BACHELORARBEIT

zur Erlangung des akademischen Grades

„Bachelor of Science in Engineering“ im Studiengang Informatik/Computer Science

Wie Lootboxen in Videospielen das Suchtverhalten von Spieler\*innen beeinflussen können.

Ausgeführt von: Oliver Müller

Personenkennzeichen: 2110257017

1. Begutachter: Mag. Dr. Christian Osterbauer

Mödling, 20. Mai 2024

Eidesstattliche Erklärung

„Ich, als Autor / als Autorin und Urheber / Urheberin der vorliegenden Arbeit, bestätige mit meiner Unterschrift die Kenntnisnahme der einschlägigen urheber- und hochschulrechtlichen Bestimmungen (vgl. Urheberrechtsgesetz idgF sowie Satzungsteil Studienrechtliche Bestimmungen / Prüfungsordnung der FH Technikum Wien idgF).

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig und nach den aktuell geltenden Regeln der FH Technikum Wien angefertigt und dass ich Gedankengut jeglicher Art aus fremden sowie selbst verfassten Quellen zur Gänze zitiert habe. Ich bin mir bei Nachweis fehlender Eigen- und Selbstständigkeit sowie dem Nachweis eines Vorsatzes zur Erschleichung einer positiven Beurteilung dieser Arbeit der Konsequenzen bewusst, die von der Studiengangsleitung ausgesprochen werden können (vgl. Satzungsteil Studienrechtliche Bestimmungen / Prüfungsordnung der FH Technikum Wien idgF).

Weiters bestätige ich, dass ich die vorliegende Arbeit bis dato nicht veröffentlicht und weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt habe. Ich versichere, dass die abgegebene Version jener im Uploadtool entspricht.“

|  |  |  |
| --- | --- | --- |
| Mödling, 20. Mai 2024 |  |  |
| Ort, Datum |  | Digitale Unterschrift |

Kurzfassung

Lootboxen, auf Deutsch Beutekisten genannt, sind mittlerweile zu einer üblichen Methode zur Monetarisierung in Videospielen geworden.

Entwicklungsstudios für Videospiele verwenden sogenannte Monetarisierungsmethoden als zusätzliche Einkommensquellen neben dem Verkauf von Videospielen. Spieler\*innen können mittels Echtgeldtransaktionen digitale Inhalte direkt in Spielen kaufen.

Lootboxen enthalten zufällig gewählte Gegenstände mit verschiedenen Seltenheitswerten, wobei die Seltensten mit der niedrigsten Wahrscheinlichkeit in Lootboxen gezogen werden. Diese Mechanik ist seit über einem Jahrzehnt in der Industrie sehr beliebt, aber in den Medien und unter Spieler\*innen als Form von Glücksspiel verschrien und wurde zum Beispiel in Belgien bereits verboten.

Viele Studien haben sich mit den Auswirkungen von Lootboxen auf Suchtverhalten bereits beschäftigt und Umfragen unter Spieler\*innen durchgeführt, wobei konkrete Daten aus Spielen, wie Spielzeit oder Echtgeldausgaben, kein Teil davon waren.

Das Ziel der Arbeit ist es, die Frage zu klären, ob ein Spiel aufgrund der Integration einer Lootbox-Mechanik die Spieldauer und die Bereitschaft von Spieler\*innen Echtgeld auszugeben, beeinflusst.

Dafür wird ein eigenes Spiel entwickelt, welches über eine Lootbox-Mechanik und andere Methoden zur Monetarisierung verfügt. Testpersonen probieren dieses Spiel zunächst aus und füllen im Nachhinein einen Fragebogen aus, in dessen Auswertung auch Daten aus dem Spieltest einfließen.

Es zeigt sich, dass ein positiver Zusammenhang zwischen der Erfahrung mit Videospielen und Lootboxen, längerer Testspielzeit und höheren investierten Geldbeträgen besteht. Dies könnte so interpretiert werden, dass eine Integration von Lootboxen in Videospielen tatsächlich die Spieldauer erhöht und zu einem Einsatz von Echtgeld verleitet.

Abstract

**Schlagwörter:** Lootbox, Monetarisierung, Videospiele, Glücksspiel, Suchtverhalten

Loot boxeshave become a common monetization method in videogames.

These methods are designed to generate additional income for game development studios through the purchase of digital content in games via real money transactions by players.

Loot boxes usually contain randomly selected items with different rarities, with the rarest items having the lowest probability of being drawn. This mechanic has been popular in the industry for over a decade but has been criticized in the media and among gamers as a form of gambling, leading to it being banned in Belgium, for example.

Numerous studies have examined the impacts of this mechanism on addictive behaviour, conducting surveys among gamers. Though specific data, like playtime or real money expenditures, from games were not included.

The aim of this thesis is to determine whether the integration of a loot box mechanic influences the duration of gameplay and the willingness of players to spend real money.

To achieve this, a game will be developed featuring a loot box mechanism among other monetization strategies. Test participants will initially try this game and subsequently complete a questionnaire that incorporates data from their gameplay experience.

It showed that there is a positive correlation between experience with video games and loot boxes, longer test play times, and higher amounts of money spent. This could be interpreted as indicating that the integration of loot boxes in video games indeed increases play duration and can entice the spending of real money.

**Keywords:** Loot boxes, Monetization, Videogames, Gambling, Addictive behaviour

Danksagung

Ich möchte mich bei meiner Familie und meiner Lebensgefährtin Katja für die großartige Unterstützung und das Vertrauen bedanken, die mich während dieses Studiums begleitet haben und auch beim Korrekturlesen der Bachelorarbeit halfen.

Weiters möchte ich mich bei meinen Freunden und Studienkollegen Felix, Gabriel, Josua, Matus und Patrick dafür bedanken, dass wir während des Studiums ein großartiges und erfolgreiches Team waren und ohne die ich die letzten 6 Semester sehr wahrscheinlich nicht so viel Spaß gehabt hätte.

Letztendlich bedanke ich mich noch bei meinem Betreuer Herrn Mag. Dr. Christian Osterbauer für seine Hilfe und Beratung während dem Schreiben der Bachelorarbeit.

Inhaltsverzeichnis

[1 Einleitung 8](#_Toc167032820)

[1.1 Motivation 10](#_Toc167032821)

[1.2 Forschungsfragen 11](#_Toc167032822)

[2 Methodik 12](#_Toc167032823)

[2.1 Entwicklung der Spielbasis 12](#_Toc167032824)

[2.2 Entwicklungsumgebung und Programmiersprache 13](#_Toc167032825)

[2.3 Auswahl passender Mechaniken/Game Patterns 14](#_Toc167032826)

[2.3.1 Balancing des Spiels 14](#_Toc167032827)

[2.3.2 Dark Patterns 14](#_Toc167032828)

[2.3.3 Existierendes Spiel als Referenz 15](#_Toc167032829)

[2.3.4 Blog über das Design von „Free-to-play“ Spielen als Referenz 16](#_Toc167032830)

[2.3.5 Gewählte Patterns 17](#_Toc167032831)

[2.3.6 Verbindung zu den Forschungsfragen 18](#_Toc167032832)

[2.4 Datensammlung 18](#_Toc167032833)

[2.5 Gewählte Tools 18](#_Toc167032834)

[3 Lösung 19](#_Toc167032835)

[3.1 Anforderungen 19](#_Toc167032836)

[3.1.1 Basisspiel 19](#_Toc167032837)

[3.1.2 Lootbox-Mechanik 19](#_Toc167032838)

[3.1.3 Premium-Shop 20](#_Toc167032839)

[3.1.4 Rangliste 20](#_Toc167032840)

[3.2 Funktionalität 20](#_Toc167032841)

[3.2.1 Basisspiel/User Interface 20](#_Toc167032842)

[3.2.2 Lootbox-Mechanik 21](#_Toc167032843)

[3.2.3 Premium-Shop 23](#_Toc167032844)

[3.2.4 Rangliste 25](#_Toc167032845)

[3.3 Benutzeroberfläche 26](#_Toc167032846)

[3.3.1 Basisspiel/User-Interface 26](#_Toc167032847)

[3.3.2 Lootbox-Mechanik 29](#_Toc167032848)

[3.3.3 Premium-Shop 30](#_Toc167032849)

[3.3.4 Rangliste 32](#_Toc167032850)

[3.4 Softwarearchitektur 33](#_Toc167032851)

[3.4.1 Grundgerüst 33](#_Toc167032852)

[3.4.2 Lootbox-Mechanik 33](#_Toc167032853)

[3.4.3 Umgesetzte Patterns 39](#_Toc167032854)

[3.5 Spieltest und Umfrage 44](#_Toc167032855)

[3.5.1 Fragenkatalog 44](#_Toc167032856)

[3.5.2 Erläuterung der Fragen und Ergebnisse der Umfrage 46](#_Toc167032857)

[4 Diskussion 53](#_Toc167032858)

[4.1 Potenzial 53](#_Toc167032859)

[4.2 Bezug auf die Forschungsfragen 53](#_Toc167032860)

[4.3 Probleme 54](#_Toc167032861)

[4.4 Mögliche Verbesserungen 55](#_Toc167032862)

[4.4.1 Testspiel 55](#_Toc167032863)

[4.4.2 Fragebogen 55](#_Toc167032864)

[4.5 Abschluss 55](#_Toc167032865)

[Literaturverzeichnis 56](#_Toc167032866)

[Abbildungsverzeichnis 58](#_Toc167032867)

[Tabellenverzeichnis 59](#_Toc167032868)

[Abkürzungsverzeichnis 60](#_Toc167032869)

[Anhang A: Ergebnisse der Umfrage 61](#_Toc167032870)

[Anhang B: Auswahl relevanter Skripts aus dem Spiel 63](#_Toc167032871)

# Einleitung

Im letzten Jahrzehnt entwickelten sich Lootboxen zu einer beliebten Praxis, Videospiele zu monetarisieren, da dadurch zusätzliche lukrative Einnahmen für Entwicklerstudios und Herausgeber gleichermaßen ermöglicht werden.

Laut Statista erwirtschaftete Activision Blizzard im Jahr 2022 5,886 Milliarden US-Dollar über In-Game Käufe, Abonnements und andere Einnahmen, verglichen mit 1,642 Milliarden US-Dollar in Produktverkäufen [1]. Als weiteres Beispiel generierte Electronic Arts 2022 4,277 Milliarden US-Dollar Umsatz allein durch den Verkauf von Zusatzinhalten [2].

Vergleicht man Daten von Videospielverkäufen mit In-Game- und In-App-Käufen aus Deutschland, zeigt sich ein ähnliches Bild. Gegenüber dem Vorjahr ist im Jahr 2023 der Umsatz mit In-Game- und In-App-Käufen um 6% von 4,458 Millionen Euro auf 4,742 Millionen Euro gestiegen.

Der Umsatzanteil liegt damit bei über 80% des Gesamtumsatzes (Abbildung 1) [3].

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

Automatisch generierte Beschreibung

Abbildung 1: Einleitung – Umsatz mit Computer- und Videospielen in Deutschland

Die Lootbox-Mechanik, die oft als Form von Glücksspiel bezeichnet wird, ist aber nicht ohne Kontroversen. Spieler\*innen können mit dem Einsatz von Echtgeld Lootboxen kaufen, die eine definierte Anzahl zufälliger virtueller Gebrauchsgegenstände mit unterschiedlichen Seltenheitsgraden enthalten. Erst nach dem Kauf und dem Öffnen wird bekannt, welche Gegenstände sich darin befinden.

Anfangs war es nicht üblich, die Wahrscheinlichkeitsverteilung der verschiedenen Seltenheitsgrade, auch Fund Rate oder im englischen „Droprate“ genannt, anzugeben. Erst im Jahr 2019 forderten drei große Hersteller von Videospielkonsolen, dass alle Spiele auf ihren Plattformen die „Droprate“ der Lootboxen offenlegen müssen [4].

Seitdem ist dies gängige Praxis in fast allen Videospielen mit Lootbox-Mechaniken. Diese willkommene Veränderung in der Transparenz hat aber wenig Einfluss auf die potenzielle Suchtgefahr, denn die Spannung, vielleicht aus der nächsten Box einen legendären Gegenstand zu erhalten, bleibt.

Selbst wenn ein Spiel keine besonders interessante Geschichte hat oder aufwändig programmiert ist, kann eine Implementierung von Lootboxen zu einem Erfolg führen. Vorausgesetzt, die Mechanik ist gut in das Spiel integriert und motiviert Spieler\*innen zum Kauf [5].

In manchen Spielen sind die Inhalte von Lootboxen nicht spielentscheidend. Das heißt, dass durch deren Kauf kein unfairer Vorteil entsteht (zum Beispiel kosmetische Gegenstände wie Bekleidung oder Frisuren). Bei anderen bekannten Spielen, wie beispielsweise FIFA Ultimate Team, ist es durch den Einsatz von Echtgeld möglich sich einen Vorteil zu erkaufen, da in den Lootboxen unterschiedlich starke und seltene Fußballspieler\*innen enthalten sind.

Sich einen Vorteil mit Geld erkaufen zu können wird in der Branche und Gamingwelt auch als „Pay-to-Win“ bezeichnet.

Neben Lootboxen finden auch andere Arten von „Mikrotransaktionen“ in Videospielen Verwendung, die es Spielenden ermöglichen, im Spiel selbst weitere Inhalte wie DLCs (Downloadable Content), Premiumwährung, Beschleunigungen oder kosmetische Gegenstände zu kaufen. DLCs sind beispielsweise meist umfassendere Erweiterungen des Basisspiels mit einer neuen Geschichte und neuen Orten, die erkundet werden können.

Gerade Lootboxen sind in den letzten Jahren häufiger in den Schlagzeilen. Diese werden mit Glücksspielmechanismen verglichen, da immer mehr Menschen, vor allem Jugendliche und junge Erwachsene, ein Suchtverhalten entwickeln. Verglichen werden kann hier die Ungewissheit. Auf der einen Seite im Glückspiel, ob bei der nächsten Runde endlich der große Geldgewinn dabei ist und auf der anderen die Lootbox, in der hoffentlich der lang ersehnte seltene und wertvolle Gegenstand enthalten ist. Auch die Inszenierung lässt sich vergleichen, denn hier wird auf beiden Seiten mit Musik, bunten Lichtern und Effekten gearbeitet, um die Wartezeit bis zur Bekanntgabe des Gewinns möglichst spannend zu gestalten [6].

Diese strukturellen und psychologischen Ähnlichkeiten führen potenziell zu problematischem Glücksspielverhalten. Während die Spieleindustrie solche Kritiken zurückweist, Lootboxen als „Überraschungsmechanik“ bezeichnet und mit dem Öffnen von Sammelkartenspielpackungen vergleicht, bleibt das Potenzial dieser Mechanismen, dafür anfällige Menschen anzuziehen und ernsthafte Probleme zu verursachen, ein umstrittenes und moralisch komplexes Thema [7].

## Motivation

Lootboxen werden meistens in „Free-to-play“-Spielen integriert. Solche Spiele können grundsätzlich gratis gespielt werden, allerdings werden diese dann entweder durch Lootboxen oder andere Angebote für Mikrotransaktionen monetarisiert. Diese werden, wie in „Monetarisierung von Computerspielen“ von Koubek beschrieben, Teil einer sogenannten „Game-Loop“ (Spielkreislauf), welche immer wieder durchlaufen wird, damit Spieler\*innen motiviert bleiben [8]. Beispielsweise müssen produzierte Ressourcen täglich eingesammelt werden, mit denen dann Gebäude immer weiter aufgewertet oder Einheiten trainiert und aufgestuft werden. Diese Einheiten werden dann für Kämpfe gegen andere Spieler\*innen verwendet, um deren Ressourcen zu stehlen und Punkte zu bekommen. Die Implementation von Premiumwährungen wie Juwelen, welche hauptsächlich durch Echtgeldeinsatz erhältlich sind, werden für die Verkürzung von Wartezeiten verwendet.

