



**HOCHSCHULE
MITTWEIDA**

University of Applied Sciences

Fakultät Angewandte Computer- und Biowissenschaften

Professur Medieninformatik

Bachelorarbeit

Praktische Validierung von Bartles Taxonomie der
Spielerklassifikation in einem vom ursprünglichen theoretischen
Rahmen divergierenden Spielumfeld

Jamison Bassett

Mittweida, den 17. Januar 2024

Erstprüfer: Prof. Dr. rer. nat. Marc Ritter

Zweitprüfer: M.Sc. Manuel Heinzig

Bassett, Jamison

Praktische Validierung von Bartles Taxonomie der Spielerklassifikation in einem vom ursprünglichen theoretischen Rahmen divergierenden Spielumfeld

Bachelorarbeit, Fakultät Angewandte Computer- und Biowissenschaften

Hochschule Mittweida — University of Applied Sciences, Januar 2024

Referat

In dieser Arbeit geht es um die Validierung von Richard Bartles Modell der Spielerarchetypen in einem Einzelspieler-Videospiel. Das Modell wurde für die Entwicklung des Spiels *Explorer Bot* angewendet. Die Validität des Modells wurde anschließend über zwei Faktoren bestimmt. Zum einen wurden die Meinungen über *Explorer Bot* der Spieler mittels einer Umfrage aufgezeichnet. Der andere Faktor besteht aus der Existenz der Archetypen unter den Spielern *Explorer Bots*.

Name: Bassett, Jamison

Studiengang: Medieninformatik und interaktives Entertainment

Seminargruppe: MIw2-B

English Title: Practical Validation of Bartles Taxonomy of Player Archetypes within an Environment diverging from the Original Theoretical Framework

Inhaltsverzeichnis

1	Einführung und Motivation	1
2	Grundlagen	3
2.1	Hintergrund	3
2.2	Bartles vier Archetypenmodell	5
2.3	Kritik an Bartles Modell	10
3	Konzeption	19
3.1	Explorer Bot's Gamedesign im Kontext zu Bartle	20
3.2	Datenansammlung	28
4	Implementierung	31
4.1	Unity Analytics Service	31
4.2	Erstellung der Spielwelt	33
4.3	Dialogimplementation	36
4.4	Quests	37
5	Evaluation	43
5.1	Auswertung der Umfrage	43
5.2	Bestimmung der Archetypen	44
6	Zusammenfassung und Ausblick	51
	Literaturverzeichnis	I
A	Konzeptionsdokumente	A1

B Evaluationsdokumente

V

1. Einführung und Motivation

Immer mehr Personen wollen Videospiele entwickeln. Mit den kontinuierlichen Fortschritten der Technik fällt der Anfang stetig leichter. Eine Hürde bleibt jedoch bestehen. Wie wird garantiert, dass das entwickelte Spiel die Spieler auch anspricht? Die meisten Personen spielen Videospiele aus dem einfachen Grund "Spaß". Dabei unterscheiden sich die Quellen dieses Spaßes enorm. Nicht nur in verschiedenen Spielen: Das Spielvergnügen in einem rasanten Shooter wie Call Of Duty¹ ist sicherlich nicht der gleiche Spaß wie in einem entspannten LifeSim wie Stardew Valley², sondern auch innerhalb eines Spiels. Ein Spieler von League of Legends³ kann definitiv Spaß haben, indem er alleine die Rängeleiter erklimmt, aber genauso viel Spaß kann der nächste Spieler haben, welcher League of Legends in einer Gruppe mit Freunden spielt.

Diese unterschiedlichen Arten, ein Videospiele zu genießen wurden schon früh erkannt. Der erste Versuch, Spieler in Gruppen aufzuteilen, wurde durchgeführt von Dr. Richard Bartle in 1996[Bar96]. Die Nutzen, Spieler in grobe Archetypen zu gruppieren, sind vielfältig. Vom Feinjustieren, welche Art von Spieler ein Entwickler in seinem Spiel ansprechen will, bis hin zu gezielten Monetarisierungsplänen.

Bartle erschuf seine Taxonomie der *Player Suits* mit dem Ziel, Spieldesigner zu unterstützen. Er stellt fest, dass Spieler aus einer von vier Quellen ihr Spielvergnügen schöpfen. Abgeleitet wurden davon die vier Archetypen, welche von Spielern verkörpert werden. Inhalt dieser Arbeit ist es, sein Modell zu beleuchten und zu prüfen. Als Leitfaden für die Spielentwicklung kann das Modell ein nützliches Werkzeug sein. Die Applikation auf moderne Zielgruppen muss daher stetig untersucht werden. Auch die Reichweite des Anwendungsfeldes des Modells ist es wert, betrachtet zu werden. Ein empirisch geprüfter, gelungener Anwendungsfall öffnet eine Vielzahl an Möglichkeiten für neue Entwickler.

¹ *Call Of Duty* veröffentlicht durch Activision <https://www.callofduty.com/>

² *Stardew Valley* entwickelt von Concerned Ape <https://www.stardewvalley.net/>

³ *League of Legends* entwickelt von Riot Games <https://www.leagueoflegends.com/>

Die Untersuchung des Modells wird anhand des Videospiele *Explorer Bot* durchgeführt. *Explorer Bot* wurde in der Unity Engine speziell für diese Arbeit entwickelt. Die Entwicklung von *Explorer Bot* wurde von den Aussagen des Modells gelenkt. Speziell im Bereich des Game-Designs kommt ein gezieltes Arbeiten in Richtung der Archetypen zum Vorschein. Neue Erkenntnisse über die Anwendung von Bartles Modell werden durch das divergierende Genre des Spiels gewonnen. Das Modell wurde für ein Multispieler-Spiel konzipiert. Daher nimmt es die Interaktionen zwischen Spielern in Acht. Bei *Explorer Bot* handelt es sich um ein Einzelspieler-Spiel. Die Anwendung des Modells bezieht sich daher vermehrt auf die Konzeption des Spiels, als auf die Dynamiken unter den Spielern. Die Resultate des Experiments begründen die Verwendung des Modells für Entwickler von einer diversen Menge an Spielen.

Das Urteil, ob Bartles Modell weiterhin effektiv angewendet werden kann, wird über zwei Faktoren gefällt. Zum einen werden während des Spielens die Aktionen des Spielers aufgezeichnet. Daraus werden Informationen extrahiert, die zur Bestimmung von Bartles Archetypen unter den Spielern führen. Durch Präsenz der Archetypen innerhalb der Probanden kann die praktische Validität des Modells gesichert werden. Die theoretische Seite des Modells wird über einen unabhängigen Prozess geprüft. Über eine anonymisierte Umfrage geben Probanden ihr empfundenes Spielvergnügen an. Die Fragen beziehen sich auf archetyp-spezifische Elemente in *Explorer Bot*. Die positive Bewertung der Spieler formt die Validierung der theoretischen Seite des Modells. Selbst wenn die einzelnen Archetypen nicht präsent sind, zählt die positive Rezeption von Elementen, die auf sie ausgerichtet sind. Damit kann Bartles Modell weiterhin als Grundlage für Game-Design verwendet werden.

2. Grundlagen

Kern dieses Kapitels ist das von Dr. Bartle erstellte *Player Suits* Modell. Das Modell ist eine Taxonomie, welche die Spieler eines Videospiele in vier Gruppen teilt. Die Gruppen bilden somit Archetypen von Spielern. Die Archetypen unterscheiden sich hinsichtlich der Spielmotivation der Spieler. Für Spielentwickler ist solch eine Taxonomie ein wichtiges Werkzeug. Designer können sich auf die Gruppierungen berufen und somit den Prozess ihrer Arbeit optimieren.

Das Kapitel beginnt mit einer Einleitung zu Bartle. Anschließend wird der originale Kontext des Modells beleuchtet. Dabei wird speziell auf das Spiel *MUD* (Multi-User-Dungeon) eingegangen. Bartle war einer der Hauptentwickler dieses Spiels und erschuf sein Modell der Player Suits auf dessen Basis.

Nachdem der zeitliche Kontext des Modells erläutert wurde, wird das Modell selbst analysiert. Dabei werden zunächst die vier Archetypen und ihre Bedeutung aufgeschlüsselt. Darauf folgen zwei Beispiele, wie das Modell in anderen Spielen mit Erfolg angewandt wurde. Die Analyse des Modells wird durch eine Anschauung seiner Schwachstellen abgeschlossen.

2.1. Hintergrund

2.1.1. Richard Bartle

Richard Allan Bartle, geboren am 10. Januar 1960, studierte Computerwissenschaften an der University of Essex in England. Obwohl er seinen Abschluss 1988 im Feld der Künstlichen Intelligenz absolvierte, liegt seine Leidenschaft im Bereich der Computerspiele[Bar99]. So entwickelte er bereits in seiner Jugend Videospiele. Während seines Studiums traf er auf gleichgesinnte Kommilitonen, allen voran Roy Trubshaw.

Als Student entdeckte Trubshaw einen Mechanismus, welcher ermöglicht, ein Programm von mehreren Maschinen zur gleichen Zeit abzurufen. Dies erschuf die Grundlage für ein Computerspiel, welches mehrere Spieler zur gleichen Zeit spielen konnten. In 1978 begannen Bartle und Trubshaw mit der Entwicklung von *MUD*, dem Multi-User-Dungeon. Trubshaw übernahm dabei die Programmierung der Physik und Bartle die Codierung der Spiellogik.[Stu14]

Als Trubshaw am Ende seines Studiums die University of Essex verließ, übergab er die alleinige Weiterentwicklung an Richard Bartle. Obwohl es nicht das erste Spiel war, welches seinen Spielern eine virtuelle Welt präsentierte, war es das erste mit größerem Erfolg. So großem Erfolg, dass es die Entwicklung weiterer Spiele inspirierte und "MUD" zur Bezeichnung eines Genres wurde.[Bar03, S. 6-7]

2.1.2. Kontextualisierung von MUDs

MUDs können mit ihren indirekten Nachfolgern verglichen werden. Die heutigen Massive Multiplayer Online Role-Playing Games (MMORPGs), wie *World of Warcraft*¹ oder *Final Fantasy XIV*² haben ihre Wurzeln in MUDs. Oberflächlich gibt es offensichtliche Unterschiede. Die Beschreibung "Massive-Multiplayer" trifft nicht auf MUDs zu. *MUD1* war für maximal 36 Spieler zur gleichen Zeit ausgelegt[Bar03, S. 9]. Der weitaus sichtbarere Unterschied liegt in der Präsentation. Während moderne MMORPGs ein videobasiertes Interface benutzen, um ihren Spielern eine Welt zu präsentieren, waren MUDs rein auf Text angewiesen[Mor06, S. 4]. Die Limitation auf Text war durchaus auch ein Vorteil. Spieler konnten das Aussehen ihres Avatars komplett eigens bestimmen, ohne von einer Auswahl von Sprites oder 3D-Assets begrenzt zu sein[Mor06, S.9]. Manche MUDs haben Spielern sogar angeboten, eigene Orte zu kreieren. Diese konnten dann mit anderen geteilt werden.

Nichtsdestotrotz füllen MUDs dieselbe Nische wie heutige MMORPGs[Mor06, S.1]. Spiele beider Genres bieten nicht nur reine "Game"-Elemente wie das Besiegen von Gegnern und Erfüllen von Quests an, sondern genauso soziale Räume[Mor06, S.2], um sich mit anderen Spielern kurzzuschließen und zu verbinden.

MUDs nahmen die technologischen Limitationen ihrer Zeit und wandelten diese in Stärken um. Auf *MUD1* folgten weitere Spiele, die das Genre ausbauten. So gab es

¹World of Warcraft, 2004, entwickelt von Blizzard Entertainment <https://worldofwarcraft.blizzard.com>

²Final Fantasy XIV, 2010, entwickelt von Square Enix <https://de.finalfantasyxiv.com/>

MUDs wie *AberMUD*, welche ihren Fokus auf das Kämpfen mit anderen Spielern und Monstern in der virtuellen Welt legten. Jedoch auch viele Pendants, wie *TinyMUD*, wo die Möglichkeit zu kämpfen gar nicht erst programmiert wurde.[Bar03, S. 13] Natürlich gab es auch MUDs zwischen diesen beiden Extremen. Der Aspekt, mit seiner Figur eine feste Rolle zu verkörpern und mit anderen Spielern zusammen in der Welt eine gemeinsame Geschichte zu erzählen, war ebenfalls in vielen MUDs vorhanden.

Dies erschuf eine Vielfalt an Optionen des Zeitvertreivs für Spieler. Doch wieso weisen manche Spieler eine Tendenz zu dem einen MUD als dem anderen hervor? Wie können Entwickler ihre virtuelle Welt designen, sodass andere Personen sie spielen wollen? Oder anders: Was wollen Leute von einem MUD? Dieser Frage ging Bartle nach und erstellte seine erste Taxonomie von Spielern im Jahr 1996.[Bar96]

2.2. Bartles vier Archetypenmodell

Bartle erschuf sein Modell aus einer Analyse von einer monatelangen Diskussion in einem Forum. Das Thema der Diskussion war die Frage: "Was wollen Leute von einem MUD?". Die Diskussion wurde geführt von Spielern des MUDs *MUD2*. Der Nachfolger des originalen *MUDs* wurde ebenfalls von Bartle entwickelt. Von seinem Vorgänger unterscheidet es sich hauptsächlich auf programmatischer Ebene.

Die Diskussionsteilnehmer besaßen den höchsten Rang innerhalb des Spiels. Folglich waren sie die Spieler mit der meisten Erfahrung. Die Anzahl dieser Spieler betrug zu der Zeit circa 30 Personen. Die Hälfte davon trug einen Großteil zur Diskussion bei, während die andere Hälfte ihre Meinungen nur ab und zu mit eingebracht hat. Nach einer Periode von sieben Monaten (November 1989 bis Mai 1990) schloss Bartle die Diskussion, da keine neuen Punkte mehr beigetragen wurden. In seiner Zusammenfassung der Diskussion abstrahierte Bartle die verschiedenen Meinungen und erkannte ein Muster.[Bar96, S. 2]

Die kurze Antwort auf die Frage "Was wollen Leute von einem MUD?" ist "Spaß". In der Tat empfanden viele der Diskussionsteilnehmer die gleichen Elemente als spaßig, es bildeten sich jedoch klare Untergruppen. Die aus den Untergruppen folgende Archetyp-Taxonomie beruht auf der Quelle des Späßes, der ausgeübten Tätigkeiten eines Spielers. Nicht allein, dass er sie am meisten ausführt[Bar09, S. 10]. Bartle erkannte folgende vier Gruppen:

◇ **Achiever**, der Erfolgsmensch

- Spaß kommt daraus, die höchste Punktzahl/Level zu erreichen
- engstirnig, zielorientiert
- *agieren* mit Spielwelt als Werkzeug ihres Fortschritts
- andere Spieler sind Mittel zum Zweck

♠ **Explorer**, der Entdecker

- Spaß kommt vom Wissen, wie die Spielwelt funktioniert/strukturiert ist
- unkonventionell, esoterisch
- *interagieren* mit der Spielwelt, um zu experimentieren
- andere Spieler kommen eher in die Quere

♡ **Socializer**, der Gesellige

- Spaß kommt vom Austausch mit Anderen
- offen, freundlich
- Spielwelt ist lediglich der Hintergrund auf dem Freunde getroffen werden
- höchstschätzung der *Interaktion* mit anderen Spielern.

♣ **Killer**, der Drahtzieher

- Spaß kommt vom Effekt auf andere Personen
- skrupellos, auferlegend
- Spielwelt ist Mittel zum Zweck
- *agieren* mit anderen Spielern, meist destruktiv

Der Name "Player Suits" Model kommt von der Verbindung zwischen den Archetypen und den Farben eines standard französischen Blatts Karten. Bartle zählte diese Verbindung, um die Typen bildlicher zu machen, für das leichtere Erinnern.[Bar96, S. 3]

Diese Archetypen existieren nicht komplett ohne Überschneidungen. Bartle abstrahierte die Interessen der Gruppen und unterteilte sie wieder (Siehe Abb. 2.1). Die Achse "Spieler—Welt" zeigt, ob das Interesse eines Spielers auf anderen Spielern oder auf der virtuellen Welt liegt. Die "Aktion—Interaktion"-Achse beschreibt, wie ein Spieler mit dem Subjekt der anderen Achse handelt.

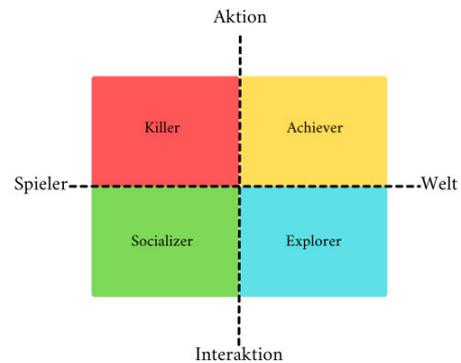


Abbildung 2.1.: Interessendiagramm der Spielertypen

So überlappen Killer mit Socializern, aber nur in Bezug darauf, dass beide Archetypen auf andere Spieler fokussiert sind. Der Unterschied liegt in der Aktion/Interaktion Dichotomie. Socializer möchten mit anderen interagieren; reden, Items austauschen, sie kennenlernen. Killer möchten auf andere Spieler einwirken. Ihre Freude kommt daraus, ihren Effekt auf andere zu wissen und oftmals die eigene Übermacht zu demonstrieren.

Dieses Bedürfnis, die eigene Wirkung zu erfahren, teilen sie mit Achievern. Spieler vom Typ Achiever wollen das Spiel meistern und sehen, wie gut sie ihre Fähigkeiten entwickelt haben. Andere Spieler sind nebenwertig. Das Gleiche gilt für Explorer. Ihr Fokus liegt ebenfalls auf der Spielwelt, aber mit einem Winkel auf Interaktion. Sie wollen von der Spielwelt überrascht werden, die simulierte Physik an ihre Grenzen treiben und ihr tiefes Wissen über die Spielwelt nutzen, um eben dieses noch weiter zu vertiefen. [Bar96, S.6]

Nach Bartles persönlichen Erfahrungen ist jeder Spieler, zu einem bestimmten Zeitpunkt, immer nur einem primären Archetypen einzuteilen. Den Überschneidungen der Archetypen und anderen Faktoren, wie zum Beispiel die Gemütslage des Spielers, sei es verschuldet, dass Spieler ihren Archetyp zu wechseln scheinen. Diese Transformation ist jedoch nur temporär und dient immer zur Bereicherung des Hauptarchetypen. [Bar96, S.3]

2.2.1. Bartles Modell im Bezug auf Gamedesign

Eine wichtige Kenntnis, die aus dem Modell gewonnen wurde, ist, wie sich die verschiedenen Archetypen untereinander beeinflussen. Speziell im Bezug auf jeweilige Spielerzahlen. Den größten Nutzen findet das Modell in der Umgebung, in und für

welche es erstellt wurde. Virtuelle Welten für mehrere Spieler. Die Spieler als Ganzes bilden die Population einer solchen Welt. Die Gewährleistung, dass die Welt weiter belebt, das Spiel weiter gespielt wird, kommt von einem Equilibrium der Archetypen. Die Playerbase braucht Spieler von jedem Archetypen. Wenn ein Archetyp zu viel oder wenig vertreten ist, leidet das Spiel darunter[Bar96, S. 7].

Jeder Spieler eines Archetyps hat gewissen Einfluss auf die Spieler der anderen Archetypen. Wenn ein Spiel zu viele Killer in der Bevölkerung hat, wird ein Absenken vor allem im Achiever- und Socializer-Sektor gesehen. Folglich, wenn keine Socializer oder Achiever vorhanden sind, haben Killer niemanden, auf den sie effektiv einwirken können. Was schlussendlich zu einem MUD führt, welches nur von Explorern populiert wird. Sobald das MUD keine neuen Kenntnisse für diese Spieler hervorbringt, spielt es niemand mehr. Diese und weitere Dynamiken hat Bartle in seinem ersten Model ebenfalls aufgeführt. [Bar96, S.20]

Somit konnte das Modell von anderen Entwicklern virtueller Mehrspieler-Welten unterstützend genutzt werden, um ihr Spiel besser in die gewünschte Balance zu bringen oder sich von der Konkurrenz abzuheben.

