



**HOCHSCHULE
MITTWEIDA**

University of Applied Sciences

Fakultät Angewandte Computer- und Biowissenschaften

Professur Medieninformatik

Bachelorarbeit

Untersuchung der Auswirkungen verschiedener Varianten der
Implementation lokaler Mehrspielermechaniken in einem
zweidimensionalen Plattformerspiel auf ausgewählte Aspekte
des Spielerlebnisses.

Carl Hunger

Mittweida, den 18. Dezember 2023

Erstprüfer: Prof. Dr.-Ing. Christian Roschke

Zweitprüfer: Manuel Heintzig M. Sc.

Hunger, Carl

Untersuchung der Auswirkungen verschiedener Varianten der Implementation lokaler Mehrspielermechaniken in einem zweidimensionalen Plattformerspiel auf ausgewählte Aspekte des Spielerlebnisses.

Bachelorarbeit, Fakultät Angewandte Computer- und Biowissenschaften
Hochschule Mittweida — University of Applied Sciences, Dezember 2023

Referat

In dieser Arbeit geht es um die Auswirkung welche unterschiedliche Mehrspielermechaniken auf das Spielerlebnis, die Spieldauer, die Schwierigkeit des Spiels und die Nutzerfreundlichkeit haben. Dabei wurden die einzelnen Mechaniken mit Probanden getestet und die Ergebnisse der Tests dann evaluiert.

Name: Hunger, Carl

Studiengang: Medieninformatik und interaktives Entertainment

Seminargruppe: MI19w1-B

English Title: Investigating the effects that different implementations of local multiplayer mechanics in a two-dimensional platformer game have on selected aspects of the game experience.

Inhaltsverzeichnis

1	Einführung und Motivation	1
1.1	Aufgabenbeschreibung	1
1.2	Zielstellung und Aufbau der Arbeit	2
2	Grundlagen	3
2.1	Plattformerspiel	3
2.2	Mehrspielermechaniken	5
2.3	Engine	7
2.4	Soziale Komponenten von Mehrspieler Spielen	10
3	Anforderungen und Spezifikation oder Konzeption	15
3.1	Spielkonzept	15
3.2	Mehrspielerkonzepte	17
3.3	Datenerhebung	23
4	Implementierung	25
4.1	Implementierung des Grundspiel	25
4.2	Mehrspielerimplementierung	35
5	Evaluation	41
5.1	Ziele der Evaluation	41
5.2	Durchführung der Evaluation	42
5.3	Auswertung der Evaluation	47
6	Zusammenfassung und Ausblick	55
6.1	Verbesserungsmöglichkeiten	55
6.2	Fazit	56

Literaturverzeichnis

I

1 Einführung und Motivation

Im Bereich der zweidimensionalen Plattformerspiele existieren bereits sehr viele Spiele. Alleine auf der digitalen Vertriebsplattform 'Steam' können in der Kategorie '2D-Plattformer' 6919 Spiele gefunden werden. Darunter befinden sich 370 Spiele in der Kategorie 'Mehrspieler' und 257 davon in der Kategorie 'Lokaler Mehrspieler'. [Sto23] Darunter sind viele verschiedene Herangehensweise an das Thema lokaler Mehrspieler zu finden. Bevor es im Entstehungsprozess eines Spiels in die Implementierungsphase geht muss festgelegt werden, wie genau der lokale Mehrspieler realisiert werden soll. Die Entscheidung für eine spezifische Variante wird dabei von verschiedenen Faktoren beeinflusst und nimmt viel Zeit in Anspruch. Die Ergebnisse dieser Arbeit sollen dazu dienen diesen Prozess zu beschleunigen und Entwicklern helfen die beste Mehrspielervariante für ihr Spiel zu finden.

1.1 Aufgabenbeschreibung

Diese Arbeit beschäftigt sich mit verschiedenen Mehrspielermechaniken und deren Auswirkung auf bestimmte Aspekte des Spielerlebnis. Dafür soll ein zweidimensionales Plattformerspiel erstellt werden. Hierfür erfolgt zunächst die Definition der zu implementierenden Mehrspielermechaniken. Anschließend wird ein Konzept für die einzelnen Mechaniken erstellt wie sie konkret im Spiel eingebaut werden. Danach soll die Umsetzung der erstellten Konzepte erfolgen. Dieses Spiel wird dann mit der Hilfe von Probanden evaluiert.

Die Ergebnisse der Evaluation werden danach analysiert, um Aussagen über die Veränderung zuvor ausgewählter Aspekte zwischen den Verschiedenen Mechaniken zu treffen.

Umgesetzt wird der praktische Teil der Arbeit in Unity. Es wird in C# programmiert.

1.2 Zielstellung und Aufbau der Arbeit

Das Ziel der Arbeit ist es, eine Empfehlung bezüglich der verschiedenen Mechaniken geben zu können, ob sie das Spiel bereichern oder ob Entwickler sich in diesem Genre eher für andere Varianten entscheiden sollten.

Im ersten Teil dieser Arbeit, in Kapitel 2, soll ein Grundverständnis für Mehrspielermechaniken vermittelt werden. Diese Grundlagen bilden die Basis für eine erfolgreiche Konzeption des Spiels und die Auswahl der zu untersuchenden Mechaniken.

Im Kapitel 3.1 wird das Grundspiel Konzept erläutert, das mit den einzelnen Mehrspielermechaniken leicht erweitert werden kann. Das Konzept der ausgewählten Mehrspielervarianten und wie diese in das Grundspiel eingebaut werden wird in Kapitel 3.2 erläutert. Kapitel 4 beschreibt anschließend die Implementierung der einzelnen Bestandteile und deren Besonderheiten.

Nachfolgend wird in Kapitel 5 über die Evaluation des Spiels und der implementierten Mehrspielervarianten berichtet. Dabei wird auf die Ziele der Evaluation und deren Ergebnisse in Bezug auf die zugrunde liegende Untersuchung eingegangen um in Kapitel 6 abschließend eine Empfehlung für die verschiedenen Mehrspielervarianten auszusprechen.

2 Grundlagen

Nach der im vorherigen Kapitel erläuterten Aufgabenstellung und den Zielen der Arbeit dienen die folgenden Seiten der Erörterung, um was es sich bei den einzelnen Komponenten des Lösungsansatz handelt. Dabei wird darauf eingegangen warum konkret ein zweidimensionales Plattformerspiel gewählt und warum sich für die einzelnen Mehrspielermechaniken entschieden wurde.

2.1 Plattformerspiel

In einem Plattformerspiel ist es die Aufgabe des Spielers ein Level zu durchqueren und das Ziel zu erreichen. Dabei muss im klassischen Sinne von einer Plattform zur nächsten gesprungen werden ohne in den Zwischenräumen zu landen, da dies einen Neustart des Levels hervorruft. Dabei kann der spielbare Charakter mit den unterschiedlichsten Fähigkeiten ausgestattet sein.



Abbildung 2.1: Super Mario Bros U, Zielflagge (rot), Spielercharakter (grün), Plattformen (gelb)

Am Beispiel der Super Mario Spielereihe¹ haben die spielbaren Charaktere alle die Fähigkeit normal zu laufen oder zu sprinten. Sie können durch aufeinander folgende Sprünge einen sogenannten Dreisprung ausführen der ihnen ein wenig zusätzliche Sprunghöhe gewährt. Die Charaktere können sich ducken um ihre Größe zu reduzieren im Austausch für eine verringerte Bewegungsgeschwindigkeit. Zusätzlich können sie langsam an einer Wand herunter gleiten und von dieser nach Belieben auch abspringen, mit einem sogenannten Wandsprung.

Der Weg zum Ziel kann dem Spieler zusätzlich mit anderen Hindernissen erschwert werden. Wieder am Beispiel der Super Mario Spielereihe sind das sehr häufig unterschiedlichste Gegner die den Charakter verletzen, wenn er mit ihnen oder von ihnen ausgestoßenen Projektilen in Berührung kommt. Dabei verliert der Charakter Leben beziehungsweise im speziellen Beispiel sein Power-Up. Ein Power-Up ist ein sammelbarer im Level versteckter Gegenstand der dem Spielercharakter zusätzliche Fähigkeiten und meist auch zusätzliches Leben verleiht. [Med96] Wird der Spieler getroffen, wenn nur noch die niedrigste Stufe des Power-Up zur Verfügung steht führt der erlittene Schaden zu einem Neustart des Level.

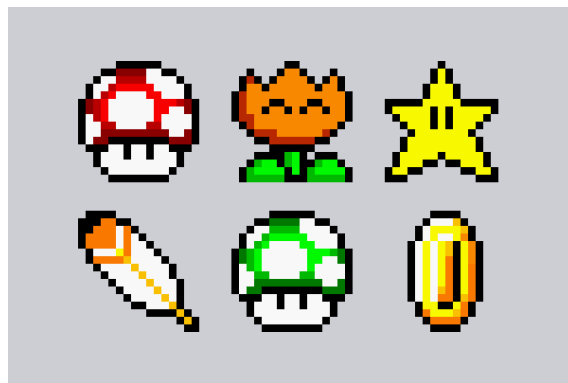


Abbildung 2.2: Super Mario Bros Power-Ups ²

Die Untersuchung beschäftigt sich mit dem Plattformer-Genre, da dieses ein leicht verständliches Spielprinzip besitzt und zudem sehr erfolgreich ist. Dadurch lässt sich eine besonders große Zielgruppe ansprechen und auch für die Tests können viele verschiedene Testgruppen akquiriert werden.

Viele der bereits vorhandenen Plattformerspiele haben trotz der Kategorisierung als Plattformerspiel ein sehr verschiedenes Spieldesign. Durch diese Abweichung in

¹<https://www.nintendo.de/Spiele/Nintendo-DS/New-Super-Mario-Bros-271969.html>

der initialen Spielmechanik ist es nicht sinnvoll für die Untersuchung eine Auswahl verschiedener Plattformerspiele mit unterschiedlichen Mehrspielerimplementationen zu untersuchen. Zudem spezialisiert sich ein Großteil der Spiele auf genau eine Mehrspielerimplementations und es ist somit nicht möglich ein Spiel zu wählen das den Anforderungen der zu testenden Kriterien erfüllt. Deshalb wird ein Spiel von Grund auf erstellt und mit ausgewählten Mehrspielermechaniken und -implementationen erweitert. Damit können die verschiedenen Implementations basierend auf der gleichen Grundmechanik des Spiels in einer homogenen Testumgebung verglichen werden.

2.2 Mehrspielermechaniken

Um über verschiedene Mehrspielermechaniken und deren Besonderheiten Aussagen treffen zu können, muss zunächst eine Grundlage geschaffen werden was der Unterschied zwischen einem Einzel- und einem Mehrspieler ist. Mittlerweile lassen sich Spiele nicht mehr vollständig in diese beiden Kategorien einteilen, da viele Aspekte aus beiden Varianten enthalten.

Als Einzelspieler wird per Definition ein Spiel genau dann bezeichnet, wenn während eines Spieldurchlaufes das Eingabegerät nur von einem Spieler bedient wird. Daher sind diese Spiele für gewöhnlich auch mit dem Hintergedanken konzipiert, dass sie nur ein Spieler zur gleichen Zeit spielt. [OF06] Als Mehrspieler werden im Gegenzug Spiele bezeichnet, bei denen mehrere Spieler die Option haben Eingaben zu tätigen oder aktiv das Spielgeschehen beeinflussen können. Dafür wird meist ein separates Endgerät für jeden Spieler benötigt, welche dann entweder über ein lokales Netzwerk oder über das Internet miteinander verbunden werden. Einige Spiele benötigen nur ein Endgerät mit mehreren Eingabegeräten und werden deswegen als lokale Mehrspieler bezeichnet. Bei Mehrspieler Spielen ist eine weitere Unterscheidung zwischen Spieler gegen Spieler oder Spieler gegen den Computer möglich. Spiele bei denen die Spieler gegeneinander spielen kommt ein weiterer großer und zeitaufwendiger Faktor dazu, die Chancengleichheit für alle Spieler zu gewährleisten. Der Prozess dahin wird als Balancing bezeichnet und kann bei Spielen über mehrere Jahre das Spiel begleiten. Dieser Prozess ist sehr gut am Beispiel des Spieles 'League Of Legends' zu erkennen. Dieses existiert bereits seit 14 Jahren und erhält in Abständen von Zwei

Wochen Anpassung in jeglichen Bestandteilen des Spieles³ [PSEN23]. Ein Mehrspielerspiel bei dem die Spieler gegeneinander antreten kann also sehr zeitaufwendig werden und um für die Evaluierung der Aspekte dieser Untersuchung eine gute Testgrundlage zu liefern wird von dieser Art eines Mehrspielerspiels abgesehen.

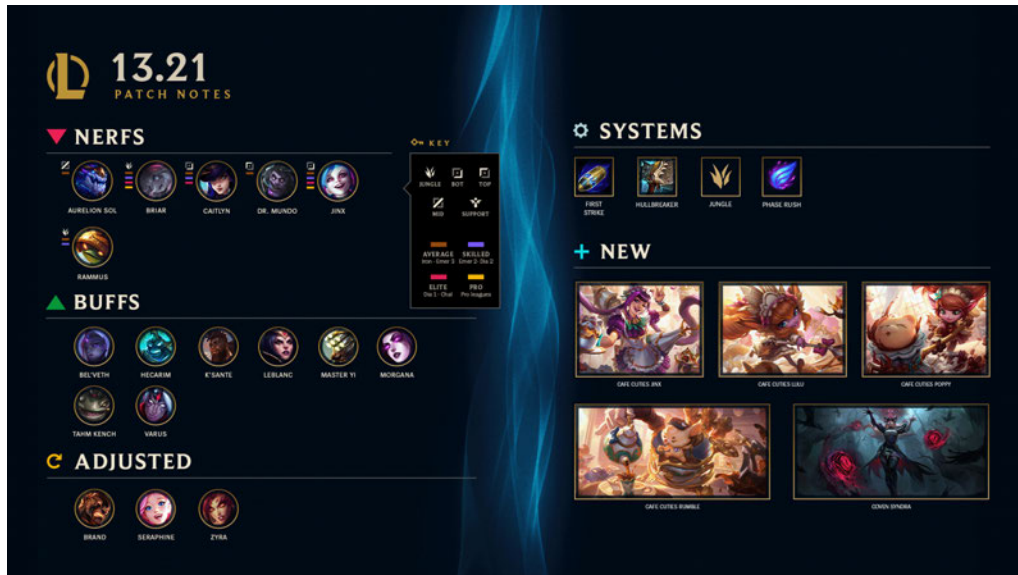


Abbildung 2.3: League of Legends, Grafik zur Darstellung welche Aspekte im Spiel verändert wurden ⁴

Bei Spielen die eine Verbindung über das Internet zwischen den einzelnen Spielern aufbauen ist ein weiterer zu betrachtender Punkt die Latenz die dabei zwischen den Spielern und dem Server und zwischen den Spielern untereinander auftreten kann. Diese ist abhängig von der Internetverbindung der einzelnen Spieler und kann von den Entwicklern des Spiels nur bedingt beeinflusst werden. In einem rundenbasierten Strategiespiel wie 'Civilization VI'⁵ sind Latenzen über 50 Millisekunden nicht besonders signifikant für das Spielerlebnis der Spieler, da die Aktionen der einzelnen Spieler wie der Name schon sagt Runde für Runde abgehandelt werden. Dabei ist es für die Spieler nicht besonders wichtig sehr schnell auf die Aktionen der anderen Spieler zu reagieren und die erhöhte Latenz führt nicht dazu, dass die Spieler ein Gefühl von Ungerechtigkeit verspüren. Im Gegenzug gibt es Kampfspiele in denen die Spieler mit zwei Charakteren gegeneinander kämpfen. Im Spiel 'Guilty Gear

³<https://www.leagueoflegends.com/en-us/news/tags/patch-notes/>

⁵<https://civilization.com/de-DE/>

Strive'⁶ haben die Spieler zum Beispiel Charaktere mit Nah- und Fernkampfangriffen um die Lebenspunkte des Gegners auf null zu reduzieren und somit den Kampf zu gewinnen. In Echtzeit können die Spieler ihre Angriffe ausführen und auf die Angriffe des Gegners mit einem Block oder einem Schritt nach hinten oder vorne reagieren. Dabei sind bereits kleinste Verzögerungen spürbar und können über Sieg oder Niederlage entscheiden.[Bai09]

Die grundlegende Art der betrachteten Mehrspielerimplementationen bildet ein lokaler Mehrspieler. Dabei sind alle teilnehmenden Parteien an einem Endgerät versammelt und spielen mit verschiedenen Eingabemöglichkeiten das gleiche Spiel zusammen oder gegeneinander. Bereits sehr früh in der Geschichte der Videospiele können Vertreter dieses Genre mit Pong aus dem Jahr 1972 gefunden werden. [Ken01] In einigen Spielen dieser Kategorie ist es auch üblich, dass sich die Spieler abwechseln und nur ein Spieler zur gleichen Zeit aktiv ist, so zum Beispiel in den Spielen der 'Dark Pictures Anthology'⁷. Diese bieten neben einem Einzelspieler und einem Online Mehrspieler den 'Movie Night Mode', bei dem sich bis zu vier Spieler einen Charakter im Spiel wählen können den sie für den Verlauf der Geschichte steuern werden. Das Spiel fordert dann zu bestimmten Zeitpunkten in der mit erlebten Geschichte die Spieler auf den Spielecontroller an einen bestimmten anderen Spieler weiterzugeben. Dabei wirken sich die Entscheidungen die ein Spieler mit dem zugewiesenen Charakter ausführt direkt auf die anderen Charaktere aus. So entsteht eine Dynamik zwischen den Spielern welche Entscheidungen sie treffen und obwohl es nur einen aktiven Spieler gibt sind alle dauerhaft involviert. Die Ausweitung der Untersuchung auf netzwerkbasierete Mehrspielervarianten wie sie zum Beispiel im Spiel 'Keep talking and nobody Explodes' verwendet werden würde diese Arbeit in Bezug auf den zeitlichen Rahmen und die Komplexität an zu untersuchenden Aspekten übersteigen. Deshalb beschränkt sie sich nur auf den lokalen Mehrspieler.