Dadurch kann nicht eindeutig gesagt werden, ob bei dieser Art von Spiel allein Lootboxen eine Sucht maßgeblich begünstigen oder ob andere Mechaniken der Monetarisierung ebenfalls eine tragende Rolle spielen.

Während der Recherche nach geeigneten Studien und Artikeln, wurden ausschließlich Auswertungen von Umfragen gefunden, die unter Personen durchgeführt wurden, die Videospiele spielen. In einer Metastudie wurden 20 Studien auf die Zusammenhänge zwischen unter anderem Lootboxen, Gaming und Glücksspiel untersucht [9]. Der Artikel befasst sich mit den psychosozialen Merkmalen von Personen, die Echtgeld für Lootboxen in Videospielen ausgeben. Es wurde ein positiver Zusammenhang zwischen dem Kauf von Lootboxen und Merkmalen der Internetspielsucht sowie Glücksspielstörungen festgestellt. Die Autoren untersuchten auch, wie Stimmung, Geschlecht, physiologischer Zustand und Motivation mit dem Kauf von Lootboxen zusammenhängen.

Ein limitierender Faktor war, dass die Qualität der einbezogenen Studien generell niedrig war, da sie größtenteils auf Selbstberichte zurückzuführen sind. Um die methodische Qualität zu erhöhen, könnten beispielsweise Protokolle von Käufen und Spielzeiten integriert werden, um objektivere und genauere Messungen durchführen zu können [9].

Aus diesem Grund wäre ein Spieltest, welcher genaue Daten liefern kann, mit anschließender Umfrage, ein neuartiger und praxisorientierterer Zugang zu dieser Thematik. Dadurch könnten potenziell neue Lösungen entwickelt werden, um solche Mechaniken zu entschärfen und um Menschen, die anfällig für eine Spielsucht sind, besser für die Thematik sensibilisieren zu können.

## Forschungsfragen

Das primäre Ziel dieser Arbeit ist es zu untersuchen, welchen Einfluss Lootboxen auf das Spielverhalten von Spieler\*innen haben. Dafür wird analysiert, in welchem Ausmaß Lootboxen die Spieldauer beeinflussen. Weiters wird erforscht, welche Auswirkungen Lootboxen auf die Bereitschaft, Geld für Transaktionen in Spielen auszugeben, haben. Möglicherweise können die Ergebnisse auch dazu beitragen zu erkennen, ob die Anziehungskraft der Lootbox-Mechanik stärker ist als andere Erlebnisse in Spielen wie die Story, die Spielwelt oder das Gameplay (wie sich das Spiel spielt).

**Folgende Forschungsfragen bilden sich aus diesen Zielen:**

1. Welche Rolle spielen Lootboxen bei der Veränderung des Spielverhaltens von Spieler\*innen hinsichtlich Spieldauer
2. Wie beeinflusst die Integration von Lootboxen in Videospielen die Bereitschaft von Spieler\*innen, Geld für In-Game-Transaktionen auszugeben?

Um die Forschungsfragen beantworten zu können, wird ein leicht verständliches Videospiel mit einer Lootbox-Mechanik sowie Möglichkeiten von Mikrotransaktionen ausgestattet. Dieses Spiel soll dann von verschiedenen Personen getestet werden. Im Anschluss beantworten die Testpersonen einen Fragebogen zu den Erfahrungen und Eindrücken des Spiels sowie generell zu deren Erfahrung mit Lootboxen in Videospielen und Glücksspielen im Allgemeinen.

# Methodik

Die fertige Lösung wird ein einfaches Spiel, das eine simple Steuerung, moderaten Inhalt und eine integrierte Lootbox-Mechanik beinhaltet. Da so eine Mechanik nicht für sich alleinsteht, wird das Spiel auch einen In-Game Shop beinhalten, der Premiumwährung gegen Geldeinsatz (fiktives Geld) anbietet. Zusätzlich werden auch andere relevante Inhalte im Spiel um diese Währung kaufbar sein, beispielsweise Beschleunigungen oder Extraenergie. Zusätzlich wird das Spiel so designt, dass ein schneller Einstieg möglich und ein leicht verständliches Spielziel definiert ist.

## Entwicklung der Spielbasis

Zu Beginn stand die Frage im Raum, auf welcher Spielbasis die Lootbox-Mechanik entwickelt wird. Ein eigenes Spiel von Grund auf zu programmieren, würde einen sehr hohen Ressourcenaufwand bedeuten. Dafür wären aber keine Lizenzprobleme vorhanden. Ebenfalls in Betracht gezogen wurde die Möglichkeit eine Modifikation für ein vorhandenes Spiel zu entwickeln wie beispielsweise Minecraft. Hier eröffnet sich allerdings das Problem, dass dieses Spiel frei zugänglich sein muss, damit die Testpersonen sich kein Spiel kaufen und zusätzlich eine Modifikation installieren müssen. Zuletzt wäre es auch möglich, vorhandene frei zugängliche Projekt-Samples zu nehmen, die beispielsweise im Marktplatz von Spiele-Engines wie Unity und Unreal zu finden sind und daraus ein Spiel zusammenzustellen.

Alle drei Vorgehensweisen bedeuten einen ähnlich großen Aufwand.

Eine umfassende Recherche nach einem passenden „open-source“ Spiel, zu dem eine Lootbox-Mechanik passt und zusätzlich leicht modifizierbar ist, blieb erfolglos. Auch die Suche nach geeigneten Projektsamples stellte sich als schwierig heraus. Es wurde zwar ein passendes „RPG-Toolkit“ (Role-Playing Game) in der Bibliothek der Unreal Engine gefunden, jedoch hätte die Einarbeitung in dieses Toolkit und die Engine zu viele zeitliche Ressourcen beansprucht.

Letzten Endes fiel die Entscheidung auf die Neuentwicklung eines Spiels. Um den Ressourcenaufwand auf die Einarbeitung in die Spiele-Engine sowie das Programmieren der Skripts zu beschränken, wurde die Verwendung von vorhandenen lizenzfreien grafischen Assets und bei Bedarf Musik und Geräuschen eingeplant.

## Entwicklungsumgebung und Programmiersprache

Der nächste Schritt war, eine geeignete Plattform für die Entwicklung auszuwählen. In der engeren Auswahl standen Unity, Unreal, Godot und Pygame.

Unity und Unreal sind umfangreiche, professionelle Werkzeuge, die auch in der Spieleindustrie eine häufige Anwendung finden. Dementsprechend hoch kann die Einstiegshürde sein. In Unity wird C# verwendet, während in Unreal mit C++ programmiert wird. Unreal bietet jedoch Blaupausen an, die den Einstieg gerade für Anfänger\*innen stark vereinfachen. Dabei ist es möglich, Gameplay-Elemente über ein visuelles Skriptsystem zu erstellen und zu modifizieren. Sogenannte „Nodes“ (Knoten, bzw. Objekte) können mit Funktionsaufrufen, Variablen oder Ereignissen ausgestattet werden, welche wiederum untereinander verbunden werden können.

Diese Funktion erleichtert zwar die Implementierung von Abläufen, die Logik dazu muss aber in Skripts trotzdem programmiert werden.

Pygame ist eine Python-Bibliothek, welche bereits in der Version 2 existiert und unter Hobby- und Indie-Entwickler\*innen beliebt ist. Sie ist „open-source“ und nicht die einzige Python Bibliothek, die für die Spieleentwicklung geeignet ist. Der Einstieg ist mit Erfahrung in Python leicht, jedoch wird das Programmieren eines Spiels durch die hohe Anzahl verschiedener Skripts und Grafiken schnell komplex und nur einen Texteditor als Entwicklungsplattform zur Verfügung zu haben kann schnell unübersichtlich werden.

Letztendlich fiel die Wahl auf die „open-source“ Spiel-Engine Godot. Godot wird stetig weiterentwickelt, arbeitet sehr stabil und ist benutzerfreundlich. Die aktuelle Version ist 4.2.1, die auch für die Bachelorarbeit gewählt wurde. Programmiert wird mit der in Godot integrierten Skriptsprache GDScript, wobei aber auch C# verwendet werden kann. Zu GDScript und Godot werden umfangreiche Dokumentationen bereitgestellt und auf diversen Plattformen wie Github und YouTube finden sich viele Inhalte und Tutorials, um schnell an Hilfe und Informationen zu kommen.

## Auswahl passender Mechaniken/Game Patterns

In diesem Kapitel wird beschrieben, welche Mechaniken und Game Patterns in Spielen vorhanden sind und welche davon später im eigens entwickelten Spiel eingebaut werden.

### Balancing des Spiels

Damit das fertige Spiel auch motivierend ist, ist es wichtig auf das richtige Balancing in der Spielerfahrung zu achten, wie in dem Buch „Monetarisierung in Computerspielen“ beschrieben wird [8].

Dabei ist unter anderem auf Fairness zu achten, sodass alle Spieler\*innen für die Herausforderungen des Spiels genügend Ressourcen zur Verfügung haben. Auch darf es nicht zu schwierig oder zu leicht sein. Die Balance zwischen Können vs. Glück, Verstand (Rätsel) vs. Geschicklichkeit (Reaktionsvermögen), Kooperation vs. Konkurrenz, Kurze vs. Lange Spieldauer, Belohnungen vs. Strafen ist ebenfalls wichtig zu beachten.

Die zuletzt genannten Punkte, auch Design Patterns genannt, werden für die Verbesserung der Spielerfahrung herangezogen und immer weiter verfeinert. Gerade bei „Free-to-play“ Spielen ist es üblich, regelmäßige Updates mit Änderungen im Balancing durchzuführen, um die Spielerfahrung möglichst vieler Spieler\*innen zu verbessern. Dabei sind auch eine gewisse Zusammenarbeit und ein Vertrauen zwischen Entwickler\*innen bzw. Game Designer\*innen und den Spieler\*innen nötig.

### Dark Patterns

Zu den genannten Design Patterns haben sich in den letzten Jahren auch Dark Patterns oder Dark Game Design Patterns einen Namen gemacht. Hierbei geht es um gezieltes Herbeiführen von negativen Erlebnissen im Spiel, die Spielende dazu verleiten sollen, diese durch Echtgeldeinsatz in positive zu wandeln. Einige dieser Muster werden ebenfalls im Buch „Monetarisierung in Computerspielen“ genannt [8].

Beispiele:

* **Wait to Play:** Spieler\*innen werden willkürlich warten gelassen. (Mit steigendem Level und Spielfortschritt werden Wartezeiten länger.)
* **Playing by Appointment:** Nicht mehr die Spieler\*innen, sondern das Spiel entscheidet, wann gespielt wird (Regelmäßiges Anmelden, um den Spielfortschritt zu sichern.).
* **Daily Rewards:** Das tägliche Anmelden wird belohnt und das Fernbleiben bestraft.
* **Grinding:** Nur das regelmäßige aktive Spielen bringt deutlichen Fortschritt.
* **Can’t Pause or Save:** Spiele werden auf zentralen Servern verwaltet und Spieler\*innen haben keine Chance ihr Spiel zu speichern oder zu pausieren.
* **Infinite Treadmill:** Es gibt keinen Endzustand und der Spielfortschritt bzw. das Ranking in der Rangliste können endlos fortgeführt werden.
* **Premium Currency:** Das Einführen einer Premium-Spielwährung beispielsweise als Juwelen, welche zwar auch in kleinen Mengen erspielbar ist, allerdings hauptsächlich gegen Geld angeboten wird.
* **Pay to Skip:** Meist in Kombination mit einer Premiumwährung. Damit lassen sich Wartezeiten verkürzen oder ganz auflösen.
* **Pay to Win:** Spielerische Vorteile durch den Kauf von stärkerer Ausrüstung oder mächtigeren Einheiten durch Premiumwährung.
* **Artifical Scarcity:** Eine künstliche Verknappung von Inhalten, die nur kurzzeitig oder in Sonderangeboten erhältlich sind. (Fear of Missing out – FOMO)

Beispiele zu Dark Social Patterns:

* **Competition:** Der Wettbewerb unter den Spieler\*innen wird durch Einsatz von Zeit und Geld erleichtert.
* **Social Obligation:** Spieler\*innen fühlen sich ihrer Community/ihrem Clan verpflichtet und wollen nicht enttäuschen.
* **Fear of Missing Out:** Die Angst, etwas zu verpassen, seien es Events oder geplante Unternehmungen des Clans.
* **Endowed Value:** Der erarbeitete Spielfortschritt, eventuell auch durch bereits ausgegebenes Geld unterstützt, wird nicht einfach so aufgegeben.

Oft werden manche der genannten Dark Patterns gar nicht als negativ empfunden. Gerade die sozialen Mechaniken wie der Wettbewerb zwischen Spieler\*innen oder Clans wird oft als positive intensive Erfahrung gesehen [8].

Dark Patterns finden allerdings nicht nur in der Spieleentwicklung ihren Platz, sondern zum Beispiel auch in sozialen Netzwerken oder im Marketing, wie ein Bericht auf onlinesicherheit.at beschreibt [10].

### Existierendes Spiel als Referenz

Best Gaem Studio hat am 15.12.2015 einen „Pay-to-Win“ Simulator mit dem Namen „Pay 2 Win: The Tricks Exposed“ veröffentlicht, der die Tricks, bzw. Methoden von Monetarisierung in Videospielen anhand der Spielerfahrung offen anspricht und erläutert [11].

Ein Bild, das Text, Screenshot, Website, Onlinewerbung enthält.

Automatisch generierte Beschreibung

Abbildung 2: Methodik – Screenshot des Spiels aus der Vertriebsplattform Steam

Die Inhalte dieses Simulators werden als Inspiration und Referenz für das eigens entwickelte Spiel herangezogen, unter anderem die Möglichkeit durch mehrmaliges Klicken auf einen Button je 40 Cent zu erhalten. Die Testpersonen geben also kein echtes Geld aus, sondern können selbst fiktives Geld generieren, welches wiederum im Spiel für diverse Inhalte ausgeben werden kann.

### Blog über das Design von „Free-to-play“ Spielen als Referenz

Ein Blog mit dem Titel „The design of Free-to-play Games“ wird ebenfalls als Referenz für einige der Mechaniken verwendet. Darin geht es hauptsächlich um allgemeine Eigenschaften von „Free-to-play“-Spielen, wie beispielsweise einen leichten und schnellen Einstieg, eine leicht verständliche Benutzeroberfläche oder die Implementierung von kleinen aber wichtigen Spielabläufen, die immer nur kurze Zeit gespielt werden, aber dafür sehr oft über einen langen Zeitraum.

Sehr wichtig sind auch offene Schleifen, die Spielende immer wieder zurückholen, da sie das Gefühl bekommen sollen, immer noch etwas erledigen zu müssen [12].

Dabei bleibt das Ende offen, sodass zu erledigende Aufgaben in einer Endlosschleife bleiben und nicht so leicht unterbrochen werden können. Sogenannte Endlosschleifen werden auch „Game-Loops“ genannt.

### Gewählte Patterns

Die folgenden Design-Patterns wurden für eine Integration in das Spiel ausgewählt. Für jedes Pattern wird eine mögliche Umsetzung erläutert.