Nicht viele Gamedesigner geben ihren Prozess bekannt. Um das damals neue MMORPG zu bewerben, tat Paul Barnett, Creative Director von *Warhammer: Age of Reckoning*³ (WAR), jedoch genau das. In einem Interview von 2008 mit der BBC gab er bekannt, dass das Design des frisch herausgebrachten Spiels, sich auf das Modell Bartles gestützt hat[Cor08]. Im Vergleich zu dem heute noch sehr erfolgreichen Spiel *World of Warcraft*(WOW) soll das von Mythic Entertainment entwickelte MMORPG den Fokus auf die Spieler legen, welche nicht vom Archetyp Killer sind. Barnett kritisierte *WOW* für die engstirnige Befriedigung eines einzigen Spielertyps. Er behauptete, dass sich das Mehrspieler-Spiel eher wie ein Einzelspieler-Spiel anfühlte, das lediglich von vielen Spielern zur gleichen Zeit gespielt wurde. Dies führte zu einem Gamedesign für *WAR*, das Kooperation zwischen Spielern derselben in-game Faktion betont. Da es sehr einfach gemacht wurde, Spielergruppen bei einer Quest beizutreten, gab es viel mehr Möglichkeiten, für Socializer Spaß zu finden. Achiever wurden durch eine flache Levelkurve bedient. Spieler sollten nicht unnötig lange ihren Charakter hochleveln, um ihren Effekt in der Welt zu spüren.

Das Interview geht nicht auf den Archetypen des Explorers ein. Eine Vermutung, wie Explorer befriedigt werden, kann dennoch aufgestellt werden. Explorer bevorzugen

³Warhammer: Age of Reckoning, entwickelt durch Mythic Entertainment, veröffentlicht 2008, eingestellt 2013

eine Welt, die sich lebendig anfühlt[Bar96, S. 6]. Eine lebende Welt zu erkunden und neue Facetten zu entdecken, ist ein starker Antrieb für Explorer. Das Franchise von *Warhammer* existiert seit 1983⁴. Auf die lange Geschichte des *Warhammer*-Universums können die Entwickler aufbauen. Somit werden nicht nur Explorer, die das Franchise zum ersten Mal entdecken, angesprochen. Auch langjährige Fans können neue Informationen finden und demnach ihr Wissensnetz ausbauen.

Laut Bartles Modell sind die meisten Spieler vom Typ Socializer[Bar03, S. 182]. Die Entwickler des Spiels *GoPets*⁵ haben sich dies zu Nutze gemacht. Folglich entwickelten sie in 2005 ein Spiel, welches gezielt auf diese Socializer ausgelegt war.[Bar12b, 5:50-6:25]

GoPets war eine virtuelle Welt, in der Spieler auf virtuelle Haustiere Acht geben konnten. Das gesamte Gameplay drehte sich um diese Haustiere. Spieler konnten ihre Katze, ihren Hund, Panda etc. beliebig anpassen und anderen Spielern oder Freunden zeigen. Finanziert wurde *GoPets* durch das Erwerben von Abonements oder einer in-game Währung mit Echt-Geld[Mik08]. Diese Premium-Währung konnte dazu benutzt werden, um Items zu kaufen. Anfänglich bezogen sich viele dieser Items weiterhin auf das individuelle Anpassen der Haustiere, beispielsweise durch Kleidungsstücke. Als die Entwickler nachgeforscht haben, welche Items sich am besten verkauften, fanden sie heraus, dass das Item, welches den meisten Gewinn brachte, ein Fruchtbaum war. Dieser Fruchtbaum unterschied sich von anderen Items in dem Aspekt, dass er den Spielern ein Ziel und eine Belohnung darbot. Wenn ein Spieler sein Haustier eine Stunde lang unter dem Baum hat sitzen lassen, dann fiel eine Nuss vom Baum, die der Spieler aufsammeln konnte. Dies war der einzige Weg, innerhalb des Spiels, an so eine Frucht zu kommen. Sollte sich das Tier vom Baum entfernen, würde sich der Countdown zurücksetzen. Durch dieses herausfordernde Element wurde die Nuss noch begehrter und weckte somit genug Verlangen in Spielern, um echtes Geld zu investieren.

Unter erneuter Betrachtung durch das Player Suits Modell kamen *GoPets*' Entwickler zu dem Entschluss, dass dieses Item sehr attraktiv für Spieler des Achiever-Typs war. Obwohl das Spiel auf Socializer ausgelegt war, gab es einige Achiever und auch Explorer in der Playerbase, welche viel bereit waren, ihr Geld für das Spiel auszugeben[Bar12b, 7:05-7:40]. Mit dieser Erkenntnis fügten die Entwickler *GoPets* einige weitere simple Aktivitäten hinzu, die in das selbe Schema des Fruchtbaums

⁴Veröffentlichung der ersten Edition von *Warhammer* veröffentlicht von Games Workshop Ltd.

⁵*GoPets* entwickelt durch *GoPets Ltd.*, veröffentlicht in 2005, eingestellt in 2009

fielen. Eine Woche später haben sich die Einnahmen von *GoPets* verdoppelt. [Bar09, S. 8]

Ungeachtet der eigentlichen Zielgruppe wurden dem Spiel Inhalte zugeführt, die andere Spielertypen ansprechen. Infolgedessen erfuhren die Entwickler großen Erfolg. Bartles These, dass seine vier Archetypen in Spielern jedes Spiels vorkommen [Bar96, S. 11] belegt sich. Demnach sollten Entwickler ihre Spiele divers gestalten.

Aus diesen zwei Beispielen ist zu erkennen, dass Bartles Player Suits Modell durchaus seine Verwendung und seinen Nutzen hat. Die theoretische Grundlage, die das Modell bereitstellt, bezieht sich auf die Frage "Warum haben Leute Spaß, wenn sie MMOs spielen?". Die Antwort sind die vier Archetypen, welche alle einen anderen Blickwinkel auf das Spielvergnügen bieten. Da das Konzept von Spaß, selbst auf vier Unterteilungen runtergebrochen, sehr facettenreich ist, stehen manche der Archetypen in Konflikt zueinander. Somit kann das Modell ebenso Aussagen über die Balance zwischen den Archetypen treffen.

Gamedesigner können das Modell verwenden, um Vorhersagen zu treffen, welche Arten von Spielern positive Reaktionen auf verschiedene Spielelemente haben werden und welche Auswirkung eine Steigerung der Spieleranzahl eines Archetypen auf die Spielerpopulation des Spiels als Ganzes hat. Das erleichtert es Gamedesignern, das Spiel in eine gewünschte Richtung zu lenken und gezielte Erfahrungen zu erwecken.

2.3. Kritik an Bartles Modell

Eine große Schwachstelle des Player Suit Modells sind die Daten, auf denen es beruht. Anstelle von empirischen Daten wurden die Kerninformationen aus Diskussionen erhoben. Das Subjekt der Diskussion war ein spezifisches MUD, *MUD2*, in einer Zeit, in der schon eine breite Masse an anderen MUDs existierte. Somit kommt die Frage auf, ob die Erkenntnisse der Diskussion nicht zu spezifisch auf die Konditionen von *MUD2* beschränkt seien.

Eine weitere Folge einer solchen Herangehensweise zur Datenerhebung ist, dass die Unterteilung des Modells unsauber erfolgt. Komponenten eines Archetypen können überflüssig sein, oder quintessentielle Eigenschaften wurden bei der Erstellung übersehen. Zwei Archetypen können auch kritische Überschneidungen aufweisen, welche die eigentliche Einteilung von Spielern erschwert. Als Konsequenz verliert Bartles Modell an Effektivität in der Praxis.

Letztlich hat Bartle selbst über die Jahre seine Theorie weiterentwickelt und verfeinert. Dabei ist er auf die Defizite des originalen Modells eingegangen und hat dieses erweitert. Das Ergebnis ist ein Modell, welches acht Spielertypen beschreibt.

In diesem Unterkapitel werden diese drei Kritikpunkte näher beleuchtet. Zunächst wird die Problematik der verschiedenen Ausrichtungen von MUDs untersucht. Nachfolgend werden die Verbesserungen, welche Bartle an seinem Modell durchgeführt hat, erläutert. Schließlich wird auf den manglenden wissenschaftlichen Prozess eingegangen. Im letzten Punkt wird dabei besonders die Arbeit von Nicholas Yee hervorgehoben.

2.3.1. Verschiedene MUDs, verschiedene Spieler?

Wie in Kapitel 2.2 erwähnt waren die Teilnehmer der Diskussion, auf die sich das Player Suits Modell bezieht, alle Spieler des Spiels *MUD2*. Dies ist wichtig zu betonen, da *MUD2* das Töten anderer Spielercharaktere erlaubte [Bar96, S. 2]. Zu dieser Zeit, 1989, über 10 Jahre nachdem *MUD1* programmiert wurde, existierten bereits einige Imitatoren. Unter anderen sticht *TinyMUD* von Jim Aspnes heraus. *TinyMUD* implementierte keinerlei Art zu kämpfen, ob gegen andere Spieler oder Nicht-Spieler-Entitäten. Stattdessen wurde der Fokus auf das spielerangetriebene Erstellen von Orten und Objekten innerhalb des Spiels gelegt. Die Abwesenheit von spielähnlichen Elementen bewirkte, dass die Hauptaktivität von Spielern sich auf das Kreieren und den Austausch über ihre Kreationen beschränkte [Bar03, S. 23].

Im Vergleich zu anderen MUDs war *TinyMUD* weitaus mehr auf die soziale Schiene ausgelegt und, laut Bartle, führte zu einer Unterteilung von MUDs in "sozialen" und "spielähnlichen". Im Bezug auf das Player Suit Modell kann durch das Interessendiagramm (siehe Abb. 2.1) bestimmt werden, wie ein MUD angelegt sein muss, um in eine der beiden Kategorien zu fallen. Soziale MUDs waren auf die untere Hälfte fokussiert. Die Interaktion mit der Welt und anderen Spielern stand im Vordergrund. Sobald das Spiel Elemente beinhaltete, welche die obere Hälfte des Interessendiagramms, also Wettstreit, unterstützen, würde es den Titel eines "spiel-ähnlichen" MUDs bekommen. Diese Unterteilung war so prävalent, dass Spieler von sozialen MUDs behaupteten, sie würden gar nicht MUDs spielen, da der Term sich nur auf kampforientierte Spielumgebungen beziehe, mit welchen sie nicht in Verbindung gebracht werden wollten. [Bar96, S. 11]

Die Existenz von MUDs, die, allem Anschein nach, keinerlei Möglichkeit bieten, einen Spieler vom Typ Killer oder Achiever zu sein, würde Bartles Theorie annullieren. Anstatt diese Sonderfälle ebenfalls einfach nicht als MUD zu bezeichnen, argumentiert Bartle jedoch für die Inklusion solcher Spiele unter dem MUD-Genre. Zusätzlich beschreibt er, wie sowohl Killer als auch Achiever in sozialen MUDs vorhanden sind[Bar96, S. 11].

Killer ist lediglich der Überbegriff für Spieler, welche Vergnügen aus ihrem Effekt auf andere erzielen. Wenn ein Kampf nicht möglich ist, gibt es dennoch eine Vielzahl anderer Arten, seine Mitspieler zu peinigen. Von Belästigung einzelner Personen bis hin zu dem gezielten Auseinanderreißen einer Community. Achiever gibt es ebenfalls in einer leicht veränderten Form. Spieler, welche das Kreieren selbst als einen Wettstreit sehen und sich daran messen, wer den "besten" Raum hat oder die meisten Kreationen, sind klare Achiever.

Nur weil das Spiel die Aktionen, die mit den beiden Archetypen verbunden sind, nicht belohnt, bedeutet das nicht, dass diese Archetypen nicht in der Playerbase vorhanden sind. Der Umkehrschluss bewahrheitet sich ebenfalls mit der Präsenz von Socializern und Explorern in für den Kampf ausgelegte MUDs.

Nichtsdestotrotz ist die Betrachtung, wie sich Bartles Playertypen in verschiedenen Auslegungen des MUD-Genres manifestieren, wertvoll. Unter der Analyse von spielähnlichen MUDs können zwei weitere Sub-Genres gefunden werden.

- PvP(Player versus Player) MUDs
 - MUDs in denen es explizit erlaubt ist, andere Spielercharactere zu töten, anstatt nur computergenerierte Gegner
 - Ziel des Spiels wird es, gegen andere Spieler zu kämpfen
- Roleplaying MUDs
 - MUDs bei welchen der "Rollenspiel"-Aspekt der virtuellen Welt im Vordergrund steht
 - Ziel des Spiels ist, eine feste Rolle durch seinen Charakter zu belegen. Der Avatar hat dabei seine eigene Hintergrundgeschichte, die ein Teil der Geschichte der Spielwelt ist

Viele MUDs enthalten Elemente, die beide Arten zu spielen unterstützen. In Isolation betrachtet haben die verschiedenen Auslegungen Einwirkung auf die Interpretation

der Spielertypen. Ein Achiever in einem auf Rollenspiel basierten MUD könnte als Ziel haben, nicht außerhalb des erstellten Charakters zu agieren, eine Eigenschaft, welche Bartle zunächst nur Socializern zu schrieb[Bar96, S.3]. Des Weiteren wäre die Unterscheidung von Achiever und Killer in einem PVP-MUD ebenfalls um einiges schwieriger, da beide in diesem Umfeld wie perfekte Killer erscheinen.[vM07, S. 15]

Diesem Kritikpunkt steht entgegen, dass Bartle betonte: Sein Modell sei speziell auf die Ursache der Taten im Spiel ausgelegt. Nicht auf die Observation eines Spielers, der eine bestimmte Tat ausführt, welche ihn zu einem dazugehörigen Archetypen macht[Bar09, S. 10].

Also kann der Archetyp eines Spielers nur bestimmt werden, wenn dieser Spieler korrekt angibt, aus welchem Grund er Spaß hat. Die Schwachstellen hier sind die Voraussetzung, dass Spieler wissen, warum ein Spiel ihnen Vergnügen bereitet und dass das Modell diese verschiedenen Gründe berücksichtigt. Eine der am häufig genantesten Antworten auf die Fragen warum ein Spiel Spaß machte, das Gefallen an Immersion in eine Welt, ist nicht im Model vorhanden[Bar12a, S. 1].

2.3.2. Die fehlende Immersion

Die fehlende Inklusion des Aspekts der Immersion ist nicht die eizige Kritik, die Bartle einige Jahre später aufgriff. Die Mängel an seinem originalen Modell zählt er selbst auf:

1. Es gab zwei verschiedene Arten von Killer
2. Spieler verändern über die Zeit ihren primären Archetypen, jedoch ohne Erklärung
3. Das Modell berücksichtig nicht das Konzept von Immersion
4. Die Theorie hat keine Zusammenhänge zu anderen, etablierten Theorien außerhalb der Games-Industrie

[Bar09, S. 9]

Um diese Punkte auszubessern, erweiterte Bartle das Interessendiagramm um eine zusätzliche Achse. Diese Achse beschreibt die Ausführung der Aktionen eines Spielers näher. Die zwei Ausprägungen des Killers waren zum einen, Spieler, welche sich anderen aus destruktiven Gründen auferlegen, und zum anderen, Spieler, die dies

aus konstruktiven Gründen, wie zur Gruppenorganisation, tun.[Bar03, S. 223]

Ähnlich subtile Unterschiede innerhalb der restlichen vier Archetypen traten nach genauerer Untersuchung ebenfalls ans Licht. Die Handlungen eines Achievers können zum Beispiel bis auf das kleinste Detail ausgeplant sein, um die höchste Punktzahl zu erreichen. Der nächste Achiever hat genug vertrauen in seine aufgebauten Fertigkeiten, dass er gar nicht vorraus plant, wenn er die Herausforderungen des Spiels annimmt[Bar03, S. 226].

Das entstehende Muster stellt geplante Handlungen gegenüber Handlungen, welche sich auf die Instinkte des Spielers beruhen. Wenn ein Spieler sich keine Gedanken über seine Handlungen machen muss, beschrieb Bartle das Verhalten als *implizit*. *Explizites* Verhalten wurde als offen, durchdacht und ordentlich beschrieben[Bar03, S. 222]. Das erweiterte Model liefert 8 distinkte Spielertypen:

IMPLIZIT		EXPLIZIT	
AKTION	INTERAKTION	AKTION	INTERAKTION
Instinktive	Experten	Macher	Lerner
●Griefers	●Friends	●Politicians	●Networkers
●Opportunists	●Hackers	●Planners	●Scientists

Tabelle 2.1.: Die neuen Spielertypen unter betrachtung "Expliziter" und "Implizierter" Handlungen. Magenta denotiert einen Typ mit Fokus auf Spieler, Cyan einen Typ mit Fokus auf die Spielwelt.[Bar03, S.227]

Die zusätzliche Dimension des Modells löst nicht nur das Problem der vorher dissonanten Unterscheidung des Killers in zwei Unterkategorien, sondern auch die anderen, aufgelisteten Schwachstellen.

Das Wechseln des Archetypen eines Spielers wurde in Bezug auf das originale, zweidimensionale Modell am häufigsten in dieser Progression beobachtet: Killer → Explorer → Achiever → Socializer. Die typische Entwicklung eines Spielers kann demnach so erklärt werden: Die Spielerin wird anfangs die Grenzen ihres eigenen Verhaltens testen (Killer), danach beginnt sie, Wissen über ihre Umwelt zu sammeln (Explorer). Folglich wird sie das gesammelte Wissen anwenden (Achiever), wobei sie Verbindungen zu anderen Spielern aufbaut(Socializer). Schlussendlich wird sie sich davon zur Ruhe setzen und ihre Zeit mit Chatten unter ihren Freunden verbringen.[Bar03, S. 220] Jedoch wurde auch ein anscheinendes Oszillieren zwischen zwei Archetypen gesehen, zum Beispiel: Killer → Socializer → Killer → Socializer[Bar12a, S. 90].

Dieses Umschlagen zwischen zwei Typen kann nun besser erklärt werden. Ein Spieler fängt im impliziten Killer-Typ, dem Griefer, an. Testet sein Verhalten, wechselt dann zum expliziten Socializer, dem Networker, um bewusst Freunde zu schließen. Danach folgt eine Transformation zu dem expliziten Killer, dem Politician, zum Beispiel um die vorher geformten Kontakte in einer gemeinsamen Gruppe zusammenzuführen. Letzendlich ist der Spieler in seiner sozialen Gruppe angekommen und wird somit zum impliziten Socializer, dem Friend, welcher sich keine Gedanken mehr über die Interaktion mit seiner Gruppe an Mitspielern machen muss.

Durch diese Aufschlüsselung erkannte Bartle ein weiteres Muster. Beide gerade beschriebenen Werdegänge folgen der gleichen Form. Vom Beginn in einem Quadrant (Aktion + Spieler und Aktion + Implizit) ist der nächste Schritt zum gegenteiligen Quadranten (Interaktion + Welt und Interaktion + Explizit). Dann ändert sich der Archetyp nur auf der Interaktion/Aktion Achse (Aktion + Welt und Aktion + Explizit) und endet im letzten, wiederum gegenteiligen Quadranten (Interaktion + Spieler und Interaktion + Implizit).[Bar03, S. 224-225] Die vier am meisten beobachteten Sequenzen von Spielertypen lauten wie folgt[Bar03, S.228]:

- Griefer → Scientist → Planner → Friend
- Opportunist → Networker → Planner → Friend
- Griefer → Networker → Politician → Friend
- Opportunist → Scientist → Planner → Hacker

Mit der Fähigkeit, den Archetyp eines Spielers zuverlässig über Zeit zu verfolgen, kann man nun auch die Immersion eines Spielers berücksichtigen. Bartle nimmt an: Je weiter ein Spieler innerhalb einer Sequenz voranschreitet, desto vertiefter ist er im Spiel, sprich desto größer ist seine Immersion[Bar12a, S. 90]. Die Anwendung des Modells für Entwickler steigt, da sie es nun benutzen können, um den Inhalt des Spiels in Zusammenhang mit dem Fortschritt eines Spielers besser zu designen. Die typischen Sequenzen enden allesamt mit einem Spieler vom Typ "Friend" oder "Hacker". Demnach ist es sinnvoll, Inhalte für diese Archetypen mit dem Hintergedanken zu designen, dass sie am meisten von langwährigen Spielern benutzt werden. Die niedrige Immersion zu Beginn der Sequenzen kann auch von Entwicklern genutzt werden. Networker werden noch nicht sehr in die Spielwelt eingetaucht sein. Daher muss das System, welches Spielern ermöglicht, sich zu vernetzen, nicht zwingend in der Spielwelt erklärbar eingebettet sein. In der Planner-Stufe ist die Immersion des

Spielers um einiges höher. Damit der Grad an Immersion gehalten werden kann, ist es wichtiger für Inhalte, die an Planner gerichtet sind, immersiv zu sein. Zum Beispiel sollte die Rüstung, welche nach 20 Stunden Spielzeit endlich ausrüstbar ist, nicht nur die "Rüstung+2" sein, sondern die "Platin-Mithril Rüstung". [Bar03, S. 230]

Letzlich kann die persönliche Entwicklung, die Spieler innerhalb einer virtuellen Welt erfahren, mit Joseph Campbells *Heldenreise* verglichen werden.