2.3 Engine

Für die Erstellung des Spiels wird eine Spieleengine verwendet. Bei einer Engine handelt es sich um eine Entwicklungsumgebung die das Entwickeln von Programmen vereinfachen soll. Für diese Arbeit wird die Unity Engine verwendet. Die Engine

⁶<https://de.bandainamcoent.eu/guilty-gear/guilty-gear-strive>

⁷<https://www.thedarkpictures.com>

wurde am sechsten Juni 2005 veröffentlicht und wird seitdem stetig erweitert und verbessert. [Haa13] Dadurch das sie bereits 18 Jahre existiert gibt es online sehr viel Material und Hilfestellungen für das Arbeiten mit ihr. Zudem besitzt Unity eine ausführliche Dokumentation und kann unter bestimmten Voraussetzungen kostenlos genutzt werden.

In der Engine wird C# als Programmiersprache verwendet. Dabei handelt es sich um eine typischer objektorientierte Allzweck-Programmiersprache die von Microsoft entwickelt wird. Diese besitzt eine ausführliche frei zugängliche Dokumentation und wurde das erste Mal im Juli 2000 als Teil von Microsofts .NET Framework Initiative veröffentlicht.

2.3.1 Entwicklung mit Unity

Bereits bei der Erstellung eines Projektes bietet Unity verschiedene Vorlagen. Diese konfigurieren die Engine für die Verwendung in einem bestimmten Projekt. Standardmäßig installiert sind die Vorlagen für ein zweidimensionales Projekt, ein dreidimensionales Projekt und ein zweidimensionales Projekt in welchem eine andere Methode für die Nachbearbeitung visueller Elemente verwendet wird. Für dieses Projekt soll die normale zweidimensionale Vorlage verwendet werden, da die zusätzlichen Optionen in der visuellen Darstellung für diese Untersuchung keine Relevanz haben.

In der Unity-Engine selbst wird in Szenen gearbeitet. Eine Szene ist ein virtueller Raum der mit Objekten gefüllt und in dem diese Objekte bewegt werden können. Abhängig von der gewählten Vorlage ist diese Ansicht zwei- oder dreidimensional und die Objekte können auf drei verschiedenen Achsen, der Horizontalen, der Vertikalen und der Tiefe im Raum platziert werden. Diese Achsen werden in der Engine mit den Buchstaben X Y und Z referenziert. Für jede dieser Achsen kann einem Objekt ein konkreter Wert zugeordnet werden und sie stellen die Position des Objektes im Raum dar. Für die Wechselwirkung zwischen den einzelnen Objekten stellt Unity ein eigenes Physiksystem zur Verfügung. Dabei können Objekte mit einem 'Collider' versehen werden. Dieser ermöglicht es die Kollision mit anderen Objekten zu erfassen und abhängig davon Aktionen auszuführen. Das Physiksystem kann die natürliche Wechselwirkung zwischen Objekten die miteinander kollidieren berechnen und wiedergeben oder Entwickler können die Konsequenz einer Kollision manuell in

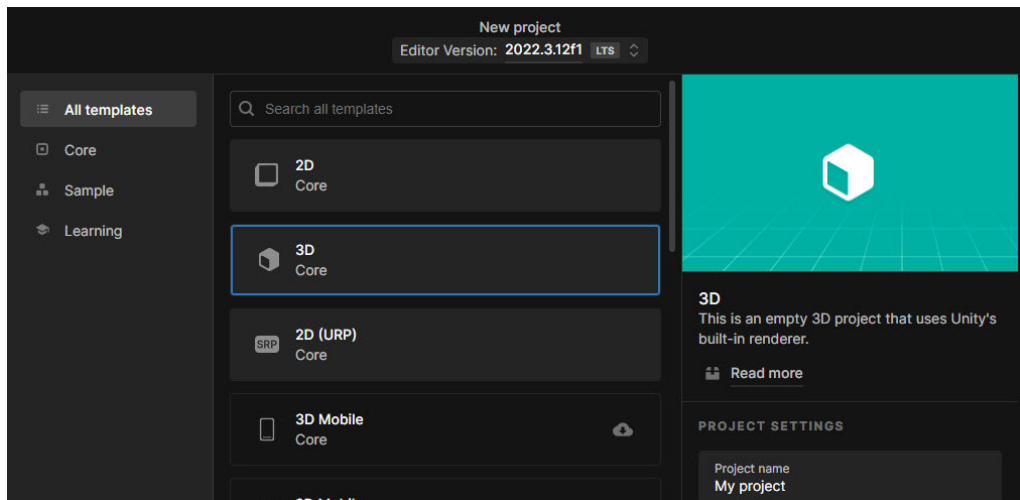


Abbildung 2.4: verschiedene Vorlagen bei der Erstellung eines Projektes in der Unity Engine

einem Skript definieren. Als Skript wird ein von einem Entwickler geschriebener Code bezeichnet. Bei diesem kann es sich zum Beispiel um eine Klasse handeln, die die komplette Logik der Gegner in einem Spiel enthält. Für Klassen bietet Unity eine Klasse von der diese erben können, die 'Monobehaviour' Klasse. Diese bietet bereits implementierte Methoden, die zu fixen Zeitpunkten ausgelöst werden, so zum Beispiel die 'Update' Methode die bei jeder Aktualisierung der Szene aufgerufen wird. Diese ermöglicht es den Entwicklern Skripte zu schreiben, um die Objekte in der Szene abhängig von der vergangenen Zeit zu steuern. [Uni23]

Generell ist es üblich eine Namenskonvention zu führen. Dafür wird zu Beginn des Projektes festgelegt wie einzelne Elemente wie Klassen, Variablen oder Objekte in einer Szene benannt werden um die Übersichtlichkeit und das Arbeiten im Projekt zu erleichtern. Diese Vorschriften für die Benennung können von Integrierten Entwicklungsumgebungen, kurz IDE, wie JetBrains Rider⁸ eingelesen werden. Bei einer IDE handelt es sich um ein Programm um den Entwickler im Entwicklungsprozess zu unterstützen. Dabei erhält der Entwickler abhängig von der verwendeten IDE unterschiedliche Hilfen, wie die automatische Vervollständigung von angefangenen Methodennamen, Finden von Variablen-Referenzen oder eben Hinweise auf inkonsistente Benennungen. Die JetBrains Rider IDE analysiert zudem den geschriebenen Code

⁸<https://www.jetbrains.com/de-de/lp/dotnet-unity/>

und gibt Vorschläge wie der Code performanter oder leichter lesbar umgeschrieben werden kann. [DS23]

2.3.2 Entwicklung eines zweidimensionalen Projekts

Für die Entwicklung eines zweidimensionalen Projekts bietet die Unity Engine die im vorherigen Abschnitt beschriebenen Funktionen speziell für diese Umgebung optimiert. Die einzelnen Elemente sind mit dem Zusatz '2D' gekennzeichnet um sie schneller finden zu können. Diese vorimplementierten Funktionen beschleunigen den Entwicklungsprozess enorm, da sie direkt nutzbar sind und nicht erst neu implementiert werden müssen.

Unity bietet für die Darstellung sehr vorteilhafte Funktionen. Bilder sind in einem Projekt in ihren Eigenschaften manuell an den Bedarf des Entwicklers anpassbar. So können zum Beispiel die Bildpunkte pro Längeneinheit eingestellt werden um einen höheren oder niedrigeren Detailgrad zu erreichen. Eine weitere sehr vorteilhafte Funktion besonders im zweidimensionalen Raum sind 'Tilemap' Elemente. Diese ermöglichen es in unterschiedlichen Rastern Elemente in der Szene zu platzieren. Dafür wird eine 'Tilepalette' erstellt und mit den gewünschten Bildern gefüllt, um sie dann auf der 'Tilemap' zu platzieren. [Hal18]

2.4 Soziale Komponenten von Mehrspieler Spielen

Im Kontext der sozialen Aspekte von Videospiele taucht immer wieder ein Name auf, Richard Bartle. Dieser veröffentlichte bereits 1996 eine Kategorisierung von Spielern in vier verschiedene Gruppen. Diese Studie basiert auf Bartle's Untersuchung von Online Mehrspieler Spielen und die Kategorisierung auf den Zielen die die einzelnen Gruppen beim Spielen verfolgen. Dabei gibt es zwei Achsen anhand die Spieler eingeordnet werden. Die erste beschreibt das Handeln der Spieler, ob sie eher agieren oder interagieren. Auf der zweiten Achse ist dargestellt womit die Spieler [Bar96]

Die umstrittenste der Spielergruppen sind die sogenannten 'Killer'. Dabei handelt es sich um Spieler die die im Spiel gegebenen Mechaniken verwenden um anderen Spielern Schaden zuzufügen oder sie zumindest in ihrer Spielweise zu stören. Wenn

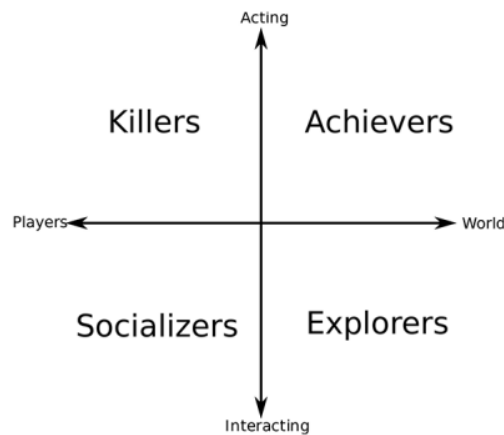


Abbildung 2.5: Bartle's Taxonomie [Ber22, S. 62]

es im Rahmen der Spielmechaniken möglich ist so involviert dieser Prozess meist die Akquise einer Waffe in irgendeiner Form und deren Einsatz um andere Spieler wie der Name schon sagt zu besiegen beziehungsweise im Spielekontext deren Charakter zu töten. In der Geschichte von Videospiele und der sich darum entwickelten Kultur kam immer wieder die sogenannte Killerspieldebatte auf. In diesem Kontext wurden Studien durchgeführt die feststellen sollten, ob Videospiele mit exzessiver Gewalt zu einer Abstumpfung der Emotionen oder verstärkter Aggression führen können.[Sch12] Da sich der Fokus dieser Untersuchung auf die kooperativen Aspekte beschränkt wird nicht weiter auf diese Thematik eingegangen.

Spieler die mit der Welt interagieren und diese bis in die letzte Ecke erkunden sind 'Explorer'. Ihr Ziel ist es alles über die virtuelle Welt zu erfahren. Darunter zählt die Topologie der Karte, die Geschichte der einzelnen Charaktere in der Welt, die Interaktion zwischen diesen Charakteren, die Geschichte der Welt als Ganzes oder aber von den Entwicklern platzierte 'Easter Eggs'⁹, kleinen Geheimnissen die die Entwickler für die Spieler versteckt haben. Des weiteren beschäftigen sich 'Explorer' mit der Physik und der generellen Funktion der einzelnen Spielmechaniken und sämtliche Wechselwirkungen zwischen ihnen, egal ob von den Entwicklern so angedacht oder nicht.

Die 'Achiever' sind Spieler die sich das Ziel gesetzt haben alle Hürden zu meistern und alle Ziele zu erreichen. Diese können sie sich selbst gesteckt haben oder von

⁹<https://gamesandimpact.org/uncategorized/the-cake-is-a-lie-easter-eggs-in-video-games/>

den Entwicklern des Spiels in der Form von Errungenschaften, besonders starken und wertvollen Gegenständen oder zum Beispiel in dem sie einen besonders starken Gegner besiegen.

Ihnen gegenüber stehen in Bartle's Modell die 'Socialiser'. Spieler die in einem Spiel daran Freude und Spielspaß finden mit anderen Spielern zu interagieren und zu kommunizieren und das Spiel wird eher Mittel zum Zweck statt dem Hauptfokus. Der Fortschritt im Spiel dient meist nur noch dazu um einen bestimmten Status im Spiel zu erhalten oder um bestimmte neue Bereiche freizuschalten damit die anderen Spieler nicht außer Reichweite geraten und die Gemeinschaft gewährt bleibt. Das übergreifende Ziel der 'Socialiser' ist es nicht das Ende des Spiels zu erreichen oder der Beste zu sein, sondern Freundschaften zu schließen und zu festigen die über die Grenzen eines einzelnen Spiel hinweg bestehen bleiben.

Mit der Thematik eines lokalen Mehrspieler Spiel stellen in dieser Untersuchung primär 'Socialiser' und sekundär 'Achiever' den Hauptfokus dar. Ausschlaggebend für diese Einschätzung sind der kooperative und kommunikative Anteil in dieser Spielekategorie. Hinzu kommt, dass lokale Mehrspieler Spiele meist direkt auf Freundschaften abzielen, also Personen die gerne etwas zusammen unternehmen.



Abbildung 2.6: Personen spielen ein Spiel mit lokalem Mehrspieler ¹⁰

Für die Spieler kann das Spielen auch abgesehen vom Spaßfaktor positiv Effekte haben. Eine Studie aus dem Jahr 2013 die sich mit der Thematik beschäftigt, ob An- oder Abwesenheit von Gewalt in Videospiele prosoziales oder aggressives Verhalten fördert. Die Ergebnisse dieser Untersuchung stellen zwar dar, dass Gewalt in den Spielen keinen Einfluss auf die untersuchten Parameter hat, jedoch konnte kooperatives Spielen unabhängig der Gewalt mit einer reduzierten Aggressivität der

Probanden in Verbindung gebracht werden. [JF13] Dieser äußerst positive Nebeneffekt kann besonders bei der Konzeption von Spielen die im pädagogischen Bereich Anwendung finden von Vorteil sein.

3 Anforderungen und Spezifikation oder Konzeption

Nachdem die Grundlagen für die Thematik dieser Arbeit im vorgerigen Kapitel definiert wurden, werden in den folgenden Abschnitten die Konzepte für das Grundspiel und die verschiedenen Mehrspielerimplementationen erläutert. Dabei wird sowohl auf das Prinzip als auch den Grund warum das Prinzip genauso angewendet wird eingegangen. Dabei orientieren sich die Konzepte an den im Kapitel Grundlagen analysierten Gegebenheiten und Voraussetzungen für ein zweidimensionales Mehrspieler Plattformerspiel.