* **Wait to Play:** Zum Beispiel könnte eine in Auftrag gegebene Herstellung eines Gegenstandes eine gewisse Anzahl Sekunden oder Minuten dauern, ehe dieser eingesammelt werden kann.
* **Grinding:** Erst das regelmäßige Spielen über längere Zeiträume, würde den Testpersonen das Erreichen eines gewissen Fortschritts ermöglichen.
* **Can’t Pause or Save:** Das Spiel könnte nur im Hintergrund automatisch gespeichert werden, sodass Testpersonen keine Chance haben, unliebsame Ereignisse rückwirkend durch das Laden eines alten Spielstandes zu ändern.
* **Infinite Treadmill:** Das Spiel könnte so aufgebaut werden, dass die Kernmechanik immer wieder wiederholt werden kann. Punkte, die durch diese Mechanik gesammelt würden, scheinen in einer Rangliste mit simulierten Computergegnern auf.
* **Premium Currency:** Die Verwendung einer Premiumwährung könnte ein zentraler Bestandteil sein, mit dem die Lootboxen, aber auch andere Inhalte bezahlt werden.
* **Pay to Skip:** Das kommt ganz auf das Genre des Spiels an, das entwickelt wird. Eventuell eignet sich stattdessen eine **Pay to Win** Mechanik besser.
* **Artifical Scarcity:** Ein Sonderangebot könnte für eine gewisse Zeit am Anfang des Spiels angepriesen werden, beispielsweise Premiumwährung um den halben Preis.
* **Competition:** Die meisten Spiele mit einer Lootbox Mechanik sind kostenlos spielbar und auf viele gleichzeitige Spieler\*innen ausgelegt. Da die Entwicklung eines solchen Spiels zu viele Ressourcen beanspruchen würde und auch das Testen erschwert, wird der Mehrspieleranteil lediglich simuliert. Beispielsweise könnte eine Rangliste implementiert werden, bei der durch die Darstellung der Punktestände von NPCs (Non Player Character) eine Art Wettbewerb suggeriert wird. Die Testpersonen werden so motiviert, Zeit, Aufwand und im Idealfall auch fiktives Geld zu investieren, um auf dieser Rangliste möglichst weit vorne zu stehen, da dies auch mit besseren Gewinnen belohnt wird.

Die Implementierung verschiedener Mechaniken wie ein beschleunigter und leichter Start, der rasch zu einer hohen Paywall führt (die Kosten steigen im späteren Spielverlauf im Vergleich zum Anfang stark an), Vergünstigungen aber auch Bundles (günstigere Kombiangebote) sollen Testpersonen zu einem Kauf von Lootboxen verleiten.

### Verbindung zu den Forschungsfragen

Lootboxen werden eine möglichst zentrale Rolle spielen, damit deren Einfluss auf das Spielverhalten auch bedeutsam ist. Um im Spiel weiterzukommen, wird die Verwendung von Lootboxen notwendig sein. Auf der anderen Seite darf dies aber auch nicht einschränken. Eine Implementation einer Lootbox, die für die gewöhnliche Spielwährung kaufbar ist, wäre eine Option. Damit die Attraktivität einer Premium-Lootbox erhöht wird, wird diese wesentlich bessere Konditionen bieten. Damit wäre diese Form der Integration von Lootboxen auch ein Faktor, der die Bereitschaft zu Geldinvestitionen beeinflussen könnte.

## Datensammlung

Während des Spiels werden Daten gesammelt. Unter anderem, wie viel fiktives Geld die Testpersonen generiert haben, wie viel sie davon ausgegeben haben und wie viel Zeit sie mit dem Spiel verbracht haben. Diese Daten werden von den Testpersonen nach dem Test bei einer Umfrage eingegeben, damit diese ausgewertet werden können.

## Gewählte Tools

Als Entwicklungsumgebung wurde die „open-source“-Spiel-Engine Godot [13] [14] gewählt, da die damit entwickelten Spiele nicht durch Lizenzen beschränkt werden. Das Spiel selbst wurde in der dazu gehörenden Programmiersprache GDScript geschrieben, da diese Sprache von den Entwicklern von Godot erstellt wurde, sehr gut dokumentiert ist und sich nahtlos in die Engine einfügt.

Es wurden keine externen Bibliotheken und Frameworks benötigt, da die Engine bereits alles Benötigte beinhaltet.

Als Grafikprogramme wurden MS Paint und Affinity Designer verwendet. Da auf ein vorhandenes frei verfügbares Asset Pack für sämtliche Grafiken im Spiel zurückgegriffen wurde, wurden die genannten Programme nur für Anpassungen verwendet.

# Lösung

Informationen zum Spiel:

Titel: Farm Island – A Lootbox Experience

Ansicht: 2D – Frontal

Grafikdesign: Pixel

Genre: Farmsimulation

Für die Grafiken wurde online nach Assets recherchiert und folgende ausgewählt:

Sprout Lands Asset Pack by Cup Nooble [15]

Sprout Lands UI Expansion by Cup Nooble [15]

Dem Spiel wurden keine Soundeffekte oder sonstige Audioelemente hinzugefügt.

In den folgenden Unterkapiteln werden zuerst die Anforderungen (3.1) für jeden relevanten Teil erklärt. Anschließend wird die genaue Ausführung textuell besprochen (3.2) und es werden Screenshots dieser Teile gezeigt und beschrieben (0).

Zuletzt wird in 3.4 die Softwarearchitektur, der für die Lootbox-Mechanik und Monetarisierung relevanten Teile, in Codeausschnitten veranschaulicht.

Für die Versionierung und Sicherung wurde das Tool Github verwendet.

Github Repository: <https://github.com/Highlander0815/BAC_Lootboxen-in-Videospielen.git>

## Anforderungen

### Basisspiel

Mittels der „WASD“-Tasten oder der Pfeiltasten kann die Spielfigur in der Spielwelt bewegt werden. Mit der Maus und der „E“-Taste kann mit Gegenständen in der Spielwelt interagiert werden. Die Spielwelt ist eine kleine Insel auf der eine eigene Hütte, Farmland, Vegetation und Nutztiere sowie der Shop eines NPC-Händlers zu finden sind. Die Samen aus Lootboxen werden auf Farmland angepflanzt. Diese wachsen über einen gewissen Zeitraum und können dann geerntet und im Shop verkauft werden. Es gibt eine Spielwährung mit der Bezeichnung „Coins“, welche durch den Verkauf von reifem Gemüse generiert wird.

### Lootbox-Mechanik

Mit der Spielwährung „Coins“ können unter anderem Lootboxen geöffnet werden, welche drei Exemplare von zufällig gewählten unterschiedlich seltenen Samen enthalten. Eine Premium-Lootbox mit doppelter Chance auf seltenere Samen kann mit der Premiumwährung „Silver“ gekauft werden.

### Premium-Shop

Im Premium-Shop können Premiumwährung, größere Farmflächen oder strukturelle Erweiterungen wie ein Brunnen (für schnelleres Pflanzenwachstum) gekauft werden.

### Rangliste

Die Rangliste soll die Spieler\*innen motivieren, mehr Zeit und bestenfalls fiktives Geld in das Spiel zu investieren, um auf die vordersten Plätze zu kommen und diese auch halten zu können. Sie zeigt den aktuellen Punktestand der Spielenden und der simulierten NPCs. Punkte werden durch den Verkauf der Feldfrüchte gesammelt. Um den Mehrspieleraspekt und einen gewissen Wettbewerb zu simulieren, werden fiktive Namen und deren Punktestände hinzugefügt. Die Spieler\*innen werden nur mit zeitlichem und monetärem Aufwand die obersten Plätze erreichen und halten können.

## Funktionalität

Die Funktionalität wird am Beispiel der Unterkapitel in Kapitel 3.1 erklärt.

### Basisspiel/User Interface

Ein Startmenü ermöglicht einen angenehmeren und entschleunigten Start in das Spiel (Abbildung 3). Nach Drücken des „Play“-Buttons öffnet sich eine Textbox mit einer kurzen Begrüßung und der Möglichkeit, einen Usernamen einzugeben (Abbildung 4).

Bevor das Spiel selbst gespielt werden kann, werden noch die Steuerung (Abbildung 5) sowie eine kurze Spielanleitung „How to Play“ als Bilderreihe eingeblendet (Screenshots des Spiels mit Beschriftung) (Abbildung 6). Diese Erklärung kann jederzeit im Pause-Menü unter „Controls“ eingesehen werden.

Direkt im Anschluss beginnt das eigentliche Spiel vor dem eigenen Haus. Davor befindet sich ein kleiner Fleck fruchtbarer Boden, der zum Anpflanzen der Samen genutzt werden kann. In der linken oberen Ecke des User-Interfaces befindet sich eine Übersicht über den derzeitigen Kontostand des fiktiven Geldes, den bereits ausgegebenen Geldbetrag und der „+“-Button, mit dem der Kontostand pro Mausklick um einen Betrag von 0,40€ erhöht werden kann.

Rechts oben informiert eine Übersicht über den aktuellen Stand der Spielwährung „Coins“ und der Premium-Spielwährung „Silver“.

Am unteren Rand befindet sich die sogenannte „Hotbar“, zu der Items hinzugefügt werden können, um sie im Spiel zu verwenden (Abbildung 6)(Abbildung 7).

Zu den Basisfunktionen zählt noch das Inventar, das über die Taste „I“ aufgerufen werden kann. Anfangs zählt das Inventar noch acht Plätze, die jedoch mit Premium-Spielwährung erweitert werden können. Über das Inventar werden Items wie Samen oder Gemüse der Hotbar zugewiesen. Ein kleines Fenster informiert mit dem Namen, der Gruppe und Seltenheit über das Item (Abbildung 8). Eine Möglichkeit, Items fallen zu lassen ist ebenfalls vorhanden.

Wurden Samen der Hotbar zugewiesen, können diese mit den Tasten „1, 2, 3, 4 und 5“ ausgewählt werden und per linkem Mausklick auf fruchtbare Erde gepflanzt werden (Abbildung 9).

### Lootbox-Mechanik

Die Lootbox-Mechanik befindet sich in einem eigenen Premium-Shop, der während des Spiels wie ein Menü mit der Taste „Q“ aufrufbar ist.

In diesem Premium-Shop werden eine Basis- und eine Premiumvariante der Lootbox angeboten. Die Basisvariante ist gegen „Coins“ erwerbbar und die Premiumvariante gegen „Silver“. Der Unterschied der beiden Boxen ist, dass die Premiumvariante eine höhere Fundrate (Droprate) bei den selteneren Samen hat (Tabelle 1: Lootboxen – Droprates/FundratenTabelle 1).

|  |  |
| --- | --- |
| Droprates (Fundraten) der  verschiedenen Seltenheitskategorien beider Lootboxen | |
| Seltenheitsstufe | Basis Lootbox | Premium Lootbox |
| 1 | 35% | 30% |
| 2 | 55% | 50% |
| 3 | 9.5% | 19% |
| 4 | 0.5% | 1% |

Tabelle 1: Lootboxen – Droprates/Fundraten

Die Unterteilungen wurden auf Basis von Recherche und eigenen Erfahrungswerten in   
„Free-to-play“-Spielen mit Lootboxen gewählt. Übliche Raten in dem Genre sind gerade für die Kategorie der seltensten Gegenstände im Bereich von 0,2% bis 0,6%.

Dass die Seltenheitsstufe 2 eine höhere Rate aufweist als 1, ist Absicht. Stufe 2 ist wertvoller als 1 aber im Vergleich zu 3 und 4 immer noch relativ wertlos. Der Gedanke ist hier, dass wenn sich in der Lootbox häufiger höherwertigere Gegenstände als die niedrigste Stufe befinden, die Motivation bei weniger Glück auch weniger stark sinkt.

Die Kosten, um eine Basis-Lootbox zu öffnen betragen fünf „Coins“ und bei der Premium-Lootbox fünf „Silver“. Ist der Betrag verfügbar, wird nach einem Klick auf die entsprechende Lootbox der Mechanismus gestartet. Während sich die Box öffnet, leuchtet das Innere und hinter der Box fliegen weiße Partikel sternförmig nach außen. Nach kurzer Zeit erscheinen die drei Items, welche per „Linksklick“ aufgehoben werden können. Verlässt die Testperson vorzeitig den Shop, werden die Items automatisch dem Inventar hinzugefügt.

Es wurde ebenfalls eine Version der Lootbox für die Interaktion in der Spielwelt erstellt. Diese hat die gleiche Droprate wie die Premium-Lootbox, der einzige Unterschied ist, dass das Öffnen der Box und das Aufheben der Items mit der Taste „E“ statt der linken Maustaste geschieht.

Insgesamt existieren im Spiel 14 verschiedene Samen, die in den Lootboxen enthalten sind (Tabelle 2).

|  |  |  |  |
| --- | --- | --- | --- |
| Samen | Seltenheit | Wert „Coins“ / „Silver“ | Punktewert |
| Cabbage | 1 | 1 „Coins“ | 1 |
| Wheat | 1 | 1 „Coins“ | 1 |
| Aubergine | 2 | 2 „Coins“ | 4 |
| Carrot | 2 | 2 „Coins“ | 4 |
| Corn | 2 | 2 „Coins“ | 4 |
| Cucumber | 2 | 2 „Coins“ | 4 |
| Tomato | 2 | 2 „Coins“ | 4 |
| White Radish | 2 | 2 „Coins“ | 4 |
| Artichocke | 3 | 4 „Coins“ | 20 |
| Cauliflower | 3 | 4 „Coins“ | 20 |
| Pumpkin | 3 | 4 „Coins“ | 20 |
| Purple Radish | 3 | 4 „Coins“ | 20 |
| Blue Poppy | 4 | 20 „Coins“ / 1 “Silver“ | 100 |
| Starfruit | 4 | 20 „Coins“ / 1 “Silver“ | 100 |

Tabelle 2: Lootbox – Mechanik Seltenheit und Wert der Samen

Die Seltenheitskategorie 2 beinhaltet die meisten Samen, damit die höchstmögliche Diversität in der Belohnung erreicht wird.

**Verbindung zu den Forschungsfragen**

Es wurden zwei verschiedene Lootboxen implementiert, wobei eine davon eine Basis-Lootbox ist, die für Testpersonen eine Art Einstieg in die Mechanik ist. Sobald eine Testperson Premiumwährung „Silver“ zur Verfügung hat, kann diese ebenso schnell und leicht ausgegeben werden, um eine Premium-Lootbox zu öffnen.

Eine schnell verfügbare Lootbox, welche direkt mit einem Klick erreichbar ist und gekauft werden kann, soll ebenfalls deren Verwendung fördern und wurde dementsprechend designt.

Die Lootbox-Mechanik wurde direkt in den Spielfluss integriert und ist dadurch auch spielrelevant geworden. Testpersonen müssen zumindest die Basisversion der Lootbox kaufen, damit ein Fortschritt möglich ist.

### Premium-Shop

Der Premium-Shop beinhaltet neben den Lootboxen auch andere kaufbare Inhalte wie Erweiterungen für Inventar und Farmland, Strukturen, eine Umtauschfunktion und den Kauf von Premium-Spielwährung.

Alle genannten Bereiche sind in einem Tab-Menü untergebracht. Das heißt, dass immer nur der aktive Tab angezeigt wird.

Mit welcher Währung eine bestimmte Erweiterung gekauft werden kann, wurde nach deren Relevanz für den Spielfluss bestimmt. Beispielsweise muss zwangsläufig mehr Farmland gekauft werden, daher werden hier „Coins“ verlangt. Eine Inventarerweiterung oder der Kauf einer Struktur sind hingegen nur Erleichterungen für den Spielfluss. Daher werden diese für „Silver“ angeboten.