Campbell war ein amerikanischer Mythenforscher. In seinem 1949 veröffentlichten Buch *Der Heros in tausend Gestalten* untersucht er eine Vielzahl von Geschichten und Überlieferungen verschiedener Kulturen. Unter anderen betrachtete er sie unter dem Aspekt der Psychoanalyse [Cam04, S. 8]. Sein Ziel war es, universelle Gemeinsamkeiten zu finden. Diese fand er in den vorhandenen Erzählstrukturen. So wird der Protagonist, der Held, immer aus seiner familiären Umwelt auf eine Reise in eine unbekannte Umgebung geschickt. Auf seiner Reise trifft er auf Helfer und Widersacher. Nach Bewältigung der Herausforderungen, die ihm gestellt wurden, kehrt er schlussendlich zu seiner vertrauten Umgebung zurück. [Cam04, S. 227]

Bartles erweitertes Modell weist Parallelen zu dieser Heldenreise auf. Unter Ergänzung des Immersionsgrades ist es nun möglich, den Fortschritt des Spielers auf der Heldenreise nachzuvollziehen. Somit wird der Spieler selbst zum Helden, anstatt die Heldenreise nur aus zweiter Hand durch einen Protagonisten zu erfahren. Die unbekannte Umgebung ist die virtuelle Welt des Spiels. Andere Mitspieler können sowohl die Rolle des Helfer als auch des Widersachers verkörpern.

Durch die starke Verbindung zur etablierten Heldenreise kann Bartles Modell nun in anderen akademischen Feldern, wie zum Beispiel der Psychologie, verankert und angewendet werden. [Bar12a, S. 90]

Diese tiefgreifende Erweiterung des Modells steigert die Nutzung für Entwickler, sowie die Anwendung in anderen, akademischen Feldern. Ein weiterer großer Kritikpunkt wird jedoch nicht damit außer Kraft gesetzt.

2.3.3. Wissenschaftlichkeit

Bartles Player Suits Modell mangelt es an empirischen Daten und Herangehensweisen. Die gesammelten Daten, auf denen das Modell basiert, beruhen auf persönlichen

Anekdoten und Meinungen von Spielern in einem Forum. Diese Methodik führt zu mehreren Schwachstellen im Modell:

1. Vorgestellte Komponenten eines Typen sind in Wahrheit unabhängig voneinander
2. Typen, welche als unabhängig voneinander beschrieben wurden, können in Abhängigkeit zueinander auftreten
3. Das Modell allein bietet keinen Weg für Spieler, ihren Typen herauszufinden

Nicholas "Nick" Yee stellte diese Limitation fest.[Yee05, S. 2]

Punkt eins führt zu einer unsauberer Definition der Typen. Bestimmte Merkmale von Typen können in Wahrheit gar nicht so markant für den Typ sein, während wichtige Eigenschaften exkludiert werden.

Die meisten Explorer legen vielleicht keinen hohen Wert auf das Kartieren der virtuellen Geografie der Spielwelt, sondern sind exklusiv auf das Explorieren von Spielmechaniken fokussiert. Für Achiever wäre es vielleicht viel interessanter, ihren Bezug zu einem Verlangen nach Macht zu betrachten als schlichtweg die Akkumulation von Punkten.[Yee02, Reiter "Bartle's Types"]

Punkt zwei beschwert das korrekte Identifizieren eines Spielers zu einem einzelnen Archeytpen. Würde ein Spieler der sehr aktiv in einer Gruppe zum effizienten Spielen des Spiels als Socializer oder Achiever zählen[Yee05, S. 2]? Wenn der Spaß auf dem Erklimmen der Herausforderungen des Spiels (für Achiever), genauso wie auf dem Dasein anderer Spieler (für Socializer), beruht, dann sind die vorher als unabhängig gesehenen Archeytpen keine schlüssige Taxonomie mehr.

Punkt drei ist neben den genannten Punkten weniger ausschlaggebend. Eben durch die vorher genannten Schwachstellen im Modell kann ein versuchtes Einordnen eines Spielers eher zur unbeabsichtigten Kreation neuer Spielertypen führen, anstatt dass bereits vorhandene Typen daran gemessen werden[Yee05, S. 2].

Der sogenannte Bartle-Test⁶, welcher zwar nicht von Bartle selbst designt wurde, versucht diese Lücke zu füllen. Diese Version des Tests besteht aus 30 Fragen mit je zwei Antwortmöglichkeiten. Das Resultat zeigt dem Nutzer anschließend, welchen

⁶Erstellt von Erwin S. Andreasen und Brandon Downey. Version gehostet von Matthew Barr
<https://matthewbarr.co.uk/bartle/>

primären Archetypen er besitzt, sowie zu wie viel Prozent er zu den anderen Archetypen zählt.

Die Präsentation der Fragen sowie deren Antwortmöglichkeiten ist der problematische Aspekt des Bartle-Tests. Durch die präsentierten Antworten wird der Nutzer zwingend in die Archtypen eingeteilt. Somit wird das Modell im Rahmen des Testes selbsterfüllend. Zudem wird eine eine Dichotomie zwischen den Typen weiter vorangetrieben, wo keine vorhanden ist[Yee02, Reiter "Bartle's Types"].

Das Fehlen eines angemessenen Einteilungssystems hindert auch das Unterscheiden von Spieleraktionen in Bezug auf die Typen. So würden Explorer, welche schlicht das gewalttätige Beeinflussen anderer Spieler als würdig der Exploration betrachten, als Killer eingeschätzt werden[Bar03, S. 186]. Was wiederum die analysierten Dynamiken zwischen Spielertypen hinterfragt. Somit wird die Effizienz dieses Aspektes des Modells für Entwickler annulliert.

Nick Yee sah diese Probleme und stellte Recherchen an, um Bartles Modell empirisch zu belegen. Daraus entstand Yees Modell über "Spielermotivationen", mit einigen Gemeinsamkeiten, aber auch Unterschieden zu dem Player Suits Modell.

3. Konzeption

Bartle erstellte das Modell der Player Suits als Hilfsmittel für Spielentwickler [Bar09, S. 4-5]. Unter Benutzung des Modells soll es Entwicklern einfacher fallen, Spiele zu entwickeln, welche nicht nur sie selbst, sondern auch andere Personen mit Freude spielen.

Ziel dieser Arbeit ist es Bartles Modell in der Praxis zu validieren. Um dies zu erreichen, wurde das Spiel *Explorer Bot* entwickelt. Das Modell erfuhr Anwendung im Bereich des Gamedesigns. Die Inklusion von Spielelementen, welche auf die vier Archetypen kalibriert sind, sollte den Spielspaß der Spieler steigern. Des Weiteren sollten sich Trends innerhalb der Playerbase bilden, welche indikativ für die Zugehörigkeit eines Archetypen sind.

Die Daten, um diese Annahme zu beweisen und Bartles Modell zu validieren, kommen aus zwei Quellen. Der erste Satz an Daten wurde automatisch von den Spielern gesammelt, während sie *Explorer Bot* gespielt haben. Ausgewählte Aktivitäten einzelner Spieler wurden verfolgt und zusammengetragen durch die Nutzung der Unity Analytics Services.

Ein unterstützender Satz an Daten wurde durch das Beantworten einer Umfrage gesammelt. Die freiwillige Umfrage wurde von Spielern nach Beendigung des Spiels beantwortet und stellte die bleibenden Eindrücke spezifischer Elemente in *Explorer Bot* in den Vordergrund.

Das folgende Kapitel wird die gewählten Gamedesign-Elemente und ihre dazugehörigen Archetypen behandeln. Anschließend werden die automatische Datenakquisition und die Umfrage beleuchtet.

3.1. Explorer Bot's Gamedesign im Kontext zu Bartle

Zunächst wird in diesem Kapitel eine grobe Übersicht über die Inhalte des Spiels gegeben. Anschließend werden die Verbindungen zwischen diesen Elementen und den vier Archetypen des Player Suits Modells erläutert.

Explorer Bot wird durch seine Nichtlinearität ausgezeichnet. Der Spieler schlüpft in die Rolle eines Roboters, welcher gerade von einer Hexe fertig gebaut wurde. Das Spiel startet mit dem Augenblick des "Erwachens" des Roboters. Seine Erbauerin erklärt ihm, dass sie vergessen hätte, wieso sie überhaupt einen Roboter gebaut hat. Bis der Grund ihr wieder einfällt, ist dem Roboter damit die Freiheit gewährt, die Welt zu erkunden. Dem Spieler signalisiert dies, dass ihm keine Einschränkungen für seine nächsten Aktionen in diesem Sandbox-Game gestellt werden.

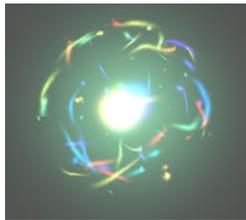


Abbildung 3.1.: Point Of Interest

Eines der ersten Objekte, die der Spieler entdeckt, nimmt die Form einer leuchtenden Kugel an. Wenn der Spieler nahe genug an sie herantritt, wird sie von ihm aufgesammelt. Die Platzierung dieser Spielobjekte dient zum Markieren von Interessenspunkten der Welt. Im HUD wird dem Spieler angezeigt, wie viele dieser Orte von Interesse er bereits gefunden hat und wie viele es insgesamt gibt. Mit dem Einsammeln einer jeden Kugel wird dem Spieler ebenfalls ein kleiner Satz angezeigt, der die Welt ausgestaltet.

Weiterhin bietet *Explorer Bot* zwei Arten an Nicht-Spieler-Charakteren (NPCs) an. Der Spieler kann auf unterschiedliche Weisen mit ihnen interagieren. Die erste Art besteht aus einer Vielzahl an Monstern, welche in der Welt verteilt sind und bei Blickkontakt mit dem Spieler anfangen diesen anzugreifen. Im Verlauf des Spiels erhält der Spieler Waffen, mit denen er sich von diesen Angriffen verteidigen und somit die Gegner besiegen kann.

Die zweite Art von NPC bilden die friedlichen Menschen, welche der Spieler hauptsächlich innerhalb des Dorfes im Spiel vorfindet. Die Interaktion mit diesen NPCs beläuft sich auf das halten von Konversationen. Je nach Fortschritt im Spiel werden Charaktere unterschiedlich auf den Spieler reagieren und ihren Dialog entsprechend ändern.

Das Reden mit NPCs ist eine essentielle Spielmechanik für die Spieler *Explorer Bots*. Während sie die Welt erkunden, werden sie auf NPCs treffen, welche ein rotes

Ausrufezeichen über ihren Köpfen schweben haben. Diese Figuren haben eine Quest an den Spieler zu geben.

Explorer Bot bietet dem Spieler drei verfügbare Quests an:

	Pilz Quest	Farm Quest	Drachen Quest
Auftraggeber	Dorfälteste	Schmiedin	Magierin
Kontext	durch den Aufruhr im Dorf kann ihr niemand beim Pilze-sammeln helfen	Monster verwüsten die Gemüsefelder und müssen vertrieben werden	Drache ist aufgewacht und gefährdet die Sicherheit der Dorfbewohner
Aufgabe	Spieler muss 5 Pilze im Wald nahe des Dorfes finden	Spieler muss alle Monster in Nähe der Farm besiegen	Den Drachen im Kampf besiegen ODER Drei in der Welt verteilte Obelisken finden und aktivieren, um den Drachen wieder zum Schlafen zu bringen
Belohnung	Keine Belohnung	größeres Schwert was mehr Schaden anrichtet	Keine direkte Belohnung
Besonderheit	Keine Besonderheiten	Falls Spieler zu Beginn der Quest noch kein Schwert hat, gibt ihm die Schmiedin eins	mehrere Lösungswege

Tabelle 3.1.: Übersicht der Quests in *Explorer Bot*

Die Quests von *Explorer Bot* beinhalten Elemente für mehrere Archetypen. Die folgenden Unterkapitel werden entsprechende Aspekte der Quests erläutern. Desweiteren spielen Quests eine essentielle Rolle für das Determinieren der Spielerarchetypen. Ihr Effekt in der Gesamtheit *Explorer Bots* ist die Vorgabe geregelter Szenarien. Innerhalb dieser Szenarien ist das unterschiedliche Verhalten von Spieler zu Spieler am stärksten zu erkennen.

3.1.1. Achiever-Elemente

Spieler vom Typ Achiever werden in *Explorer Bot* vielen Elementen begegnen, welche den Spielspaß für sie steigern. Der Platz von Achievern auf Bartles Inter-

essendiagramm 2.1 beschreibt, dass ihr Fokus auf ihrer Wirkung auf der Welt liegt. Achiever wollen ihre Fähigkeiten entwickeln und dadurch die virtuelle Welt meistern. Der Fortschritt in ihren Fähigkeiten und somit auch im Spiel an sich wird auf verschiedene Weisen dargestellt. Außerdem steht die Akkumulation von Punkten im Vordergrund. In *Explorer Bot* wird der Spielfortschritt mit dem Sammeln von Punkten verknüpft.

Die in Abbildung 3.1 gezeigten Point Of Interests sind Collectibles, welche der Spieler in der Welt verstreut findet. Im HUD können Spieler zu jeder Zeit sehen, wie viele Points bereits eingesammelt wurden, und wie viele insgesamt in der Welt existieren. Dieser Zähler im HUD wurde primär für Achiever in das Spiel eingebunden. Da es sich bei *Explorer Bot* um ein nicht-lineares Sandbox-Spiel handelt, besteht eine der Hauptmechaniken aus dem Herumlaufen in der virtuellen Welt. Für Achiever ist das nicht genug, um Spaß daraus zu schöpfen. Daher wurden die Point of Interests und der Zähler als eine Art Ziel implementiert. Achievern wird es dadurch ermöglicht, nur durch Benutzung der achiever-fernen Hauptmechanik etwas innerhalb des Spiels zu erreichen.

Gesammelte Collectibles sind nicht das Einzige, was für den Spieler festgehalten wird. Im Pausemenü werden ebenfalls die Status der Quests angezeigt. Für jede der drei Quests wird dem Spieler die Information gegeben, ob die jeweilige Aufgabe bereits erledigt ist, oder das Problem noch besteht. Wenn eine Quest noch nicht angefangen wurde, können Spieler in die Richtung gelenkt werden, um diesen Spielinhalt zu entdecken. Nach dem Erledigen einer Quest wird dieser Fortschritt langanhaltend für Spieler veranschaulicht. Somit wird das Abschließen von Quest zu einer anderen Art von "Punkten", welche Achiever sammeln können.

Explorer Bots Quests beinhalten weitere Elemente für Achiever. So besteht zum Beispiel die Pilz Quest alleinig aus dem Sammeln einer Ressource. Innerhalb dieser Quest wird der Fortschritt ebenfalls über einen Zähler im HUD nachverfolgt. Der Spieler wird von einer alten Frau gebeten, fünf Pilze für sie zu finden und zu sammeln. Diese Pilze befinden sich in dem Wald abseits des Dorfes, in dem sie die Quest angenommen haben. Neben den gesuchten Pilzen beherbergt der Wald eine Vielzahl von pilzähnlichen Monstern. Sollte der Spieler diese Quest ohne ein Schwert anfangen, muss er den Monstern gekonnt ausweichen, um nicht angegriffen zu werden. Nachdem er ein Schwert erhält, kann er den Wald beruhigter absuchen und sich gegen die näher kommenden Gegner wehren. Beide dieser Szenarien for-

dern die Fähigkeiten des Spielers heraus und geben Achievern die Befriedigung, ihre Fertigkeiten zu verbessern und zu beweisen.

Ähnlich verhält es sich mit der Farm Quest. Die dem Spieler gegebene Aufgabe handelt sich hierbei um das Verteidigen der Gemüsegelder des Dorfes. An der Farm angekommen, wird der Spieler mit sechs Monstern konfrontiert. Diese Monster unterscheiden sich von denen, die der Spieler im Wald antreffen kann. Sie sehen nicht nur anders aus, sondern besitzen auch mehr Angriffskraft und Lebenspunkte. Zusätzlich wurde ein Monster größer und gefährlicher gestaltet. Dieses Monster fungiert als eine Art Boss-Monster und stellt eine weitere Herausforderung für den Spieler dar. Im Kampf wird der Fortschritt des Spielers ebenfalls grafisch dargestellt. Den Gegnern wurde eine Lebensanzeige verliehen. Mit jedem Hieb seines Schwertes richtet der Spieler Schaden an und verringert somit den Füllstand der Anzeige. Um dem Spieler das Gefühl von steigender Macht durch seinen Fortschritt zu geben, wurde ebenfalls die Anzeige von Schadenspunkten implementiert. Über den Verlauf des Spiels kann der Spieler eins von drei Schwerten in seinem Besitz haben. Das schwächste Schwert bekommt er zu Beginn der Farm Quest, das stärkste wird ihm als Belohnung nach Abschluss dieser Quest gegeben. Das dritte Schwert besitzt einen Angriffswert zwischen den beiden und kann in der Welt gefunden werden. Somit dient es zusätzlich als Belohnung für Achiever, welche sich tiefer mit der Welt auseinandersetzen.

Im Bezug auf Achiever baut die dritte Quest des Spiels auf dem Machtverhalten der Farm Quest auf. Die Drachen Quest präsentiert den Spieler mit mehreren Lösungswegen. Achiever können ihre Macht nach einem langen, anstrengenden Kampf mit dem Drachen beweisen. Alternativ können sie auch die Welt benutzen, um den Kampf leichter zu gestalten. In beiden Fällen werden die Fähigkeiten des Spielers auf die Probe gestellt. Wenn der Achiever auf eine direkte Konfrontation verzichtet und den Drachen erst zum Einschlafen bringt, kann er den Drachen sehr leicht besiegen. Dadurch wird das Gefühl von beherrschter Macht verstärkt, was sich lohnend für Achiever anfühlt.

Um das Spiel auf Achiever zu kalibrieren, wurden eine Vielzahl an Designmaßnahmen getroffen. Die Zeit zwischen dem Erledigen von Quests wird für Achiever durch das Sammeln von Collectibles ausgeschmückt. Während Achiever die Quests von *Explorer Bot* abschließen, werden ihre Fähigkeiten entwickelt und getestet.

3.1.2. Explorier-Elemente

Explorierer, wie der Name andeutet, wollen explorieren. Damit sie Spaß in einer virtuellen Welt haben, muss diese Welt etwas bieten, was entdeckt werden kann. Deswegen ermöglicht *Explorer Bot* dem Spieler ab Beginn, die gesamte Spielwelt zu erkunden.

Das Entdecken der Welt wird auf verschiedenste Weisen anerkannt. So kann der Spieler beispielsweise ein Schwert versteckt in der Höhle hinter einem Wasserfall finden. Infolgedessen werden dem Spieler mehr Möglichkeiten zur Interaktion mit der Welt gegeben, da er nun ein Schwert besitzt. Somit werden Explorierer, welche sich die Zeit genommen haben, diesen Winkel der Welt zu entdecken, mit noch mehr Optionen ihrem Spielspaß zu folgen, belohnt.

Explorer Bots Welt hat noch weitere Orte und Sachen zu entdecken. Die meisten von ihnen werden mit den in Abbildung 3.1 gezeigten leuchtenden Kugeln markiert. Diese Point of Interests sind für Explorierer ein kleiner Leitfaden für die Erkundung. Wie im Achiever-Abschnitt unter 3.1.1 beschrieben, trägt das Einsammeln der Point Of Interest Kugeln zu Achiever-Inhalten bei. Das Sammeln der Kugeln hat jedoch eine weitere Funktion. Sie geben dem Spieler Informationen über die Welt. Jedes dieser Collectibles gibt dem Spieler einen Satz über die Welt. Im Allgemeinen führt dies zu einer Welt, die sich realer und durchlebter anfühlt. Speziell für Explorierer hat es die Wirkung, dass sie durch ihre Exploration mit Wissen belohnt werden.