3.1 Spielkonzept

Das Grundgerüst für das Spiel bietet ein zweidimensionales Plattformerspiel. In diesem besitzt der Spieler zwei verschiedene Fähigkeiten dauerhaft und kann durch das aufsammeln eines Power-Up eine weitere Fähigkeit erlangen. Die Basisfähigkeiten bestehen aus Laufen und Springen. Das Laufen zeichnet sich dadurch aus, dass die Spieler sich auf der horizontalen Ebene fortbewegen können. Mit dem Springen erhalten sie zusätzlich die Fähigkeit sich eingeschränkt in der vertikalen Ebene zu bewegen. Dabei erhalten sie beim Einsatz der Fähigkeit einen Antrieb nach oben. Dabei wirkt eine simulierte Gravitationskraft den Spielern entgegen und führt dazu, dass die Spieler abhängig vom initialen Antrieb eine maximale Sprunghöhe besitzen. Mit diesen beiden Fähigkeiten in Kombination ist es den Spielern möglich sich innerhalb des Spiels fortzubewegen. Diese Fähigkeiten sind die grundlegenden Fähigkeiten die in jedem Plattformerspiel vorhanden sind. Die Fähigkeit welche die Spieler in den Leveln erlangen können ist das Schießen. Diese erlangen sie durch das aufsammeln einer im Level platzierten Waffe. Hat ein Spieler eine solche Waffe erlangt kann der Spieler ein Projektil schießen und mit diesen Wechselwirkungen im Level hervorrufen. Die Parameter wie genau sich das Schießen gestaltet gehen dabei

von der im Level platzierten Waffe aus. Das Spiel besteht aus mehreren Leveln die es für die Spieler zu durchqueren gilt. Die Level sind dabei mit einer durchschnittlichen Spieldauer von zwei Minuten pro Level angedacht, da der Fokus nicht auf den Leveln, sondern den Mehrspielermechaniken liegen soll. In den Leveln werden verschiedene Typen an Hindernissen verwendet, die es den Spielern erschweren das Ende zu erreichen.

Das erste Hindernis stellt dabei ein simples Jump'n'Run dar. Dabei müssen die Spieler in einem vertikalen Layout des Levels von Plattform zu Plattform springen, um die Spitze und somit das Ziel zu erreichen. Dabei ist die mentale Herausforderung den richtigen Weg zu finden und die motorische Herausforderung die einzelnen Sprünge zu absolvieren.

Im zweiten Level wird das Spiel dann mit einem Rätsel erweitert. Dieses muss gelöst werden um im Level zu können und den Weg zum Ziel zu öffnen. Dadurch werden die Spieler in einem weiteren Aspekt mental gefordert, wenn sie das Level bewältigen wollen. In den Mehrspielerimplementationen dieses Level ist die Kommunikation zwischen beiden Spielern noch eine zusätzliche Schwierigkeit die es zu bewältigen gilt. Zudem bietet ein Rätsel bei dem sich der Spieler auf eine Druckplatte stellen muss um diese zu betätigen und die damit verbundene Tür zu öffnen viele verschiedene Möglichkeiten der Ausarbeitung. Die Druckplatte könnte mit einer Zeitsteuerung versehen sein und die Spieler müssen die geöffnete Tür innerhalb der vorgegebenen Zeit durchqueren. Es könnten im Level Objekte enthalten sein die die Spieler auf bestimmte Art und Weise bewegen können und somit die Druckplatte mit einem der Objekte aktivieren. Dies sind zwei Beispiele die voraussetzen dass die Spieler das Level verstehen und dann koordiniert auf die Lösung hinarbeiten können. Wie sie das bewerkstelligen kann sich dabei je nach Mehrspielermechanik stark unterscheiden und bietet somit einen guten Grundsatz für eine Untersuchung.

Die Spieler werden im dritten Level mit mehreren Gegnern konfrontiert. Wenn die Spieler mit den Gegnern in Kontakt kommen verlieren sie Lebenspunkte. Sollten die Lebenspunkte eines Spielers auf null fallen führt dies zu einem Neustart des Level. Die Gegner müssen besiegt werden um im Level voranzuschreiten und die Tür zum Ziel zu öffnen. Dabei wird vor allem die Kommunikations- und Reaktionsfähigkeit der Spieler auf die Aktionen der Gegner gefordert. Einige dieser Gegner besitzen eine kleine Menge an Lebenspunkten und können schnell besiegt werden. Diese dienen dazu, den Spielern die Kampfmechanik beizubringen und diese anwenden

zu können. Danach begegnen die Spieler mehreren Gegnern mit größerer Menge an Lebenspunkten um die Spieler herauszufordern bevor sie am Ende des Levels auf einen Bossgegner treffen. Dieser ist mit weiteren Fähigkeiten ausgestattet um den Spielern zu schaden und es diesen zu erschweren das Level zu absolvieren. Hier müssen die Spieler beweisen das sie die Kampfmechaniken verstanden haben und auch einsetzen können.

Diese drei verschiedenen Level bieten die Grundlage für die verschiedenen Mehrspielerimplementation. Alle Mechaniken sind leicht erweiterbar und können auch leicht an die Jeweiligen Voraussetzung der Mehrspielerimplementationen angepasst werden. Dabei werden zudem die am weitesten verbreiteten Mechaniken von Plattformerspielen, das genreddefinierende über Plattformen zum Ziel springen, das lösen von Rätseln und das besiegen von Gegner, abgedeckt und in der Untersuchung berücksichtigt. Dies ermöglicht besonders aussagekräftige Ergebnisse über die untersuchten Aspekte.

Um diese Level zu absolvieren besitzen die Spieler einige grundlegende Fähigkeiten. Zunächst haben sie die Hauptmechaniken eines Plattformerspiels, Rennen und Springen. Diese dienen der generellen Bewegung und der Überwältigung von Abgründen oder dem Erreichen höherliegender Positionen. Zusätzlich ist es den Spielern möglich ein Projektil zu schießen, das mit dem Level interagiert. Dabei kann die Interaktion aus zum Beispiel der Aktivierung eines Schalters oder dem Treffen und somit Schaden eines Gegners bestehen. Die Verwendungsweise und die korrekte Nutzung dieser Fähigkeiten gilt es für die Spieler zu verstehen und korrekt anzuwenden um die Level zu absolvieren.

Die Komplexität der Fähigkeiten die die Spieler zur Verfügung haben wird dabei gering gehalten, da sie nicht den Hauptbestandteil dieser Untersuchung darstellen.

3.2 Mehrspielerkonzepte

Da diese Arbeit der Untersuchung der Implementation verschiedener Mehrspielermechaniken dient muss das Konzept des Grundspiels in seinen einzelnen Bestandteilen auf einen Mehrspieler angepasst werden. Dafür werden verschiedene Ansätze gewählt, um diese miteinander vergleichen zu können.

3.2.1 Zweispielermodus ohne zusätzliche Veränderung

In diesem Ansatz wird ermöglicht das Spiel zu zweit zu spielen und ein weiterer spielbarer Charakter wird den Spielern zur Verfügung gestellt. Dieser besitzt grundlegend die gleichen Fähigkeiten wie der erste Charakter und kann durch ein zweites Eingabegerät gesteuert werden. Bei dieser Implementation wird keine Änderung an den Levels vorgenommen. Die Spieler spielen lediglich zu zweit das Spiel und können auf den Kopf des anderen springen um eine erhöhte Position zu erlangen.



Abbildung 3.1: Mehrspieler Modus in Super Mario Bros U

Diese Mehrspielervariante ist in vielen Spielen des Super Mario Franchise vertreten, mit einer kleinen Anpassung der Anzahl an Power-Ups die im Level zu finden sind. Auch hier können die Spieler erhöhte Positionen erreichen indem sie auf den Kopf des jeweils anderen springen. Die Spielercharaktere blockieren sich gegenseitig und können nicht durch den jeweils anderen hindurch laufen. Von dieser Funktion wird in diesem Konzept abgesehen um die Interaktionsmöglichkeiten der Spieler auf ein Minimum zu beschränken. Damit soll das Gefühl erreicht werden, dass die Spieler nebeneinander und nicht miteinander spielen. Trotz dieses schlichten Design handelt es sich um eine sehr erfolgreich Mehrspielervariante, wie andere erfolgreiche lokale Mehrspielerspiele wie Unrailed¹ beweisen. [Ber22] Durch die Simplizität dieser Implementation soll sie als Kontrollwert beziehungsweise als Referenz dienen anhand

¹<https://unrailed-game.com/unrailed.html>

derer die anderen Implementationen gemessen und die Ergebnisse der Evaluation dahingehend analysiert werden.

3.2.2 Modifizierung der Levels

Wie bereits im ersten Ansatz wird ein weiterer Spieler hinzugefügt mit den gleichen grundlegenden Fähigkeiten. Der Unterschied zur ersten Implementation besteht darin, dass die Level abhängig vom Fokus des jeweiligen Levels modifiziert werden.

Im ersten Level in dem der Fokus auf dem gezielten springen und erklimmen des Levels liegt werden einige Passagen des Levels angepasst. Da die Spieler aufeinander stehen und voneinander abspringen können sollen diese Anpassungen sie ermutigen, diese Funktion mehr auszunutzen. Dabei sollen die beiden Spieler motiviert werden sich gegenseitig helfen um schneller das gemeinsame Ziel zu erreichen. [RMP08] Zum Beispiel könnte ein Spieler der keine Probleme mit dem Erklimmen des Level hat dem Mitspieler helfen, eine besonders schwierige Passage zu überspringen. Auch die Kommunikation des optimalen Weges oder kleinen Hinweise zur besseren Steuerung des Charakter ist ein erhofftes Ziel des Design.

Wenn die Spieler ihre Geduld und Kommunikation im ersten Level erprobt haben werden sie im zweiten Level zusätzlich noch auf ihre kognitiven Fähigkeiten getestet. Der in diesem Level bestehende Rätselfokus wurde so angepasst, dass es für einen Spieler alleine nicht möglich ist das Ende zu erreichen. Am Anfang des Levels werden sie mit einer verschlossenen Tür konfrontiert. Diese wird mit einer Druckplatte geöffnet die nur aktiviert bleibt, wenn ein Spieler auf dieser steht. Auf der anderen Seite der Tür gibt es dann eine zweite Druckplatte die nach dem gleichen Prinzip wie die erste funktioniert. Die Spieler müssen sich gegenseitig die Tür öffnen um vereint fortfahren zu können. Das nächste Hindernis bildet eine weitere verschlossene Tür. Diese kann durch einen Schalter geöffnet werden den man mit einem Projektil abschießen muss. Das Problem besteht darin, dass der Schalter von einer Barriere verdeckt wird. Diese kann nur durch eine Druckplatte geöffnet werden. Von der Position der Druckplatte ist es dem Spieler nicht möglich den Schalter abzuschießen. Hier müssen die Spieler zuerst erkennen welches Element die Tür öffnen kann und sich danach so koordinieren, dass ein Spieler bereit ist den Schalter abzuschießen während der andere Spieler auf der Druckplatte steht und die Barriere vor dem Schalter öffnet. Wenn sie auch dieses Rätsel gelöst haben erwartet sie das letzte Rätsel. Dieses nutzt

die Elemente aus den vorherigen zwei Rätseln und fügt noch Plattformen hinzu. Hier werden die Spieler mit angepassten Zeitfenster dazu gebracht ihre Aktionen aufeinander abzustimmen. Tun sie dies nicht reicht die Zeit nicht aus um die Plattformen zu erklimmen und die Spieler stecken fest. Diese Entscheidung im Leveldesign basieren auf dem hoch dekorierten drei dimensional Puzzle Spiel 'Portal 2'². Das Design der Rätsel kann erwiesenermaßen dazu beitragen die Fähigkeit zur Problemlösung zu steigern und hat somit auch außerhalb des Spielspaß einen Mehrwert für die Spieler. [SW15]

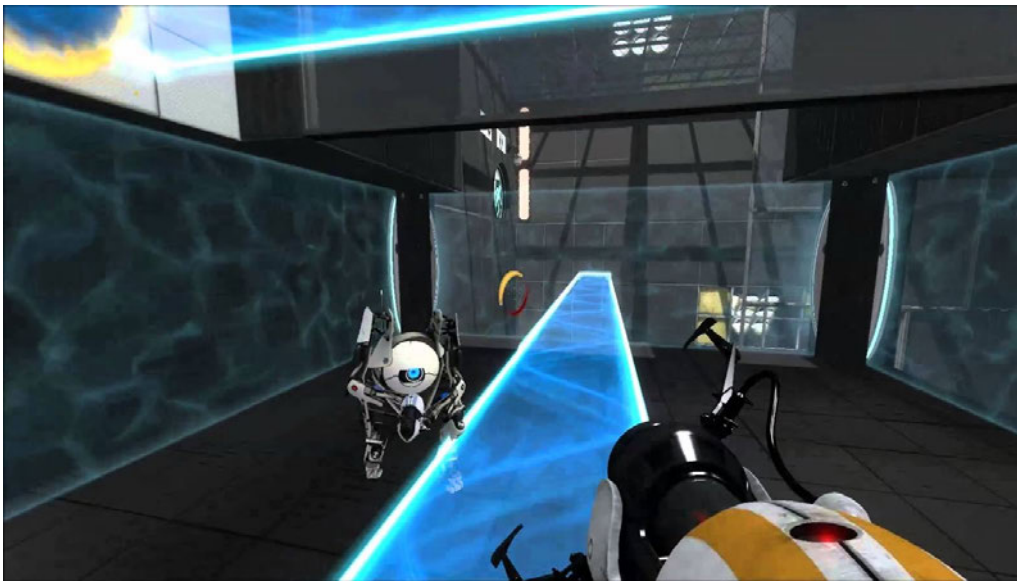


Abbildung 3.2: Mehrspieler Modus in Portal 2

Im dritten, dem auf Gegner fokussierten Level müssen sich beide Spieler zusammen koordinieren, damit sie sich bei dem Besiegen der Gegner nicht gegenseitig behindern. Dabei spielt besonders das sich gegenseitig blockieren eine große Rolle. Zusätzlich werden die Gegner mehr Leben besitzen und können mehr Treffer einstecken. Der Bossgegner stellt in diesem Level die größte Änderung dar. Dessen Fähigkeiten werden so abgeändert, dass jeder Spieler gleichzeitig angegriffen wird. Dadurch entsteht für die Spieler die zusätzliche Herausforderung, dass sie auch den Angriffen auf ihren Mitspieler ausweichen müssen. Sie müssen daher gute Kommunikation und motorische Fähigkeiten besitzen, um neben dem Ausweichen den Boss auch anzugreifen und zu besiegen.

²<https://www.thinkwithportals.com>

3.2.3 Tandem

In der Tandem Variante wird den Spielern kein neuer spielbarer Charakter zur Verfügung gestellt. In diesem Ansatz wird die Kontrolle des einen spielbaren Charakter auf beide Spieler übertragen. Einer der Spieler erhält dabei die Kontrolle über das Laufen und der andere Spieler erhält die Kontrolle über das Springen und das Schießen von Projektilen. Die Level bleiben bei dieser Implementierung unverändert. In diesem Konzept soll untersucht werden, ob das Aufteilen eines Einzelerlebnis auf zwei Spieler zu einem Mehrspielererlebnis führen kann. Dabei soll besonders analysiert werden, ob es die Kommunikation zwischen den Spieler fördert oder hindert und ob es die Spielzeit mit einer positiven oder negativen Bewertung verlängert. Die Inspiration für diese Implementierung stammt aus dem Spiel 'League of Legends' und spezifischer dem All Star Event 2015³. Dabei steuerten jeweils zwei professionelle Spieler einen Charakter, einer an der Maus und einer an der Tastatur. Dabei besitzt der Spieler mit der Maus die Kontrolle über die Bewegung und kann das Ziel der Fähigkeiten bestimmen. Der Spieler an der Tastatur hat die Kontrolle über den Einsatz der verschiedenen Fähigkeiten und Gegenstände. Bei diesem Modus handelt es sich aber um ein künstlich erzeugtes Szenario und keiner im Spiel integrierten Möglichkeit. Spieler können diese Art zu spielen nur reproduzieren, wenn sie zusammen vor einem Gerät sitzen und sich die Peripherie teilen.



Abbildung 3.3: Tandem Bike aus Trials Rising

³<https://www.youtube.com/watch?v=zEUK2qle25w>

Einen integrierten Tandem Modus besitzt hingegen das Spiel 'Trials Rising', ein Motorradrennspiel bei dem der Fahrer seinen Schwerpunkt verlagern und so die Reibung und Rotation des Motorrads steuern kann. Der Tandem Modus fügt hier ein neues auswählbares Motorrad siehe Abbildung 3.3 hinzu, das im lokalen Mehrspieler von zwei Fahrern bemannt hast. Dabei können beide Spieler ihr Gewicht verlagern und direkten Einfluss auf das Fahrverhalten nehmen⁴.

3.2.4 Verbindung der Charaktere

Dieser Ansatz kombiniert das Konzept 'Tandem' und das Konzept 'Zweispielermodus ohne zusätzliche Veränderung'. Es werden den Spielern zwei spielbare Charaktere zur Verfügung gestellt. Jeder Spieler kann einen dieser Charaktere nach links und rechts laufen lassen wie in 'Zweispielermodus ohne zusätzliche Veränderung'. Der Unterschied liegt darin, dass der jeweils andere Spieler den Charakter Springen und Projektile schießen lassen kann. Die Herausforderung bei diesem Konzept liegt in der Koordination des eigenen Charakters in Kombination mit dem Charakter des Mitspielers, da ein Spieler für beide Charaktere relevante Aktionen ausführen kann.