**Inventory:**

Im Tab „Inventory“ kann das Inventar gegen „Silver“ um weitere vier Plätze erweitert werden. Anfangs bietet das Inventar acht Plätze. Für jede weitere Stufe verdoppeln sich die Kosten (Tabelle 3).

|  |  |
| --- | --- |
| Plätze (maximal) | Kosten („Silver“) |
| 12 | 2 |
| 16 | 4 |
| 20 | 8 |
| 24 | 16 |
| 28 | 32 |
| 32 | 64 |

Tabelle 3: Premium-Shop – Kosten der Inventarerweiterung

**Farmland:**

Im Tab „Farmland“ kann das verfügbare fruchtbare Farmland gegen „Coins“ vergrößert werden. Für jede gekaufte Stufe erhöht sich der Preis um das Zweifache (Tabelle 4).

|  |  |
| --- | --- |
| Stufe | Kosten („Coins“) |
| 1 | 25 |
| 2 | 50 |
| 3 | 100 |
| 4 | 200 |
| 5 | 400 |
| 6 | 800 |
| 7 | 1600 |

Tabelle 4: Premium-Shop – Kosten der Farmlanderweiterung

Für jede Stufe bis einschließlich der 6. Stufe werden neue Farmland-Flächen in der Spielwelt generiert. Die letzte Stufe entfernt eine Absperrung für einen vordesignten Bereich im Westen der Insel, in dem sich eine größere Fläche Farmland befindet.

**Structure:**

Im Tab „Structure“ kann ein Brunnen um 50 „Silver“ gekauft werden, welcher in der Spielwelt neben dem Haus platziert wird und das Pflanzenwachstum um 20% beschleunigt.

**Exchange:**

Im Tab „Exchange“ kann die Premiumwährung „Silver“ jederzeit gegen „Coins“ im Verhältnis 1:5 getauscht werden.

**Silver Ingots:**

Im Tab „Silver Ingots“ kann das generierte fiktive Geld für „Silver“-Pakete ausgegeben werden. Dies ist eine in der Spieleindustrie übliche Praktik, um „Free-to-play“-Spiele zu monetarisieren. Die Preise variieren sehr stark zwischen den Spielen und reichen in manchen Fällen für einzelne Pakete von 0,99€ bis zu 999€. Für das Spiel wurden allerdings niedrigere Preise gewählt (Tabelle 5).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Paket | Kosten | Grundmenge | Multiplikator | Bonus | Finale Menge |
| Tiny Pack | 1,99€ | 5 | - | - | 5 „Silver“ |
| Small Pack | 7,99€ | 20 | 4 | 10% | 22 „Silver“ |
| Medium Pack | 19,99€ | 50 | 2,5 | 10% | 55 „Silver“ |
| Large Pack | 49,99€ | 125 | 2,5 | 10% | 138 „Silver“ |

Tabelle 5: Premium-Shop – Kosten der "Silver" Pakete

**Erklärung der Berechnung:**

Für das „Tiny Pack“ erhält man für 1,99€ 5 „Silver“. Für das nächsthöhere Paket „Small Pack“ wurde mit dem Multiplikator 4 gerechnet. 1,99€ \* 4 ergibt ungefähr 7,99€ und 5 \* 4 ergibt 20 „Silver“. Diese Berechnung setzt sich für die beiden letzten Pakete mit dem Multiplikator 2,5 fort. Für alle Pakete außer dem „Tiny Pack“ erhält man zusätzlich 10% Bonus auf das „Silver“. Das wären beim „Small Pack“ also 2 für insgesamt 22 Stück.

Diese Art von Bonus hat sich in der Spieleindustrie durchgesetzt, da sie Spieler\*innen dazu verleiten soll, für einen besseren Deal mehr Geld auszugeben.

### Rangliste

Zu Beginn des Spiels liegt der eigene Punktestand bei 0, wobei bereits fiktive NPCs (Non Player Character) mit einem höheren Spielstand in der Liste vorhanden sind. Punkte werden beim Verkauf von Gemüse im Shop vergeben, Tabelle 2 erläutert den Punktewert für jedes einzelne Gemüse.

Anfangs erhalten die NPCs einen unterschiedlichen Punktefortschritt zwischen 0 und 126 Punkten. Erhalten Spielende aus jeglicher Quelle Punkte, erhöhen sich die der NPCs mit 50-prozentiger Wahrscheinlichkeit ebenfalls um einen zufälligen Wert (1, 4, 20 oder 100 Punkte). Die Chance wird für jeden NPC extra berechnet. Erhalten Spielende aber über 50 Punkte auf einmal, bekommen die NPCs eine zweite Chance ihre Punkte zu erhöhen, damit der/die Spieler\*in nicht zu schnell auf dem ersten Platz landet.

## Benutzeroberfläche

### Basisspiel/User-Interface

Hier sind einige Screenshots des User Interfaces (UI) aufgelistet, beginnend mit dem Startmenü (Abbildung 3), dann den Textfenstern (Abbildung 4) und Anleitungen (Abbildung 5)(Abbildung 6), die den Testpersonen beim Start des Spiels angezeigt werden.

Ein Bild, das Text, Screenshot, Plan, Karte enthält.

Automatisch generierte Beschreibung

Abbildung 3: Benutzeroberfläche – Startmenü

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 4: Benutzeroberfläche – Begrüßung

Ein Bild, das Text, Screenshot, Schrift, Rechteck enthält.

Automatisch generierte Beschreibung

Abbildung 5: Benutzeroberfläche – Steuerung

Ein Bild, das Text, Screenshot, Karte enthält.

Automatisch generierte Beschreibung

Abbildung 6: Benutzeroberfläche – Anleitung

Die folgenden Screenshots zeigen und erklären das User Interface im Spiel (Abbildung 7), welche Informationen die Testpersonen über Items erhalten können (Name, Bezeichnung und Seltenheit) und das Optionsmenü der Items (**Assign** Item to Hotbar, **Drop** Item) (Abbildung 8)(Abbildung 9).

Ein Bild, das Text, Screenshot, Karte enthält.

Automatisch generierte Beschreibung

Abbildung 7: Benutzeroberfläche – User Interface

Ein Bild, das Screenshot, Text enthält.

Automatisch generierte Beschreibung

Abbildung 8: Benutzeroberfläche – Item Information

Ein Bild, das Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 9: Benutzeroberfläche –   
Item Optionen

### Lootbox-Mechanik

Die Lootbox-Mechanik wurde in einen Premium-Shop integriert. Auf dieser Oberfläche sind der derzeitige Stand der Spielwährung „Coins“ und der Premiumwährung „Silver“ ersichtlich (Abbildung 10). Die Basis-Lootbox und Premium-Lootbox stehen nebeneinander und können direkt hier per linkem Mausklick geöffnet werden, solange die entsprechende Währung ausreichend verfügbar ist. Die Animation beginnt und nach kurzer Zeit erscheinen die drei zufällig ausgewählten Samen.

Die Fragezeichen neben den Kosten sind Buttons, die nach Betätigung ein Info-Fenster mit den entsprechenden Fundraten anzeigen.

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Abbildung 10: Benutzeroberfläche – Lootbox-Mechanik

### Premium-Shop

Der wichtigste Teil des Premium-Shops ist der Kauf von Premiumwährung mit Echtgeld (Abbildung 11). Wie bereits erwähnt, können Spielende im User-Interface Geld mit einem Button generieren. Dieses Geld kann ausschließlich hier ausgegeben werden.

Die Flags „!!!Most Popular!!!“ und „!!!Best Value!!!“ wurden an ähnliche Praktiken aus der Spieleindustrie aber auch an andere Bereiche angelehnt. Diese werden bei manchen Angeboten im Marketing hinzugefügt, dass diese die beliebtesten oder meistverkauften seien.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 11: Benutzeroberfläche – Premiumwährung Shop

Folgende Teile des Premium-Shops erweitern die Möglichkeiten „Coins“ wie auch „Silver“ auszugeben:

Die Erweiterung des Inventars um 4 Slots für „Silver“ (**Fehler! Verweisquelle konnte nicht gefunden werden.**).

Der Kauf von Farmlanderweiterungen für „Coins“ (weil spielrelevant) (Abbildung 13).

Der Kauf von Strukturen (hier der Brunnen) für „Silver“ (Abbildung 13) (Abbildung 14).

Der Umtausch von „Silver“ in „Coins“, für höheren Anreiz, Geld auszugeben (Abbildung 15).

Ein Bild, das Text, Screenshot, Software, Betriebssystem enthält.

Automatisch generierte Beschreibung

Abbildung 12: Benutzeroberfläche – Inventarerweiterung

Ein Bild, das Text, Screenshot, Software, Betriebssystem enthält.

Automatisch generierte Beschreibung

Abbildung 13: Benutzeroberfläche – Farmlanderweiterung

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Abbildung 14: Benutzeroberfläche – Strukturerweiterung

Ein Bild, das Text, Screenshot, Software, Multimedia enthält.

Automatisch generierte Beschreibung

Abbildung 15: Benutzeroberfläche – Umtausch

### Rangliste

Hier abgebildet ist die Rangliste, die den Wettbewerb zwischen den Spieler\*innen und NPCs simuliert (Abbildung 16). Dadurch bekommen Spielende den Anreiz, mehr Ressourcen zu investieren.

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 16: Benutzeroberfläche – Rangliste

## Softwarearchitektur

Anhand der in der Methodik im Kapitel 2.3.5 beschriebenen und gewählten Patterns wird der für die Monetarisierung relevante Teil der Softwarearchitektur erklärt.

Die Lootbox-Mechanik wird ebenfalls beschrieben.

### Grundgerüst

Die zentrale Logik, in der die meisten Funktionen zusammengeführt werden, befindet sich in einem globalen Skript mit dem Namen „Global.gd“. Dieses wird als Autoload-Skript geführt. Dadurch wird es als Erstes im Programm geladen, von allen anderen Skripts gesehen und kann sehr einfach referenziert werden. Es braucht keine eigene Instanz mehr erstellt zu werden, um auf Variablen und Funktionen zuzugreifen.

Um verschiedene Skripts miteinander zu verbinden, wurden auch Signale verwendet. Wird ein Signal von einem Skript ausgelöst, kann dieses in anderen Skripts empfangen werden, wenn ein Empfänger definiert wurde. So können zum Beispiel bei Datenänderungen im Hintergrund alle verbundenen Elemente informiert werden, um beispielsweise einen Wert an mehreren Stellen im User-Interface gleichzeitig zu aktualisieren.

### Lootbox-Mechanik

Da die Lootbox-Mechanik der zentrale Bestandteil dieser Arbeit ist, wird diese auch detaillierter beschrieben. In Godot wird jeder Gegenstand, jedes Level oder jedes Menü als eine eigene Szene erstellt.

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Jede Szene enthält eine sogenannte Root-Node (zu Deutsch: Knotenwurzel) hier „*Chest*“ (Abbildung 17). Diese Node ist ein Parent aller darunter liegenden Child-Nodes (hier zum Beispiel „*Chest*\_*Body*“, welche wiederum Parent von „CollisionShape2D“ ist).

Dadurch kann jedes dargestellte Objekt sehr individuell aufgebaut werden. Wird nun so eine Szene wie diese Lootbox in einem Level benötigt, muss nur die Szene selbst als neue Child-Node in das Level integriert werden. Sie ist dadurch sehr flexibel einsetzbar und kann leicht an mehreren Stellen gleichzeitig verwendet werden, ohne jedes Mal eine ganz neue Kiste programmieren zu müssen.

Abbildung 17: Lootbox Mechanik – Node-Übersicht

#### Erläuterung der einzelnen Nodes

* **Chest**

Art: Sprite2D

Aufgabe: Beinhaltet die Sprites (Grafiken) für das Aussehen der Lootbox und der Animation.

Außerdem wurde dieser Node das zugehörige Skript angehängt, welches sämtliche Logik beinhaltet (erkennbar an dem Pergamentsymbol rechts neben dem Namen).

* **Chest\_Body**

Art: StaticBody2D

Aufgabe: Definiert mithilfe der Child Node **CollisionShape2D** einen Bereich direkt über der Lootbox, der von Spielfiguren (auch NPCs) nicht betreten werden kann.

* **Area2D**

Art: Area2D

Aufgabe: Definiert mithilfe der Child Node **CollisionShape2D** einen Bereich vor der Lootbox. Befindet sich eine Spielfigur in diesem Bereich, kann die Lootbox geöffnet werden.

* **AnimationPlayer**

Art: AnimationPlayer

Aufgabe: Beinhaltet alle Animationen der Lootbox. Im Fall der Lootbox nur für das Öffnen. Hierfür werden die Sprites verwendet, die in Sprite2D hinzugefügt wurden.

* **AnimationTree**

Art: AnimationTree

Aufgabe: Enthält die erstellten Animationen als Baumstruktur mit einem Startknoten, der zur „Öffnen“-Animation führt.

* **CPUParticles2D**

Art: CPUParticles2D

Aufgabe: Enthält die Einstellungen des Partikeleffekts der Lootbox, der über die CPU berechnet wird. Dieser Partikeleffekt wird vom Skript gesteuert.

* **PointLight2D**

Art: PointLight2D

Aufgabe: Enthält die Einstellungen des Lichteffekts der Lootbox. Dieser Effekt wird vom Skript gesteuert.

#### Lootbox-Mechanik Skript

Zum Programmstart werden vor dem ersten Durchlauf des Skripts alle Variablen initialisiert und vorbereitet, die etwas von außerhalb des Skripts referenzieren oder holen (Abbildung 18).

Alle *„@onready*“ Variablen referenzieren alle Nodes der „*Chest*“-Szene, auf die während des Skripts zugegriffen wird. Außerdem wird eine neue Instanz der *ItemData*-Klasse erstellt, welche sämtliche Informationen über Samen und Gemüse enthält.

Die „*items*“ Variable referenziert eine Node von außerhalb der Szene, an die die nach dem Öffnen der Lootbox erzeugten Samen-Szenen angehängt werden, damit diese in der Spielwelt erscheinen.

Die Variable „*seeds*“ enthält später die Daten für die Samen.

„*player\_in\_range*“ wird immer dann auf „true“ gesetzt, wenn sich die Spielfigur in der **CollisionShape2D** der **Area2D** befindet.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 18: Lootbox Mechanik – Skript Variableninitialisierung

Die *„\_ready()*“ Funktion wird nur bei Initialisierung des Skripts einmal ausgeführt und weist der Variable „seeds“ den Rückgabewert der „*get\_seeds()*“-Methode zu, welche die Daten der Samen enthält.

„\_*process()*“ prüft, ob „*player\_in\_range*“ auf „true“ gesetzt ist und gleichzeitig die Interaktionstaste von der Testperson gedrückt wird. Ist dies der Fall, wird die Methode „*spawn\_items()*“ ausgeführt.

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 19: Lootbox Mechanik – \_ready- und \_process-Funktionen

Die beiden Funktionen “\_*on\_area\_2d\_body\_entered()*” und “*\_on\_area\_2d\_body\_entered()*” ändern die Variable “*player\_in\_range*” entweder zu „true“ oder „false“, je nachdem ob die Spielfigur den Bereich „**Area2D**“ betritt oder verlässt. Zusätzlich wird noch die UI aus- oder eingeblendet (Abbildung 20).

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 20: Lootbox Mechanik – Spielfigur im Bereich

„*spawn\_items()*“ ist die zentrale Funktion, die den Ablauf des gesamten Öffnungsmechanismus abarbeitet. Zuerst wird „*open\_chest\_animation()*“ aufgerufen.

Als Nächstes wird die Position der Items in der Spielwelt über die Position der Lootbox selbst definiert, damit die Saatgutpakete neben dieser erscheinen.