Abbildung 3.2.: Einer der im Spiel zu findenden Obelisken

Das Aspekt des Experimentierens wurde ebenfalls berücksichtigt. In der Welt verteilt sind drei Obelisken. Ihr Zweck dient dem Abschließen der Drachen Quest. Sobald jede der drei Säulen aktiviert ist, schläft der Drache ein und die Quest gilt als abgeschlossen. Obwohl die Magierin, ein abgelegener NPC, dem Spieler von den Obelisken erzählt, ist diese Information nicht notwendig, um die Obelisken aktivieren zu können.

Spieler haben eine limitierte Anzahl an Möglichkeiten, mit den Objekten in der Welt zu interagieren. Um mit NPCs zu

reden, wird den Spielern angezeigt, welche Taste sie drücken müssen, um eine Konversation zu starten. Die zweite Möglichkeit, zu interagieren, wird vom Spiel gar nicht signalisiert. Nur durch Ausprobieren seiner Möglichkeiten findet der Spieler heraus, dass die Obeliskten durch einen Schwerthieb aktiviert werden können.

Obeliskten wurden in der Welt so platziert, dass sie Spieler mit einer großen Wahrscheinlichkeit finden, bevor sie von ihnen und ihrem Zweck in der virtuellen Welt erzählt bekommen. Dies unterstützt eine größere Belohnung, wenn Spieler durch Ausprobieren die Obeliskten bereits aktiviert haben. Sie sparen nicht nur Zeit dadurch, sondern auch ihr Einfallsreichtum wird von der Magierin durch zusätzlichen Dialog anerkannt.

Die Natur des Spiels, den Spieler nicht auf einen geradlinigen Weg zu lenken, formt die Basis für das Ansprechen von Spielern des Typen Explorer. Durch das Sammeln der Point of Interests können Explorer ihr Wissensnetz über die Welt aufbauen und vertiefen. Schließlich wird ihr Bedarf an Experimentation mit dem Rätsel um die Obeliskten befriedigt.

3.1.3. Killer-Elemente

Das Einbringen von Elementen, die einem Spieler von Typ Killer Freude bereiten, fiel schwerer, als die in 3.1.1 und 3.1.2 beschriebenen Elemente. Wie im Interessengraph 2.1 dargestellt, liegt das Interesse von Killern nicht bei der virtuellen Welt, sondern bei anderen Spielern. Da es sich bei *Explorer Bot* um ein Einzelspieler-Spiel handelt, besteht von anderen Spielern ein Mangel. Als Approximation wurden humanoide NPCs verwendet.

Des Weiteren haben Killer Interesse daran, auf andere einzuwirken. Diese Einwirkung muss nicht unbedingt destruktiv sein, wie der Name zunächst anscheinend lässt. Positive Reaktionen auf den Spieler sind genauso ausschlaggebend für das Genießen des Spiels. Um den Aktionen des Spielers eine tiefere Wirkung zu verleihen, wurden die Dialoge der NPCs dynamisch gestaltet. Über den Verlauf einer Quest werden die auftraggebenden NPCs nicht immer die gleichen Zeilen an Konversation dem Spieler übergeben. Die Kulmination dieser reaktiven Dialog-Stadien wird nach dem Abschluss der Drachen Quest erreicht. Wie in Tabelle 3.1 beschrieben, gibt es zwei Lösungswege für diese Quest.

Wenn der Spieler auf das Plateau des Drachen kommt, begegnet er der Magierin. Zunächst verhält sie sich ihm kalt gegenüber, erzählt ihm dennoch über den Drachen und die friedliche Lösung der Quest. Nun kann der Spieler sich entscheiden, ob er den gewaltlosen Weg verfolgt oder den Drachen in einem Kampf tötet. Zu diesem Zeitpunkt ist ein Kampf mit dem Drachen schwer, aber nicht unmöglich. Wenn der Spieler die passive Route einschlägt, ändert sich die Meinung der Magierin ihm gegenüber und sie wird freundlicher in ihren Konversationen. Nach Erreichen des Einschläferns des Drachen bittet sie den Spieler einfach wieder ins Dorf zu gehen. Dort erwarten ihn dann eine Reihe neuer Dialoge der Dorfbewohner, welche sich auf das Einschläfern des Drachen beziehen. Einige sind froh, dass die Gefahr vorüber sei, aber manche bleiben ängstlich für den Fall des erneuten Erwachens des Drachen. Dem Spieler steht es frei, zu dem Plateau zurückzukehren. Der nun schlafende Drache ist wehrlos und der Spieler kann ihn mit einem einzigen Hieb niederschlagen. Diese Aktion verändert ein weiteres Mal alle Dialoge der NPCs mit dem Spielercharakter. Die Magierin ist erzürnt, dass der Spieler ein wehrloses Tier angriff. Die Dorfbewohner werden den Spieler jedoch als Helden bezeichnen.

Diese Tiefen an NPC-Reaktionen geben Killern das Feedback, dass ihre Aktionen einen messbaren Effekt auf andere Leute haben. Selbst wenn diese "Leute" gänzlich virtuell und computergesteuert sind.

3.1.4. Socializer-Elemente

Der in 3.1.3 beschriebene Mangel an Mitspielern ist für den Socializer noch spürbarer, da Spieler dieses Typen nicht nur auf andere Spieler, sondern besonders auf die Interaktion mit diesen fokussiert sind.

Die Mechanik, welche am meisten für Socializer ausgelegt ist, sind die dynamisch gestalteten Dialoge. Die meisten NPCs ändern nach dem erstmaligen Ansprechen den Inhalt der Konversation mit dem Spielercharakter. Die tiefsten Schichtungen der Dialoge erfahren Socializer mit den NPCs, die ihnen Quests geben. Während der Quest, die Pilze zu sammeln, wird im Dialog mit der Dorfältesten die genaue Anzahl der noch benötigten Pilze kalkuliert. Wenn der Spieler die Farm von Monstern befreit hat, ohne vorher mit der Schmiedin zu reden, reagiert sie positiv überrascht, bevor sie die Belohnung übergibt. Die Magierin reagiert darauf, ob der Spieler bereits die Farm Quest abgeschlossen und mindestens einen Obelisk aktiviert hat. Im Verlauf

der Drachen Quest, wird die Magierin ebenfalls aktiv erkennen, welche Obelisken der Spieler aktiviert hat. Folglich gibt sie nur Hinweise, wo inaktive Obelisken zu finden sind. Diese Vielfalt an Dialogen mit dem Spieler hilft, Socializern Beziehungen mit den Figuren zu knüpfen.

Auch gibt es zwei Instanzen von Gruppen aus NPCs. Diese Gruppen reden untereinander mit sich, der Spieler hört lediglich zu. Das verleiht diesen NPCs einen Grad an Eigenständigkeit und simuliert somit die Präsenz von Spielern, welche nicht an den Socializer gebunden sind. Das baut das Gefühl einer "Community" innerhalb der virtuellen Welt auf. Durch den Drang nach Verbindung von Socializern wird *Explorer Bot* dadurch für sie attraktiver. Socializer, die Teil dieser Community werden wollen, erreichen dies durch das Erledigen der Quests. NPCs reagieren mit Freude auf die geleistete Hilfe. Folglich wird die Verbindung zu ihnen vertieft. So werden Socializer zu den anderen Inhalten des Spiels gelenkt.

Neben dem Gefühl, Teil einer Community zu sein, begehren Socializer ebenso den Aufbau von direkten Beziehungen mit anderen Spielern. In *Explorer Bot* kann die tiefste Beziehung mit der Hexe aufgebaut werden. Die Hexe ist der erste NPC, den Spieler antreffen. In der Geschichte des Spiels ist sie diejenige, die den robotischen Spielercharakter erschaffen hat. Durch diese Vorgeschichte haben Spieler einen direkten Draht zu der Figur. Obwohl sie den eigentlich Grund für die Kreation des Spielercharakters vergas und ihn folglich direkt nach ihrer ersten Konversation in die Welt rausschickt. Spieler, die weitere Male mit ihr reden, erfahren, dass sie sich um den Spielercharakter sorgt. Im Verlauf des Spieles können Spieler immer wieder zu ihr zurückkehren. In den resultierenden Konversationen stehen die Aktionen des Spielers im Vordergrund. Nach erfolgreichem Abschluss der Pilz quest erinnert sie sich beispielsweise, dass sie den Roboter für genau diesen Zweck erschuf. Wenn der Spieler über seine Taten auf der Farm berichtet, hegt sie Befürchtungen über den Fall, dass dem Spielercharakter etwas hätte zustoßen können. Entsprechend dieser Sorge spielt die Hexe auch die Rolle der Heilerin. Im Dialog bemerkt sie, ob dem Spieler Lebenspunkte fehlen, und heilt ihn entsprechend. Sollte der Spieler im Rahmen des Spiels sterben, wacht er wieder vor der Hexe auf und sie erzählt ihm, wie sie ihn reparierte.

Socializer werden von der Beziehung, welche der NPC zu ihrem Charakter fühlt, angesprochen. Die Beziehung kommt nur zum Vorschein, durch die Interaktion mit der Hexe. Somit gleicht das Reden mit ihr dem Aufbau dieser Beziehung.

Ähnlich wie für Killer spielt das Ausmaß an NPC-Interaktion eine bedeutende Rolle für Socializer in *Explorer Bot*. Sie bilden das Gerüst für alle sozialen Aktionen, die der Spieler ausführen kann. Folglich sind sie auch der Katalyst für Socializer, sich mit den weniger sozialen Aspekten des Spieles auseinanderzusetzen.

3.2. Datenansammlung

Wie in Abschnitt 2.3.3 beschrieben, war ein großer Kritikpunkt des Player Suits Modells die Quelle von Bartles Daten. Anstelle einer freien Diskussion, die Spieler von *Explorer Bot* über ihre Erfahrung halten, wurde für diese Arbeit ein gezielter Fragebogen erstellt. Das Hauptziel des Fragebogens verfolgt die Evaluation der in 3.1 besprochenen Gamedesign-Elemente.

Des Weiteren wurden während der Spielzeit Daten über die Aktionen des Spielers gesammelt. Dieser Prozess wurde im Hintergrund mit Hilfe der Unity Analytics Services ausgeführt. Dies half, Daten zu sammeln, ohne den Spielfluss des Spielers zu unterbrechen. Das Wissen über die Spielerhandlungen hilft dem Aufbau eines Modells, analog zu Bartles Playertypes, für *Explorer Bot*.

Diese zwei Datensätze konstituieren die Evaluation der Validierung von Bartles Modell. Der automatisch gesammelte Datensatz ermöglicht eine Analyse der Spieler im Kontext des Spiels. Die Umfrage setzt diese Analyse in einen größeren, sowie persönlicheren Kontext des Spielers und seiner sonstigen Präferenzen in Videospiele.

3.2.1. Automatische Datensammlung

Der von Unity bereitgestellte Service zur Datenakquise arbeitet mit Events. Diese Events können während der Entwicklung festgelegt werden. Somit konnte gezielt abgefragt werden, wann ein Spieler etwas tat. Zum Beispiel, wann welche Quest abgeschlossen oder welcher Point of Interest eingesammelt wurde. Für *Explorer Bot* wurden zwölf spezifische Events festgelegt. Jedes Event mit Beschreibung ist im Anhang unter A.2 gelistet. Das Fehlen eines vorher festgelegten Events ist ebenso aufschlussreich für das Verhalten des Spielers. Wenn ein Spieler keinen Eintrag mit dem Event seines Spieltodes aufweist, könnte das indikativ für einen hohen Skill sein.

Ein weiterer Vorteil der Unity Analytics Services ist das genaue Protokollieren des eingehenden Events. Somit wird unter anderem der Zeitstempel des Events festgehalten. Dies erleichtert das Verfolgen der Aktionen eines Spielers über den Verlauf des Spiels sowie einen tieferen Einblick in die Motivationen des Spielers. So können Interessen von Spielern ermittelt werden. Spieler, deren Fokus auf dem Sammeln von Point Of Interests lag, werden das dazugehörige Event öfter auslösen. Dieses erkennbare Muster deutet demnach auf die Tendenz des Spielers, zum Achiever- oder Explorer-Archetyp zu gehören. Unter Betrachtung davon, wie schnell hintereinander die Points gesammelt wurden, kann zusätzlich zwischen den beiden weltfokussierten Archetypen entschieden werden. Eine kürzere Zeit zwischen den Sammlungsevents deutet auf ein gezielteres Suchverhalten, was auf einen Achiever zurückzuführen ist. Im Gegensatz kann eine größere Zeit zwischen den Events bedeuten, dass der Spieler länger an einem Ort der Welt verweilt und diese einnimmt oder anderweitig analysiert. Dieses Verhalten wäre mit Explorern assoziiert.

3.2.2. Probandenumfrage

Bartle betonte, dass der Archetyp eines Spielers nicht aus dem alleinigen Ausführen einer Tätigkeit zu erkennen ist. Viel wichtiger sei der Grund der ausgeführten Tätigkeit.[Bar09, S.10] Um den Datensatz der faktischen Tätigkeiten der Spieler auszubauen und dies zu berücksichtigen, wurde eine Umfrage erstellt. Die Umfrage wurde von den Spielern auf freiwilliger Basis nach Abschluss des Spiels ausgefüllt. Die Liste aller Fragen ist in A.1 nachzulesen. Da die automatische Datenakquise anonymisiert erfolgte, konnten keine direkten Verbindungen zwischen den Datensätzen gezogen werden. Die Antworten auf die Umfrage geben dennoch eine wertvolle Übersicht auf die Einstellung der Spieler. Die Umfrage erwähnte gezielt keinen Bezug zu Bartles Archetypen. Spieler sollten nicht unbewusst in eine der vier Kategorien gelenkt werden, falls sie von dem Modell oder ihrem eigenen Archetyp wussten.

Der Fragebogen beinhaltet zehn Fragen. Diese beziehen sich auf die Inhalte von *Explorer Bot* und wie der Spieler sie in anderen Spielen wahrnimmt. Die Probanden werden zunächst befragt, wie viel Spaß ihnen das jeweilige Element bereitet hat. Die Antwort bildet eine Einstufung auf einer Skala von eins bis fünf. Eine Antwort von eins beschreibt dabei, dass das Element keinen Spaß bereitete. Die Antwort von fünf sagt aus, dass der Spieler große Freude durch dieses Element im Spiel erfuhr. Gekoppelt zu diesen Fragen wurde Probanden die Möglichkeit gegeben, ihre Antwort

mit eigenen Worten zu erklären. Sowie ihre Einstellung zu dem bestimmten Element in anderen Spielen. Dies half zu unterscheiden, ob Spieler, denen ein Element nicht gefiel, es generell ungern in Spielen sehen, oder lediglich die spezielle Implementation in *Explorer Bot* sie nicht ansprach.

Des Weiteren wurde gefragt, wie Spieler die Entscheidung trafen, was sie in *Explorer Bot* machen. Durch die nichtlineare Art des Spiels ist diese Information ausschlaggebend. Socializer würden sich auf den "Befehl" des Hexen-NPCs, die Welt zu erkunden, beruhen. Explorer hielten sich weiterhin daran und beschrieben, dass sie lediglich in der Welt umherliefen. Während Killer und Achiever gezielter die Orte aufsuchten, die für sie interessant erschienen.

Letzlich wurde den Spielern die Möglichkeit gegeben, sämtliche Gedanken zu ihrer Erfahrung mit *Explorer Bot* niederzuschreiben. Diese Impressionen halfen, das Bild des Spielers weiter aufzubauen. Durch die Offenheit der Frage beinhaltet die Antwort das Wichtigste für den Probanden. Somit ist selbst ein Wiederholen schon gegebener Informationen eine wertvolle Einsicht, da dieser Aspekt nun noch mehr Gewicht für die Bestimmung des Archetypen hat.

4. Implementierung

Das folgende Kapitel behandelt die Entwicklung des Spiels *Explorer Bot*. Insbesondere wird auf die Erstellung und Einbindung der in Kapitel 3 beschriebenen Elemente eingegangen.

Explorer Bot wurde mithilfe der Unity Engine entwickelt. Die Entscheidung fiel durch eine Vielzahl an Vorteilen, welche die Entwicklungszeit einfacher gestalteten. Zum einen entstand das Projekt auf der von Unity bereitgestellten *Third Person Core*-Vorlage. Diese Vorlage beinhaltet einen steuerbaren Spielercharakter und eine auf ihn gerichtete Kamera aus der dritten Person. Damit wurde schnell die Grundlage für die weitere Entwicklung erschaffen.

Weitere Vorteile beziehen sich auf die Verfügbarkeit kostenloser 3-D-Modelle im Asset shop und dem eingebauten Terrain-Tool zur Erstellung der Spielwelt. Sowie eine große Community, welche Add-Ons bereitstellt, wie *Yarn Spinner*, welches zur Implementation der Dialoge verwendet wurde. Schließlich war Unitys Service zur Datensammlung ein ausschlaggebender Punkt für die Nutzung der Engine.

4.1. Unity Analytics Service

Der Unity Analytics Service (UAS) ist ein Unterbereich der Unity Cloud¹, welcher sich auf das Sammeln von Daten und deren Analyse fokussiert.

Die Informationen, welche UAS gesammelt werden, sind in der Unity Cloud abrufbar. Die Nutzung des Service stellt voraus, dass das Entwicklungsprojekt mit einem Projekt der Cloud verbunden wird. Aus dem Editor der Engine heraus kann diese Verbindung unter `Edit>Project Settings>Services` erstellt werden. Je nachdem, welcher Service implementiert werden soll, muss das entsprechende Package mit

¹Unity Cloud und Unity Engine entwickelt von Unity Technologies

Unitys *Package Manager* heruntergeladen werden. Für *Explorer Bot* wurde das Analytics Package verwendet. Mit Abschluss der Vorbereitungen kann der Service durch eine Initialisierung gestartet werden. Diese kann an einem beliebigen Ort im Projekt stattfinden und wird mit dem Befehl `await UnityServices.InitializeAsync();` ausgeführt. Sobald dieser Prozess erfolgreich beendet wurde, beginnt die Datensammlung.

Wie in Kapitel 3.2.1 beschrieben, arbeitet der Unity Analytics Service mit programmierbaren Events. Dafür muss zu einem das Event im Cloud-Projekt definiert werden. Dort bekommt es einen Namen sowie eine Beschreibung versehen. Neben den vorgeschriebenen Parametern, welche mit jedem Event festgehalten werden, können auch benutzerdefinierte Parameter für das Event festgelegt werden. Zum Beispiel wurde für das Event *CollectedPointOfInterest* drei spezielle Parameter festgelegt.

- Der Name des jeweiligen Points
- Wie viel Zeit im Spiel bereits vergangen ist
- Wie viel Zeit seit dem letzten Einsammeln eines Points vergangen ist

Das Event wird jedes Mal ausgelöst, wenn der Spieler einen Point Of Interest einsammelt. Die spezifizierten Parameter werden dabei als ein `Dictionary<string, object>` übergeben. Zu beachten ist die genaue Übereinstimmung aller String-Werte. Das `object` hält den ermittelten Wert. Zusammengesetzt ergibt das den Inhalt folgender Methode:

Listing 4.1: Beispiel für den Aufbau eines UAS Events

```
1 public void LogPointOfInterest(string pointName){
2     var eventData = new Dictionary<string, object>{
3         {"PointOfInterest", pointName},
4         {"TimeBetween", TimeOfCollection()},
5         {"CollectionTime", collectionTime},
6     };
7     AnalyticsService.Instance.CustomData(
8         "CollectedPointOfInterest", eventData);
9 }
```

Dieser Aufbau wurde analog für jedes benutzerdefinierte Event verwendet. Events, welche keine zusätzlichen Daten benötigen, wurden ebenfalls mit der Methode

`AnalyticsService.Instance.CustomData()`; registriert. Dabei wird lediglich das erste Argument, der Name des Events, übergeben.

4.2. Erstellung der Spielwelt

4.2.1. Terrain

Zunächst wurde der Aufbau der Spielwelt digital skizziert (siehe A.1). Basierend auf dieser Skizze begann die Arbeit innerhalb der Engine auf Grundlage des Unity Terrain-Tools. Zunächst wurde eine Gruppe an flachen Terrains erstellt. Anschließend konnte mit den Brushes des Terrain-Tools gearbeitet und der Welt Form gegeben werden.