Abbildung 3.4: Shift Happens, Masse wird von einem Charakter zum anderen transferiert

⁴<https://www.rockpapershotgun.com/trials-risings-tandem-mode-preview>

Hier soll untersucht werden ob damit eine zusätzliche und meisterbare Herausforderung geschaffen werden kann. Das Konzept der geteilten Kontrolle wird damit in einem Extrembeispiel getestet. Eine einfachere Variante dieser Implementierung ist im Spiel 'Shift Happens'⁵ zu finden. Hier steuern zwei Spieler in einem lokalen Mehrspieler zwei Charaktere, die sich eine Masse teilen. Diese kann per Knopfdruck von einem Charakter zum anderen transferiert werden und je nachdem wer gerade wenig und wer gerade viel Masse hat haben die Charaktere unterschiedliche Stärke und Größe. Auch hier müssen durch clevere Absprachen der Spieler, wer zu welchem Zeitpunkt wie viel Masse hat, Rätsel gelöst werden.

3.3 Datenerhebung

Während dem Spiel sollen Daten erhoben werden. Diese Daten sollen die Auswertung der Evaluation unterstützen. Die zu erhebenden Daten sollen dabei Aussagen darüber treffen wie schnell die Probanden den Einzelspieler und auch die einzelnen Mehrspielerimplementationen absolvieren konnten. Zusätzlich werden die Rückschläge in der Form von Toden in Level drei erfasst. Wenn die Probanden alle Implementationen gespielt haben sollen von ihnen zusätzliche Aussagen erhoben werden. Da es sich bei dem Spielerlebnis um eine sehr abstrakte Größe handelt lässt sich diese nicht direkt messen. Um trotzdem ein möglichst aussagekräftiges Ergebnis zu erhalten werden die erhobenen quantitativen Daten mit den Aussagen der Probanden verglichen und ein Kontext zwischen ihnen hergestellt. [CWK11]

Werden die Daten manuell erhoben gibt es immer eine Fehlerquote. Um diese zu umgehen wird die Erhebung der Zeitdaten direkt in der Implementierung mit eingeplant. Die Tode werden manuell erfasst, da es sich bei diesen um absolute Werte handelt und die Fehlerquote dabei vernachlässigbar ist.

⁵<http://www.shifthappensgame.com>

4 Implementierung

Dieses Kapitel der Arbeit beschreibt die Implementierung des Spielkonzeptes und der einzelnen Mehrspielervarianten. Dabei wird auf die verwendeten Bibliotheken und Hilfestellungen eingegangen und Fremdleistungen werden angezeigt. Zusätzlich wird der Prozess und die Vorgehensweise bei der Implementierung erläutert und die zugrundeliegenden Gedankengänge offengelegt.

4.1 Implementierung des Grundspiel

Den Beginn der Implementierung bildet die Erstellung eines neuen Projektes mit der Unity Engine. Diese bietet dem Nutzer dafür bereits verschiedene Voreinstellungen abhängig davon, welche Art Projekt erstellt werden soll. Darunter gibt es die Option '2D Core' die für dieses Projekt gewählt wurde. Diese Voreinstellung konfiguriert die Engine automatisch für die Entwicklung einer zweidimensionalen Anwendung.

4.1.1 Charakterbewegung

Es wurde damit begonnen einen Spielercharakter zu erstellen. Dieser sollte sich nach links und rechts bewegen, sowie springen können. Dafür wurde ein bereits bestehendes Package installiert. Ein Package ist dabei eine Ansammlung an verschiedenen Assets die als eine einzelne Datei transferiert werden können. Dieses Package kann dann im Unity Projekt importiert werden und fügt die einzelnen Assets in einer bei Erstellung des Packages definierten Struktur dem Projekt hinzu. Das verwendete Package ist 'Ultimate-2D-Controller' von Matthew Spencer und ist über GitHub frei zugänglich bereitgestellt. [Spe23] GitHub ist eine Plattform für die Versionskontrolle auf der ein Nutzer ein Projekt erstellen kann. Das Projekt kann dabei privat oder wie in diesem Fall öffentlich gestellt werden.[Tsi19] Dieses bereitgestellte Package

verwendet die Unity interne Physikberechnung für Kollisionsabfragen nicht, stattdessen implementiert es Kollisionsabfragen selbst. Dies bietet den Vorteil das sie sich leicht anpassen lässt, um besondere Anforderungen an die Kollision zwischen zwei Objekten zu implementieren. Zudem bietet dieses Package die Funktionen: Variable Sprunghöhe, 'Coyote Time' und Jump Buffering, die in sehr gut bewerteten zweidimensionalen Plattformerspielen wie 'Super Meat Boy' verwendet werden. 'Super Meat Boy' erreicht auf der Website 'Metacritic' eine Kritikerwertung von 90/100 und eine Nutzerbewertung von 7,9/10.¹ Diese etablierten Funktionen sollen dazu dienen, dass die Spielerfahrung für die Probanden möglichst nah an einen vollentwickelten zweidimensionalen Plattformer heran reicht.

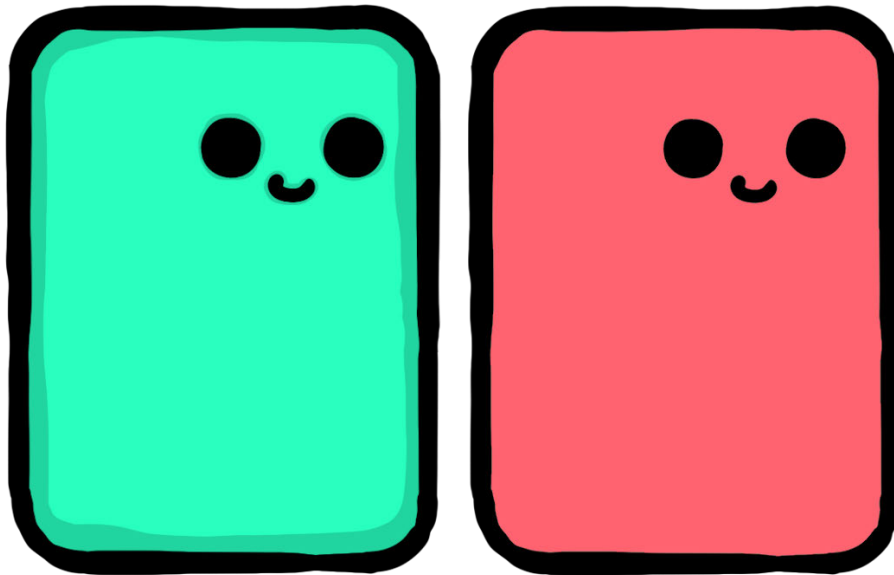


Abbildung 4.1: Die beiden spielbaren Charaktere

Variable Sprunghöhe

Diese Funktion ermöglicht den Spielern unterschiedliche hoch zu springen. Dabei ist die Sprunghöhe abhängig von der Dauer die der Spieler die Taste betätigt und erlaubt dem Spieler eine präzisere Kontrolle über den gesteuerten Charakter. Um diesen Effekt zu erreichen wird überprüft wann die Spieler die Sprungtaste los lassen. In dem Moment wird dem Charakter eine zusätzliche Kraft hinzugefügt, die ihn zum

¹<https://www.metacritic.com/game/super-meat-boy/>

Boden drückt. Die Stärke dieser Kraft kann manuell eingestellt werden und so beim Testen auf den Wert angepasst werden, der sich für die Spieler am besten anfühlt. [Pho23]

'Coyote Time'

Bei 'Coyote Time' handelt es sich um ein Zeitfenster, in welchem es dem Spieler noch möglich ist den Charakter springen zu lassen, obwohl dieser bereits von einer Plattform heruntergelaufen ist und eigentlich nicht mehr springen dürfte. Die Größe dieses Zeitfensters lässt sich dabei manuell einstellen und dient dazu die Spieler nicht dafür zu bestrafen, wenn sie nur leicht zu spät die Eingabe zum Springen geben. [Pho23]

'Jump Buffering'

'Jump Buffering' beschreibt das Speichern der letzten Sprungeingabe und das ausführen dieser Eingabe, wenn sie kurz vor dem Erreichen des Bodens geschieht. Diese Funktion wird verwendet, um die Spieler nicht dafür zu bestrafen, wenn sie die Eingabe nur leicht zu früh betätigen. Damit wird ein flüssigeres Spielen ermöglicht, da eine kleine Fehleingabe nicht zu einem Stoppen der Bewegung führt. Je nachdem wie vergebend das Spiel sein soll kann das Zeitfenster in dem die Eingabe zwischen gespeichert wird größer oder kleiner gewählt werden. [Pho23]

Charakterkollision

Die Kollisionsabfragen des installierten Package überprüfen bei jeder Aktualisierung der Charakterpositionen ob sich über, unter, links oder rechts vom Charakter ein Objekt auf einer bestimmten Ebene befindet. Ebenen dienen in Unity dazu die Interaktionen verschiedener Objekte zu bestimmen. So zum Beispiel welches Objekt zu sehen sein soll, wenn sich zwei Objekte auf unterschiedlichen Ebenen an der gleichen Stelle befinden. Im Fall der Kollisionsabfrage wird nach Objekten auf der 'Ground' Ebene überprüft. Wenn in einer Richtung eines dieser Objekte gefunden wurde so wird die Geschwindigkeit in diese Richtung auf null gesetzt, damit der Charakter sich nicht in diese Objekte hinein bewegt. Um die Wechselwirkung zwischen den beiden Spielern zu fördern, wurde die Überprüfung auf eine weitere Ebene implementiert,

die 'Playerhead' Ebene. Auf diese Ebene wurde jedoch nur nach unten überprüft, um festzustellen ob sich ein Charakter auf dem Kopf eines anderen befindet und dann auf diesem stehen bleiben kann. Durch diese Wechselwirkung besitzt einer der Spielercharaktere eine erhöhte Position und kann somit auch andere erhöhte Positionen erreichen, für die vorher die eigene Sprunghöhe nicht ausreichte. Dies bietet sowohl in der Konzeptionierung von Leveln für den Leveldesigner, als auch beim Spielen der Level für die Spieler weitere Möglichkeiten und Optionen das Level zu bewältigen.

4.1.2 Spielereingaben

Die Eingaben der Spieler werden über das 'New Inputsystem' Package von Unity gesammelt. Dieses Package ermöglicht es ein 'InputActionAsset' zu erstellen. In diesem können mehrere 'ActionMap' Elemente erstellt werden. Dies ist ein Element in dem eine Zusammenstellung aus Eingabeoptionen angelegt und diese dann unterschiedliche Tasten zugewiesen werden können.

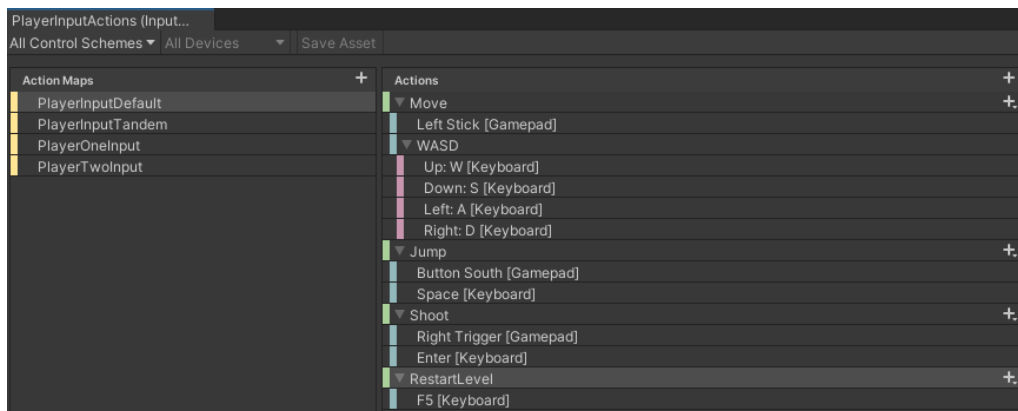


Abbildung 4.2: Überblick über eine 'ActionMap' im 'InputActionAsset'

Dabei besteht die Möglichkeit für unterschiedliche Eingabegeräte separat Tastenbelegungen einzurichten. Dies ist für den lokalen Multiplayer wichtig, damit mehrere Eingabegeräte am gleichen Endgerät verwendet werden können. Wenn eine der zugewiesenen Tasten einer Eingabeoption betätigt wird, dann löst diese ein Event aus. Im Code können Methoden an die Events angehängt werden, um die Eingaben auszulesen und zu verwerten. An jedes Event können beliebig viele Methoden angehängt und zu einem späteren Zeitpunkt wieder abgelöst werden. Diese Modularität

ist für die Implementierung der verschiedenen Mehrspielervarianten in Kapitel 4.2 von Vorteil.

4.1.3 Visualisierung

Für die visuelle Darstellung der einzelnen Level Elemente wurde eine rechteckige 'Tilemap' verwendet. In der Szenenansicht wird dadurch ein Raster aus Vierecken dargestellt. Auf einer 'TilePalette' werden die im Level verwendeten Grafiken eingefügt und können wie auf einer Farbpalette ausgewählt und in das Level gemalt werden. Das Raster ermöglicht eine sehr präzise Arbeit und verhindert ungewollte Überschneidung einzelner Elemente. In einer Szene können mehrere Objekte mit einer 'Tilemap' versehen werden, um unterschiedliche Elemente oder Konstellationen von Elementen einzeln ansprechen zu können. Zusätzlich bietet Unity einen 'Collider' speziell für 'Tilemap' Objekte. Dieser approximiert die Form der dargestellten grafischen Elemente und bildet daraus einen 'Collider' der leistungssparend ist und möglichst genaue Kollisionsabfragen ermöglicht.

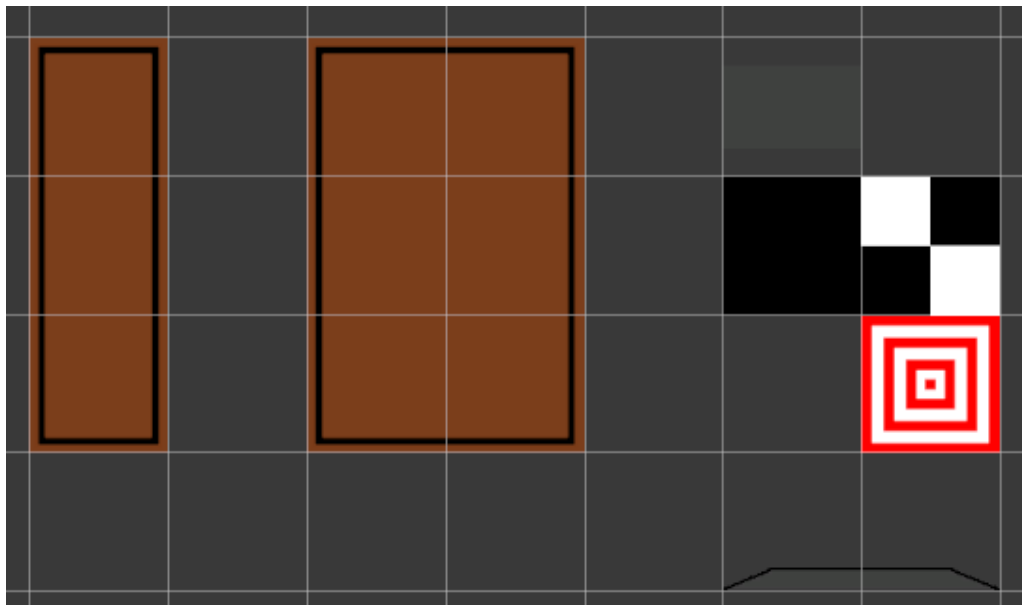


Abbildung 4.3: Verschiedene Level Elemente auf der 'TilePalette'

4.1.4 Levelemente

Für die verschiedenen Level werden verschiedenen funktionale Elemente benötigt. Diese sind das Ziel, Druckplatten, Schalter die durch abschießen aktiviert werden, Waffen, Plattformen und Wände, sowie Gegner, der Bossgegner und Türen die durch das Töten von Gegnern geöffnet werden können.