Dann werden einzeln die drei Saatgutpakete von der Funktion „*random\_seed\_generator()*“ erzeugt und mit den Positionsdaten der Funktion „*spawn\_items()* übergeben, welche nun die Instanziierung übernimmt und die Items in der Spielwelt platziert.

Im Anschluss wird noch eine Logik ausgeführt, die die Lootbox graduell unsichtbar macht und dann aus der Spielwelt löscht (Abbildung 21).

Ein Bild, das Text, Screenshot, Software enthält.

Automatisch generierte Beschreibung

Abbildung 21: Lootbox Mechanik – Zentrale Logik des Lootbox Mechanismus

„*open\_chest()*“ übernimmt die Animation und die Aktivierung der Licht- und Partikeleffekte (Abbildung 22).

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 22: Lootbox Mechanik – Lootbox Animation

“*random\_seed\_generator()*” erzeugt eine zufällige Zahl zwischen 1 und 100, ruft die Funktion „*get\_seeds\_by\_rarity()*“ mit der zugehörigen Rarität auf und gibt den Rückgabewert dieser (ein zufälliges Saatgutpaket einer bestimmten Rarität) zurück (Abbildung 23).

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 23: Lootbox Mechanik – Zufallsgenerator Saatgutrarität

In „*get\_seeds\_by\_rarity()*“ wird zuerst das Array “seeds” durch die gegebene Rarität gefiltert. Anschließend wird ein zufälliger Eintrag im neuen Array „filtered seeds“ ausgewählt und zurückgegeben (Abbildung 24).

Ein Bild, das Text, Screenshot, Software, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 24: Lootbox Mechanik – Zufällige Saatgutauswahl

In der Funktion „*spawn\_item()*“ wird eine neue Szene für ein Saatgutpaket vorbereitet, mit den nötigen Daten wie Name, Typ, Rarität, Position ausgestattet und an die externe Node „items“ angehängt und initialisiert. Dadurch erscheint das Paket in der Spielwelt und kann verwendet werden (Abbildung 25).

Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 25: Lootbox Mechanik – Instanzierung der einzelnen Saatgut-Szenen

### Umgesetzte Patterns

* **Wait to Play:** Die Wartezeiten wurden beim Pflanzenwachstum implementiert. Jede Instanz eines gepflanzten Samens erhält je nach Rarität einen eigenen Zeitwert, der als Float übergeben wird (10s für Rarität 1, 15s für Rarität 2, 30s für Rarität 3 und 45s für Rarität 4). Bis auf eine Ausnahme (5 Stufen) hat jede Pflanze 4 Stufen und für jede Stufe wird ein Timer mit der übergebenen Zeitspanne gestartet. Somit warten die Testpersonen bei einer Pflanze der Rarität 1 insgesamt 30s, bis diese geerntet werden kann.
* **Grinding:**

Das Grinding konnte nur in sehr geringem Ausmaß realisiert werden. Bedingt durch die fehlende Möglichkeit das Spiel zu speichern, konnten nur niedrige Wartezeiten beim Pflanzenwachstum gewählt werden, damit die Testpersonen das Spiel auch in einer Sitzung zielführend testen können. Durch die Game Loop (Abbildung 26**Fehler! Verweisquelle konnte nicht gefunden werden.**) wird aber trotzdem ein länger andauernder Test ermöglicht.

* **Can’t Pause or Save:**

Aufgrund zu hoher Komplexität und zu hohem damit verbundenen Zeitaufwand bei der Integration einer automatischen Speicherfunktion, wurde darauf verzichtet. Zusätzlich müsste das Spiel selbst nach dem erneuten Start den Zeitraum zwischen den Spielsessions berechnen und eventuell Pflanzen fertig wachsen lassen und Punktestände der NPCs anpassen. Daher müssen die Testpersonen das Spiel in einer einzigen Sitzung testen und können zwischendurch das Spiel nicht beenden.

* **Infinite Treadmill:**

Eine Game Loop wurde so realisiert, dass immer wieder von Neuem Samen gekauft und angepflanzt werden müssen. Anfangs erhalten Testpersonen eine Grundmenge an der Spielwährung „Coins“, um Basis-Lootboxen zu kaufen und die ersten Samen anzupflanzen. Nach der Ernte wird das Gemüse verkauft, die Testpersonen erhalten „Coins“ und Punkte für die Rangliste und die erhaltenen „Coins“ können wieder für neue Lootboxen oder andere Inhalte im Premium-Shop ausgegeben werden (Abbildung 26**Fehler! Verweisquelle konnte nicht gefunden werden.**).

Ein Bild, das Text, Diagramm, Entwurf, Kreis enthält.

Automatisch generierte Beschreibung

Abbildung 26: Softwarearchitektur – Grinding Cycle

* **Premium Currency:**

Eine Premiumwährung „Silver“ wurde in das Spiel implementiert (Abbildung 11). Am Beispiel der beiden kleinsten kaufbaren Packs ist ersichtlich, dass zuerst der Kontostand, welcher im „Global.gd“-Skript zentral gespeichert wird, mit dem zum Kauf nötigen Betrag verglichen wird. Anschließend wird je ein Signal mit dem neuen Echtgeld-Kontostand und dem Bestand an der Premiumwährung „Silver“ ausgesendet. Somit erhalten alle Empfänger ein Update. Zuletzt wird noch der „Spent“-Betrag erhöht, welcher den Gesamtwert des ausgegebenen Echtgeldes repräsentiert (Abbildung 27).

Ein Bild, das Text, Screenshot, Software, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 27: Softwarearchitektur – Premiumwährung

* **Pay-to-Skip** oder **Pay-to-Win:**

Für das Spiel wurde eine Pay-to-Win-Mechanik ausgewählt, da Pay-to-Skip wesentlich mehr zeitliche Ressourcen erfordert hätte.

Es wurden zwei Varianten der Lootbox implementiert, eine Basisvariante, kaufbar durch die Spielwährung „Coins“, und eine Premiumvariante, kaufbar durch die Premiumwährung „Silver“ (Abbildung 10).

Die Premiumvariante hat eine doppelt so hohe Chance auf Rarität 3 und 4 Samen wie die Basis-Lootbox.

* **Artifical Scarcity:**

Eine Form von künstlicher Verknappung wurde ebenfalls aufgrund zu hoher Ressourcenkosten nicht implementiert. Geplant gewesen wäre aber ein einmaliges Sonderangebot, welches nach ungefähr 5 Minuten im Spiel getriggert worden wäre und der Testperson ein besonders günstiges Angebot unterbreitet hätte, wie 5 Premium-Lootboxen zum Preis von einer oder doppelte Premiumwährung auf den nächsten Einkauf.

* **Competition:**

Eine sinnvolle Komponente, die eine Art Wettbewerb ausreichend simuliert, ist ein Leaderboard (zu Deutsch: Rangliste). Um diese Rangliste sinnvoll umzusetzen, ist es wichtig darauf zu achten, dass sie sich so organisch wie möglich anfühlt. Starten die Testpersonen zum ersten Mal das Spiel und öffnen die Rangliste, zeigt der eigene Punktestand 0 Punkte und 15 andere simulierte NPCs haben zum Großteil bereits Punkte gesammelt und auch schon einen hohen Punktestand (Abbildung 16).

Sämtliche Daten bezüglich der Punkte werden in einem Dictionary-Array gespeichert. Jeder Eintrag beinhaltet den Namen und die dazugehörigen Punkte (Abbildung 28Abbildung 28: Softwarearchitektur – Leaderboard Dat**Fehler! Verweisquelle konnte nicht gefunden werden.**).

Ein Bild, das Text, Screenshot, Schrift, Karte Menü enthält.

Automatisch generierte Beschreibung

Abbildung 28: Softwarearchitektur – Leaderboard Data

Videospiele werden in Schleifen ausgeführt, immer ein Frame (Bild) nach dem anderen. Die *\_process(\_delta*) Funktion wird bei jedem einzelnen Frame einmal aufgerufen (Abbildung 29). Hier wird in jedem Frame bei *testLB()* zuerst geprüft, ob die Taste für die Rangliste gedrückt wurde und zwischen den Zuständen „Anzeigen“ und „Nicht anzeigen“ wechselt. Der Code darunter überprüft bei jedem Durchlauf, ob sich die Spielerpunkte geändert haben. Falls ja, wird der Punktestand jedes NPCs mit einer Chance von 50% um einen zufälligen Wert erhöht. Verdient eine Testperson über 50 Punkte auf einmal, wird *update\_leaderboard\_data()* ein zweites Mal aufgerufen.

Am Ende wird das Array mit den Spielerdaten geordnet, damit die höchste Punktezahl am Anfang und die niedrigste am Ende stehen.

Ein Bild, das Text, Screenshot, Schrift, Software enthält.

Automatisch generierte Beschreibung

Abbildung 29: Softwarearchitektur – Update Leaderboard Data

Von *show\_leaderboard()* und *hide\_leaderboard()* werden zentral die Erzeugung und Darstellung beziehungsweise das korrekte Auflösen und Verbergen der Rangliste gesteuert.

*clear\_leaderboard()* ist dafür verantwortlich, dass alle Zeilen in der Rangliste gelöscht werden, damit die Rangliste beim nächsten Aufruf neu befüllt werden kann (Abbildung 30).

Ein Bild, das Text, Screenshot, Software enthält.

Automatisch generierte Beschreibung

Abbildung 30: Softwarearchitektur – Show and Hide Leaderboard

Ein weiterer wichtiger Teil der Ranglisten-Logik ist das Erstellen der neuen Zeilen mit den aktuellen Daten. Das Ordnen der Daten passiert beim Prüfen des Punktestandes, sodass in *add\_label()* bei jedem Aufruf der Rangliste nur jede Zeile neu eingefügt wird (Abbildung 31).

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 31: Softwarearchitektur – Labels hinzufügen

## Spieltest und Umfrage

Das Spiel wurde als Windows, MacOS und Linux Testversion zur Verfügung gestellt. Diese wurden in einen Google-Drive-Ordner hochgeladen, wobei der Link zu diesem bei der Ausschreibung angegeben wurde.

Für die Umfrage wurde die Onlinebefragungsplattform SoSci Survey verwendet [16]. Diese bietet eine kostenfreie Onlineumfrage für nicht kommerzielle Hintergründe. Dieses Tool ist sehr hochwertig und bietet viele verschiedene Möglichkeiten, die Umfrage zu gestalten sowie eine sehr detaillierte Datenerhebung.

In 3.5.1 werden alle gestellten Fragen und die zugehörigen Antwortmöglichkeiten aufgelistet. In 3.5.2 werden diese Fragen erläutert und die Ergebnisse der Umfrage präsentiert.

Der Spieltest und die Umfrage wurden zwischen dem 27.4.2024 und einschließlich dem 6.5.2024 durchgeführt.

### Fragenkatalog

**Demografische Fragen**

1. Wie alt sind Sie? (12-99)

Antwortmöglichkeiten: Zahl zwischen 12 und 99

**Fragen zu Erfahrung mit Lootboxen in Videospielen**

1. Haben Sie jemals ein oder mehrere Spiele mit einer Lootbox-Mechanik gespielt?

(z.B. FiFa UT, Fortnite, Genshin Impact, ...)

Antwortmöglichkeiten: Ja, Nein, Weiß ich nicht

1. Haben Sie jemals eine Lootbox geöffnet (auch ohne Echtgeld-Einsatz)?

Antwortmöglichkeiten: Ja, Nein, Weiß ich nicht

1. Haben Sie jemals eine Lootbox mit Echtgeld gekauft?

Antwortmöglichkeiten: Ja, Nein, Keine Angabe

1. Haben Sie allgemein schon einmal Geld in Videospielen ausgegeben, Lootboxen ausgeschlossen (z.B. Kleidung, Skins für Charaktere, neue Geschichten, Beschleunigungen, ...)?

Antwortmöglichkeiten: Ja, Nein, Keine Angabe

1. Wie viel Zeit verbringen Sie pro Woche in Videospielen allgemein?

Antwortmöglichkeiten: 00:00 bis 56:59 (Stunden:Minuten)

1. Wie viel Zeit davon ebenfalls wöchentlich in Videospielen mit einer Lootbox-Mechanik (Stunden:Minuten)?

Antwortmöglichkeiten: 00:00 bis 56:59 (Stunden:Minuten)

**Fragen zur Erfahrung mit Glücksspielen**

1. Haben Sie für Glücksspiele jemals Geld ausgegeben?

Antwortmöglichkeiten: Ja, Nein, Keine Angabe

1. Wie oft spielen Sie Glücksspiele?

Antwortmöglichkeiten: täglich, wöchentlich, monatlich, nie

**Fragen zum Spieltest**

1. Haben sie sich im Spiel schnell zurechtgefunden?

Antwortmöglichkeiten: Ja, Eher ja, Eher nein, Nein

1. War die Lootbox-Mechanik Ihrer Meinung nach ansprechend gestaltet?

Antwortmöglichkeiten: Ansprechend, Weniger ansprechend, Nicht ansprechend

1. Waren Sie wegen einer oder mehrerer Spielmechaniken dazu verleitet, Echtgeld auszugeben (Egal ob Sie dann auch wirklich welches ausgegeben haben)?

Antwortmöglichkeiten: Ja, Nein, Kann ich nicht sagen

1. Falls Sie noch nie Spiele mit einer Lootbox-Mechanik gespielt oder Lootboxen gekauft haben, können Sie sich nach dem Spieltest vorstellen, dass Sie für diese Mechanik offen wären und eventuell sogar Geld dafür ausgeben würden?

Antwortmöglichkeiten: Ja, Eher ja, Eher nein, Nein

1. Falls Sie noch etwas zum Spiel anmerken wollen (maximal 255 Zeichen):

(keine Pflichtfrage)

**Dateneingabe zum Spieltest**

1. Abschließend bitte noch die Daten eintragen, die nach dem Beenden des Spiels angezeigt werden. (Im Pause Menü auf "Finish" klicken)

Abgefragte Daten: Achieved Points, Money Spent, Played Time, Premium-Shop Time

1. Vielen Dank für die Teilnahme am Spieltest und der Umfrage!

Sie können unten gerne noch allgemeine Anmerkungen zum Fragebogen anfügen.

(keine Pflichtfrage)

### Erläuterung der Fragen und Ergebnisse der Umfrage

Insgesamt haben 13 Personen am Spieltest und der Umfrage teilgenommen, wobei eine Person den Fragebogen vor der Dateneingabe beendet hat.

Das durchschnittliche Alter betrug 28.

#### Demografische Fragen

Es wurde bewusst nur nach dem Alter gefragt, um den Fragebogen möglichst kurz und simpel zu halten. Fragen nach anderen Daten wie beispielsweise dem Geschlecht der Testperson oder dem Verdienst wurden daher nicht gestellt, hätten aber potenziell interessant sein können, um genauere Ergebnisse zu erhalten und diese besser vergleichen und besser bewerten zu können.

**Ergebnisse:**

Jüngste Testperson: 21

Älteste Testperson: 37

Testpersonen wurden hauptsächlich im Studiengang Informatik der Fachhochschule gesucht, wobei auch Freund\*innen, Bekannte und Verwandte inkludiert wurden. Daher ist die Altersverteilung wie erwartet ausgefallen.

#### Fragen zu Erfahrungen mit Lootboxen in Videospielen

Zuallererst werden Fragen nach den bisherigen Erfahrungen mit Spielen, die eine Lootbox-Mechanik implementiert haben, gestellt. Auch, ob die Testpersonen bereits Echtgeld für Lootboxen ausgegeben haben. Wie viele Stunden und Minuten pro Woche mit Videospielen verbracht werden, wird ebenfalls abgefragt.