Zuerst wurde mit der `Set Height`-Brush gearbeitet. Da *Explorer Bot* einen höhergelegenen Bereich enthält, wurde das obere Drittel des Terrains einheitlich angehoben. Um den Fluss, der durch die Welt fließt, zu berücksichtigen, wurde das gesamte Terrain ein weiteres Mal erhöht. Daraufhin wurde der Flusspfad eingezeichnet. Der Aufstieg zu dem oberen Plateau wurde zunächst mit dem gleichen Pinsel als eine Treppe geformt. Nachfolgend wurde mit den Pinseln `Smooth Height` sowie `Raise or Lower Terrain` gearbeitet, um den Aufstieg abzurunden. Letzterer Pinsel wurde zusätzlich verwendet, um dem Untergrund eine schwache dreidimensionale Textur zu geben. Somit wirkte die Welt natürlicher. Letzlich wurden dem Boden passende Farben verliehen, um entsprechend Gras, Sand oder Ähnliches anzudeuten.

4.2.2. Assets

Sämtliche Assets, welche für die Darstellung der Natur von *Explorer Bot* verwendet wurden, stammen aus dem *Low Poly Nature Asset Pack* von *BOKI*. Die Flora konnte mithilfe des Terrain-Tools platziert werden. Unter dem Reiter `Paint Trees` beziehungsweise `Paint Details` konnten Bäume sowie Gras und Blumen in die Welt gezeichnet werden. Durch die verschiedenen Pinselformen und -einstellungen konnten sehr natürlich aussehende Blumenfelder erstellt werden.

Die Bergkette sowie alle anderen Steine wurden per Hand in die Welt gesetzt. Neben den 3D-Modellen bot das Asset Pack ebenfalls einen Shader für Wasser sowie Partikelsysteme zum Emulieren eines Wasserfalls an. Letzteres wurde leicht verändert, um besser in den Stil des Spieles zu passen. Der Wasserfall besteht lediglich aus einem Mesh, welches anhand eines Shaders die Illusion fließendem Wassers verliehen wird. Aufgrund der Tatsache, dass die beiden Ebenen von *Explorer Bot* visuell sehr unterschiedlich sind, und daher verschiedene Farben für das Wasser vorzeigen, musste der Shader erweitert werden. Durch die neue Funktion erhält der Wasserfall farblich einen Gradient über zwei Farben. Dies verbindet die Flüsse beider Ebenen. Der Shader wurde in Unitys Shadergraph zusammengestellt. Zum Erreichen des Gradienten wurden die Farben über die UVs des Meshes mithilfe der `Lerp`-Funktion gemischt.

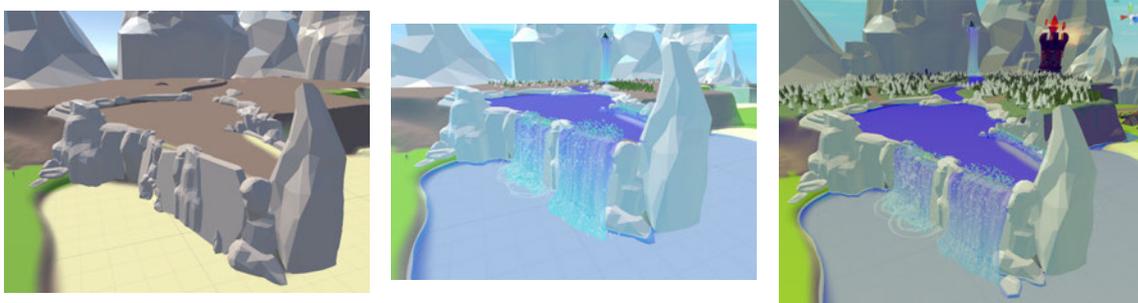


Abbildung 4.1.: Entwicklungsprozess des Wasserfall Areal

Das Modell des Obelisk wurde bereitgestellt von *JUHWAN*. Damit Spieler erkennen, wann sie einen Obelisk aktivierten, wurde audio-visuelles Feedback implementiert. Unter Benutzung des *LeanTween* Add-Ons wurde den einzelnen Steinsäulen eine Animation gegeben. Zunächst drehen sie sich um ihre eigene Achse. Dem Hauptobelisk wird zusätzlich der Befehl gegeben, sich leicht auf seiner Y-Achse auf und ab zu bewegen. Dies signalisiert dem Spieler, dass die Obelisk übernatürlich sind.

Listing 4.2: Animationsbefehl des Obelisk

```
1 LeanTween.rotateAround(tinyPillars, Vector3.up, 180,  
2     spinSpeed).setOnComplete(InitializeBeam);  
3 LeanTween.rotateAround(mainPillar, Vector3.up, 180,  
4     spinSpeed);  
5 LeanTween.moveLocalY(mainPillar, 1.5f, floatSpeed).
```

```
6 setEaseInOutSine().setLoopPingPong();
```

Diese Animation wird untermalt mit dem Soundeffekt von Steinen, die aneinander reiben. Sobald die Animation beendet ist, wird eine Coroutine gestartet, die das Abspielen weiterer Sounds steuert.

Listing 4.3: Sequentielles Abspielen von Sounds

```
1 IEnumerator Sounds(){
2     _audio.loop = false;
3     _audio.clip = PowerUp;
4     _audio.Play();
5     while(_audio.isPlaying){
6         yield return null;
7     }
8     _audio.clip = ActiveSound;
9     _audio.loop = true;
10    _audio.Play();
11 }
```

Die While-Schleife aus Zeile 5 garantiert das komplette Abspielen des *PowerUp* Sounds, bevor der Audioplayer den nächsten Sound abspielt. Innerhalb dieser Zeit wird als weiteres Feedback ein Visual Effect gestartet. Dieser nimmt die Form eines Laserstrahls an, der auf den Drachen gerichtet ist. *ActiveSound* ist ein tiefes Summen, was den Strahl unterstreicht.

4.2.3. Points of interests

Das Asset der Point Of Interests besteht aus einem visuellen Effect, erstellt mit Unitys *Visual Effects Graph* (VFX-Graph). Der Graph besteht aus 3 Hauptsystemen. Zwei Systeme sind für das Spawnen des leuchtenden Kerns des Punktes verantwortlich. Das Dritte übernimmt die Trails, welche um diesen Kern schwirren.

Das visuelle Feedback des Einsammelns besteht hauptsächlich aus dem Ausschwärmen der Trails. Programmatisch wurde dies über freigelegte Variablen des VFX-Graphs ermöglicht. Über den Graph-Block **Conform to Sphere** ist es möglich, den Trails einen bestimmten Radius zu geben, um welchen sie sich bewegen. Wenn der

Spieler den Bereich des Point Of Interests betritt, werden zwei Variablen verändert. Zuerst wird die Variable `SpawnMult` auf 0 gesetzt. Somit wird verhindert, dass neue Partikel durch den VFX-Graph initialisiert werden. Des Weiteren wird die Variable `TrailMult` um ein Vielfaches erhöht. Im Graphen beeinflusst diese Variable den Radius des `Conform to Sphere`-Blocks. Folglich entfernen die existierenden Trails sich von dem Zentrum des Gameobjects.

4.3. Dialogimplementation

Für das Einarbeiten der Dialoge wurde das Add-On *Yarn Spinner* von Secret Lab verwendet. *Yarn Spinner* übernimmt eine Vielzahl der technischen Notwendigkeiten, die ein Dialogsystem benötigt, um so reaktionsfähig zu sein, wie es in Abschnitt 3.1.4 beschrieben wurde.

Dialoge werden als `.yarn`-Datei gespeichert, können aber in jedem Texteditor geschrieben und bearbeitet werden. Dabei hat jeder alleinstehende Konversationsstrang, auch Node genannt, den gleichen Aufbau, bestehend aus einem Header und Body. An erster Stelle im Header steht der Titel der Konversation. Dieser kann benutzt werden, um von Konversation zu Konversation zu springen. Zusätzlich kann der Header weitere Informationen in der Form von "tags" beinhalten. Diese Funktion fand jedoch keine Verwendung in *Explorer Bot*.

Im Body der Node wird der Dialog geschrieben. Dabei symbolisiert Text auf einer neuen Zeile den Anfang einer neuen Textbox, die dem Spieler präsentiert wird. Ebenso findet die Funktion von *Yarn Spinner* Variablen zu evaluieren hier ihren Nutzen. So kann während des Dialogs eine If-Abfrage über die Lebenspunkte des Spielers gestellt werden und entsprechend reagiert werden. Realisiert wird diese Funktion über *Yarn Spinners In Memory Variable Storage*-Komponente. Diese Komponente ist für das Registrieren neuer Variablen sowie das Aktualisieren bereits existierender Variablen verantwortlich.

Zuletzt wurde ein weiteres Feature von *Yarn Spinner* in *Explorer Bot* verwendet. Aus dem Dialog heraus können Funktionen innerhalb von C#-Skripten ausgeführt werden. Diese Nutzweise kann durch das im letzten Abschnitt erwähnte Beispiel der Lebenspunkte des Spielers beleuchtet werden. Sollte die Anzahl der Lebenspunkte zu niedrig sein, wird während des Dialogs der Befehl `<<Heal Player>>` ausgeführt.

”Player” designiert hier den Namen des GameObjects, auf welchem das Skript mit der ”Heal”-Methode zu finden ist. Die Methode sieht wie folgt aus:

```

1  [YarnCommand("Heal")]
2  public void Heal(){
3      var eventData = new Dictionary<string, object>{
4          {"CurrentHitpoints", _currentHealth},
5          {"Difference", maxHealth - _currentHealth}
6      };
7      AnalyticsService.Instance.
8          CustomData("PlayerHeal", eventData);
9      _currentHealth = maxHealth;
10     _variableStorage.UpdatePlayerHealth(maxHealth);
11     healthBar.UpdateHealthbar(maxHealth, _currentHealth);
12 }

```

Zu beachten ist das Attribut `[YarnCommand("Heal")]`. Es signalisiert, dass genau diese Methode auszuführen ist, wenn im Dialog der `<<Heal>>` Befehl aufgerufen wird. Um festzuhalten, wann sich Spieler heilten, wurde die Eventstruktur aus 4.1 verwendet. Letztlich werden die aktuellen Lebenspunkte auf den Wert der maximalen Lebenspunkte gesetzt und das UI, in Zeile 11, sowie das `In Memory Variable Storage`, in Zeile 10, werden aktualisiert.

4.4. Quests

Programmatisch arbeiten Quests eng mit dem Dialogsystem zusammen. Neben dem Akzeptieren von Quests während eines Dialogs werden auch Informationen in beide Richtungen aktualisiert. Wie in Abschnitt 3.1.4 beschrieben, reagieren NPCs dynamisch in Bezug auf den Status der Quest.

4.4.1. Pilz Quest

So kann die Quest des Pilzesammelns auch nur über den Dialog mit der Dorfältesten abgeschlossen werden. Für den Abschluss spielen zwei Teile eine Rolle. Die Rolle des `Mushroom Quest Manager`-Skripts ist es festzuhalten, wann ein Pilz aufgesammelt

wurde, und wie viele Pilze der Spieler insgesamt besitzt. Die Rolle des Dialogs ist es, die Logik zu verarbeiten, ob der Spieler zum Zeitpunkt der Unterhaltung genug Pilze mit sich trägt. Der Dialog wurde wie folgt geschrieben:

Listing 4.4: Body einer *Yarn Spinner* Node

```
1 <<if $MushroomsCollected <= 0>>
2 Elder: Would you help me collect some mushrooms [...]
3 <<elseif $MushroomsCollected < 5>>
4 <<set $MushroomsNeeded = 5 - $MushroomsCollected>>
5 Elder: Ohhh you already collected some mushrooms, thanks!
6     I only need {$MushroomsNeeded} more.
7 <<else>>
8 Elder: You've brought me 5 mushrooms! Thank you!
9 <<finishQuest Forest>>
10 <<endif>>
```

Die Evaluationslogik besteht aus mehreren If-Abfragen. Je nach Anzahl der gesammelten Pilze werden dem Spieler andere Zeilen des Dialogs gezeigt. Für den Fall, dass der Spieler bereits Pilze gesammelt hat, aber noch nicht genug, wird ihm eine besondere Dialogzeile präsentiert. In Zeile vier wird eine neue Variable deklariert, welche die Differenz der benötigten und schon gesammelten Pilze wiedergibt. Im darauffolgenden Dialog kann diese Variable ohne Probleme eingesetzt werden. Dem Spieler wird anschließend der numerische Wert angezeigt.

Nur wenn der Spieler fünf oder mehr Pilze gesammelt hat und die Dorfälteste anspricht, wird der letzte Fall des Dialogs erreicht. Nur hier wird der Befehl in Zeile neun ausgeführt und somit die Quest abgeschlossen.

4.4.2. Farm Quest

In der Farm Quest erfährt die Kopplung mit dem Dialogsystem ebenso eine Verbindung mit der Datensammlung der UAS. Ziel der Quest bestand aus dem Besiegen jeglicher Monster auf dem Gelände der Farm. Dabei existieren die Gameobjects der Gegner als Kinder des **Farm Quest Managers**. Das verbundene Skript überprüft, wie viele Kinder vorhanden sind. Wenn die Anzahl der Kinder null erreicht, hat der Spieler alle erforderlichen Gegner besiegt. Anschließend wird folgende Methode aufgerufen:

```

1 public void FinishQuest() {
2     _variableStorage.variableStorage.TryGetValue(
3         "$questReady", out questReady);
4     questFulfilled = true;
5     _variableStorage.UpdateFarmQuest(questFulfilled);
6     var eventData = new Dictionary<string, object> {
7         {"AcceptedQuestBeforehand", questReady}
8     };
9     AnalyticsService.Instance.CustomData(
10        "FarmQuest", eventData);
11 }

```

In Zeile zwei wird der Wert einer Variable aus dem Speicher der Dialog-Variablen gelesen. Die Variable `$questReady` beschreibt hierbei, ob der Spieler vorher mit der Schmiedin geredet hat. Diese Information wird anschließend mit dem `FarmQuest`-Event verschickt. Somit wird Rücksichtnahme auf den Fall, dass ein Spieler, unabhängig einer gestellten Aufgabe, das Schwert findet und die Farm betritt, genommen. Dadurch wird die freie Form von *Explorer Bot* aktiv unterstützt.

4.4.3. Drachen Quest

Im Zusammenhang mit aktivierten Obelisken wurde ebenfalls ein komplexer Dialogbaum mit der Magierin erstellt. Dieser Abschnitt wird sich jedoch auf die anderen Systeme der Quest beziehen. Anders als bei der Pilz- und Farmquest gab es keinen zentralen Quest-Manager. Die Aufgaben der Evaluation und der entsprechenden Datensammlung wurden auf den Drachen und die Obelisken aufgeteilt.

Jeder Obelisk hat zwei Aufgaben. Einerseits das unter Abschnitt 4.2.2 beschriebene audio-visuelle Feedback. Andererseits die Registratur der eigenen Aktivierung. Dieser Vorgang wird über zwei Events geregelt, da ein aktivierter Obelisk in zwei Stellen aufgenommen wird. Zuerst sendet ein Obelisk ein Event an die UAS. Als benutzerdefinierter Parameter wird der Name des Obelisk mitgesendet. Das zweite Event, `OnActivation`, wird innerhalb *Explorer Bots* ausgestrahlt. Empfänger dieses Events ist der Drache.

Der Drache ist ein besonderer Gegner. Im Verlauf der Quest kann er den speziellen Zustand "schlafend" einnehmen. Um diesen Zustand zu erreichen, müssen alle Obelisken aktiviert sein. Zunächst wird in der `Start()`-Methode des Drachen die benötigte Anzahl an Obelisken ermittelt. Dies geschieht durch den Befehl `GameObject.FindGameObjectsWithTag("Obelisk").Length`; In der Engine wurde jedes Obelisk-Objekt mit dem Tag "Obelisk" versehen. Dementsprechend ist das Array, welches `FindGameObjectsWithTag()` zurückgibt, gefüllt mit jedlichen Obelisken in der Szene. Dieser Weg ist praktischer, als das simple Festlegen einer Anzahl, da er besonders hilfreich ist für das Hinzufügen neuer Obelisken. Der Nachteil einer längeren Abwicklungszeit durch den Aufruf der imperformanten Methode ist in *Explorer Bot* nicht zu bemerken.

Das `OnActivation`-Event des Obelisk ist verbunden mit der `ReigsterObelisk()`-Methode.

```
1 void RegisterObelisk(string obelisk_name){
2     activatedObelisks.Add(obelisk_name);
3     if(activatedObelisks.Count == _neededObelisks
4         && !pacified){
5         AnalyticsService.Instance.CustomData(
6             "DragonPacified");
7         Weaken();
8     }
9 }
```

Diese Methode fügt den aktivierten Obelisken der Liste `activatedObelisks` hinzu. Danach wird geprüft, ob die Länge der Liste der Anzahl der benötigten Obelisken entspricht. Insofern dies stimmt und der Drache nicht bereits zum Schlafen gebracht wurde, wird ein UAS-Event für die Pazifizierung des Drachen gesendet. Letztlich wird die `Weak()`-Methode aufgerufen. Sie übernimmt alle Funktionen, um den Drachen als eingeschlafen zu erkennen.

Durch die nicht-finale Natur dieser Quest gibt es keinen einzelnen "Quest abgeschlossen"-Status. Stattdessen werden zwei Booleans für die Bestimmung der Dialoge festgelegt. Die zwei Variablen heißen jeweils `$dragonAsleep` und `$dragonDead`. Da der Drache nach seinem Tod nicht mehr als schlafend angesehen werden kann, wird in entsprechenden Dialogen immer zuerst nach dem Wert von `$dragonDead` gefragt. Im Wahrheitsfall springt der Dialog in eine spezielle Node. Damit wird umgangen,

dass zwei sich widersprechende Dialogstränge abgespielt werden, sollten beide Werte "wahr" betragen.

5. Evaluation

Ziel dieses Kapitel ist es, die Forschungsfrage zu beantworten. Bartles Taxonomie gilt als validiert, wenn die in Kapitel 3.1 beschriebenen Methoden zu einem Spiel führen, welches den Spielern messbar gefällt. Zudem soll das Verhalten der Probanden innerhalb des Spiels untersucht werden. Die Analyse soll hervorbringen, ob Verhaltensmuster, analog zu Bartles Archetypen, zu erkennen sind.

Jegliche Daten wurden im Zeitraum vom 24. November bis 26. Dezember 2023 gesammelt.

Die freiwillige Google Umfrage wurde von elf Personen beantwortet. Im Experimentationszeitraum wurde das Spiel von 13 Probanden gespielt (siehe B.1). Innerhalb dieser Gruppe befinden sich jedoch zwei Nutzer, welche keine nennenswerten Aktionen in *Explorer Bot* ausführten. Folglich bezieht die Archetyp-Analyse ebenfalls auf elf Personen.

5.1. Auswertung der Umfrage

Die Fragen und Antworten der Umfrage beschrieben in B.12, B.13 und B.14 beziehen sich auf das allgemeine Spaßempfinden der Kerninhalte von *Explorer Bot*. Mit jeweils elf Antworten ist die höchste Punktzahl, die erreicht werden kann, 55. Die erreichten Summen sind jeweils 43($\approx 78\%$) für das Sammeln von Collectibles in der Welt, 31($\approx 56\%$) für das Kämpfen mit Gegnern sowie 41($\approx 75\%$) für das Erledigen von Quests.

Die niedrige Summe in Bezug auf die Gegner hat zwei Quellen. In B.18 wurden die ausführlichen Antworten auf diese Facette des Spiels mit den jeweilig verliehenen Punkten gegenübergestellt. In den Fließtextantworten sieht man, dass vier Probanden (in der Abbildung rot hinterlegt) allgemein nicht gerne in Spielen kämpfen. Des Weiteren gibt es drei Personen, welche Combat zwar mögen, aber die Exekution

der Mechanik innerhalb von *Explorer Bot* als mangelhaft empfanden (in der Abbildung gelb hinterlegt). Im Bezug auf die Archetypenbestimmung kann dies zu einem verminderten Auftreten des Killer-Typs führen.