Ziel

Um ein Level beenden zu können müssen die Spieler den Zielbereich erreichen. Dieser ist visuell mit dem schwarz-weißen Muster gekennzeichnet, welches aus dem Motorsport bekannt ist. Das Ziel wurde als eigenes Element mit einer 'Tilemap' abgebildet und mit einem 'Collider' versehen. Wenn ein Spieler sich in den Bereich dieses Collider begibt, dann wird eine Methode ausgelöst die zuerst überprüft welcher Modus aktuell gespielt wird. Abhängig vom Ergebnis dieser Überprüfung wird dann eine Spieleranzahl ermittelt die sich im Zielbereich befinden müssen, um das Level erfolgreich zu absolvieren. Die aktuelle Anzahl an Spielern im Zielbereich wird dabei zwischengespeichert und jedes Mal, wenn ein Spieler den Zielbereich verlässt oder betritt aktualisiert. Ist die benötigte Spieleranzahl im Zielbereich erreicht wird ein Event ausgelöst und die Nutzeroberfläche gibt den Spielern eine visuelle Rückmeldung, dass sie das Level erfolgreich beendet haben und gibt ihnen die Möglichkeit in das nächste Level zu gehen oder zum Hauptmenü zurückzukehren.

Druckplatten und Schalter

Diese Elemente werden für die Rätsel im zweiten Level benötigt. Für deren Implementierung wurden zunächst die Druckplatten und auf diesen basierend dann die Schalter zum Abschießen implementiert. Die Druckplatten funktionieren dabei ähnlich zum Zielbereich. Für jede Druckplatte wird ein separates Objekt über eine 'Tilemap' im Level platziert. Dann wird jedes dieser Elemente mit einem Collider versehen und wenn sich ein Objekt in diesen Bereich bewegt wird eine Methode ausgeführt. Diese überprüft ob es sich bei dem Objekt um einen Spielercharakter handelt und aktiviert oder deaktiviert dann die ihr im Inspektor zugewiesenen Objekte. Zudem wird für die Druckplatten eine Zeitsteuerung implementiert, die über eine 'Coroutine' zu einem späteren Zeitpunkt die zugewiesenen Objekte wieder in

ihren Ursprungszustand zurück setzt. Eine 'Coroutine' ist dabei ein Prozess der gestartet wird und parallel zu den Skripten läuft. Diese besitzt die Möglichkeit eine bestimmte Zeit abzuwarten, bevor sie den in ihr enthaltenen Code ausführt. Diese Zeitsteuerung schafft neue Optionen beim Leveldesign und ermöglicht es besonders auch im Einzelspieler Zeitdruck auf die Spieler aufzubauen und die Koordination ihrer kognitiven und motorischen Fähigkeiten zu fordern.

Um diese Funktionalität zu auf die Schalter zu erweitern, wurde das Skript um eine Variabel erweitert die entweder wahr oder falsch sein kann. Diese dient dem umschalten zwischen der Druckplatten- und der Schalterfunktion und kann im Inspektor in Unity geändert werden. Wenn der Schalterfunktion aktiviert ist überprüft das Skript nicht, ob ein Spieler den Bereich des 'Collider' betritt sondern ob es ein Projektil ist. Die restliche Funktionalität bleibt gleich. Durch die erweiterte Reichweite mit der diese Schalter aktiviert werden können werden noch einmal neue Möglichkeiten im Leveldesign geschaffen.



Abbildung 4.4: Waffen Power Up

Waffen

Als Waffe wird ein im Level sammelbares Objekt bezeichnet, das dem Spieler die Möglichkeit gibt Projektile horizontal zu schießen und mit diesen Schalter zu aktivieren und Gegner zu besiegen. Diese können aufgesammelt werden indem die Spieler

diese berühren und führen diese dann mit sich. Die Waffe ist ein eigenständiges Objekt und ist mit einem 'Weapon'-Skript versehen, das die Schusslogik und die Parameter der Waffe und das zu instanzierende Projektil enthält. Betätigen die Spieler die Schusseingabe und halten eine Waffe, dann wird die Schussmethode dieser Waffe ausgeführt und in einem kleinen Abstand vor dem Spieler wird das Projektil instanziiert. Dies dient dazu, dass das Projektil nicht direkt mit dem Spieler kollidiert. Das Projektil besitzt ein eigenes Skript, welches die Bewegung des Projektil steuert und die Kollision mit anderen Objekten im Level regelt.

Plattformen und Wände

Den Grundbaustein für die Level bilden Plattformen und Wände. Diese werden mit Hilfe einer 'Tilemap' im Level platziert und der 'Ground' Ebene zugeordnet. Dadurch werden sie von der Kollisionsabfrage erfasst und die Charakter können auf den Plattformen stehen sich nicht durch die Wände hindurchbewegen. Bei den Plattformen gibt es eine Besonderheit, dass einige Plattformen nicht auf der 'Ground' sondern der 'Playerhead' Ebene liegen. Den Spielern ist es bei diesen Plattformen möglich von unten durch sie hindurch zu springen und auf diesen zu stehen. Somit sind sie in eine Richtung durchlässig und es können in den Leveln verschiedene Passagen basierend auf dieser Tatsache entworfen werden.

Gegner, Bossgegner und Gegnertüren

Bei den Gegnern handelt es sich optisch um eine Abwandlung des Spielercharakters in lila, mit einem abgewandelten Gesicht das immer in die Bewegungsrichtung schaut. Diese Darstellung soll den Spielern ermöglichen die Gegner und deren voraussichtliche Bewegung klar zu erkennen sowie eine Verwechslung mit dem Charakter des Mitspielers zu vermeiden. Die Visualisierung des Bossgegner ist ausgehend von den normalen Gegnern verändert, behält aber dabei lila als Grundfarbe bei. Damit kann eine visuelle Zuordnung zu den Gegnern und eine klare Unterscheidung von den normalen Gegner gewährleistet werden.

Da der Fokus dieser Untersuchung auf den Mehrspielervarianten und nicht auf dem Design der Gegner liegen soll wurde sich für ein sehr eindimensionales Design der Gegner entschieden. Diese bewegen sich abhängig von einem Vektor der ihnen im Inspektor übergeben werden kann. Zusätzlich zu dieser Bewegungsrichtung können

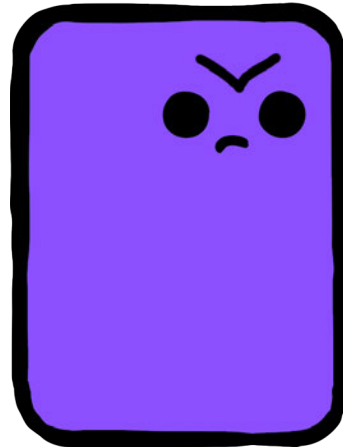


Abbildung 4.5: ein Gegner

im Inspektor eine Zeit und eine Geschwindigkeit übergeben werden. Der Gegner wird sich in den angegebenen Zeitintervallen um 180 Grad drehen und so auf einer von diesen Werten vorgegebenen Strecke patrouilliert. Die Distanz die der Gegner dabei zurück legt kann direkt mit der Zeit und der Geschwindigkeit beeinflusst werden. Damit die Gegner für die Spieler auch eine Gefahr darstellen können, kann ihnen im Inspektor ein Schadenswert zugewiesen werden, den sie bei Kontakt mit einem Spielercharakter an diesem verursachen. Um zu erkennen wann ein Gegner mit einem Spielercharakter kollidiert wird der Gegner mit einem 'Collider' versehen, der als 'Trigger' eingestellt wird. Dies sorgt dafür, dass der 'Collider' keine direkte Kollision hervorruft und die physikalischen Konsequenzen daraus auf das Objekt angewendet, sondern überprüft welche Objekte sich innerhalb des 'Collider' befinden. Im Skript kann dann die 'OnTriggerStay2D' Methode verwendet werden. Diese ist von Unity vordefiniert und wird in geringen Zeitintervallen aufgerufen und enthält als Übergabeparameter das Objekt, das sich innerhalb dieses 'Trigger' befindet. In dieser Methode kann dann ein Event ausgelöst werden, das übergibt welcher Charakter getroffen wurde und wie viel Lebenspunkte dieser Charakter dadurch verliert. Dadurch würde der Spielercharakter jedes Mal wenn diese Methode aufgerufen wird Lebenspunkte verlieren. Durch den geringen Zeitintervall in dem die Methode aufgerufen wird, würde der getroffene Charakter allerdings fast sofort sterben. Um dies zu verhindern muss ein Zeitfenster implementiert werden, das dem getroffenen Charakter Unverwundbarkeit verleiht. Durch das ausgelöste Event kann diese Unverwundbarkeit im 'EntityHealth' Skript geregelt werden. In diesem wird eine Methode an dieses Event angehängt die überprüft, ob seit dem Zeitpunkt

an dem der Charakter das letzte Mal getroffen wurde mehr Zeit vergangen ist, als der Charakter nach einem Treffer unverwundbar ist. Die Zeit der Unverwundbarkeit kann dabei im Inspektor eingestellt werden, um das finden und anpassen für die perfekte Zeit möglichst einfach zu gestalten.

Damit der Bossgegner für die Spieler eine größere Herausforderung darstellt als die normalen Gegner, wird seine Bewegung um eine weitere Richtung erweitert. Dieser patrouilliert nicht nur anhand eines Vektors, sondern alterniert zwischen zwei verschiedenen Vektoren. Zusätzlich zu dieser Erweiterung des Bewegungsmuster wird der Bossgegner mit einem Projektil erweitert. Dieses wird in definierten Zeitintervallen auf den ihm am nächsten Spielercharakter abgefeuert. Dadurch ist es den Spielern nicht möglich ihre Charaktere so zu positionieren, dass sie dem Bossgegner schaden zufügen, gleichzeitig aber nicht von diesem getroffen werden können. Die Projektile können von Plattformen blockiert werden, um den Spielern die Möglichkeit zu bieten, diese zu ihrem strategischen Vorteil zu nutzen.

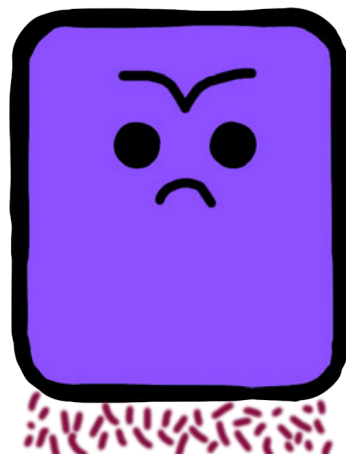


Abbildung 4.6: der Bossgegner

Da die Gegner ihre festgelegten Wege haben und von diesen nicht abweichen wäre es für die Spieler theoretisch möglich dem Konflikt mit diesen gänzlich zu entgehen. Damit dies nicht möglich ist wurde das dritte Level in drei verschiedene Abschnitte gegliedert und diese wurden mit sogenannten Gegnertüren getrennt. In jedem dieser Abschnitte wurden Gegner platziert, im ersten einer, im zweiten vier und im dritten dann der Bossgegner. Für die Gegnertüren wurde das 'EnemyDoor' Skript angelegt. In diesem wird ein Methode an das 'EntityKilled' Event angehängt um zu überprüfen, wie viele Gegner von den Spielern besiegt wurden. Im Inspektor kann

dann festgelegt werden, wie viele Gegner insgesamt besiegt werden müssen, damit die entsprechende Tür geöffnet wird. Im Level sind das konkret für Tür eins ein Gegner, für Tür zwei fünf Gegner und für Tür drei hinter das Ziel liegt alle sechs Gegner.

4.2 Mehrspielerimplementierung

Für die verschiedenen Mehrspielervarianten werden im 'InputActionAsset' verschiedene 'ActionMaps' erstellt. Damit lassen sich für jede Mehrspielervariante unterschiedliche Tastenbelegungen für die gleichen Aktionen festlegen. Dies ist besonders für die Tandem Variante siehe Abschnitt 4.2.3 und die Variante in der die Charaktere verbunden sind im Abschnitt 4.2.4 relevant. Um das Spiel in den verschiedenen Varianten zu initialisieren wurde eine statische Einstellung eingeführt, in der der aktuelle Spielmodus gespeichert wird. Diese Variable wird abhängig von der Auswahl des Spielmodus im Hauptmenü geändert. Da es abhängig von diesem Spielmodus einen oder zwei spielbare Charaktere gibt wurde ein Manager-Script, fortlaufend als Playermanager bezeichnet, erstellt, das beim Laden des Levels die richtige Anzahl an Charakteren initialisiert.

Die Mehrspielervarianten sind in der Implementierung wie folgt bezeichnet: 'Zweispielermodus ohne zusätzliche Veränderung' als 'Multiplayer', 'Modifizierung des Levels' als 'Multiplayer Plus', 'Tandem' als 'Tandem' und 'Verbindung der Charaktere' als 'Shared Controls'.

4.2.1 Zweispielermodus ohne zusätzliche Veränderung

Um das Spiel in dieser Mehrspielervariante mit einem zweiten individuell steuerbaren Charakter zu erweitern wurden zwei weitere 'ActionMaps' erstellt. Dabei wurde eine 'ActionMap' für die Eingaben von Spieler eins und eine für die Eingaben von Spieler zwei erstellt. Mit diesen wurde der Playermanager für den Spielmodus Mehrspieler erweitert. Wenn dieser ausgewählt ist initialisiert der Playermanager für jeden Charakter eine neue Instanz des 'InputActionAsset' und hängt die Methoden für die Steuerung des an die Events der zuständigen 'ActionMap' an. Damit kann gewährleistet werden, dass beide Spieler nicht ungewollt den Charakter des anderen

beeinflussen können. Zudem bietet diese Variante den Vorteil, dass nachträgliche Änderungen an der Tastenbelegung für die einzelnen Charaktere in Unity und nicht im Code angepasst werden müssen. Diese Variante wurde als erstes implementiert und als Basis für die Implementierung im nachfolgenden Abschnitt verwendet.



Abbildung 4.7: Layout des ersten Levels

4.2.2 Modifizierung des Levels

In dieser Variante wird die gleiche Steuerung wie im vorherigen Abschnitt verwendet und konnte somit komplett übernommen werden. Dabei werden im Playermanager bei der Initialisierung im modifizierten Mehrspielermodus die gleichen Methoden an die gleichen Events angehängt. Die Änderung in diesem Modus stellten die Level und die Objekte in den Levels dar, mit denen die Spieler interagieren können.

Level eins erhielt ein neues Layout. Dabei war die Aufgabe für die Spieler immer noch von unten nach oben zu gelangen und die Spitze des Levels durch gekonnte Sprünge von Plattform zu Plattform zu erreichen. Die Position der verschiedenen Plattformen und der Weg durch das Level der sich aus ihnen ergibt wurde geändert. Damit sollte untersucht werden, wie sich eine Veränderung des Levellayout auf die Spieler auswirkt.

Im zweiten Level wurde das generelle Layout beibehalten und auch die Position der Elemente im Level, mit denen die Spieler interagieren können blieb gleich. Die

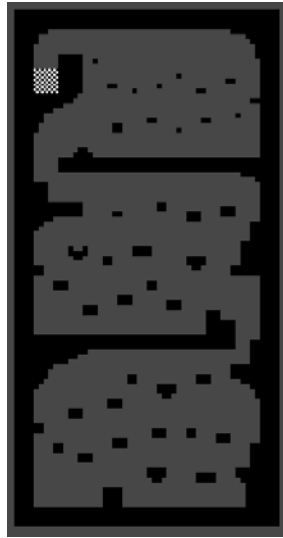


Abbildung 4.8: neues Layout für das erste Level

Veränderung in Level bildeten die Funktion der Level Elemente wie Druckplatten und Schaltern. Einige Druckplatten wurden so angepasst, dass einer der Spieler auf ihnen stehen bleiben musste, damit ihr Effekt aktiv blieb. Dieser war entweder das Öffnen einer Tür oder das Entfernen einer Barriere vor einem Schalter. Zudem wurden die Zeitfenster die die Spieler haben um bestimmte Aktionen im Level durchzuführen so angepasst, dass es für einen Spieler alleine nicht möglich ist diese Passagen zu überwinden. Dies sollte die Kommunikation der Spieler untereinander fördern und sie die Rätsel so noch einmal aus einer neuen Perspektive erleben lassen.

Aufgrund dieser umfangreichen Anpassungen in Level eins und zwei wurde für beide Level eine weitere Szene erstellt. In dieser wurde das Level initial kopiert und anschließend modifiziert. Bei der Navigation durch die einzelnen Level wurden dann im modifizierten Mehrspielermodus eben diese modifizierten Level anstelle ihrer Basisversionen geladen.