**Zu 2.:** Um herauszufinden, ob die Testpersonen schon Erfahrung mit Lootboxen sammeln konnten und daher eventuell anders an das Testspiel herangehen. Sie können potenziell sämtliche Mechaniken schneller verstehen und testen.

**Zu 3.:** Ist dahingehend interessant, ob die Testpersonen Lootbox-Mechaniken aktiv nutzen, egal ob ohne oder mit Bezahlung.

**Zu 4.:** Haben Testpersonen bereits Geld investiert, sind diese mit höherer Wahrscheinlichkeit auch bei anderen Spielen mit Lootbox-Mechanik eher bereit, Geld auszugeben.

Der Gesamtbetrag an Geld, den Spielende bereits in Lootboxen investiert haben, sind sensible Daten und wurde daher nicht abgefragt. Die Frage hätte aber entschärft werden können, indem eine Auswahl an Bereichen erfragt worden wäre, wie 0€, 0€ - 10€, 10€ - 50€, über 50€ pro Woche oder Monat.

**Zu 5.:** Haben die Testpersonen überhaupt schon einmal Geld innerhalb eines Spiels ausgegeben, kann ebenfalls erkannt werden, ob sie dies gezielt für bestimmte Inhalte gemacht haben oder aber auch in Lootboxen investiert haben.

**Zu 6. und 7.:** Auch die Spielzeit der Testpersonen ist von Wichtigkeit, da diese einen Hang zu Videospielen aufweisen kann, mit dem auch eine gewisse Verpflichtung einhergehen kann.

**Ergebnisse:**

Bis auf zwei Testpersonen haben alle bereits Erfahrung mit Videospielen, die eine Lootbox-Mechanik aufweisen.

Bis auf eine Testperson haben auch alle bereits Lootboxen geöffnet, wobei 7 von 13 Testpersonen noch nie für Lootboxen bezahlt haben.

Zusätzlich haben nur 3 von 7 dieser Personen auch noch nie für andere Inhalte in Spielen Geld ausgegeben.

Die Spielzeit unter den Testpersonen variiert stark. Die drei Testpersonen, die kein Geld für digitale Inhalte ausgegeben haben, sind auch diejenigen, die die wenigste Zeit mit Videospielen verbringen, von gar nicht bis zu 2 Stunden. Der Rest der Testpersonen verbringt ungefähr zwischen 10 und 35 Stunden pro Woche mit Videospielen, wobei diese Zeit bei Einzelnen bei bis zu 50 bis über 57 Stunden liegt.

Die verbrachte Zeit in Spielen mit einer Lootbox-Mechanik fällt weitaus niedriger aus. Hier spielen die meisten gar keine derartigen Spiele und wenn, dann einzelne bis zu 5 und 10 Stunden pro Woche.

#### Fragen zur Erfahrung mit Glücksspielen

Hier wird nach den Glücksspielgewohnheiten der Testpersonen gefragt. Es geht darum, ob die Testpersonen bereits Geld dafür ausgegeben haben oder regelmäßig spielen.

**Zu 8. und 9.:** Diese beiden Fragen ermöglichen das Finden eines Zusammenhangs zwischen dem Spielen von Glücksspielen und der Verwendung von Lootboxen in Videospielen.

**Ergebnisse:**

Für Glücksspiele haben 7 von 13 Testpersonen bereits Geld ausgegeben, wobei eine einzige Testperson monatlich Glücksspiele spielt. Diese eine Person ist Teil der Gruppe, die am seltensten Videospiele spielt.

#### Fragen zum Spieltest

In der dritten Kategorie werden die Erfahrungen und die Meinung zum Spieltest abgefragt.

**Zu 10.:** Diese Frage klärt, ob der Einstieg in das Spiel als entspannt und leicht verständlich empfunden wurde. Dies ist wichtig, damit Testpersonen eher dabeibleiben und das Spiel spielen/testen, weil sie durch komplizierte Mechaniken und wenig Erklärungen eher abgeschreckt werden könnten.

**Zu 11.:** Eine ansprechende Lootbox-Mechanik ist wichtig, damit diese auch aktiv genutzt wird und die Testpersonen dazu ermutigt, Geld für Spielfortschritt zu investieren.

**Zu 12.:** Die Frage, ob die implementierte Lootbox-Mechanik zu Echtgeldeinsatz verleitet, wurde ebenfalls als wichtig erachtet. Dazu gehört auch, wie essenziell die Verwendung dieser Funktion, sei es auch nur die Basisvariante, ist.

**Zu 13.:** Ergänzend zu Frage 12 wird hier noch gefragt, ob die Testpersonen sich vorstellen können, dass sie für eine solche Mechanik der Monetarisierung offen wären und sogar Geld ausgeben würden. Dies könnte ebenfalls wichtig sein, um zu verstehen, wie viele Testpersonen sich auf derartige Methoden einlassen.

**Ergebnisse:**

Im Spiel fanden sich 7 Testpersonen im Spiel eher gut zurecht und 5 sehr gut. Nur eine, die auch Erfahrung mit Videospielen hat, fand sich weniger gut zurecht.

6 Testpersonen empfanden die Umsetzung der Lootbox-Mechanik als ansprechend gestaltet, 6 weniger ansprechend und eine Person nicht ansprechend.

6 Testpersonen gaben an, wegen einer oder mehrerer Spielmechaniken dazu verleitet worden zu sein, Geld auszugeben, wobei 6 Testpersonen dies verneinten. Eine Person konnte es nicht sagen.

3 Testpersonen gaben an, sich vorstellen zu können, für Lootbox-Mechaniken offen zu sein und eventuell sogar Geld auszugeben. 4 Testpersonen gaben an, eher nicht dafür offen zu sein und 6 gar nicht.

Die 10 Testpersonen, die „Eher nicht“ und „Nein“ angekreuzten, gaben auch an, keine Spiele mit Lootboxen zu spielen.

#### Anmerkungen der Testpersonen

**Wörtliche Zitate aus dem Fragebogen:**

„Es ist ein nettes kurzweiliges Spiel bis man auf Platz eins ist“

* Grundsätzlich war geplant, dass das Spiel nur für eine kurze Spielphase interessant ist und Inhalte bietet. Vielleicht hätten ein paar kleinere Anpassungen in der Punktevergabe ausgereicht, aber der Fokus lag nicht auf einem ausgeklügelten Simulationsalgorithmus der Rangliste.

„Inventar ist startk begrenzt, ich hab aufgehört als meine Früchte nicht mehr reingepasst haben“

* Dass das Inventar von vornherein stark begrenzt ist, war eine bewusste Designentscheidung, um die Testpersonen zu einem langen Grind oder dem Einsatz von Geld zu bewegen. Das Inventar ist im Premiumshop-Menü mit der Premiumwährung erweiterbar.

„Sehr nett gestaltetes Spiel, hat Spaß gemacht; Mir war es im Spiel sehr wichtig, das Inventory zu vergrößern, um dann auch möglichst viele Pflanzen auf einmal verkaufen zu können-habe schnell gemerkt, dass mir auch mein Feld zu klein ist - habe es vergr.“

* Anhand dieser Anmerkung kann nachvollzogen werden, dass die Testperson sich mit den verschiedenen Mechaniken auseinandergesetzt und aktiv genutzt hat. Sie hat auch etwas Geld investiert.

„fürn Test immer nur 0,40c adden zu können war ein bisschen nervig“

* Die Funktion, einen kleinen Geldbetrag auf Knopfdruck hinzuzufügen, ist bewusst mit einem gewissen Aufwand verbunden. Testpersonen werden dadurch auch ein wenig daran gehindert, Geld auszugeben.

„Ich habe mir schwer getan in das Spiel hineinzufinden und war manchmal nicht in der Lage, die Setzlinge anzupflanzen. Die Spielestatistik könnte irreführend sein, ich habe Geld ausgegeben, um zu testen was passiert, ohne die gekauften Setzlinge zu nutzen.“

* Leider konnte keine umfassende Testphase durchgeführt werden und dadurch sind einige „Quality of Life“-Fehler im Gameplay zurückgeblieben. Beispielsweise konnten keine reifen Pflanzen geerntet werden, wenn in der Hotbar ein mit einem Item belegter Platz ausgewählt war.

Eventuell bezieht sich die Testperson auch auf den kleinen Aktionsradius der Spielfigur. Das Anpflanzen von Saatgut selbst gibt dem User wenig visuelles Feedback.

Auch wenn eine Testperson das gekaufte Saatgut nicht nutzt oder nutzen kann. Das Geld wurde trotzdem dafür ausgegeben.

„Die Pflanzen sind zu schnell gewachsen um mich zum kauf von silver zu bewegen. Am Weg zu geschäft und zurück waren die pflanzen wieder erntereif -> somit kein bedarf für den brunnen. Die seltenheits wslkeit war zu niedrig für echtgeld imo“

* Vielleicht hätten die Werte für das Pflanzenwachstum etwas erhöht werden können, allerdings war das Ziel dieses Spieltests auch eine kurze Testphase von ungefähr 15 bis 30 Minuten. Und da das Spiel keine Form von Spielstandspeicherung enthält, wäre eine wesentlich länger Wachstumsphase nicht sinnvoll gewesen.

Um aktuelle „Free-to-Play“-Spiele realitätsnaher nachzubilden, wurde die Wahrscheinlichkeit für die seltensten Samen bewusst so niedrig angesetzt. Allerdings wäre aufgrund der kurzen Testphase auch eine höhere Wahrscheinlichkeit sinnvoll gewesen, um die Testbarkeit des Spiels zu erhöhen.

„Gutes Spiel, benötigt etwas rebalancing. Hohe Punkte wegen exploit. Feedback bezieht sich jedoch auf pre-exploit zeit.“

* Das Balancing hätte durch eine kleine Testphase im Vorhinein verbessert werden können, um eine sinnvollere Testumgebung für die tatsächliche Umfrage bieten zu können.

Die Beseitigung von Exploits, mehr Feedback bei Spielerinteraktion und generelle Verbesserungen in der Quality-of-Life wären dadurch ermöglicht worden.

#### Erhobene Spieldaten

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr.** | **Achieved Points** | **Money spent (€)** | **Played time (sec)** | **Premium-Shop time (sec)** | **Percentage of Premium-Shop time to Player time** |
| 1 | 584,00 | 49990,00 | 1189,00 | 312,00 | 0,26 |
| 2 | 579,00 | 0,00 | 3349,00 | 480,00 | 0,14 |
| 3 | 76,00 | 0,00 | 887,00 | 73,00 | 0,08 |
| 4 | 181,00 | 0,00 | 885,00 | 47,00 | 0,05 |
| 5 | 204,00 | 349,93 | 1036,00 | 194,00 | 0,19 |
| 6 | 8,00 | 1,99 | 702,00 | 38,00 | 0,05 |
| 7 | 370,00 | 27,98 | 2472,00 | 261,00 | 0,11 |
| 8 |  |  |  |  |  |
| 9 | 181,00 | 11,97 | 1790,00 | 416,00 | 0,23 |
| 10 | 0,00 | 149,97 | 581,00 | 161,00 | 0,28 |
| 11 | 14,00 | 7,99 | 678,00 | 62,00 | 0,09 |
| 12 | 0,00 | 0,00 | 129,00 | 0,00 | 0,00 |
| 13 | 8863,00 | 0,00 | 1749,00 | 212,00 | 0,12 |

Tabelle 6: Spieltest und Umfrage - Erhobene Spieldaten

**Berechnete Werte:**

Durchschnittliche Ausgaben aller Testpersonen, die Geld investiert haben: 85,69€

Durchschnittliche Spielzeit aller Testpersonen: 21,45 Minuten

**Potenziell fehlerhafte Eingaben:**

Dass Testperson 1 hier wirklich 49990€ ausgegeben hat, ist eher unwahrscheinlich, außer es wurde eine Art Clickbot verwendet (Ein Programm, dass für den User automatisch sehr schnell klickt.). Daher kann davon ausgegangen werden, dass es sich um einen Eingabefehler handelt und wahrscheinlich 49,99€ ausgegeben wurden.

Testperson 8 hat das Spiel vielleicht getestet, aber in den letzten Teil des Fragebogens die Daten aus dem Spieltest nicht eingefügt.

**Interpretationen:**

Da eine Testperson keine Spieldaten angegeben hat, wurde diese exkludiert. Daher wird im Folgenden Teil von maximal 12 Testpersonen gesprochen.

2 Testpersonen haben keine Punkte durch den Verkauf von Feldfrüchten erhalten, wobei eine davon 149,97€ ausgegeben hat.

5 Testpersonen haben das Spiel ohne Echtgeldeinsatz getestet, von denen eine den höchsten Punktestand erreicht hat, ohne die höchste Spielzeit aufzuweisen.

Der prozentuelle Anteil der im Premiumshop verbrachten Zeit zu der gesamten Spielzeit liegt zwischen 5% und 28%. 8 von 12 Testpersonen haben über 10% der Spielzeit im Premiumshop verbracht. Daraus kann geschlossen werden, dass der Shop ein wichtiger Teil des Spiels ist, der auch wirklich genutzt wurde.

#### Bezug auf die Forschungsfragen:

Wird bedacht, dass es sich um einen Spieltest handelt, bei dem kein echtes Geld investiert wurde, muss das Ergebnis mit Vorsicht betrachtet werden. Hier wären Daten aus der Industrie bei echten Spielen wesentlich aussagekräftiger. Womöglich würde kein Unternehmen diese Daten aber veröffentlichen, einerseits aus Datenschutzgründen, aber auch, da damit vermutlich geheimes firmeninternes Gedankengut öffentlich werden würde.

Im Durchschnitt haben 7 Testpersonen (welche auch wirklich Geld ausgegeben haben), 86,69€ ausgegeben, wobei hier aber Ausreißer Richtung 149,97€ und 349,93€ vorhanden sind. Vergleicht man hier die Höhe des investierten Geldes mit der Spieldauer, kann Folgendes erkannt werden:

Bis auf zwei Ausnahmen haben Testpersonen, die mehr Geld investiert haben, auch mehr Zeit im Spiel und im prozentuellen Anteil mehr Zeit im Premiumshop verbracht. Die zwei Ausnahmen sind Testpersonen, die nichts investiert haben und trotzdem sehr lange getestet haben. Hier könnte angenommen werden, dass diese durch die fehlende Premiumwährung mehr „grinden“ (durch viel Zeit und eintönige Arbeit wird Fortschritt erlangt) mussten, um einen ähnlichen Fortschritt zu erlangen.

Testpersonen, die auch angegeben haben, dass sie für eine Investition in Lootboxen offen wären, haben im Vergleich zu anderen mehr Geld während des Spieltests ausgegeben.

Die Testpersonen, die wiederum angegeben haben, nicht für solche Mechaniken offen zu sein, haben bis auf zwei Testpersonen kein bis wenig Geld im Spieltest ausgegeben. Diese zwei Testpersonen haben im Vergleich mehr Geld ausgegeben, haben aber im Fragebogen angegeben, dass sie sehr wenig bis gar keine Zeit für Videospiele aufwenden. Daher kann in diesem Fall eher davon ausgegangen werden, dass die Mechaniken lediglich ausprobiert wurden.

# Diskussion

## Potenzial

Die Ergebnisse und Erkenntnisse dieser Bachelorarbeit können zu einer Weiterentwicklung von Umfragen zum Thema Videospiel- und Glücksspielsucht beitragen, indem diese mit einem vorangehenden Spieltest erweitert werden und die Daten mit in die Auswertung einfließen.

Dieser neue Ansatz in diesem Bereich der Forschung könnte zu stichhaltigeren Ergebnissen führen, die mehr Bewusstsein schaffen würden und einen besseren Schutz und für Spieler\*innen ermöglichen könnten.