Aus B.17 und B.19 geht hervor, dass die restlichen zwei Hauptaspekte des Spiels generell positiv empfunden wurden. Vor allem das Sammeln der Point Of Interests wurde gut aufgenommen. Mit nur einer Person, der explizit die Welt von *Explorer Bot* nicht gefiel.

Währenddessen wurde dem Gesichtspunkt der Quests zwar eine hohe Punktzahl gegeben, die detaillierteren Antworten geben jedoch eine gemischtere Impression wider. Ursache dafür kann die Auslage der Quests auf die spezifischen Archetypen sein. Die unterschiedlichen Quests sprechen verschiedene Archetypen an, was zu einem gemischten Ergebnis führt.

Die Ergebnisse der Umfrage deuten auf ein generelles Vergnügen mit *Explorer Bot*. Das Arbeiten nach Bartles Modell sorgte für ein Spiel, welches mehrere Leute auf verschiedene Ebenen ansprach. Ungeachtet der Existenz einzelner Archetypen innerhalb der Spielerbasis boten Bartles Player Suits ein Gerüst, an welchem Spielinhalte konzipiert und entwickelt werden konnten. Somit gilt das Modell, zumindest theoretisch angewendet, als validiert.

5.2. Bestimmung der Archetypen

Die gesammelten Daten des UAS wurden per SQL-Abfragen aus der Cloud-Datenbank abgerufen. Anschließend wurden sie mit dem Programm Libre Office Calc¹ verarbeitet.

Um den elf Spielern einen Archetyp zuschreiben zu können, wurde eine Metrik erstellt. Diese Metrik besteht aus den aufgezeichneten Aktivitäten des Spiels, welche jeweils mit den zutreffenden Archetypen und einer Punktzahl versehen wurden. Je größer die Punktzahl, desto aussagekräftiger ist die einzelne Aktion für den dazugehörigen Archetyp. So gab es zum Beispiel für jeden Spieler einen Punkt für den Killer Archetyp für jeden Gegner, der besiegt wurde (Abb.: B.6). Wenn ein Spieler den Drachen besiegte, sogar zehn Punkte. Für das Besiegen der meisten Gegner

¹Libre Office Calc entwickelt von The Document Foundation <https://de.libreoffice.org/discover/calc/>

wurden ebenfalls Bonuspunkte an die Spieler verliehen.

Bei Socializern galt als am aussagekräftigsten die Anzahl, wie oft ein Spieler mit NPCs geredet hat (Abb.: B.5), sowie mit wie vielen unterschiedlichen NPCs gesprochen wurde (Abb.: B.7). Für jeden Eintrag dieser Events gab es jeweils einen Punkt. Für das Führen einer Konversation mit dem NPC, der am wenigsten angesprochen wurde, gab es einen Bonuspunkt. (NPC "LaundryGuy" B.8)

Für Achiever sowie Explorer wurden die gesammelten Point Of Interests gezählt (Abb.: B.9). Wie in 3.1.2 beschrieben, gelten POIs als Inhalt für beide Archetypen. Des Weiteren kann von den gefundenen POIs abgeleitet werden, ob ein Spieler sich die Zeit genommen hat, mit der Welt zu interagieren, um so viel wie möglich von ihr zu entdecken. Im gleichen Zuge wurde die Spielzeit als ein Indikator für Explorer benutzt (Abb.: B.10). Da *Explorer Bot* keine feste Spielzeit oder einen Countdown besitzt, wurde von jedem Spieler die Gesamtspielzeit kalkuliert. Anschließend wurden die ermittelten Werte auf einen Bereich von 0 zu 1 normalisiert. Die längste Spielzeit erhielt den Wert von 1.

Die Quest um den Drachen und die Obelisken nahm, wie angenommen, die facettenreichste Ausführung an. Es wurden 4 Interaktionen bestimmt, welche jeweils mit mehreren Archetypen assoziiert sind. Da die verschiedenen Aktionen jedoch nicht alle gleich demonstrativ für die verbundenen Archetypen sind, wurden unterschiedliche Punktzahlen pro Archetyp vergeben.

Interaktion	Assoziierte Archetypen	Vergebene Punktzahl pro Archetyp
Drachen töten	Killer, Achiever, Explorer	10, 5, 2
Tod durch Drachen	Explorer, Achiever, Killer	10, 2, 2
Einschläfern des Drachen	Achiever, Explorer, Socializer	10, 5, 2
Nicht-Tötung des Drachen nach Einschläferung	Socializer	10

Tabelle 5.1.: Schlüsselinteraktionen mit dem Drachen und der verbundene Archetyp

Der zuerst genannte Archetyp ist am stärksten mit der Interaktion verbunden. Für eine Übersicht, welcher Spieler welche Aktionen vollzog, dient das Diagramm B.4.

Ähnlich wurde für zahlreiche andere Bereiche des Spiels vorgegangen, um so viele Aktionen der Spieler wie möglich zu analysieren. Die finalen Archetypen wurden

ermittelt, indem die erreichte Punktezah als prozentualer Anteil der maximal zu erreichenden Punkte berechnet wurde. Dies führte zu folgenden Werten:

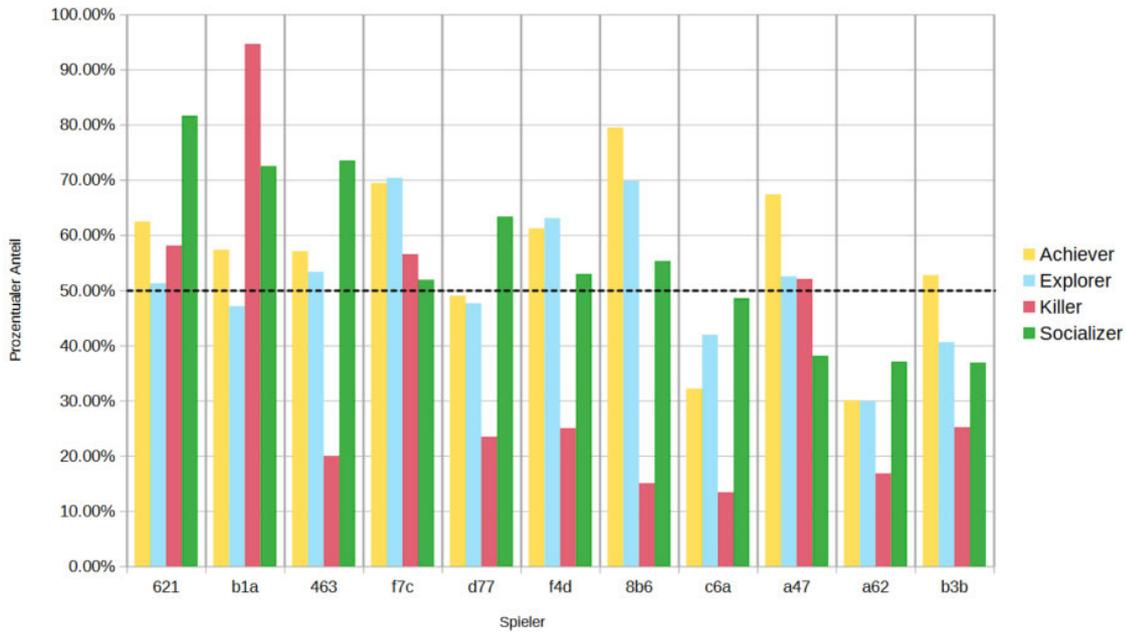


Abbildung 5.1.: Erzielter Anteil der maximal erreichten Punktzahl pro Archetyp pro Spieler

Archetyp	Spieler
Achiever	8b6, a47, b3b
Explorer	f4d, f7c
Killer	b1a
Socializer	621, 463, d77

Tabelle 5.2.: Verteilung der Spieler auf die vier Archetypen

Die Spieler, abgebildet auf der X-Achse, wurden nach den ersten drei Zeichen ihrer jeweiligen Unity-ID benannt. Die Maximale Punktzahl betrug gerundet jeweils für Achiever 67, Explorer 76, Killer 61 und Socializer 88. Insgesamt existieren 23 Kategorien, von welchen Spieler Punkte für einen oder mehrer Archetypen erzielen konnten. Abbildung B.11 zeigt die erreichten Punktzahlen in Ganzzahlen sowie prozentualen Anteilen. Tabelle 5.2 liefert einen zusätzlichen Überblick, welche Spieler sich für welchen Archetypen qualifiziert haben.

Direkt zu erkennen ist die grobe Verteilung der Archetypen. Achiever und Socializer bilden die zwei populiertesten Archetypen mit jeweils drei von elf Spielern. Die Spieler mit den IDs 8b6, a47 und b3b befassten sich während ihrer Spielzeit am

meisten mit Achiever-Inhalten. Der Archetyp des Socializers wurde erreicht von den Spielern 621, 463, d77. Explorer sind zweimal vertreten durch Probanden f4d und f7c. Spieler b1a ist der einzige des Killer-Archetyps. Schließlich ist zu sehen, dass zwei Spieler, c6a und a62, in keinem Archetypen eine Punktzahl von über 50% erzielten. Die niedrige Punktzahl ist indikativ für ein fehlendes Auseinandersetzen mit dem Spiel. Dadurch verlieren die aufgezeichneten Aktionen an Authentizität und es wurde sich dafür entschieden, diesen Spielern keinen Archetypen zuzuordnen.

Die große Anzahl an Socializern stimmt mit Bartles These überein, dass in der Allgemeinheit die Menge an Socializer die anderen Archetypen übertrifft[Bar03, S. 182]. Für den speziellen Fall unter *Explorer Bot* war dieses Ergebnis dennoch überraschend. Socializer legen den größten Wert auf die Interaktion mit anderen Spielern. Folglich war davon auszugehen, dass sie in einem Spiel ohne andere Spieler sich nicht als Socializer ausleben können. In Diagramm B.5 kann man erkennen, dass durchschnittlich ≈ 4 mal mit jedem der zehn NPCs geredet wurde. Sogar nach Ausschluss der Quest-NPCs, mit welchen Spieler mehrere Male reden müssen, bleibt der Durchschnitt über zwei Konversationen pro NPC. *Explorer Bot* gibt keine Indikation, dass normale NPCs ihren Dialog wechseln. Diese Durchschnittswerte zeigen demnach, dass Spieler generell den Kontakt zu NPCs suchen. Für Spielentwickler stellt dies einen klaren Fokuspunkt dar. Spieler suchen erstellte Inhalte in Bezug auf NPCs schnell und leicht auf. Die darin investierte Arbeit wird somit in mehr Spielzeit und Spielvergnügen umgewandelt.

Wie im vorangegangenen Kapitel 5.1 erwähnt, empfanden viele der Spieler das Kampfsystem in *Explorer Bot* als ungenügend. Diese Empfindung spiegelt sich in den gemessenen Werten wieder. In über der Hälfte aller Probanden manifestierte sich der Killer-Archetyp als derjenige Archetyp mit der niedrigsten Punktzahl. Dennoch gab es einen Spieler, welcher mit Zuversicht dem Killer zugeschrieben werden kann. Dieser Fakt beleuchtet die Notwendigkeit für Entwickler, ein manigfaltiges Spiel zu kreieren. Selbst in einer geringen Spieleranzahl existiert genug Diversität, um ein breites Spektrum an Inhalten zu fordern. Ebenfalls wird die zeitintensive Erweiterung und Verbesserung dieser Systeme gerechtfertigt.

Die Mehrheit an Achievern war ein zu erwartendes Ergebnis. *Explorer Bot* ist ein Einzelspieler-Spiel. Dies allein lädt Spieler vom Typ Achiever und Explorer ein. Zudem sind Achiever der Archetyp, welcher am meisten Wert auf die Elemente einer virtuellen Welt legt, welche sich am meisten wie ein Spiel anfühlen[Bar03, S.

173]. Das bedeutet, dass in dem Großteil aller Spiele Achiever positiv angesprochen werden. Somit ist von einer Vielzahl von Achievern unter Spielern als Ganzes auszugehen. Diese Behauptung wird in den Ergebnissen widerspiegelt.

Explorer sind der zweite Archetyp, der sich mehr mit der Welt als anderen Spielern befasst. Wie Achiever erfahren sie keinen Nachteil in einem Einzelspieler-Spiel. Zudem unterstützt die Grundstruktur *Explorer Bots*, wie in Kapitel 3.1.2 beschrieben, die Spielweise von Explorern. Folglich liegt bei zwei von elf Spielern die Anzahl von Explorern niedriger als erwartet. Dies kann einmal zu Stande kommen, durch den kleinen Probenumfang des Experiments. Zum anderen kann es daran liegen, dass *Explorer Bot* nicht genügend aktiven Inhalt für Explorer bietet. Folglich verwirklichen sich Spieler, welche eigentlich Explorer sind, durch anderweitig gekennzeichnete Inhalte. Bartle nannte das Bedürfnis des Erkundens der Tiefe[Bar96, S.3] eines Systems als genauso wichtig für Explorer wie die Erkundung der Breite. Die offene Welt von *Explorer Bot* bedient zwar die Erkundung in die Breite, bietet aber nicht viele Möglichkeiten für die Erkundung in die Tiefe. Wie zum Beispiel das Experimentieren mit komplexeren Systemen wie das einer simulierten Physik.

Einen Einblick in die niedrige Anzahl von Explorern können die anderen zwei Quests geben. Die Farmquest, anders als erwartet, wurde sehr homogen durchgeführt. Jeder Spieler sprach zuerst mit der Schmiedin, bevor er anfang, die Gegner auf der Farm zu besiegen(Abb.: B.3). Dieser Fund kann mit der niedrigen Anzahl an Explorern innerhalb der untersuchten Spieler korrelieren.

Ebenso erwies sich die Pilzquest als wenig aufschlussreich. Durch einen unbemerkten Fehler in der Datensammlung konnte nicht festgestellt werden, ob Spieler vor der Konversation mit der Dorfältesten bereits Pilze gesammelt hatten oder nicht. Dieser Fehler kann dazu geführt haben, dass Spieler fälschlicherweise nicht als Explorer erkannt werden konnten. Jedoch wurden bei keinem Spieler genug Punkte für den Explorer Archetyp gesammelt, sodass dieser Archetyp nur knapp den zweiten anstatt den ersten Platz erreichte. Demnach würde sich höchstwahrscheinlich nichts an der finalen Einteilung ändern.

Schlussendlich gibt das Abschließen der Quests ebenfalls nur wenig Information bekannt, da jeder Spieler alle Quests abschließ(Abb.:B.2). Lediglich Spieler c6a und a62 setzten sich nicht tiefer mit der Drachenquest auseinander. Das Abschließen so ziemlich jeder Quest deutet auf eine Homogenität in den Spielern verschiedener Archetypen hin. Dies ist eine wichtige Erkenntnis für Spieleentwickler. Wenn jeder

Spieler ungeachtet des Archetyps dazu geneigt ist, die ihm präsentierten Quests zu erledigen, sollten diese Elemente des Spiels jeden Archetyp so gut wie möglich ansprechen.

Alle vier Spielerarchetypen kristallisierten sich aus den gemessenen Daten. Folglich bewährt sich Bartles Modell als theoretische Grundlage von *Explorer Bot*. Die Inklusion von archetypspezifischen Elementen produzierte distinktives Spielerverhalten. Für Entwickler, welche für einen größtmöglichen Markt produzieren wollen, ist diese Erkenntnis wertvoll. Ein Spiel, welches einen großen Fokus auf Socializer-Inhalte legt, kann weiterhin Spieler anderer Archetypen abholen. In 5.1 kommt zum Vorschein, dass die Mehrheit der Spieler sich mit mehr als 50% aller Inhalte befasste. Bartle argumentiert für einen einzelnen, primären Archetyp für Spieler eines MUDs[Bar96, S.3]. In einem Spiel fern von der MUD-Umgebung sieht dies anders aus. In einer finiten Spielumgebung leben Spieler eine Mischung der Archetypen aus.

Bartles Modell beweist Nutzen in der Praxis. Nur liegt der Fokus bei Einzelspielerspielen weniger auf den Dynamiken zwischen den Archetypen. Sondern mehr auf die Aussagen der Motivationsgründe einzelner Spieler.

6. Zusammenfassung und Ausblick

Die Untersuchung einer über 20 Jahre alten Theorie scheint auf den ersten Blick überflüssig. Nach Bartle wurden eine Vielzahl von Spielertaxonomien aufgestellt. Jedoch, während andere Modelle sich mit der reinen Kategorisierung von Spielern beschäftigen, versucht Bartles Modell direkt Spielentwicklern zu helfen. Die Player Suits nutzen die Archetypen, um die Bedürfnisse und Auswirkungen verschiedener Spieler zu kommunizieren. Somit wird es zu einer wertvollen Ressource für Spieleentwickler. Damit dieser Wert bestehen bleibt, benötigt es stetige Untersuchung.

In Kapitel 2 wurde Bartles Modell vorgestellt. Die vier Spielerarchetypen und ihre jeweiligen Bedürfnisse werden erläutert. Ebenso werden sie in Kontext zum Game-Design schon vorhandener Spiele gestellt. Der spielerische Kontext, für welchen das Modell erstellt wurde, wird ebenfalls beleuchtet. MUDs sind virtuelle Welten, bespielt von mehreren Spielern. Diese Beschaffenheit rief die Dynamiken unter den Spielern hervor, welche schlussendlich zu Bartles Modell führten.

Unter dem Aspekt der Archetypen wird in Kapitel 3 das Game-Design von *Explorer Bot* analysiert. Das Modell dient als die theoretische Grundlage, auf welcher die Inhalte des Spiels erschaffen wurden. Es wird ausführlich darüber geschrieben, welche Wirkung spezifische Spielelemente auf die jeweiligen Archetypen haben sollen.

Anschließend wird in Kapitel 4 die Erstellung *Explorer Bots* beschrieben. Wie in Kapitel 2 erwähnt war die Umgebung des MUDs ausschlaggebend für das Auftreten der Spielerarchetypen. Daher wurde in *Explorer Bot* Wert darauf gelegt, eine möglichst lebendige Welt zu erschaffen. Ohne ein MUD zu rekreieren, wurde die Atmosphäre eines Mehrspieler-Spiels approximiert. Durch sich verändernde Dialoge wurde den NPCs des Spiels mehr Leben eingehaucht. Auch während der Quests reagierten sie deatilreich auf die Aktionen der Spieler.

Die Auswertung dieser Spieleraktionen findet in Kapitel 5 statt. Diese Analyse bildet einen von zwei Punkten, unter welchem Bartles Modell letztendlich validiert wurde. Der zweite Punkt bilden Meinungen der Spieler über Elemente von *Explorer Bot*.

Die positive Rezeption des Spiels, gekoppelt mit der Präsenz aller vier Archetypen unter den Probanden, beweist den Nutzen von Bartles Modell.

Die Ermittlung der Archetypen brachte eine wichtige Erkenntnis zum Vorschein: Selbst in Spielen mit einem finiten Inhalt finden Spieler Wege, sich auf grundlegend unterschiedliche Weisen zu verwirklichen. Für Entwickler ist das Ausmaß dieser Wege unmöglich vorher zu wissen. Deshalb sind Werkzeuge wie das Player Suits Modell unentbehrlich. Sie geben eine grobe, dennoch weitreichende Einsicht in die Interessen möglicher Spieler.

Die Evaluation bemerkte, wie Archetypen anfangen, weniger eindeutig für den einzelnen Spieler aufzutreten. Eine zukünftige Studie könnte diesen Fund weiter untersuchen. Eine mögliche Erkenntnis wäre, dass Spieler eines Spiels mit begrenztem Inhalt nicht nur archetypspezifische Inhalte aufsuchen. Über ihre Spielzeit werden sie den gesamten Inhalt des Spiels erfahren. Ein Spieler würde somit nicht mehr nur in einen Archetypen fallen. Folglich müsste Bartles Modell anders angewendet werden.

Zum jetzigen Zeitpunkt kann gesagt werden, dass Bartles Modell der Player Suits weiterhin effizient in der Spielentwicklung genutzt werden kann. Dabei deuten die Ergebnisse dieser Arbeit auf einen effektiveren Nutzen des Modells als theoretische Grundlage für Spielinhalte.

