *Research on motivation in education: The classroom milieu*, 2:13–51, 1985.
- [53] Richard M Ryan and Edward L Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, 55(1):68, 2000.
- [54] Richard M Ryan, Julius Kuhl, and Edward L Deci. Nature and autonomy: An organizational view of social and neurobiological aspects of self-regulation in behavior and development. *Development and psychopathology*, 9(4):701–728, 1997.
- [55] Richard M Ryan, Robert W Plant, and Stephanie O’Malley. Initial motivations for alcohol treatment: Relations with patient characteristics, treatment involvement, and dropout. *Addictive behaviors*, 20(3):279–297, 1995.
- [56] Martin EP Seligman. Helplessness san francisco. *H. Freeman*, 1975.
- [57] John L Sherry. Flow and media enjoyment. *Communication theory*, 14(4):328–347, 2004.
- [58] Bernard Weiner. An attributional theory of achievement motivation and emotion. *Psychological review*, 92(4):548, 1985.
- [59] Bernard Weiner. Intrapersonal and interpersonal theories of motivation from an attributional perspective. *Educational psychology review*, 12:1–14, 2000.

A Anhang - Umfrageergebnisse

Zeitstempel	08.02.2025 14:25:59	10.02.2025 20:19:58	11.02.2025 20:30:14	11.02.2025 20:35:00
Hast du Version A oder B gespielt?	A	A	A	A
Die Schwierigkeitskurve über die Wellen hinweg war angemessen	4	2	4	5
Du konntest dich auf das Spiel konzentrieren	5	5	5	5
Dir war klar, was zu tun ist	5	5	5	5
Das Spiel hat dir ausreichend Feedback über den Spielverlauf geboten	4	5	4	5
Du warst in das Spiel vertieft	5	5	5	4
Die Reaktion des Spiels auf deinen Input war entsprechend schnell und nach deiner Erwartung	4	5	5	5
Du konntest in das Spiel eintauchen	4	5	4	4
Du konntest dich im Spiel verlieren	4	5	4	4
Wieviel Erfahrung hast du mit Videospielen?	5	1	5	4
Achievements und dergleichen sind dir in Videospielen wichtig	2	1	4	5
Die Möglichkeit zu Erforschen ist dir in Videospielen wichtig	3	4	4	4
Du bevorzugst Multiplayerspiele gegenüber Singleplayerspielen	4	1	2	3
Es ist wichtig für dich in Videospielen möglichst "gut" zu sein	4	2	3	5
Wie oft spielst du Videospiele?	täglich	Seltener als monatlich	täglich	Alle zwei bis drei Tage
Wie alt bist du?	25-30	25-30	25-30	18-25
Wie identifizierst du dich?	Weiblich	Männlich	Weiblich	Divers
Gibt es sonst noch Anmerkungen?		Da ich in Videospielen nicht gut bin, hat mich das Spiel herausgefordert, ohne dass ich besonders weit vorgedrungen wäre	Das Spiel weckt auf jeden Fall den Ehrgeiz, sodass man doch noch mal eine weitere Runde startet, weil man neue Möglichkeiten sieht, wie man sich verbessern kann.	

11.02.2025 20:58:05	16.02.2025 13:38:37	16.02.2025 13:39:42	16.02.2025 13:52:58	16.02.2025 13:53:54	08.02.2025 14:32:44	08.02.2025 15:24:00
A	A	A	A	A	B	B
4	5	4	5	5	5	2
5	4	5	5	4	5	3
5	5	5	5	5	5	5
4	5	4	5	5	4	3
4	5	4	5	4	5	1
5	4	4	5	5	5	4
4	5	5	5	4	4	1
5	5	4	5	4	5	1
5	5	4	5	5	5	5
3	3	5	4	4	2	4
5	4	3	3	2	4	4
4	5	3	5	4	3	4
2	5	3	5	5	4	4
Alle zwei bis drei Tage	täglich	Wöchentlich	täglich	Alle zwei bis drei Tage	täglich	täglich
25-30	30-40	25-30	18-25	25-30	25-30	25-30
Männlich	Weiblich	Männlich	Männlich	Männlich	Weiblich	Weiblich
						Ich hatte den dash lieber auf shift und party auf E oder so, das hat mich sehr verwirrt

08.02.2025 16:22:43	10.02.2025 20:30:10	11.02.2025 20:32:45	11.02.2025 20:36:22	11.02.2025 20:59:35	16.02.2025 13:54:51	16.02.2025 13:55:38
B	B	B	B	B	B	B
4	2	2	2	4	3	3
2	5	3	3	5	4	4
2	5	5	3	5	5	5
2	5	4	3	4	5	5
2	5	3	3	5	4	4
4	5	4	5	5	3	3
4	5	4	3	4	5	3
1	5	3	3	5	4	3
5	1	5	3	5	3	2
4	1	4	4	2	1	1
4	4	3	4	4	3	1
1	1	2	4	3	5	3
1	2	3	4	2	4	4
täglich	Seltener als monatlich	täglich	Wochenentlich	Alle zwei bis drei Tage	Wochenentlich	Monatlich
25-30	25-30	25-30	25-30	25-30	25-30	30-40
Männlich	Männlich	Weiblich	Divers	Männlich	Männlich	Weiblich
Spiele selten und war dementsprechend relativ schnell überfordert (meist 4. Welle), Spielerfahrung war trotzdem bis zum jeweiligen Ende packend						

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit mit dem Titel:

Effekte eines adaptiven Schwierigkeitsgrades auf Flow in Videospielen

selbständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich wörtlich aus der Literatur oder aus anderen Quellen wie z. B. Internetseiten übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

22.02.2025

Datum

Unterschrift