Im dritten und letzten Level wurden keine Anpassungen am Layout oder den Elementen im Level vorgenommen. Stattdessen wurde das Skript, das die Leben eines Gegners kontrolliert angepasst. Diesem kann man in Unity im Inspektor zwei Werte übergeben wobei einer die normalen Leben des Gegners darstellt und der andere die Leben im modifizierten Mehrspielermodus. Wenn das Level dann geladen wird überprüft das Skript in welchem Modus sich das Spiel aktuell befindet und setzt entsprechend die Lebenspunkte die der Gegner besitzt auf die normalen Leben-

spunkte oder die modifizierten Lebenspunkte. Diese Änderung soll als Ausgleich dienen, dass die Spieler im Mehrspieler zu zweit auf die Gegner schießen können und somit auch die doppelte Menge an Schaden zur Verfügung haben um die Gegner zu besiegen. Eine weitere Anpassung wurde beim Bossgegner vorgenommen. Dieser schießt im normalen Modus ein Projektil auf den Spieler, der gerade die geringste Distanz zu ihm besitzt. Das Skript für den Boss wurde so angepasst, dass er im modifizierten Mehrspielermodus auf jeden der Spieler ein Projektil schießt und somit beide Spieler dauerhaft unter Druck setzt. Dies soll verhindern, dass ein Spieler den Bossgegner alleine bekämpft während der andere Spieler nur auf Abstand bleibt und vom Bossgegner gar nicht ins Visier genommen wird. Zudem soll es den Zweck erfüllen, dass Spielen miteinander und nicht nebeneinander her zu fördern, da die Spieler sich auch untereinander koordinieren müssen um sich nicht versehentlich gegenseitig abzuschießen.

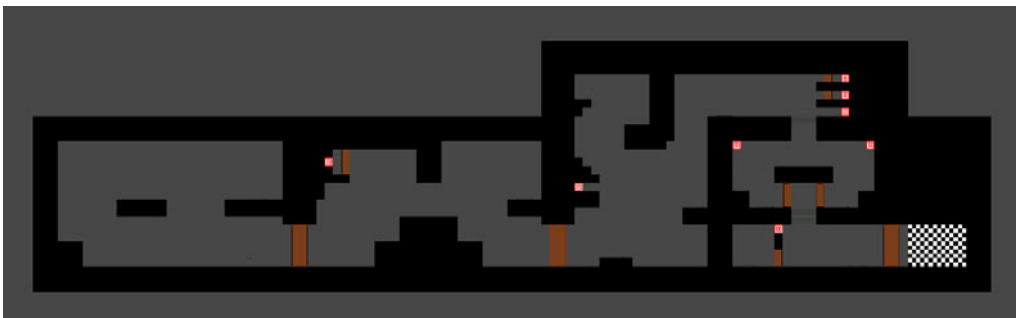


Abbildung 4.9: Layout des zweiten Levels

4.2.3 Tandem

Für den Tandemmodus wird vom Playermanager nur ein Charakter instanziiert und dessen Bewegungsmethoden getrennt an die Eingaben beider Spieler gekoppelt. Um diese Trennung der Eingaben vorzunehmen wurde eine weitere 'ActionMap' erstellt. Auf dieser wurden anders als in der 'ActionMap' für den Einzelspieler bei den einzelnen Eingabeoptionen nur Tastenbelegungen für entweder Tastatur oder Spielecontroller festgelegt. Dabei wurde für die Eingabe der Bewegung nach links und rechts die Tasten 'A' und 'D' auf der Tastatur und für das Springen 'Button South' und für das Schießen 'Right Trigger' auf einem Spielecontroller festgelegt. Die Tastatur stellt in diesem Fall das Eingabegerät für Spieler eins und ein Spielecontroller das Eingabegerät für Spieler zwei dar. Mit dieser Zuweisung der ver-

schiedenen Funktionen des Charakters können beide Spieler weiterhin bequem ihr eigenes Eingabegerät verwenden. Dabei werden sie kognitiv gefordert sich entweder mit ihrem Mitspieler abzusprechen oder über Hand-Augen-Koordination schnell auf die Aktionen des Mitspielers zu reagieren, um das angestrebte Ziel zu erreichen.

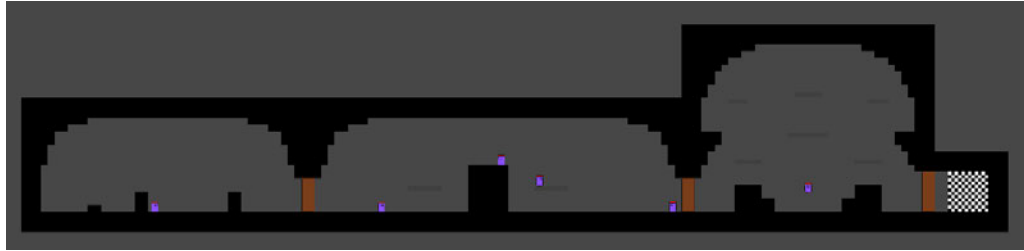


Abbildung 4.10: Layout des dritten Levels

4.2.4 Verbindung der Charaktere

Dieser Modus stellt eine Erweiterung des Tandemmodus dar, um die Spieler in ihrer Hand-Augen-Koordination noch stärker zu fordern. Dabei wird für beide Spieler ein eigener Charakter instanziiert, die Steuerung erfolgt jedoch wie im vorherigen Kapitel zum Tandemmodus beschrieben. Das heißt Spieler eins gibt die Eingaben für zum laufen für Charakter eins und die Eingaben zum Springen und Schießen für Charakter zwei und umgekehrt gibt Spieler zwei die Eingaben zum laufen für Charakter zwei und die Eingaben zum Springen und schießen für Charakter eins. Dies stellte bei der Implementierung eine kleine Herausforderung dar, da weder die Eingabe für den normalen Mehrspielermodus noch die Eingabe für den Tandemmodus direkt verwendet werden konnten. Um diese zu bewältigen wurde die jeweilige 'ActionMap' für die Eingabe von Spieler eins und die Eingabe von Spieler zwei im Playermanager den Spielern zugeordnet. Die Methoden für die Bewegung nach links und rechts wurden dabei wie bereits im Mehrspielermodus an die jeweiligen Events angehängen. Für das Springen und das Schießen wurde nach dem Instanzieren der beiden Spielercharakter auf das jeweilige 'PlayerController' Script zugegriffen und die Methoden 'OnShoot' und 'OnJump' wurden wie in Abbildung 4.11 zu sehen an die Eingaben des jeweils anderen Spielers gekoppelt. Damit wurde vermieden, dass Eingaben zu einer Aktion bei beiden Charakteren führen können.

```
private void InitializeSharedControls() {
    PlayerController playerOneController = _playerOne.GetComponent<PlayerController>();
    GameObject playerTwo = Instantiate(playerTwoPrefab, _spawnTransform.position, _spawnTransform.rotation);
    PlayerController playerTwoController = playerTwo.GetComponent<PlayerController>();
    GameEvents.Instance.OnPlayerSpawn(playerID: 1, playerTwo.transform);

    _playerInputActions.PlayerOneInput.Move.performed += playerOneController.OnMove;
    _playerInputActions.PlayerOneInput.Move.canceled += playerOneController.OnMove;
    _playerInputActions.PlayerOneInput.Jump.performed += playerTwoController.OnJump;
    _playerInputActions.PlayerOneInput.Jump.canceled += playerTwoController.OnJump;
    _playerInputActions.PlayerOneInput.Shoot.performed += playerTwoController.OnShoot;
    _playerInputActions.PlayerOneInput.Enable();

    _playerInputActions.PlayerTwoInput.Move.performed += playerTwoController.OnMove;
    _playerInputActions.PlayerTwoInput.Move.canceled += playerTwoController.OnMove;
    _playerInputActions.PlayerTwoInput.Jump.performed += playerOneController.OnJump;
    _playerInputActions.PlayerTwoInput.Jump.canceled += playerOneController.OnJump;
    _playerInputActions.PlayerTwoInput.Shoot.performed += playerOneController.OnShoot;
    _playerInputActions.PlayerTwoInput.Enable();
}
```

Abbildung 4.11: Anheften der Methoden an die entsprechenden Eingaben

5 Evaluation

Mit der Implementierung wurde die technische Vorbereitung für die Evaluation abgeschlossen und kann durchgeführt werden. Im folgenden Kapitel werden die Ziele der Evaluation definiert und die Durchführung der Evaluation geplant. Anschließend erfolgt die Analyse der erhobenen Daten und die Auswertung der Ergebnisse.

5.1 Ziele der Evaluation

Die Erweiterung eines lokalen Mehrspieler Plattformers durch verschiedene Mehrspielermechaniken kann unter bestimmten Betrachtungspunkten einen Mehrwert für den entsprechenden Titel generieren. Eine umfassende Aussage darüber zu treffen wurde, wie eingangs in Kapitel 1 beschrieben als Ziel dieser Arbeit definiert. Dafür muss geklärt werden, welche Betrachtungspunkte betrachtet werden und wie sich diese erfassen lassen.

Als ersten quantifizierbaren Punkt ist die Zeit zu nennen. Diese lässt sich wie in Kapitel 3.3 beschrieben direkt im Spiel erfassen. Die erfassten Werte können dann tabellarisch dargestellt und miteinander verglichen werden. Ein Vergleich der Wert ermöglicht eine direkte Aussage über die gesamte Spielzeit und ob diese kürzer, länger oder ähnlich der Zeit aus dem Einzelspieler Modus ist.

Die Zeit alleine kann allerdings keine ausreichende Aussage über den Mehrwert liefern, da es sich bei Videospielen um Medien der Unterhaltungsbranche handelt. Deshalb muss das Spielerlebnis der einzelnen Spieler quantifiziert und in Verbindung zur erfassten Zeit gebracht werden. Dafür wird eine Befragung der Probanden in der Form einer Umfrage mit 'Google Forms'¹ vorbereitet. Auf den konkreten Aufbau dieser Umfrage wird im Abschnitt zur Durchführung der Evaluation eingegangen.

¹<https://www.google.com/forms/about/>

5.2 Durchführung der Evaluation

Das Experiment wird in Gruppen von jeweils zwei Personen durchgeführt. Die Testgruppen wurden dabei nicht zufällig zusammengestellt, sondern haben sich jeweils als Paar für die Evaluation angemeldet. So wird der Rahmen in dem das Experiment durchgeführt wird am besten an die Realität angepasst, dass sich zwei Personen treffen um ein lokales Mehrspieler Spiel zu spielen. Das Experiment selbst wird lokal unter Anwesenheit des Evaluierenden durchgeführt.

5.2.1 Ablauf des Experiments

Eine Person spielt zuerst den Einzelspieler Modus um eine Vergleichsgrundlage zu den Mehrspielerimplementationen zu schaffen. Danach beginnt das Testen der einzelnen Mehrspielerimplementationen. Zuerst spielen die Probanden den Modus 'Zweispielermodus ohne zusätzliche Veränderung', da dieser in seiner Funktionsweise dem Einzelspieler am nächsten liegt. Über den Verlauf des Experiments kann die Wahl der Eingabegeräte beliebig häufig getauscht werden, es spielt aber immer eine Person an der Tastatur und eine mit einem Spielecontroller. Diese Freiheit wird den Probanden erlaubt um die Testsituation so nah wie möglich an die Realität anzupassen. Danach wird der 'Tandem' Modus gespielt. Dieser wird direkt vor dem Modus 'Verbindung der Charaktere' gespielt, damit sich die Probanden an die Steuerung gewöhnen können und sie von der Mechanik nicht überfordert werden. Als letzten Modus spielen die Probanden dann den 'Modifizierung des Levels' Modus. Dieser ist bewusst der letzte getestete Modus, da sich so die Level zwischen den verschiedenen Modi nur einmal am Ende ändern.

Während die Probanden die einzelnen Modi spielen wird am Ende jedes Level die erreichte Zeit notiert. Zusätzlich werden für Level drei die Anzahl der Tode aufgenommen, unabhängig davon von welchem Charakter die Lebenspunkte auf null reduziert werden. Dieser Wert soll eine Aussage über die Herausforderung in den verschiedenen Modi ermöglichen.

Wenn alle Modi von den Probanden gespielt wurden erhalten sie Zugriff auf die vorbereitete Umfrage. In dieser werden zuerst allgemeine Daten wie Alter, Geschlecht und wie häufig sie Mehrspielerspiele spielen erhoben. Mit diesen soll eine

Aussage über die Probanden und anhand dessen über die Allgemeingültigkeit dieser Untersuchung getätigt werden.

Welchem Geschlecht fühlst du dich zugehörig? *

männlich

weiblich

divers

Wie alt bist du? *

< 18 Jahre

18 - 24 Jahre

25 - 30 Jahre

30 - 50 Jahre

> 50 Jahre

Wie häufig spielst du Mehrspielerspiele? *

1 2 3 4 5

nie fast ausschließlich

Abbildung 5.1: Fragen zu den allgemeinen Daten

In diesem ersten Teil der Umfrage werden die Probanden zudem gefragt ob sie den Einzelspieler gespielt haben oder nicht. Diese Frage dient lediglich dazu, dass die Probanden die den Einzelspieler Modus gespielt haben zu einem separaten Teil der Umfrage weitergeleitet werden. Dort werden diese Probanden gebeten ihr Spielerlebnis im Einzelspieler Modus zu bewerten und mit den verschiedenen Mehrspieler Modi zu vergleichen. In der gesamten Umfrage werden die Modi mit den gleichen Namen bezeichnet wie sie in der Implementierung enthalten und in Abschnitt 4.2 aufgeführt sind.

Der Abschnitt der Umfrage danach ist wieder an alle Probanden gerichtet. Die Probanden werden darin gebeten ihr von den anderen Modi losgelöstes Spielerlebnis

zu bewerten. Zudem werden sie befragt ob eine der Mehrspielervarianten ausschlaggebend für sie sein kann, ein Spiel mit der entsprechenden Variante zu spielen.

The figure displays three separate Likert scales, each with five response options and four rows of variants. The first scale asks for an overall evaluation of the player's experience. The second scale asks if the respondent would play a game because of a specific multiplayer variant. The third scale asks if the respondent would replay a single-player game with a specific multiplayer variant.

Wie würdest du dein Spielerlebnis bewerten? *					
	sehr schlecht	schlecht	ok	gut	sehr gut
Multiplayer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tandem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Shared Controls	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Multiplayer Plus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Würdest du ein Spiel wegen dieser Mehrspielervariante spielen? *					
	niemals	eher nicht	unsicher	wenn mich jem...	ich würde jema...
Multiplayer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tandem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Shared Controls	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Multiplayer Plus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Würdest du ein Singleplayer Spiel mit dieser Mehrspielervariante erneut spielen? *					
	niemals	eher nicht	unsicher	wenn mich jem...	ich würde jema...
Multiplayer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tandem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Shared Controls	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Multiplayer Plus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Abbildung 5.2: Fragen zu den einzelnen Mehrspielervarianten

Eine Erweiterung des Spielerlebnis kann auch durch die Erweiterung eines Einzelspieler Spiel mit einem Mehrspieler Modus geschehen. Um zu überprüfen ob die untersuchten Varianten dies erreichen können werden die Probanden gebeten die Varianten diesbezüglich zu bewerten.

Abschließend werden die Probanden noch gebeten andere Mehrspielervarianten und Beispiele für diese anzugeben, die von dieser Untersuchung nicht abgedeckt wurden.

5.2.2 Probleme

Bei der Durchführung der Experimente wurden einige Probleme mit dem Prototyp festgestellt. Diese konnten teils behoben werden, teils musste eine andere Möglichkeit gefunden werden um diese Probleme zu umgehen. Ein kleineres Problem stellten fehlende 'Collider' dar, wodurch es den Probanden teils möglich war einfach durch Türen hindurch zu laufen ohne das nötige Rätsel zu lösen oder die Gegner zu töten. Da das erste Experiment direkt in der Unity Engine durchgeführt wurde, konnte dies direkt während der Durchführung des Experimentes behoben werden. Die Probanden wurden danach gebeten die Level in denen das Problem auftrat erneut zu spielen und so die benötigten Testdaten zu erhalten.

Besonders im zweiten Level wurde ein größeres Problem festgestellt. Es trat teils auf, dass der Spielcharakter in die Plattformen oder Wände geriet. Dadurch verlor der Spieler die Kontrolle über seinen Charakter und dieser hat sich entweder von alleine in der Plattform oder Wand bewegt oder bleibt komplett stecken. Die Bewegung des Charakters war dabei abhängig von der Bewegungsrichtung des Charakters bevor er in die Wand oder Plattform geriet.