Literaturverzeichnis

- [Bar96] Richard Bartle: *Hearts, clubs, diamonds, spades: Players who suit MUDs*, *Journal of MUD research*, Bd. 1(1):S. 19, 1996.
- [Bar99] Richard Bartle: *MUD Writings Archive*, 01 1999, URL: <https://mud.co.uk/richard/academic.htm>, besucht am 15.01.2024.
- [Bar03] Richard A Bartle: *Designing virtual worlds*, New Riders, 7 2003, ISBN 0-13-101816-7.
- [Bar09] Richard Bartle: *Understanding the limits of theory*, 2009.
- [Bar12a] Richard Bartle: *On Player Types*, Bd. 3, Kap. 4, S. 89–90, Delmar Cengage Learning, 2012.
- [Bar12b] Richard Bartle: *Player Type Theory: Uses and Abuses*, 2012, URL: <https://www.youtube.com/watch?v=ZIzLbE-93nc>, abgerufen am 27.11.2023.
- [Cam04] Joseph Campbell: *The Hero With A Thousand Faces*, Princeton University Press, 41 William Street, Princeton, New Jersey 08540, commemorative Aufl., 2004, ISBN 0691119244.
- [Cor08] British Broadcasting Corporation: *Stepping into a World of hurt*, 9 2008, URL: <http://news.bbc.co.uk/2/hi/technology/7618626.stm>, besucht am 23.11.2023.
- [Mik08] Mike: *What is GoPets?*, 3 2008, archiviert am 01.05.2010 unter URL: <https://web.archive.org/web/20100501070354/http://www.virtualworlddigest.com/what-is-gopets/>, abgerufen am 23.11.2023, originale URL: <https://www.virtualworlddigest.com/what-is-gopets/>.
- [Mor06] Torill Elvira Mortensen: *WoW is the new MUD: Social gaming from text to video*, *Games and Culture*, Bd. 1(4):S. 397–413, 2006.

- [Stu14] Keith Stuart: *Richard Bartle: we invented multiplayer games as a political gesture*, 11 2014, URL: <https://www.theguardian.com/technology/2014/nov/17/richard-bartle-multiplayer-games-political-gesture>, besucht am 15.01.2024.
- [vM07] Richard L. M. van Meurs: *How to play the game? A study on MUD player Types and their real life personality traits*, Diplomarbeit, Tilburg University, Warandelaan 2 5037 AB Tilburg The Netherlands, 8 2007.
- [Yee02] Nicholas Yee: *Facets: 5 Motivation Factors for Why People Play MMORPG's*, 3 2002, URL: <http://www.nickyee.com/facets/home.html>, besucht am 27.11.2023.
- [Yee05] Nicholas Yee: *Motivations of Play in MMORPGs*, 11 2005, URL: <http://www.nickyee.com/daedalus/archives/pdf/3-2.pdf>, besucht am 03.12.2023.

Anhang

A. Konzeptionsdokumente

Tabelle A.1.: Liste der evaluierten Fragen der Google Umfrage

Frage	Antwortmöglichkeit
The world was filled with colorful, floating orbs. Did you like looking for them?	Punktvergabe 1 bis 5
Did you like exploring the game's world? why or why not? How about in other games	Fließtext
Did you like battling the enemies?	Punktvergabe 1 bis 5
Do you like doing combat in other games? why or why not?	Fließtext
Did you enjoy doing the game's quests?	Punktvergabe 1 bis 5
What quests in Explorer Bot did you complete?	Checkboxen
Any thoughts on the quests in Explorer bot or quests in general?	Fließtext
Did you talk to NPCs outside of quests?	Ja/Nein
How did you decide what to do in the game?	Fließtext
Any other things in the game that stood out to you or thoughts you want to share?	Fließtext

Tabelle A.2.: Liste der registrierten Unity Analytics Services Events

Event Name	Beschreibung	Parameter
EnemyKill	The player killed an enemy	name of killed enemy
CollectedPointOfInterest	The player has collected one of the POI in the world	name of POI, collectionTime, time since last POI collection
FarmQuset	Event is received when the player killed all the enemies around the farm	information on whether the player spoke to the quest-giving NPC
EngagedNPCdialogue	Fired when the player starts dialogue with the npc	name of the NPC
Obelisk	the activation of an obelisk in the game	name of the obelisk
DragonPacified	Event that showcases when a player has pacified the dragon	n/a
PlayerDeath	Event to count how many times a player died and by what enemy	name of enemy that killed the player
PlayerHeal	Event to track when the player heals themselves	current hitpoints at healing, difference to max hitpoints
PlayerSave	Event that tracks when the player saves	save data as JSON string
PlayerLoad	Event that tracks when the player loads their save	n/a
MushroomQuestDone	Event that tracks the Completion of the mushroom quest	n/a
PikedUpItem	tracks when the player picks up an item	name of the item



Abbildung A.1.: Erste Skizze des Layouts von *ExplorerBot*

B. Evaluationsdokumente

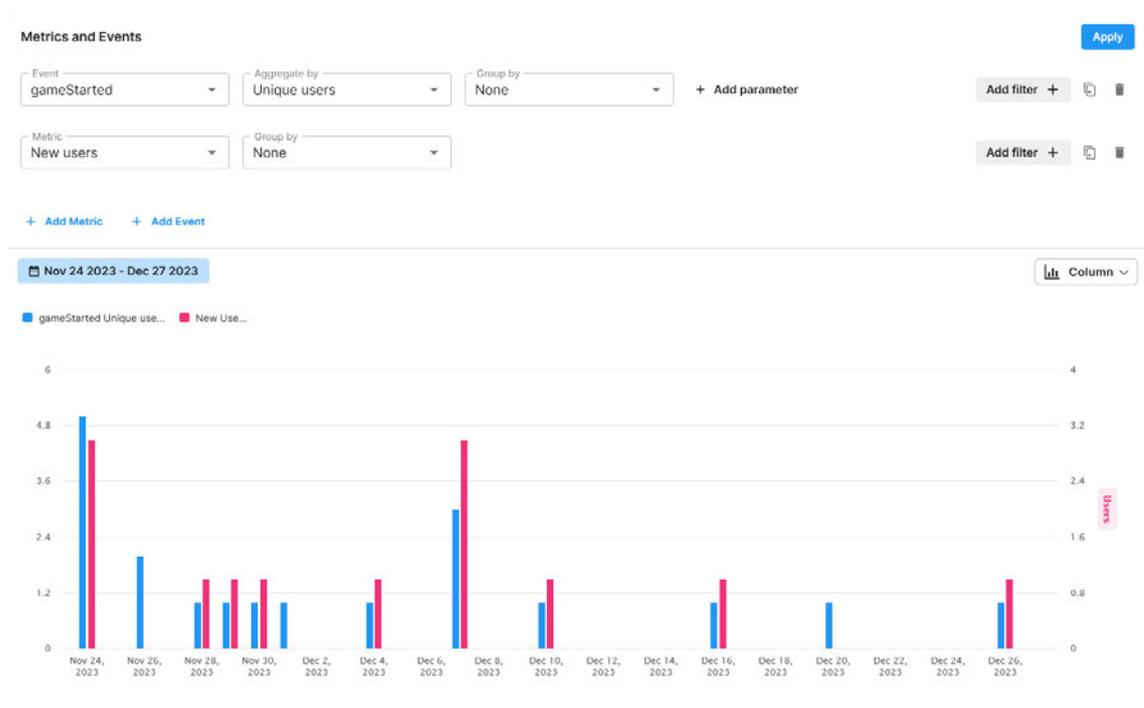


Abbildung B.1.: Erfassung neuer Spieler (rot) sowie wie oft das Spiel einmalig von einem Spieler gestartet wurde (blau)

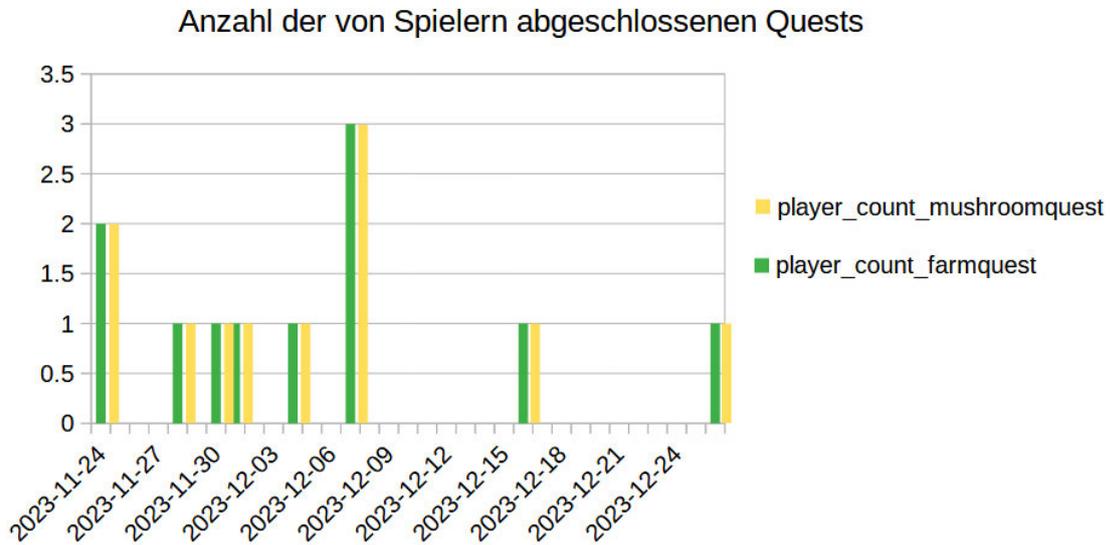


Abbildung B.2.

EVENT_DATE	EVENT_TIMESTAMP	EVENT_NAME	USER_ID	ACCEPTED_QUEST_BEFORE
2023-12-26	2023-12-26 10:11:57.311	FarmQuest	a47	1
2023-12-16	2023-12-16 12:16:46.181	FarmQuest	463	1
2023-12-07	2023-12-07 14:04:00.390	FarmQuest	8b6	1
2023-12-07	2023-12-07 14:02:30.455	FarmQuest	b3b	1
2023-12-07	2023-12-07 09:40:05.864	FarmQuest	f4d	1
2023-12-04	2023-12-04 22:54:44.621	FarmQuest	c6a	1
2023-12-01	2023-12-01 17:36:39.089	FarmQuest	d77	1
2023-11-30	2023-11-30 13:22:37.508	FarmQuest	b1a	1
2023-11-28	2023-11-28 20:21:51.865	FarmQuest	621	1
2023-11-24	2023-11-24 13:36:43.442	FarmQuest	a62	1
2023-11-24	2023-11-24 23:04:37.868	FarmQuest	f7c	1

Abbildung B.3.: Datensatz, welcher aufzeigt, dass jeder Spieler vor Abschluss der Farmquest mit der Schmiedin geredet hat. (Wert "1" der letzten Spalte entspricht "wahr")

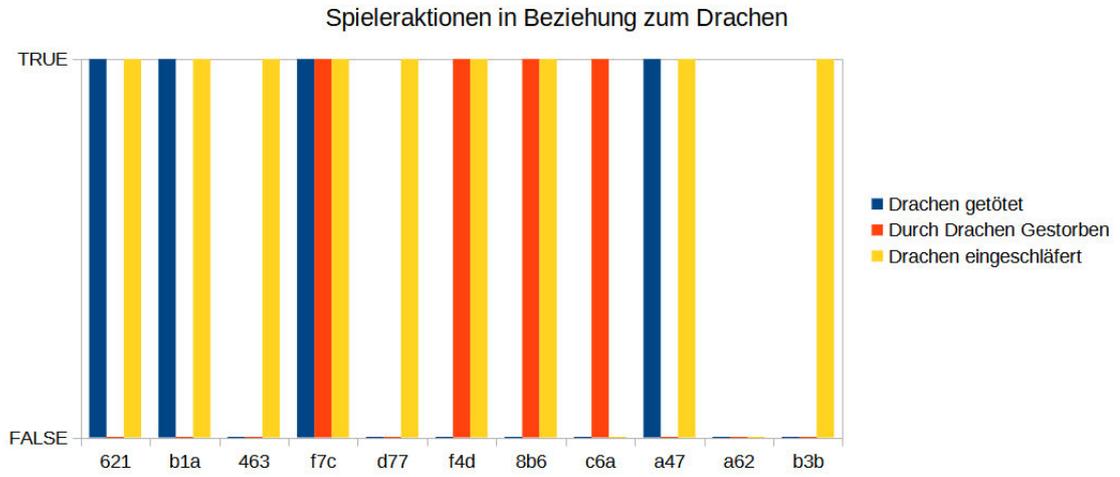


Abbildung B.4.: Übersicht der jeweiligen Spieleraktionen im Bezug auf den Drachen

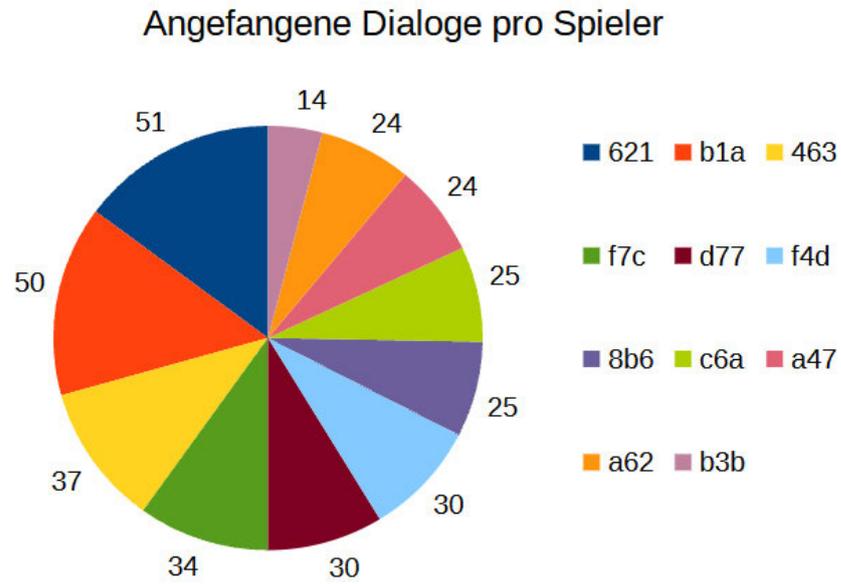


Abbildung B.5.

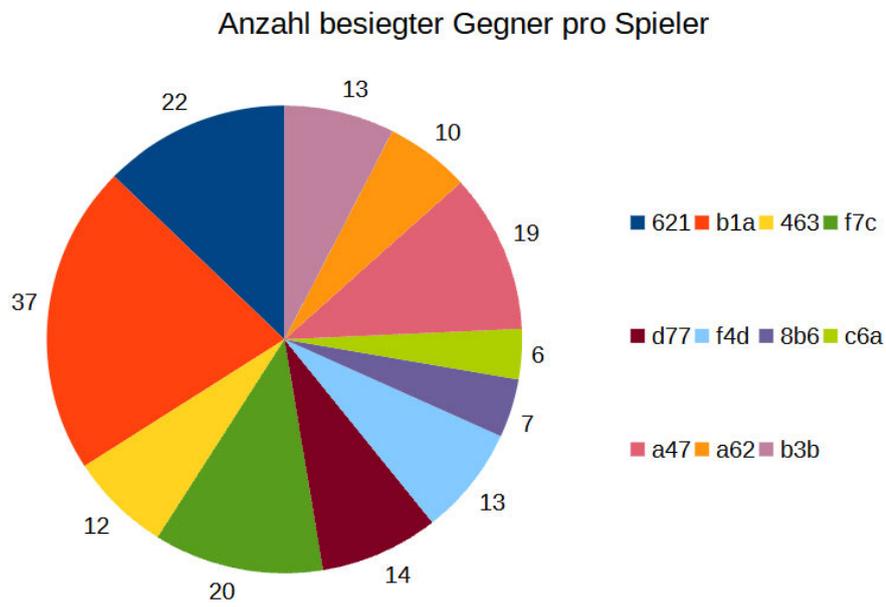


Abbildung B.6.

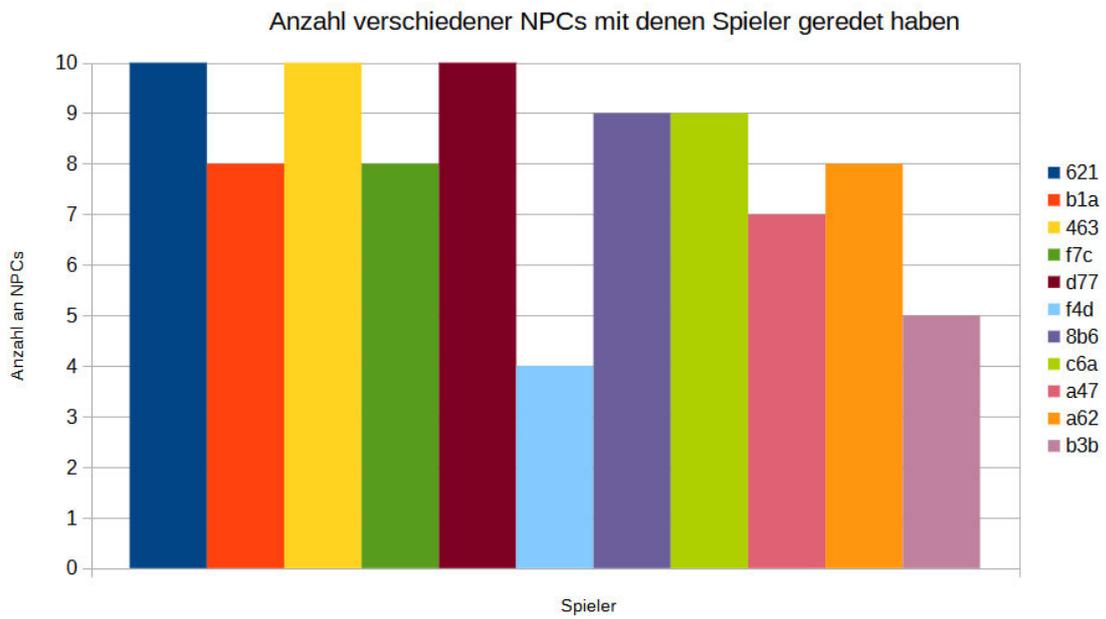


Abbildung B.7.

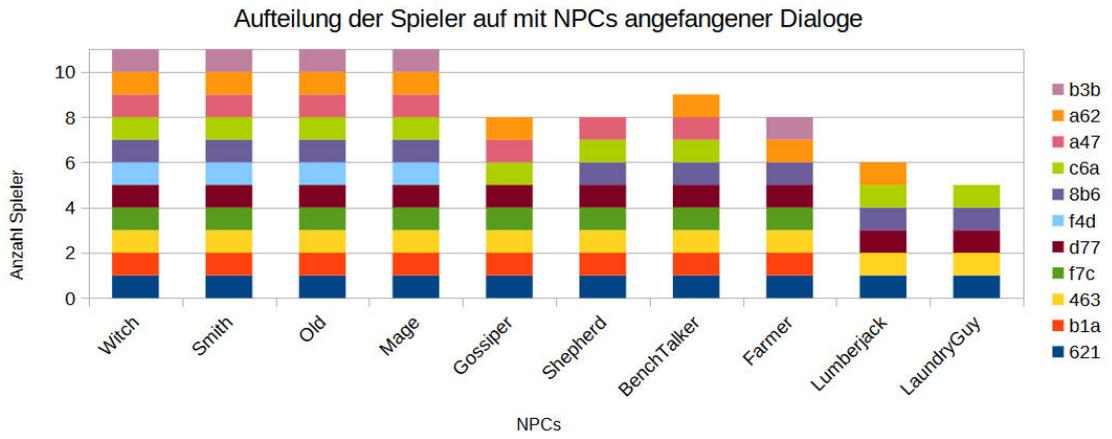


Abbildung B.8.