In den Ergebnissen des durchgeführten Spieltests und der anschließenden Umfrage konnten Zusammenhänge erkannt werden, die wahrscheinlich in größerem und professionellerem Umfang deutlicher sind.

## Bezug auf die Forschungsfragen

1. **Welche Rolle spielen Lootboxen bei der Veränderung des Spielverhaltens von Spieler\*innen hinsichtlich Spieldauer?**

Testpersonen, die Geld ausgegeben haben und im Vergleich auch mehr investierten, haben mehr Zeit im Spiel, sowie einen höheren Anteil davon im Premium-Shop verbracht. Eine Testperson mit der höchsten Spielzeit hat kein Geld ausgegeben und eine andere hat im Vergleich mehr Geld ausgegeben, aber eine geringere Spielzeit.

Aus diesen Daten kann darauf geschlossen werden, dass eine Integration von Lootboxen die Spielzeit von Spieler\*innen erhöht, welche bereits Erfahrung mit Lootboxen haben und dafür auch offen sind.

1. **Wie beeinflusst die Integration von Lootboxen in Videospielen die Bereitschaft von Spieler\*innen, Geld für In-Game-Transaktionen auszugeben?**

Aus den Daten kann darauf geschlossen werden, dass Spieler\*innen, die bereits Erfahrung mit Videospielen und einer Lootbox-Mechanik haben, sowie für diese Form der Monetarisierung offen sind und Geld investieren würden, auch im Spieltest im Vergleich mehr Geld ausgegeben haben als andere Testpersonen.

Daher kann davon ausgegangen werden, dass eine Integration einer Lootbox-Mechanik oder generell eines Premiumshops dafür zugängliche Spieler\*innen verleitet, Geld auszugeben.

## Probleme

Da im Rahmen dieser Bachelorarbeit nur ein kurzes Testspiel mit simplen Mechaniken aufgrund knapper Zeitressourcen in der Entwicklung und Testung programmiert werden konnte, sind die Ergebnisse lediglich eingeschränkt interpretierbar.

Hier müsste ein größeres, professionelles Entwicklungsteam mit der Hilfe von Expert\*innen in der Glücksspiel- und Gaming-Psychologie ein solides Spiel konzeptionieren und entwickeln. Dieses müsste dann wahrscheinlich von mehreren Tausend Personen aus verschiedenen Ländern getestet werden, um wirklich aussagekräftig zu sein und reale Zusammenhänge zu finden.

**Konzeptionierung, Entwicklung und Release**

In der Konzeptphase war die Herausforderung, Quellen und Beispiele zu finden, wie „Free-to-play“-Spiele und Lootboxen funktionieren. Hier half auf jeden Fall die eigene Erfahrung mit diesem Spielgenre und der Lootbox-Mechanik. Trotzdem war es nicht einfach, ein solches Spiel, das komplett auf Monetarisierung ausgerichtet ist, zu planen und zu entwickeln.

Während der Entwicklung des Spiels kamen immer wieder Probleme auf, da die Erfahrung in der Spielentwicklung und vor allem auch mit der Engine fehlte. Viele Ideen wurden aufgrund der Komplexität verworfen oder wesentlich simpler umgesetzt. Beispiel hierfür sind im Code vorhandene spezielle Animationen der Spielfigur, wie verschiedene Werkzeuge benutzt werden. Diese fehlen im fertigen Spiel, da die Entwicklung der dahinterliegenden Funktionen eine hohe Zahl der vorhandenen Ressourcen vereinnahmt hätte.

In der Release-Version war zumindest ein Exploit vorhanden, der es erlaubte, Items im Inventar zu duplizieren.

Manche Designentscheidungen waren beim Spieltest eher hinderlich. Zum Beispiel konnte kein Gemüse geerntet werden, wenn in der Hotbar ein Gegenstand ausgewählt war, obwohl Platz im Inventar gewesen wäre.

## Mögliche Verbesserungen

### Testspiel

Unter anderem wäre für die Entwicklung des Spiels ein größeres Team und mehr Zeit notwendig gewesen, um mehr der in Kapitel 2.3.2 beschriebenen Patterns in das Spiel integrieren zu können und diese auch besser aufeinander abzustimmen und zu optimieren.

Vor dem Start des Spieltests und der Umfrage wäre ein umfangreicher Softwaretest mit 2-3 Testpersonen sinnvoll gewesen, um Softwarefehler und Exploits zu beheben sowie die Spielerfahrung im Sinne der Spielbarkeit und der Zugänglichkeit zu verbessern.

Wahrscheinlich wäre eine Integration des Spiels auf einer Website sinnvoll gewesen, um den Fragebogen leichter mit dem Spiel zu verbinden und die erhobenen Daten, ohne den Zwischenschritt der Eingabe durch die Testperson, direkt zu speichern. Dies hätte Eingabefehler vermieden.

Ein weiterer Vorteil wäre die Kompatibilität. Testpersonen müssten sich keine Software herunterladen, sondern könnten von fast jedem Betriebssystem aus mit einem aktuellen Internetbrowser auf den Spieltest zugreifen.

### Fragebogen

Eine Aufwertung des Fragebogens durch mehr demografische Fragen wie Geschlecht und Einkommen könnte mehr Aufschluss über Ergebnisse liefern, um bessere Vergleiche unter den Testpersonen anstellen zu können.

Ebenso könnten tiefergehende Fragen zu Erfahrungen mit Lootboxen und den damit verbundenen Ausgaben gestellt werden.

## Abschluss

Die gewonnenen Erkenntnisse und Daten des Spieltests und Fragebogens bieten einen kleinen Einblick in das, was in einem professionelleren und größeren Umfang möglich wäre.

Interne Daten von Videospiel-Publishern über Lootboxen und das Spielverhalten sowie die Ausgaben der Spieler\*innen würden neu durchgeführte Studien aufwerten und aussagekräftiger machen. Da diese internen Daten aber fehlen ist der Ansatz, ein Testspiel zu entwickeln und die Daten daraus mit denen von Umfragen zu kombinieren sicher eine Möglichkeit, viele Wissenslücken im Bereich des Suchtverhaltens im Gaming und im Vergleich zur Glücksspielsucht zu schließen.

Literaturverzeichnis

|  |  |
| --- | --- |
| [1] | J. Clement, „Activision Blizzard Revenue by Composition 2020,“ Statista, 05 2023. [Online]. Available: https://www.statista.com/statistics/274761/electronic-arts-ea-extra-content-revenues/. [Zugriff am 01 12 2023]. |
| [2] | J. Clement, „Activision Blizzard Revenue by Composition 2020,“ Statista, 02 2023. [Online]. Available: https://www.statista.com/statistics/1208560/activision-blizzards-revenue-by-composition/. [Zugriff am 01 12 2023]. |
| [3] | M. Wille, „game,“ game Verband der deutschen Games-Branche, 09 04 2024. [Online]. Available: https://www.game.de/marktdaten/umsatz-mit-computer-und-videospielen-steigt-weiter/. [Zugriff am 29 04 2024]. |
| [4] | A. Castro und J. Alexander, „Nintendo, Microsoft, and Sony commit to cisclose Drop Rates for Loot Boxes,“ The Verge, 08 2019. [Online]. Available: https://www.theverge.com/2019/8/7/20758626/nintendo-microsoft-sony-loot-box-drop-rate-disclosure-video-games. [Zugriff am 01 12 2023]. |
| [5] | C. J. Larche, K. Chini, C. Lee, M. J. Dixon und M. Fernandes, „Rare Loot Box rewards trigger larger arousal and reward responses, and greater urge to open more Loot Boxes,“ *Journal of Gambling Studies,* Bd. 37, pp. 141-163, 11.2019. |
| [6] | B. Abarbanel, „Gambling vs. gaming: a Commentary on the Role of regulatory, industry, and Community Stakeholdersin the Loot Box Debate,“ *Gaming Law Review,* Bd. 22, pp. 231-234, 05.2018. |
| [7] | D. Zendle und P. Cairns, „Video Game Loot Boxes are Linked to Problem gambling: Results of a large-scale Survey,“ *PLOS ONE,* Bd. 13, Nr. 11, pp. 1-12, 11.2018. |
| [8] | J. Koubek, Monetarisierung von Computerspielen, Berlin: Nomos, 2021. |
| [9] | K. Yokomitsu, T. Irie, H. Shinkawa und M. Tanaka, „Characteristics of Gamers who Purchase Loot Box: a Systematic Literature Review,“ *Current Addiction Reports,* Bd. 8, Nr. 4, pp. 481-493, 01 12 2021. |
| [10] | „onlinesicherheit.at,“ A-SIT Zentrum für sichere Informationstechnologie – Austria, 16 05 2023. [Online]. Available: https://www.onlinesicherheit.gv.at/Services/News/Dark-Patterns.html#:~:text=Der%20Begriff%20%E2%80%9EDark%20Patterns%E2%80%9C%20bezeichnet,in%20ihren%20Entscheidungen%20beeinflusst%20werden.. [Zugriff am 14 05 2024]. |
| [11] | Best-Gaem-Studio, „Steam,“ Pay2Win: The Tricks Exposed, 2015. [Online]. Available: https://store.steampowered.com/app/416760/Pay2Win\_The\_Tricks\_Exposed/. [Zugriff am 30 01 2024]. |
| [12] | P. Luban, „Game Developer,“ 22 11 2011. [Online]. Available: https://www.gamedeveloper.com/design/the-design-of-free-to-play-games-part-1. [Zugriff am 17 04 2024]. |
| [13] | J. Linietsky, A. Manzur und Contributors, „Godot Engine,“ Godot Foundation, [Online]. Available: https://godotengine.org/. [Zugriff am 17 04 2024]. |
| [14] | J. Linietsky, A. Manzur und Community, „Godot Documentation,“ Godot Foundation, [Online]. Available: https://docs.godotengine.org/de/4.x/index.html. [Zugriff am 17 04 2024]. |
| [15] | C. Nooble, „Sprout Lands Asset Pack,“ [Online]. Available: https://cupnooble.itch.io/sprout-lands-asset-pack. [Zugriff am 17 04 2024]. |
| [16] | „SoSci Survey,“ SoSci Survey GmbH, [Online]. Available: https://www.soscisurvey.de/. [Zugriff am 02 05 2024]. |
| [17] | „Games Wirtschaft,“ 21 02 2024. [Online]. Available: https://www.gameswirtschaft.de/wirtschaft/usk-statistik-2023-altersfreigaben-lootboxen/. [Zugriff am 16 04 2024]. |

Abbildungsverzeichnis

[Abbildung 1: Einleitung – Umsatz mit Computer- und Videospielen in Deutschland 8](#_Toc167024366)

[Abbildung 2: Methodik – Screenshot des Spiels aus der Vertriebsplattform Steam 16](#_Toc167024367)

[Abbildung 3: Benutzeroberfläche – Startmenü 26](#_Toc167024368)

[Abbildung 4: Benutzeroberfläche – Begrüßung 26](#_Toc167024369)

[Abbildung 5: Benutzeroberfläche – Steuerung 27](#_Toc167024370)

[Abbildung 6: Benutzeroberfläche – Anleitung 27](#_Toc167024371)

[Abbildung 7: Benutzeroberfläche – User Interface 28](#_Toc167024372)

[Abbildung 8: Benutzeroberfläche – Item Information 28](#_Toc167024373)

[Abbildung 9: Benutzeroberfläche – Item Optionen 28](#_Toc167024374)

[Abbildung 10: Benutzeroberfläche – Lootbox-Mechanik 29](#_Toc167024375)

[Abbildung 11: Benutzeroberfläche – Premiumwährung Shop 30](#_Toc167024376)

[Abbildung 12: Benutzeroberfläche – Inventarerweiterung 31](#_Toc167024377)

[Abbildung 13: Benutzeroberfläche – Farmlanderweiterung 31](#_Toc167024378)

[Abbildung 14: Benutzeroberfläche – Strukturerweiterung 31](#_Toc167024379)

[Abbildung 15: Benutzeroberfläche – Umtausch 31](#_Toc167024380)

[Abbildung 16: Benutzeroberfläche – Rangliste 32](#_Toc167024381)

[Abbildung 17: Lootbox Mechanik – Node-Übersicht 33](#_Toc167024382)

[Abbildung 18: Lootbox Mechanik – Skript Variableninitialisierung 35](#_Toc167024383)

[Abbildung 19: Lootbox Mechanik – \_ready- und \_process-Funktionen 35](#_Toc167024384)

[Abbildung 20: Lootbox Mechanik – Spielfigur im Bereich 36](#_Toc167024385)

[Abbildung 21: Lootbox Mechanik – Zentrale Logik des Lootbox Mechanismus 36](#_Toc167024386)

[Abbildung 22: Lootbox Mechanik – Lootbox Animation 37](#_Toc167024387)

[Abbildung 23: Lootbox Mechanik – Zufallsgenerator Saatgutrarität 37](#_Toc167024388)

[Abbildung 24: Lootbox Mechanik – Zufällige Saatgutauswahl 38](#_Toc167024389)

[Abbildung 25: Lootbox Mechanik – Instanzierung der einzelnen Saatgut-Szenen 38](#_Toc167024390)

[Abbildung 26: Softwarearchitektur – Grinding Cycle 40](#_Toc167024391)

[Abbildung 27: Softwarearchitektur – Premiumwährung 40](#_Toc167024392)

[Abbildung 28: Softwarearchitektur – Leaderboard Data 42](#_Toc167024393)

[Abbildung 29: Softwarearchitektur – Update Leaderboard Data 42](#_Toc167024394)

[Abbildung 30: Softwarearchitektur – Show and Hide Leaderboard 43](#_Toc167024395)

[Abbildung 31: Softwarearchitektur – Labels hinzufügen 43](#_Toc167024396)

Tabellenverzeichnis

[Tabelle 1: Lootboxen – Droprates/Fundraten 21](#_Toc167024397)

[Tabelle 2: Lootbox – Mechanik Seltenheit und Wert der Samen 22](#_Toc167024398)

[Tabelle 3: Premium-Shop – Kosten der Inventarerweiterung 23](#_Toc167024399)

[Tabelle 4: Premium-Shop – Kosten der Farmlanderweiterung 23](#_Toc167024400)

[Tabelle 5: Premium-Shop – Kosten der "Silver" Pakete 24](#_Toc167024401)

[Tabelle 6: Spieltest und Umfrage - Erhobene Spieldaten 51](#_Toc167024402)

Abkürzungsverzeichnis

|  |  |
| --- | --- |
| NPC | Non Player Character |
| DLC | Downloadable Content |
| RPG | Role-Playing Game |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Anhang A: Ergebnisse der Umfrage

**Umfrage Teil 1:**

![Ein Bild, das Text, Zahl, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung]()

**Umfrage Teil 2:**

Ein Bild, das Text, Zahl, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

**Umfrage Teil 3:**

Ein Bild, das Text, Screenshot, Zahl, parallel enthält.