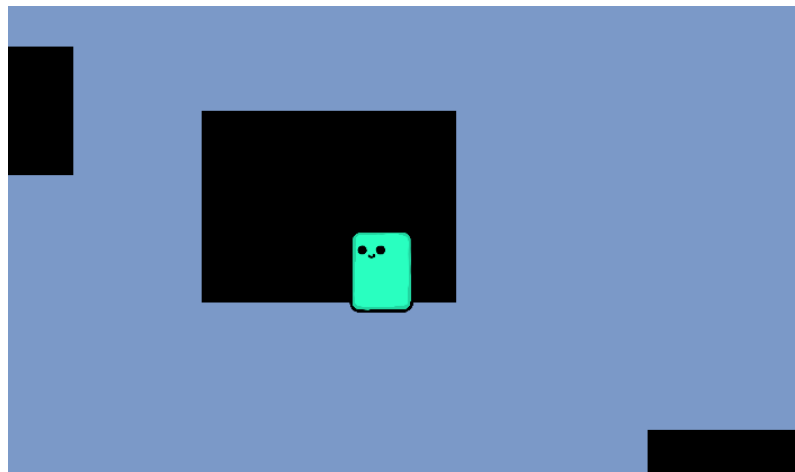


Abbildung 5.3: Charakter(türkis) stecht in einer Plattform(schwarz) fest

Dieses Problem konnte nicht so leicht behoben werden wie das Problem mit den Türen. Damit das Problem die Evaluation nicht weiter beeinflusst wurde eine Funktion implementiert, um das aktuelle Level neu zu laden und somit alle Charakter wieder an ihre Startposition zu setzen. Dies hat allerdings zur Folge das auch die integrierte Zeitmessung wieder auf null zurückgesetzt wird. Da es ungewiss war wann dieses Problem im Level auftritt erhalten den Spieler mit jedem Neustart die Möglichkeit ihr erlangtes Wissen über das Level zu nutzen. Damit ist es ihnen möglich das Level schneller als zuvor zu absolvieren und somit werden die Testdaten beeinflusst. Um dies zu umgehen wurde von der Neustartfunktion abgesehen. Stattdessen wurde beim Auftreten des Problems die Anwendung in der Unity Engine pausiert. Dann wurde der Charakter der feststeckte in der Szenenansicht manuell aus der Plattform oder der Wand heraus bewegt. Anschließend konnte das Experiment normal fortfahren. Der Vorteil bei dieser Variante ist das die Spieler den Weg bis zu ihrem aktuellen Punkt im Level nicht erneut spielen müssen. Zudem bleibt die Zeit die die Probanden für das Level benötigen möglichst akkurat, da das pausieren der Anwendung in der Engine auch die Funktion mit der die Zeit erfasst wird pausiert.

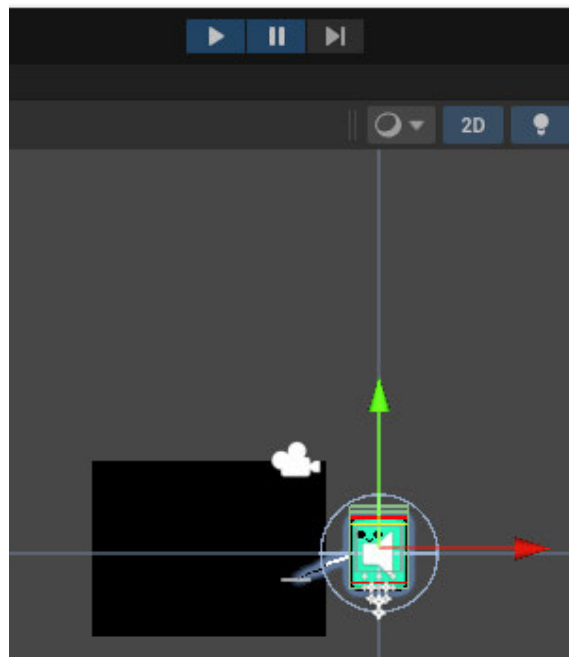


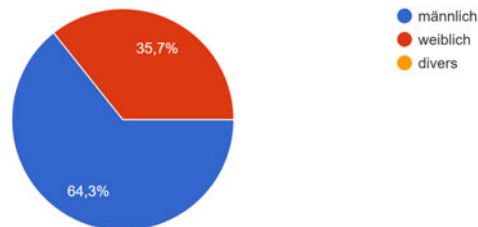
Abbildung 5.4: Charakter(türkis) wurde aus der Plattform(schwarz) bewegt

5.3 Auswertung der Evaluation

Im Rahmen der Evaluation haben 14 Probanden den Prototyp in Gruppen aus jeweils zwei Personen gespielt. Dabei waren neun Probanden männlich und fünf weiblich und elf dieser im Altersspektrum von 18 bis 24 Jahren und drei im Spektrum 25 bis 30 Jahre. Im Kontext der Evaluation wird der Begriff Spiel stellvertretend für ein zweidimensionales Plattformerspiel mit einem lokalen Mehrspieler Modus verwendet.

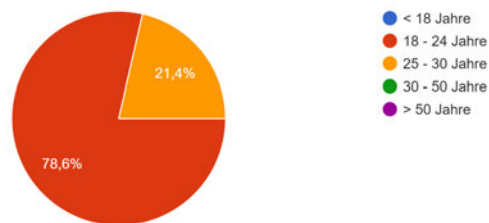
Welchem Geschlecht fühlst du dich zugehörig?

14 Antworten



Wie alt bist du?

14 Antworten



Wie häufig spielst du Mehrspielerspiele?

14 Antworten

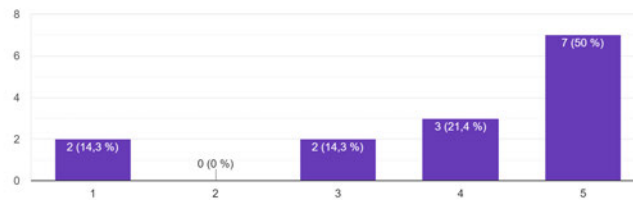


Abbildung 5.5: Allgemeine Daten über die Probanden

Ein Großteil der Probanden spielt sehr häufig Mehrspieler Spiele. Es haben aber auch Probanden teilgenommen, die wenig bis gar keine Mehrspieler Spiele spielen.

5.3.1 Erhobene Spieldaten

Als erstes wurden die in den Experimenten erhobenen Daten tabellarisch dargestellt. Dabei wurden die Namen der Modi für eine kompaktere Darstellung in Tabelle 5.1 wie folgt abgekürzt: 'Einzelspieler' als 'SP', 'Zweispielermodus ohne zusätzliche Veränderung' als 'MP', 'Tandem' blieb gleich, 'Verbindung der Charaktere' als 'Shared' und 'Modifizierung des Levels' als 'MP+'. Zusätzlich zu den erfassten Daten wird in dieser Tabelle auch der Durchschnitt für die Werte in den jeweiligen Modi dargestellt.

Die durchschnittliche Zeit in der die Probanden den Einzelspieler Modus absolvierten wird als Referenzwert verwendet und mit den Zeiten der anderen Modi verglichen. Bei der Betrachtung der Zeiten für Level eins kann festgestellt werden, dass die Modi 'Tandem' und 'Modifizierung des Levels' eine geringere Spielzeit besitzen. Dabei überrascht die geringere Spielzeit im 'Tandem' Modus besonders, da dieser in seinem Konzept als Herausforderung an die Spieler angedacht war. Dies lässt sich wahrscheinlich darauf zurück führen, dass die Probanden zuvor das Level einmal im Einzelspieler und einmal im 'Zweispielermodus ohne zusätzliche Veränderung' gespielt haben. Dadurch ist ihnen das Layout des Levels bereits bekannt und die Probanden konnten direkt zielstrebig das Level beenden. Die geringere Zeit im 'Modifizierung des Levels' Modus lässt sich entweder auf ein weniger anspruchsvolles Leveldesign oder darauf, dass die Probanden beim Test dieses Modus am meisten Übung mit der Steuerung hatten, da er als letztes getestet wurde. In den anderen beiden Modi kann eine höhere Spielzeit festgestellt werden. In beiden Fällen handelt es sich um einen Modus der direkt zuvor in einer ähnlichen Variante mit nur einem Charakter gespielt wurde. Im Fall des 'Zweispielermodus ohne zusätzliche Veränderung' kann sich die erhöhte Spielzeit damit erklären lassen, dass einer der Probanden zum ersten Mal das Spiel spielt und sich somit an die Steuerung gewöhnen muss. Hinzu kommt das die Probanden das erste Mal zwei Charaktere steuern und somit die Wechselwirkungen zwischen diesen erkunden. Beim 'Verbindung der Charaktere' Modus ist anzunehmen, dass die sogar deutlich erhöhte Zeit auf die erfolgreiche Verbindung an kognitiver und motorischer Herausforderung zurückzuführen ist. Während des Experimentes konnte beobachtet werden, dass die Probanden gehäuft versuchten

Modus		Daten der Gruppen							\ominus
SP	Level 1	130,6	196,8	136,1	720,9	356,2	197,8	620,7	337
	Level 2	140	77,5	111,7	471,1	179,8	105,8	146	176
	Level 3	90	76,8	140	155,8	187,5	118,9	99,5	124,1
	Deaths	0	1	0	8	10	0	4	3,3
MP	Level 1	240	459,1	611,1	931,4	252,8	479,2	273,2	463,8
	Level 2	174	115,9	75,8	480,6	123	96,5	253	188,4
	Level 3	110	62,6	79,5	93	79,4	81,9	72,4	82,7
	Deaths	1	0	3	1	2	4	1	1,7
Tandem	Level 1	414	180,6	333,7	200,2	218,8	231,8	351	275,7
	Level 2	149,7	65,6	147,9	183,5	214,2	293	196,5	178,6
	Level 3	96	87,2	77,3	123,2	154	108,3	124,6	110,1
	Deaths	1	0	3	2	2	7	4	2,7
Shared	Level 1	1067,7	691,4	589,9	1869,3	873,9	163,4	649,2	843,5
	Level 2	474,6	148,9	318,7	389	569,8	161,4	173,4	319,4
	Level 3	151,3	178,3	119,2	296,2	238,6	85,2	253,1	188,8
	Deaths	1	3	1	0	5	1	3	2
MP+	Level 1	218,3	105,1	217,6	384,4	216,4	245,2	286	239
	Level 2	80,6	354,4	192,7	324,6	112,6	142,7	186,9	199,2
	Level 3	93	79,9	89	275,4	81,9	130,2	165,7	130,7
	Deaths	0	2	1	0	0	0	3	0,9

Tabelle 5.1: Testdaten: Zeit für die Level in Sekunden, Tode als absoluter Wert, Durchschnitt \ominus der Werte aller Testgruppen

ihren eigenen Charakter Aktionen ausführen zu lassen, obwohl ihr Partner die Kontrolle über diese Aktionen besaß. Dies führte meist dazu, dass der entsprechende Charakter von seiner Plattform fiel und einige Passagen erneut überwältigt werden mussten.

Im zweiten Level sind die Zeiten der einzelnen Modi sehr nah beieinander bis auf die Zeit im 'Verbindung der Charaktere' Modus. Dieser ist ebenso wie in Level eins mit einer deutlich erhöhten Zeit erfasst worden. Dabei kann die Eingewöhnung in die Steuerung nur noch bedingt als Faktor genannt werden, da diese bereits in Level eins geschehen ist. Die Herausforderung gleichzeitig Eingaben für zwei verschiedene Charaktere zu tätigen führte dazu, dass einzelne Passagen des Levels nacheinander

einzelnen absolviert wurden. Dies benötigt entsprechend mehr Zeit und ist als ein Grund für die erhöhte Zeit aufzuführen.

Bei der Betrachtung von Level drei ist die Anzahl der Tode als zusätzlicher Parameter nicht außer Acht zu lassen. Diese sind in jedem Modi geringer als im Einzelspieler. Zusätzlich fällt auf, dass die Tode in den Modi in denen die Probanden zwei Charakter zur Verfügung haben besonders gering sind. Mit einem separaten Pool an Lebenspunkten haben die Spieler in diesen Modi die doppelte Menge an Lebenspunkten. Es konnte beobachtet werden, dass meist nur ein Spieler die Gegner bekämpft während der andere sich in sicherer Entfernung befindet. Wenn die Lebenspunkte des aktiven Spielers sehr niedrig sind wurde gewechselt um das Risiko das ein Charakter stirbt möglichst gering zu halten. Diese Spielweise führte allerdings auch dazu, dass im 'Verbindung der Charaktere' Modus die Zeit in Level drei deutlich höher und im 'Modifizierung des Levels' Modus geringfügig erhöht ist. Im 'Zweispelermodus ohne zusätzliche Veränderung' konnte jedoch eine deutlich niedrigere Spielzeit erfasst werden. Level drei wurde im 'Tandem' Modus nur leicht schneller absolviert und weißt weniger Tode vor.

5.3.2 Ergebnisse der Umfrage

Wie würdest du den Singleplayer bewerten?

7 Antworten

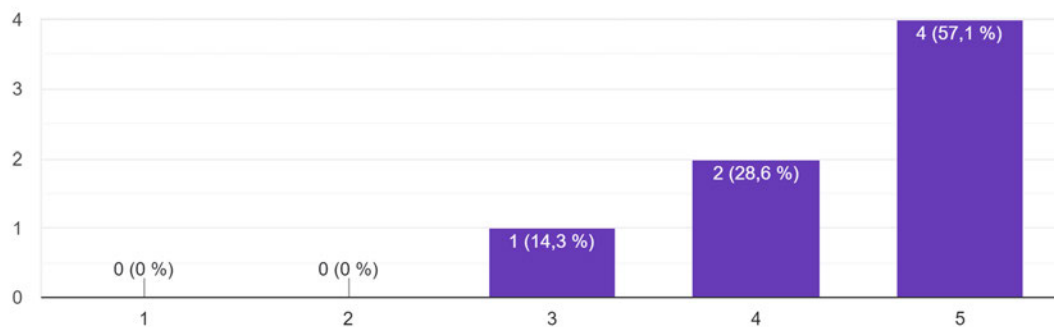


Abbildung 5.6: Probandenbewertung des Einzelspieler Modus

In Abbildung 5.6 ist zu sehen das der Einzelspieler Modus des Prototyp positiv bewertet wurde. Damit bietet er eine gute Grundlage um mit den anderen Modi verglichen zu werden. Direkt im Anschluss zur Bewertung des Einzelspieler Modus wurden die Probanden die diesen gespielt haben gebeten, ihr Spielerlebnis in den anderen Modi mit dem aus dem Einzelspieler zu vergleichen.

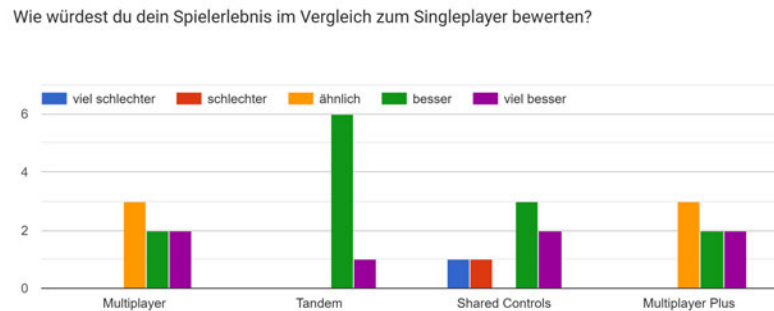


Abbildung 5.7: Bewertung des Spielerlebnis der verschiedenen Mehrspielervarianten im Vergleich zum Einzelspieler

Dem Ergebnis dieser Befragung in Abbildung 5.7 kann entnommen werden, dass trotz der bereits guten Bewertung des Einzelspieler die Mehrspielervarianten alle gleich bis besser bewertet wurden. Der Modus bei dem die Probanden ihren eigenen und den Charakter ihres Partners steuerten wurde dabei am schlechtesten bewertet und der 'Tandem' Modus am besten. Es kann also angenommen werden, dass die gezwungene Teamarbeit beider Spieler im 'Tandem' Modus eine gute Erweiterung des Einzelspieler bildet, diese Mechanik auf zwei verschiedene Charakter bezogen die Spieler aber ein wenig überfordert und dementsprechend als schlechter empfunden wird. Dabei ist der Spaß den die Probanden beim spielen haben ein großer Faktor, denn wenn Menschen überfordert sind führt dies häufig zu Frustration und der Punkt an dem die Spieler Spaß empfinden wird nicht erreicht. [Pre01]

Dieselbe Tendenz in der Bewertung ist auch in Abbildung 5.9 zu finden. Dies spricht von den weiteren Fragen losgelöst gegen eine Empfehlung dieser Variante. Am besten wurde der 'Zweispielermodus ohne zusätzliche Veränderung' wahrgenommen und der 'Tandem' Modus fast genauso gut auf Platz zwei.