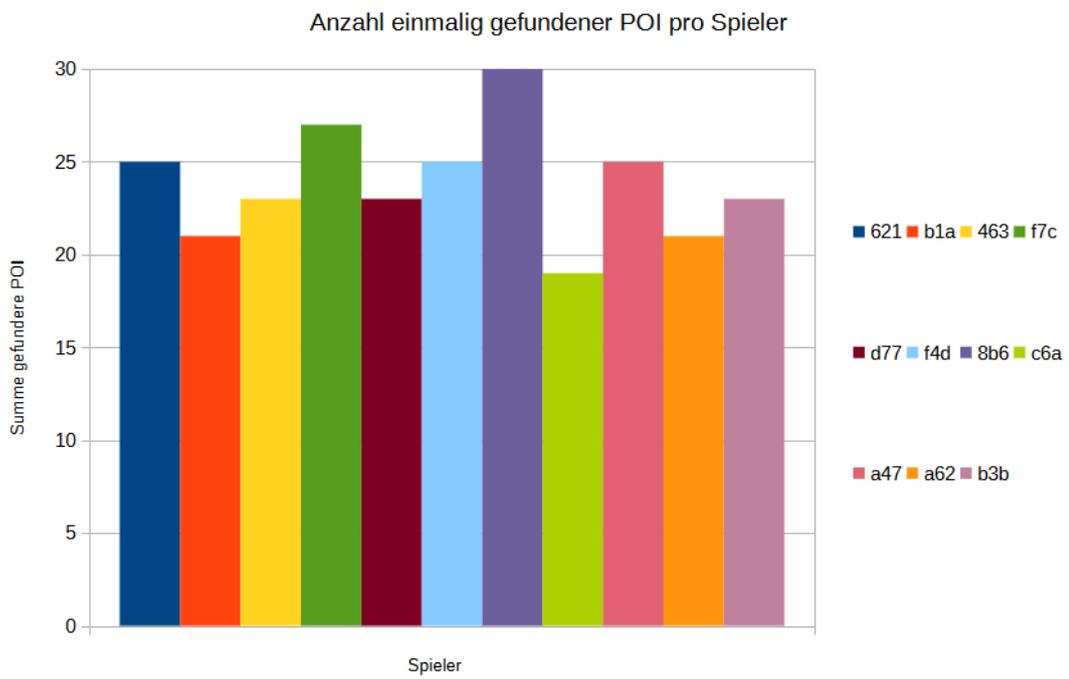


Abbildung B.9.

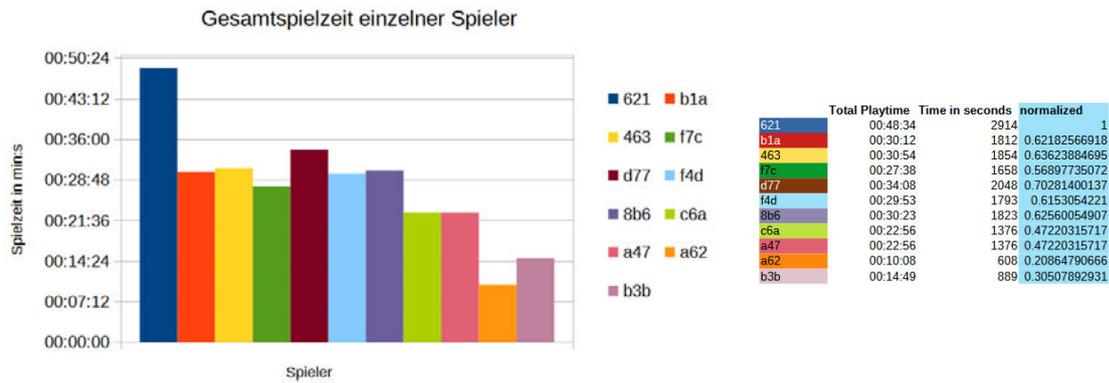


Abbildung B.10.

Archetype	Max Points	points reached			
		Achiever	Explorer	Killer	Socializer
Achiever	71.05				
Explorer	75.9				
Killer	60.57				
Socializer	87.5				
621		44.3761	38.90146932	35.188	71.4359
b1a		40.75	35.76176278	57.319	63.431
463		40.5545	40.49017605	12.1091	64.3364
f7c		49.3415	53.40880938	34.2439	45.4146
d77		34.8433	36.1651575	14.209	55.4478
f4d		43.5057	47.8856855	15.1494	46.3448
8b6		56.4839	52.99296609	9.1129	48.4032
c6a		22.88	31.83701921	8.12	42.5
a47		47.8676	39.8689033	31.5294	33.3529
a62		21.3818	22.68129971	10.1818	32.4364
b3b		37.5	30.81473652	15.26	32.28

	% reached				highest archetype
	Achiever	Explorer	Killer	Socializer	
621	62.45%	51.25%	58.10%	81.64%	81.64%
b1a	57.35%	47.12%	94.63%	72.49%	94.63%
463	57.08%	53.35%	19.99%	73.53%	73.53%
f7c	69.44%	70.37%	56.54%	51.90%	70.37%
d77	49.04%	47.65%	23.46%	63.37%	63.37%
f4d	61.23%	63.09%	25.01%	52.97%	63.09%
8b6	79.49%	69.82%	15.05%	55.32%	79.49%
c6a	32.20%	41.95%	13.41%	48.57%	48.57%
a47	67.37%	52.53%	52.06%	38.12%	67.37%
a62	30.09%	29.88%	16.81%	37.07%	37.07%
b3b	52.78%	40.60%	25.19%	36.89%	52.78%

Grau da kein Archetyp ermittelbar

Abbildung B.11.

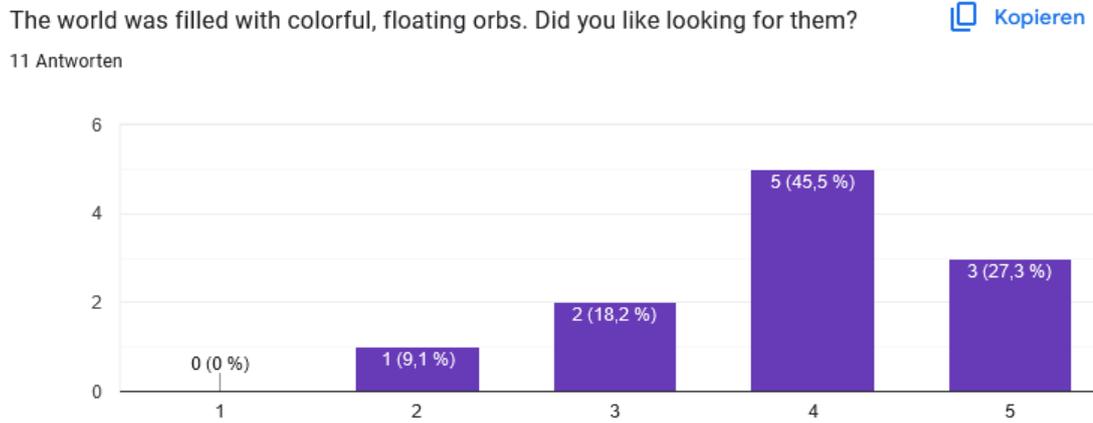


Abbildung B.12.: Antwortverteilung auf die Frage ob das Suchen der Point of Interests Spaß bereitete

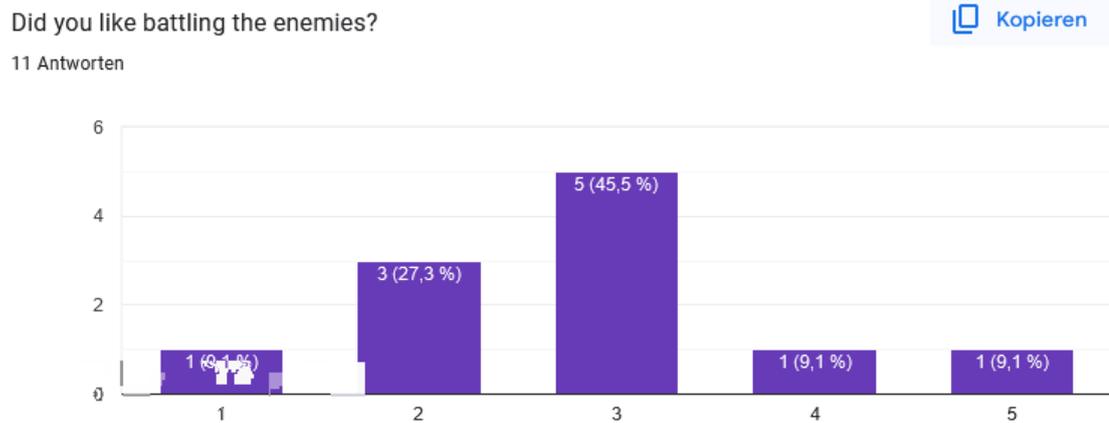


Abbildung B.13.: Antwortverteilung bezüglich des Kämpfens mit Gegnern

Did you enjoy doing the game's quests?

 Kopieren

11 Antworten

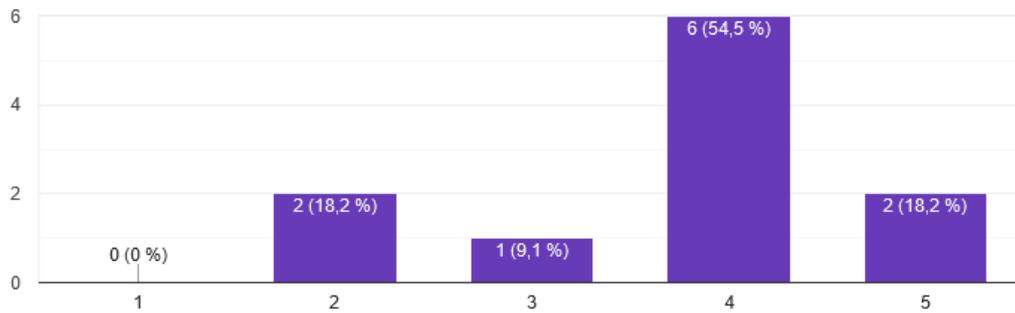


Abbildung B.14.: Antwortverteilung auf die Frage ob das Abschließen der Quests Spaß bereitete

What quests in Explorer Bot did you complete?

 Kopieren

11 Antworten

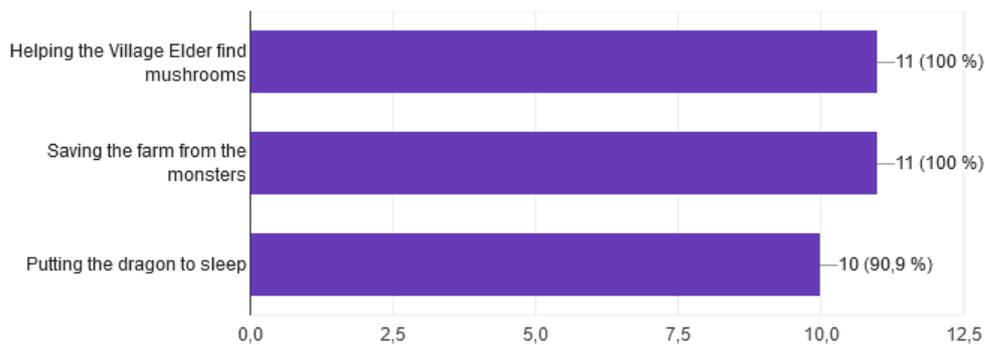


Abbildung B.15.: Angabe der Abgeschlossenen Quests

Did you talk to NPCs outside of quests?

 Kopieren

11 Antworten

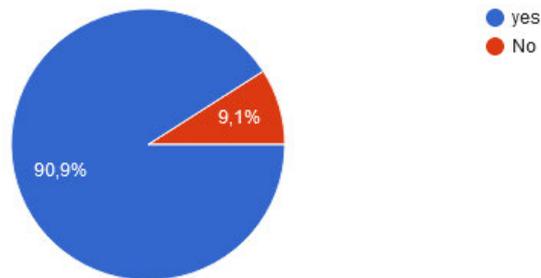


Abbildung B.16.: Angabe wie viele Probanden mit NPCs über das Mindeste hinaus geredet haben

1	Zeitstempel	The world was filled with colorful, floating orbs. Did you like looking for them?	Did you like exploring the game's world? why or why not? How about in other games
2	24.11.2023 13:43:39		2 there were people bathing in normal clothing in the pond
3	24.11.2023 23:36:23		5 I generally LOVE exploring maps in Videogames because i HATE not to have a fully uncovered map and im generally somewhat of a completionist
4	28.11.2023 20:41:37		4 I liked exploring the games world however some hitboxed from different trees or objects were a bit weird sometimes making navigating it a bit clunky. In other games I mostly enjoy exploring more than anything else
5	29.11.2023 17:08:42		4 enjoyed exploring, didnt enjoy getting stuck on and inside (groups of) trees >:]
6	30.11.2023 13:36:36		4 The Orbs where great points of interest, it was a lot of fun locking for them, i also liked, that you had the ability to go wherever you wanted, not restricted to one path
7	01.12.2023 17:52:03		3 I didn't actively go out to look for the orbs but when I ran into them it was a fun morsel of worldbuilding. The game world I enjoyed exploring most in recent years was that of Sable. The movement was fun, the limitless exploration too, and it was very chill overall.
8	07.12.2023 10:06:49		4 Yes, it was interesting. There were some things that made me curious like the hexagonal flower bed or the path that was overgrown by trees. I wondered what they might be used for or if I would have to return at a later point.
9	07.12.2023 14:14:30		3 Exploration is not what I mainly look for in Videogames
10	07.12.2023 14:54:03		5 It was really a lot of fun collecting the orbs. I was very happy when I had collected them all. the score definitely encouraged me to look for them all
11	16.12.2023 12:29:23		4 It was quite straight forward but I kinda liked it maybe because I am a gamer too I know how a open world is built ^^
12	26.12.2023 10:36:09		5 I overallly enjoyed it, however I got "stuck" in-between the hitboxes of trees on the side of the river where the third obelisk is while looking for it. I only found it by running up the side of the Dragon's Cave mountain and looking down on the game world.

Abbildung B.17.: Meinungen zsm mit gegebener Punktzahl bzgl des Erkundens

1	Zeitstempel	Did you like battling the enemies?	Do you like doing combat in other games? why or why not
2	24.11.2023 13:43:39		2 yes because they usually dont have a time frame in the basic combo where they can be attacked for no reason + combat is more satisfying
3	24.11.2023 23:36:23		5 Depends on the Combat. SoT PvP i strongly dislike cause ur always facing sweats!!! I love the League combat because u need to properly land ur spells/ work around certain spells of ur teamcomposition to really thrive and outplay ur opponents ~~is what a sweaty ass LoL player would say i just love ZOOMIN around the map with Bard and being a nuisance to everybody~~ And in DRG and Roboquest u just try and kill everything that comes ur way as fast as possible so u dont die and thats just a really great adrenaline kick
4	28.11.2023 20:41:37		2 I like it if its something that comes up naturally and not too often. I don't enjoy games where combat is the main focus/mechanic
5	29.11.2023 17:08:42		3 was a bit wonky and inconsistent but would be more fun if optimised of course
6	30.11.2023 13:36:36		3 I love fluid and dynamic combat which gives you a limited set of options to deal with each situation, as well as games where you fight many enemies quickly after one another
7	01.12.2023 17:52:03		1 Only if it flows well and fits the game.
8	07.12.2023 10:06:49		4 I like the challenge. Maybe even a little too much.
9	07.12.2023 14:14:30		2 Yes, I usually enjoy doing combat in videogames. But here it felt quite clunky
10	07.12.2023 14:54:03		3 in itself, yes. it depends a lot on the game and the type of fighting and whether it fits into the game. my favorite is turn-based fighting in rpgs, for example. it's risky but not too stressful
11	16.12.2023 12:29:23		3 My wish is a more complicated combat gameplay but i guess it is hard to get into that
12	26.12.2023 10:36:09		3 I'm not a big fan of third-person combat games in general, though in this game the enemies seemed to have particularly weird hitboxes

Abbildung B.18.: Gegebene Gründe warum das Kämpfen mit Gegnern gefallen/- nicht gefallen hat

1	Zeitstempel	Did you enjoy doing the game's quests?	What quests in Explorer Bot did you complete?	Any thoughts on the quests in Explorer bot or quests in general?
2	24.11.2023 13:43:39		Helping the Village Elder find mushrooms, 2 Saving the farm from the monsters	why did the Points of Interest Quest reset when i restarted the game
3	24.11.2023 23:36:23		Helping the Village Elder find mushrooms, 5 Saving the farm from the monsters, Putting the dragon to sleep	In hindsight petting the dragon was a Mistake ...
4	28.11.2023 20:41:37		Helping the Village Elder find mushrooms, 4 Saving the farm from the monsters, Putting the dragon to sleep	The quest were cute, nothing too challenging Generally I like quests that encourage exploration and getting to know the world, as well as small puzzles
5	29.11.2023 17:08:42		Helping the Village Elder find mushrooms, 4 Saving the farm from the monsters, Putting the dragon to sleep	everybody needs more shrooms in general
6	30.11.2023 13:36:36		Helping the Village Elder find mushrooms, 4 Saving the farm from the monsters, Putting the dragon to sleep	In my first run, i just ran past the mushroom quest, i think it would be nice to make them a little bit more in your face
7	01.12.2023 17:52:03		Helping the Village Elder find mushrooms, 3 Saving the farm from the monsters, Putting the dragon to sleep	Some quests in Explorer Bot didn't feel super great because there was no tangible payoff. Both the mushrooms and dragon quests had no reward at all I think? And I did get a cool sword skin from the monster quest but I dont think it did anything besides the visual change. (Which is fine btw, I just would've expected that to maybe have some tangible effect?)
8	07.12.2023 10:06:49		Helping the Village Elder find mushrooms, 4 Saving the farm from the monsters, Putting the dragon to sleep	Pretty standard fetch quests. Usually story incentive helps like for a example creating a potion with mushrooms and monster parts to activate the three pillars necessary to kill the dragon. That way there's more cohesion.
9	07.12.2023 14:14:30		Helping the Village Elder find mushrooms, 2 Saving the farm from the monsters, Putting the dragon to sleep	I was a bit lost on the monsters quest, I think there should've been a marker to guide me to the monsters
10	07.12.2023 14:54:03		Helping the Village Elder find mushrooms, 4 Saving the farm from the monsters, Putting the dragon to sleep	
11	16.12.2023 12:29:23		Helping the Village Elder find mushrooms, 4 Saving the farm from the monsters, Putting the dragon to sleep	Maybe just more variety of quests or little primary and subquests
12	26.12.2023 10:36:09		Helping the Village Elder find mushrooms, 5 Saving the farm from the monsters, Putting the dragon to sleep	I somewhat rushed through the quest dialogue (as people tend to do :D) and thought I had to kill the mushrooms in the forest for the "saving the farm from the monsters" quest, because I hadn't seen the farm when I took the quest. So for players like me it could be helpful if the Smith said "The monsters are still on the loose at the farm" (Instead of just "The monsters are still on the loose" when one talks to her after taking the quest

Abbildung B.19.: Gedanken der Probanden zu den Quests in *Explorer Bot*. Die zweite Spalte ist nur grün wenn alle drei Quests abgeschlossen wurden

1	Zeitstempel	How did you decide what to do in the game?
2	24.11.2023 13:43:39	i followed the path
3	24.11.2023 23:36:23	Go to every corner to look for shiny orbs-> ahh quest -> do quest -> repeat
4	28.11.2023 20:41:37	Just went exploring, following the paths and venturing into the woods looking for NPCs or orbs
5	29.11.2023 17:08:42	evil witch hag tells me to explore -> i explore (tried to find all 30 orbs but game froze on me so i had to alt+f4 and then lost my progress so i didnt find the last 3 because im too lazy to start from the beginning sorry about that) (i couldnt find them anyway)
6	30.11.2023 13:36:36	"Ouh this looks interesting" -> Goes there Monkey sees orb -> neuron activation
7	01.12.2023 17:52:03	If I saw a quest marker I went there, otherwise just generally where glowing and / or moving things were.
8	07.12.2023 10:06:49	I started with exploring, picking up all the orbs along the way and enjoying the scenery. Exclamation marks of NPCs and the wonderous world helped arouse my interest. I wanted to know more about what's going on and what secrets where hidden in that world.
9	07.12.2023 14:14:30	I just started walking to what looked the most interesting
10	07.12.2023 14:54:03	at the beginning you were told to explore, so i ran straight out behind the house and then saw the first orb and from then on i usually continued until i saw new orbs and then in the respective direction. when i arrived in the city i saw the typical exclamation mark and i went straight there. and so on.
11	16.12.2023 12:29:23	I think it through and as I already told i new games so i follow the path
12	26.12.2023 10:36:09	I saw there was a preset/well-trodden path and immediately went off, exploring in a spiral outward from the witch's house to see what else there was :D

Abbildung B.20.: Ansätze der Erkundung des Spiels. Unterschied zwischen zielstrebigiger Exploration(lila) und freier Erkundung (blau)

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 17. Januar 2024



Jamison Bassett