Automatisch generierte Beschreibung

**Umfrage Teil 4:**

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

Anhang B: Auswahl relevanter Skripts aus dem Spiel

**Skript „shop\_chest\_basic.gd“ (Basis-Lootbox):**

extends Sprite2D

class\_name lootbox\_basic

@onready var particles = $CPUParticles2D

@onready var light = $PointLight2D

@onready var animation = $AnimationTree

@onready var state\_machine = animation.get("parameters/playback")

@onready var item\_data = ItemData.new()

@onready var items = $"../Items"

@onready var button = $Button

@onready var InfoMessage = $InfoMessage

@onready var Prompt = $InfoMessage/Prompt

@onready var timer = $InfoMessage/Timer

signal update\_chest\_coins

var seeds

var player\_in\_range = false

var fade\_out\_duration = 2.0 # Duration of the fade-out in seconds

var fade\_out\_timer = 0.0 # Tracks the progress of the fade-out

func \_ready():

InfoMessage.hide()

seeds = item\_data.get\_seeds()

# handles the opening of the chest

func open\_chest():

state\_machine.travel("opening")

await get\_tree().create\_timer(0.5).timeout

light.enabled = true

print("light enabled: ", light.enabled)

await get\_tree().create\_timer(0.2).timeout

particles.emitting = true

# generates a random number and returns a seed with the sidgnated rarity level

func random\_seed\_generator():

var lucky\_number = randf\_range(0.0, 99.0)

if lucky\_number <= 0.5:

return get\_seeds\_by\_rarity(4)

elif lucky\_number <= 9.5:

return get\_seeds\_by\_rarity(3)

elif lucky\_number <= 55.0:

return get\_seeds\_by\_rarity(2)

else:

return get\_seeds\_by\_rarity(1)

# returns a random seed of the given rarity level

func get\_seeds\_by\_rarity(rarity):

var filtered\_seeds = []

for item in seeds:

if item.item\_rarity == str(rarity):

filtered\_seeds.append(item)

if filtered\_seeds.size() == 0:

return {} # Return an empty dictionary if no vegetables match the rarity

# Seed the random number generator

RandomNumberGenerator.new().randomize()

# Select a random index in the range of the filtered list

var random\_index = randi() % filtered\_seeds.size()

# Return the randomly selected vegetable

return filtered\_seeds[random\_index]

# handles the creation of the 3 selected random seeds

func spawn\_items():

button.disabled = true

# generate rewards

var rewards = []

for i in range(0, 3):

var reward = random\_seed\_generator()

rewards.append(reward)

# Define position of items

var position\_array = [Vector2(global\_position.x -15, global\_position.y + 5), Vector2(global\_position.x, global\_position.y + 8), Vector2(global\_position.x + 15, global\_position.y + 5)]

# Instantiate and spawn Items from chest

for i in range(0, 3):

spawn\_item(rewards[i], position\_array[i])

await open\_chest()

await get\_tree().create\_timer(2.0).timeout

for item in items.get\_children():

item.get\_child(0).visible = true

# spawns one specific seed in the gameworld

func spawn\_item(data, item\_position):

var item\_scene = preload("res://Scenes/Shop\_Item.tscn")

var item\_instance = item\_scene.instantiate()

item\_instance.initiate\_items(data["item\_type"], data["item\_name"], data["item\_rarity"], data["texture"], data["item\_value"], data["item\_sellable"])

item\_instance.global\_position = item\_position

item\_instance.z\_index = 1

item\_instance.get\_child(0).visible = false

items.add\_child(item\_instance)

# handles the functionality of the buy button

func \_on\_button\_pressed():

if Global.coins >= 5:

update\_chest\_coins.emit(Global.coins - 5)

await spawn\_items()

await get\_tree().create\_timer(2.5).timeout

reset\_chest()

else:

show\_prompt("Sry, you don't have enough coins to buy a chest...")

# resets the chest after it has been opened

func reset\_chest():

light.enabled = false

particles.emitting = false

state\_machine.travel("Idle")

button.disabled = false

# shows a prompt with changeable information

func show\_prompt(text, duration = 3.0):

# Set the message

Prompt.text = text

# Start the timer with the specified duration

timer.start(duration)

# Show the panel

InfoMessage.show()

# hides the info message after timer runs out

func \_on\_timer\_timeout():

InfoMessage.hide()

**Skript „ingame\_shop\_menu.gd“ (Premium Shop):**

extends Control

@onready var premium\_label = $PanelContainer/VBoxContainer/HBoxContainer/Premium\_Amount

@onready var premium\_Items = $"PanelContainer/VBoxContainer/TabContainer/Seed Box/HBoxContainer/Premium\_Chest/PanelContainer/Items"

@onready var premium\_Chest = $"PanelContainer/VBoxContainer/TabContainer/Seed Box/HBoxContainer/Premium\_Chest/PanelContainer/Shop\_Chest"

@onready var premium\_Chest\_Container = $"PanelContainer/VBoxContainer/TabContainer/Seed Box/HBoxContainer/Premium\_Chest/PanelContainer"

@onready var coins\_label = $PanelContainer/VBoxContainer/HBoxContainer/Coin\_Amount

@onready var basic\_Items = $"PanelContainer/VBoxContainer/TabContainer/Seed Box/HBoxContainer/Basic\_Chest/PanelContainer/Items"

@onready var timer = $InfoMessage/Timer

@onready var InfoMessage = $InfoMessage

@onready var Prompt = $InfoMessage/Prompt

@onready var farmland\_label = $PanelContainer/VBoxContainer/TabContainer/Farmland/MarginContainer/VBoxContainer/HBoxContainer/Price\_farmland

@onready var well\_structure\_btn = $PanelContainer/VBoxContainer/TabContainer/Structure/PanelContainer/VBoxContainer/Button

@onready var exchange\_text = $PanelContainer/VBoxContainer/TabContainer/Exchange/MarginContainer/VBoxContainer/exchange\_text

signal update\_shop\_wallet

signal update\_ingots

signal update\_farmland

signal well\_bought

signal update\_exchange\_coins

var UI\_node

var inv\_space\_price = 2

var farmland\_price = 25

var elapsed\_time = 0

func \_ready():

InfoMessage.hide()

coins\_label.text = str(Global.coins)

premium\_label.text = str(Global.silver\_ingots)

%Price\_invspace.text = str(inv\_space\_price)

farmland\_label.text = str(farmland\_price)

UI\_node = get\_tree().get\_first\_node\_in\_group("UI")

func resume():

get\_tree().paused = false

func pause():

get\_tree().paused = true

func open\_shop\_menu():

if Input.is\_action\_just\_pressed("shop\_menu") and get\_tree().paused == false:

show()

UI\_node.hide()

pause()

$GameTimer.start()

elif Input.is\_action\_just\_pressed("shop\_menu") and get\_tree().paused == true:

hide()

UI\_node.show()

on\_resume()

resume()

$GameTimer.stop()

func on\_resume():

var loot = premium\_Items.get\_children()

var basic\_loot = basic\_Items.get\_children()

if loot != null:

for item in loot:

# send loot to Inventory and delete the items

item.pickup\_item()

if basic\_loot != null:

for item in basic\_loot:

# send loot to Inventory and delete the items

item.pickup\_item()

resume()

visible = false

func \_process(\_delta):

open\_shop\_menu()

premium\_label.text = str(Global.silver\_ingots)

coins\_label.text = str(Global.coins)

var items = premium\_Items.get\_children()

var basic\_items = basic\_Items.get\_children()

if !$"PanelContainer/VBoxContainer/TabContainer/Seed Box".visible and (items != null or basic\_items != null):

for i in items:

i.hide()

for i in basic\_items:

i.hide()

elif $"PanelContainer/VBoxContainer/TabContainer/Seed Box".visible and (items != null or basic\_items != null):

for i in items:

i.show()

for i in basic\_items:

i.show()

# Price\_invspace

# Label Name for inventory space price

func \_on\_btn\_add\_inv\_space\_pressed():

if Global.silver\_ingots >= inv\_space\_price:

# increase space by 4 and reduce player silver ingots

Global.increase\_inventory\_size(4)

update\_ingots.emit(Global.silver\_ingots - inv\_space\_price)

# set up new price and update label

inv\_space\_price \*= 2

%Price\_invspace.text = str(inv\_space\_price)

premium\_label.text = str(Global.silver\_ingots)

show\_prompt("Congratulations! You just bought 4 more inventory slots!")

else:

show\_prompt("Sry, you don't have enough silver ingots...")

# Functions for buying ingame premium currency

func \_on\_btn\_buy\_tiny\_pressed():

if Global.wallet >= 1.99:

update\_shop\_wallet.emit(Global.wallet - 1.99)

update\_ingots.emit(Global.silver\_ingots + 5)

premium\_label.text = str(Global.silver\_ingots)

Global.player\_spent += 1.99

show\_prompt("Thank you for buying the tiny pack to support us a little.")

else:

show\_prompt("You don't have enough money in your wallet...")

func \_on\_btn\_buy\_small\_pressed():

if Global.wallet >= 7.99:

update\_shop\_wallet.emit(Global.wallet - 7.99)

update\_ingots.emit(Global.silver\_ingots + 22)

premium\_label.text = str(Global.silver\_ingots)

Global.player\_spent += 7.99

show\_prompt("Thank you for buying the small pack to support us a little more.")

else:

show\_prompt("You don't have enough money in your wallet...")

func \_on\_btn\_buy\_medium\_pressed():

if Global.wallet >= 19.99:

update\_shop\_wallet.emit(Global.wallet - 19.99)

update\_ingots.emit(Global.silver\_ingots + 55)

premium\_label.text = str(Global.silver\_ingots)

Global.player\_spent += 19.99

show\_prompt("Thank you for buying the medium pack to support us.")

else:

show\_prompt("You don't have enough money in your wallet...")

func \_on\_btn\_buy\_large\_pressed():

if Global.wallet >= 49.99:

update\_shop\_wallet.emit(Global.wallet - 49.99)

update\_ingots.emit(Global.silver\_ingots + 138)

premium\_label.text = str(Global.silver\_ingots)

Global.player\_spent += 49.99

show\_prompt("Thank you for buying the large pack to support us like a boss!")

else:

show\_prompt("You don't have enough money in your wallet...")

func show\_prompt(text, duration = 3.0):

# Set the message

Prompt.text = text

# Start the timer with the specified duration

timer.start(duration)

# Show the panel

InfoMessage.show()

func \_on\_timer\_timeout():

InfoMessage.hide()

# handle farmland shop

func \_on\_btn\_add\_farmland\_pressed():

if farmland\_price <= 2000:

if Global.coins >= farmland\_price:

update\_farmland.emit(Global.coins - farmland\_price)

farmland\_price \*= 2

farmland\_label.text = str(farmland\_price)

coins\_label.text = str(Global.coins)

show\_prompt("Congratulations on your new farmland! Happy farming!")

if farmland\_price >= 2000:

farmland\_label.text = "---"

else:

show\_prompt("You don't have enough coins for buying new farmland.")

else:

show\_prompt("Wow, you bought all the available farmland on this small island!")

func \_on\_info\_basic\_pressed():

show\_prompt("Drop rates:\nRarity 1: 35%\nRarity 2: 55%\nRarity 3: 9.5%\nRarity 4: 0.5%")

func \_on\_info\_premium\_pressed():

show\_prompt("Drop rates:\nRarity 1: 30%\nRarity 2: 50%\nRarity 3: 19%\nRarity 4: 1%")

func \_on\_button\_pressed():

if Global.silver\_ingots >= 50:

update\_ingots.emit(Global.silver\_ingots - 50)

well\_structure\_btn.disabled = true

well\_structure\_btn.text = "sold"

Global.well = true

well\_bought.emit()

show\_prompt("You just bought access to the well!")

else:

show\_prompt("You don't have enough silver ingots for buying new farmland.")

func \_on\_btn\_exchange\_pressed():

var change = int(exchange\_text.text.strip\_edges())

if change != null and change > 0:

if Global.silver\_ingots >= change:

update\_ingots.emit(Global.silver\_ingots - change)

update\_exchange\_coins.emit(Global.coins + (change \* 5))

premium\_label.text = str(Global.silver\_ingots)

show\_prompt("Exchange completed!")

else:

show\_prompt("Exchange failed, not enough silver ingots")

else:

show\_prompt("Exchange failed, input invalid")

func \_on\_game\_timer\_timeout():

elapsed\_time += 1

print("Elapsed Time: ", elapsed\_time)

func \_exit\_tree():

$GameTimer.stop()

Global.premium\_timer = elapsed\_time

**Skript “leaderboard.gd” (Rangliste):**

extends Control

@onready var leaderboard\_list = $MarginContainer/PanelContainer/VBoxContainer/List

var player\_data = [{"name" : "GiantChestnut", "points" : 126},

{"name" : "Adventurer\_1123095", "points" : 105},

{"name" : "GreenGecko", "points" : 99},

{"name" : "InnocentLazy", "points" : 88},

{"name" : "ShadyDeep", "points" : 76},

{"name" : "LittleTangerine", "points" : 61},

{"name" : "HellBeast", "points" : 52},

{"name" : "StormCloud", "points" : 48},

{"name" : "FitEros", "points" : 33},

{"name" : "DevilsHand", "points" : 19},

{"name" : "Adventurer\_3451827", "points" : 12},

{"name" : "LittleBig", "points" : 8},

{"name" : "Lazarus", "points" : 5},

{"name" : "RandomName", "points" : 0},

{"name" : "Pea", "points" : 0}]

func \_ready():

player\_data.append({"name" : Global.player\_name, "points" : Global.player\_points})

func initialize\_list():

var lb\_element = preload("res://Scenes/leaderbord\_element\_menu.tscn")

var element = lb\_element.instantiate()

element.set\_variables("Rank", "Player Name", "Points")

leaderboard\_list.add\_child(element)

func show\_leaderboard():

clear\_leaderboard() # Clear existing labels

initialize\_list() # Initialize list with "Rank", "Player Name" and "Points".

add\_label() # Add new labels

show() # Show the Leaderboard

pause() # Pause the game

func hide\_leaderboard():

clear\_leaderboard() # Clear existing labels

hide() # Hide the Leaderboard

resume() # Resume the game

func clear\_leaderboard():

# Clear all children in the VBoxContainer.

for child in leaderboard\_list.get\_children():

child.queue\_free()

func calculate\_points(offset):

# probaility 20% if points are added

var add\_points = randi\_range(1, 100)

if add\_points <= 50:

var rnd\_points = randi\_range(1, 100)

if rnd\_points < (10 + offset):

return 100

if rnd\_points < 30:

return 20

if rnd\_points < 60:

return 4

return 1

return 0

func update\_leaderboard\_data():

var cnt = 10

for npc in player\_data:

if npc["name"] == Global.player\_name:

npc["points"] = Global.player\_points

npc["points"] += calculate\_points(cnt)

if cnt > 0:

cnt -= 1

# Custom comparison function

func compare\_points(a, b):

return a["points"] > b["points"] # Return true if 'a' has fewer points than 'b'

func sort\_array():

player\_data.sort\_custom(compare\_points)

func add\_label():

var cnt = 1

for i in player\_data:

# Create a new Label node

var lb\_element = preload("res://Scenes/leaderbord\_element.tscn")

var player\_lb\_element = preload("res://Scenes/leaderbord\_element\_player.tscn")

var element = lb\_element.instantiate()

var player\_element = player\_lb\_element.instantiate()

# Add the label to the VBoxContainer

if i["name"] == Global.player\_name:

player\_element.set\_variables(str(cnt), i["name"], i["points"])

leaderboard\_list.add\_child(player\_element)

else:

element.set\_variables(str(cnt), i["name"], i["points"])

leaderboard\_list.add\_child(element)

cnt += 1

func resume():

get\_tree().paused = false

func pause():

get\_tree().paused = true

# handles calling of menu

func testLB():

if Input.is\_action\_just\_pressed("leaderboard") and get\_tree().paused == false:

show\_leaderboard()

elif Input.is\_action\_just\_pressed("leaderboard") and get\_tree().paused == true:

hide\_leaderboard()

func \_process(\_delta):

testLB()

# every time the player updates own points, npcs get the chance to add points too.

if Global.player\_points > Global.old\_player\_points:

if (Global.player\_points - Global.old\_player\_points) >= 50:

update\_leaderboard\_data()

update\_leaderboard\_data()

sort\_array()

Global.old\_player\_points = Global.player\_points