Die Mehrspielervarianten 'Tandem' und 'Zweispielermodus ohne zusätzliche Veränderung' sind die beiden Varianten die alleine und im Vergleich zum Einzelspieler

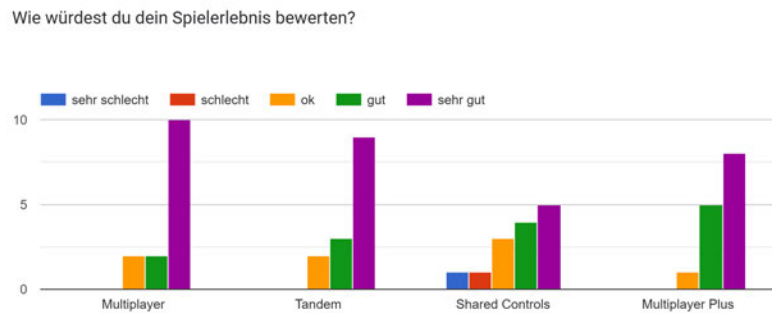


Abbildung 5.8: Bewertung des Spielerlebnis in den verschiedenen Modi

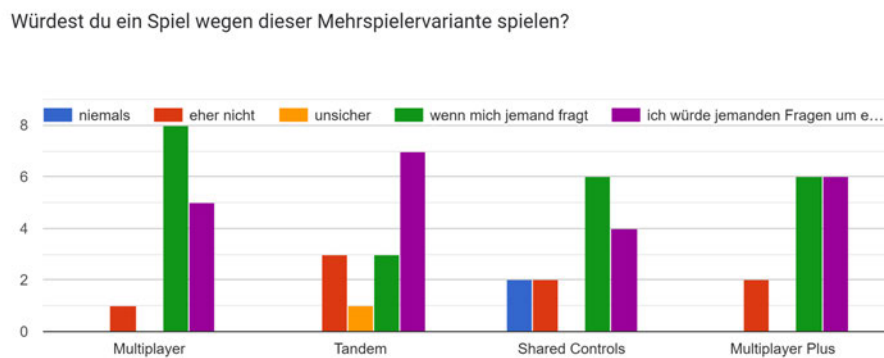


Abbildung 5.9: Interesse der Probanden ein Spiel mit der entsprechenden Mehrspieler Variante zu spielen, lila: Ich würde jemanden fragen um es mit der Person zu spielen

am besten von den Probanden bewertet wurden. Dies spiegelt sich allerdings nicht direkt im Interesse der Probanden ein Spiel mit genau dieser Mehrspielervariante zu spielen in Abbildung 5.9 wieder. Obwohl der 'Zweispielermodus ohne zusätzliche Veränderung' hier am besten abschneidet ist der 'Modifizierung des Levels' Modus fast genau so beliebt und beliebter als der 'Tandem' Modus. Die 'Verbindung beider Charaktere' ist auch hier der unbeliebteste Modus, jedoch besteht dennoch bei zehn der Probanden das Interesse den Modus zu spielen, damit wird die Annahme das der Modus als Erweiterung nicht zu empfehlen ist verworfen. Als Variante für die Implementierung eines lokalen Mehrspieler Modus in einem zweidimensionalen Plattformerspiel sind damit alle der untersuchten Varianten geeignet.

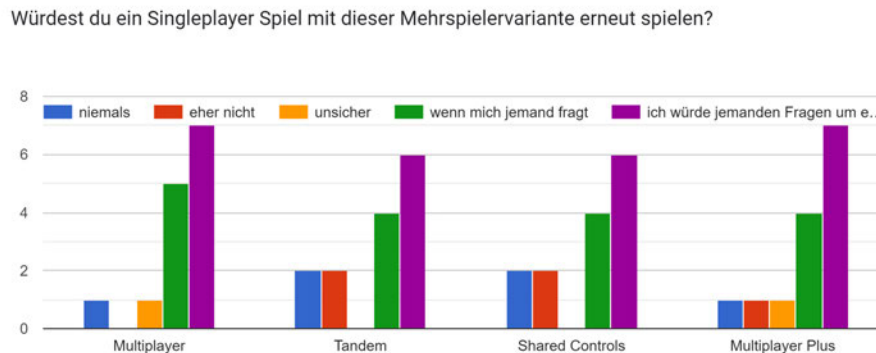


Abbildung 5.10: Bewertung des Wiederspielwert für ein Einzelspieler Spiel wenn es mit einer der Varianten erweitert wird, lila: Ich würde jemanden fragen um es mit der Person zu spielen

Spiele werden nicht immer direkt mit einem Mehrspieler Modus veröffentlicht, sondern teils erst später mit einem solchen erweitert oder dieser ist über einen zusätzlichen Kauf erwerbbar. In Abbildung 5.10 wird dargestellt wie hoch die Probanden den Wiederspielwert eines Spiels in einer der getesteten Mehrspielervarianten wäre. Dabei wurde der 'Zweispielermodus ohne zusätzliche Veränderung' erneut am besten bewertet, danach die 'Modifizierung des Levels' Variante und die verbleibenden Varianten gleichwertig auf Platz drei. Ein Großteil der Probanden würde jede der Varianten als Erweiterung eines Einzelspieler spielen und ungefähr die Hälfte würde sogar aktiv eine andere Person suchen um ein entsprechendes Spiel zu spielen.

5.3.3 Schlussfolgerung

Bei der Betrachtung der quantitativen Daten konnte festgestellt werden, dass der Modus 'Verbindung der Charaktere' die längste Spielzeit in allen Leveln vorweist. Mit einer durchschnittlichen Anzahl an Toden von zwei in Level drei befindet sich die Variante genau in der Mitte von allen fünf Modi. In der Probandenbefragung schnitt diese Variante am schlechtesten ab. Der Modus bietet also eine deutlich erhöhte Spielzeit auf Kosten des Spielerlebnis.

Mit der zweithöchsten Spielzeit und der durchschnittlichen besten Bewertung bei der Probandenbefragung sticht der 'Zweispielermodus ohne zusätzliche Veränderung'

am meisten heraus. Zusätzlich ist dieser in seiner Implementierung recht schnell erreichbar. Er bietet den Spielern die Möglichkeit die Level des Spiels gemeinsam zu erkunden, ohne dass ein Spieler Vor- oder Nachteile gegenüber dem anderen hat. Die Variante wurde von den Probanden schnell verstanden und bietet damit eine in sich nicht extra herausfordernde Mehrspielervariante was sich in der Anzahl der Tode widerspiegelt. Ähnlich dazu kann der 'Modifizierung des Levels' Modus eingeordnet werden. Dieser bietet im Vergleich eine etwas geringere Spielzeit und wurde etwas schlechter bewertet.

Die größte Herausforderung für die Probanden stellte der 'Tandem' Modus dar. Dennoch wurde der Modus sehr positiv von den Probanden bewertet und weist eine ähnliche Spielzeit wie der Einzelspieler Modus auf. Er bietet also die ideale Variante für ein herausforderndes Spiel. Er kann auch als eine herausfordernde Mehrspielervariante eines Einzelspieler Spiel verwendet werden, da die Eingaben die von einem Spieler alleine gegeben werden können lediglich auf zwei Spieler aufgeteilt werden.

Damit wurden alle Daten betrachtet und ausgewertet und die eingängliche Forschungsfrage kann anschließend beantwortet werden.

6 Zusammenfassung und Ausblick

Eingehend wurde als Ziel dieser Arbeit angestrebt eine Aussage darüber zu geben, welche Mehrspielervarianten besonders für die Implementierung eines lokalen Mehrspieler Modus in einem zweidimensionalen Plattformerspiel geeignet sind. Von den untersuchten Modi kann allgemein der 'Zweispielermodus ohne zusätzliche Veränderung' empfohlen werden. Dieser bietet einen leichten Einstieg für die Spieler und ermöglicht den Spielern das Spiel zu zweit zu erkunden und zu erleben. Er wurde von allen Probanden als äußerst positiv empfunden.

Wenn ein herausfordernder Mehrspieler Modus das Ziel sein soll, dann geht aus dieser Untersuchung der 'Tandem' Modus als Empfehlung hervor. Dieser fordert die Koordination der beiden Spieler heraus und zwingt sie miteinander zu kommunizieren. Zusätzlich ist der Modus ideal für einen lokalen Mehrspieler, da in diesem keine Eingabeverzögerung zwischen den Spielern besteht. Wenn diese Variante in einem Online Spiel verwendet werden würde könnte diese nämlich zu einem wesentlich schlechteren Spielerlebnis führen, wie in Abschnitt 2.2 beschrieben.

Damit lässt sich für die Wahl der Mehrspielerimplementation je nach angestrebtem Schwierigkeitsgrad eine Variante empfehlen. Diese Empfehlung beschränkt sich dabei auf einen lokalen Mehrspieler mit exakt zwei Spielern. Ansätze um eine größere Bandbreite an Varianten von Mehrspielerimplementationen zu untersuchen und eine allgemein gültige Empfehlung aussprechen zu können werden im Folgenden kurz erläutert.

6.1 Verbesserungsmöglichkeiten

Das Spielerlebnis kann sich mit der Addition weiterer Spieler verändern und um Spieleranzahl unabhängig eine Empfehlung auszusprechen sind weitere Untersuchungen notwendig. Dafür können die Modi 'Zweispielermodus ohne zusätzliche Veränderung'

und 'Modifizierung des Levels' problemlos mit zusätzlichen Spielern erweitert und die Tests erneut durchgeführt werden. Im 'Tandem' Modus kann entweder zusätzliche Charaktere hinzugefügt werden die von jeweils zwei weiteren Spielern gesteuert werden oder die Steuerung des einen Charakter wird noch weiter aufgeteilt. Der letzte verbleibende Modus 'Verbindung der Charaktere' lässt sich ähnlich erweitern. Dabei können für jeweils zwei weitere Spieler zwei weitere Charaktere hinzugefügt werden die in gleicher Weise in ihrer Steuerung verbunden sind wie die initialen Charaktere. Alternativ können weitere Charaktere hinzugefügt werden und die Steuerungsoptionen teilen sich auch auf diese auf, sodass immer jeder Spieler auf jeden Charakter einen Einfluss ausübt.

Des weiteren ist es möglich die Untersuchung auf bestimmte Arten von Plattformer-spielen wie sie mit den unterschiedlichen Leveln simuliert wurden zu fokussieren. Dementsprechend kann die Implementierung um weitere Mehrspielervarianten für speziell diese Anwendungsfälle erweitert und diese dann evaluiert werden. Dazu wurden die Probanden zusätzlich befragt, welche Mehrspielervarianten sie noch kennen und für gut befinden. Dabei wurde gehäuft die physische Verbindung der Spielcharaktere wie im Spiel 'Bread and Fred'¹ genannt. Speziell in diesem Spiel sind die Charaktere mit einem Seil verbunden, wie in Abbildung 6.1 zu sehen ist.

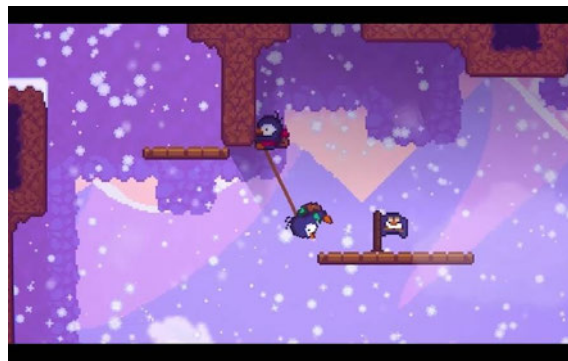


Abbildung 6.1: Bread and Fred, zwei physisch verbundene Charaktere

6.2 Fazit

Die Untersuchung bietet für Entwickler einen Einblick in die ausgewählten Mehrspielervarianten. Sie erläutert die Vor- und Nachteile der verschiedenen Varianten

¹<https://www.breadandfred.com>

und bietet für Entwickler eine Grundlage für ihre Entscheidung für eine Variante. Wenn das Spiel allerdings eine sehr einzigartige Mechanik aufweist oder sich an mehr als zwei Spieler gleichzeitig richten soll sind jedoch weitere Untersuchungen nötig.

Literaturverzeichnis

- [Bai09] William Sims Bainbridge: *Online Multiplayer Games*, Springer Cham, 1. Aufl., 12 2009, ISBN 9783031011405.
- [Bar96] Richard Bartle: *Hearts, clubs, diamonds, spades: Players who suit MUDs*, *Journal of MUD research*, Bd. 1(1):S. 19, 1996.
- [Ber22] Simon Bernard: *Local Multiplayer Games*, Diplomarbeit, Alpen-Adria-Universität Klagenfurt, 9020 Klagenfurt am Wörthersee, Österreich, 4 2022.
- [CWK11] Kimberly Chu, Chui Yin Wong und Chee Weng Khong: *Methodologies for Evaluating Player Experience in Game Play*, in *HCI International 2011 – Posters’ Extended Abstracts* (herausgegeben von Constantine Stephanidis), S. 118–122, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, ISBN 978-3-642-22098-2.
- [DS23] Anna Derezińska und Dawid Sygocki: *Review of Automated Code Refactoring of C# Programs*, in *18th Conference on Computer Science and Intelligence Systems*, Institute of Electrical and Electronics Engineers, 2023.
- [Haa13] John Haas: *A History of the Unity Game Engine*, Diplomarbeit, 100 Institute Rd, Worcester, MA 01609, Vereinigte Staaten, The address of the publisher, 12 2013.
- [Hal18] Jared Halpern: *Developing 2D Games with Unity*, Apress Berkeley, CA, 1. Aufl., 11 2018, ISBN 9781484237717.
- [JF13] Jessica M. Jerabeck und Christopher J. Ferguson: *The influence of solitary and cooperative violent video game play on aggressive and prosocial behavior*, *Computers in Human Behavior*, Bd. 29(6):S. 2573–2578, 2013, ISSN 0747-5632.

- URL <https://www.sciencedirect.com/science/article/pii/S0747563213002264>
- [Ken01] Steven L. Kent: *The Ultimate History of Video Games*, Three Rivers Press, 10 2001, ISBN 0761536434.
- [Med96] Imagine Media: *A definitive guide to gaming terminology, NEXT Generation Issue #15*, Bd. 2(15):S. 41, 3 1996, abgerufen über URL: <https://archive.org/details/nextgen-issue-015/page/n39/mode/2up>, am 30.11.2023.
- [OF06] Kas Oosterhuis und Lukas Feireiss: *The architecture co-laboratory: GameSetandMatch II : on computer games, advanced geometries, and digital technologies*, 09 2006, ISBN 9789059730366.
- [Pho23] Photon: *Showcased Mechanics*, 11 2023, URL: <https://doc.photonengine.com/quantum/current/technical-samples/2d-platformer/showcased-mechanics#main>, besucht am 25.11.2023.
- [Pre01] Marc Prensky: *Fun, play and games: What makes games engaging, Digital game-based learning*, Bd. 5(1):S. 5–31, 2001.
- [PSEN23] Johannes Pfau und Magy Seif El-Nasr: *Balancing Video Games: A Player-Driven Instrument*, in *Companion Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY Companion '23, S. 187–195, Association for Computing Machinery, New York, NY, USA, 2023, ISBN 9798400700293.
URL <https://doi.org/10.1145/3573382.3616097>
- [RMP08] José Bernardo Rocha, Samuel Mascarenhas und Rui Prada: *Game mechanics for cooperative games, ZON Digital Games 2008*, S. 72–80, 2008.
- [Sch12] Daniel Schwarz: *Virtuelle Gewalt-Computerspiele und deren Wirkungsdebatte im pädagogischen Kontext*, Dissertation, 2012.
- [Spe23] Matthew Spencer: *Ultimate 2D Controller*, 10 2023, URL: <https://github.com/Matthew-J-Spencer/Ultimate-2D-Controller>.
- [Sto23] Steam Store: *Steam-Marktsansicht, Kategorie 2D Plattformer*, 11 2023, URL: <https://store.steampowered.com/tags/en/2D%20Plattformer>, besucht am 29.11.2023.

- [SW15] Valerie J. Shute und Lubin Wang: *Measuring Problem Solving Skills in Portal 2*, S. 11–24, Springer International Publishing, Cham, 2015, ISBN 978-3-319-05825-2.
URL https://doi.org/10.1007/978-3-319-05825-2_2
- [Tsi19] Mariot Tsitoara: *Beginning Git and GitHub*, Apress Berkeley, CA, 1 Aufl., 12 2019, ISBN 9781484253120.
- [Uni23] Unity: *Unity Documentation*, 11 2023, URL: <https://docs.unity3d.com/Manual/index.html>.

Selbstständigkeitserklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbstständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche wissentlich verwendete Textausschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

Mittweida, den 18. Dezember 2023

A solid black rectangular box used to redact the signature of the author.

Carl Hunger