

Game Design Dokument

Stand: 03.07.2022

Arbeitstitel:

Hugo's Mission



Autor Christian Kitte, HS Emden-Leer

Modul Game Development – SS 2022 – Medieninformatik Online, Master

Inhaltsverzeichnis

Zusammenfassung	1
Zielformat (Version 1)	1
Spielgenre.....	1
Zielgruppe (Version 2)	1
Unique Selling Proposition USP (Version 1)	1
Technologien	1
Game Engine (Version 2)	1
CRYENGINE.....	2
Godot	3
Unity.....	3
Unreal Engine	4
Ergebnis.....	5
Technik Stack?	5
Spieldetails	6
Kurzbeschreibung (Version 2)	6
Charaktere und Zielobjekte (Version 4)	7
Hugo	7
Aussehen.....	7
Eigenschaften	7
Aktionen und Animationen.....	7
Bewegungsmöglichkeiten	7
Geister	7
Aussehen.....	7
Eigenschaften	8
Aktionen und Animationen.....	8
Bewegungsmöglichkeiten	8
Zwischenziele und finales Levelziel	8
Zwischenziel „Banane“	8
Zwischenziel „Halloween Kürbis“	9
Zwischenziel „Wand“	9
Finales Levelziel.....	10

Punkteberechnung.....	10
Level (Version 3)	11
Startlevel	12
Beschreibung des Levels	12
Aufgabe des Spielers	12
Auftretende Charaktere/Spielfiguren/Elemente	12
Wie wird das Level beendet.....	13
Skalierung der Folgelevel	13
Menü (Version 2).....	13
Haupt-Menü	13
Untermenü Tastatur	14
Untermenü Einstellungen	15
Interaktions- und Navigationsdiagramme.....	16
Ingame Menü	Fehler! Textmarke nicht definiert.
Head Up Display	17
Steuerung (in Bearbeitung).....	19
Gameplay (Version 2)	19
Abbildungen	21
Tabellen	22

Zusammenfassung

Zielplattform (Version 1)

Um die im Abschnitt Unique Selling Proposition USP ausgeführten Ziele zu erreichen ist die Vermeidung eines zu grossen Aufwandes sinnvoll. Hierbei liegen vor allen Aufwendungen wie die einer Installation oder Einrichtung im Fokus.

Auch wenn grundsätzlich die Generierung einer lokalen Anwendung auf Basis der existierenden Assets möglich ist, wird aus den genannten Gründen das zu entwickelnde Spiel primär online und Browserbasiert als WebGL Anwendung verfügbar gemacht werden.

Als ein weiterer Vorteil wird hierbei auch die einfachere Umsetzung und Veröffentlichung von Bugfixes, Patches sowie Erweiterungen gesehen.

Spielgenre

Bei dem Spiel handelt es sich um ein 3D Geschicklichkeitsspiel.

Zielgruppe (Version 2)

Als Zielgruppe werden Personen allen Alters und Geschlechts gesehen, welche eine nicht zu komplexe und somit leicht zugängliche Ablenkung haben möchten. Auf Grund fehlender Gewalt und in Game Verkäufen, wird von einer Altersfreigabe ab 6 Jahren ausgegangen.

Insbesondere soll die Gewaltfreiheit auch durch das im Spiel verwendete Wording und Storytelling befördert werden. So stirbt der Spieler nicht, sondern schläft ein, um in den vorherigen Level wieder aufzuwachen.

Unique Selling Proposition USP (Version 1)

Das Spiel verfolgt keine komplexen Strategien oder Ziele. Primäres Ziel ist das Erreichen einer hohen Punktzahl und Verewigung in der Bestenliste. Ein typisches, einfaches Spiel, um sich „zwischendurch“ zu entspannen und abzuschalten.

Über die Möglichkeit, sich Vorteile zu verschaffen und der Einbeziehung des Zufalls werden Spielverläufe immer unterschiedlich, können jedoch vom Spieler beispielsweise durch Zwischenziele beeinflusst werden. Es existiert immer eine Möglichkeit, noch weitere Level zu meistern (im Prinzip ohne Ende), was zu einem guten Reply Wert führt.

Technologien

Game Engine (Version 2)

Für die Umsetzung des Spiels existieren verschiedene Optionen. Durch die Vorgabe zur Verwendung

einer Game Engine scheiden alle Arten der manuellen Umsetzung aus. Durch die Entscheidung der primär webbasierten Veröffentlichung, treten hierfür förderliche Funktionalität in den Vordergrund.

Bei der hier vorgestellten Auswahl erfolgt ein kurzer und nicht zwingend objektiver Vergleich möglicher Game Engines. Für einen formlosen Vergleich sollen hierbei insbesondere die folgenden sechs Dimensionen näher betrachtet werden:

1. **Bedeutung:** Die Game Engine sollte etabliert sein und eine gewisse Bedeutung in der Industrie haben, um das erworbene Wissen verwertbarer zu machen.
2. **Support:** Im Umfeld der Game Engine sollte eine gute Community existieren. Ebenso sollten Dokumentation und Tutorials von Seiten des Herstellers verfügbar sein.
3. **Assets:** Vom Hersteller sollten Assets und Erweiterungen kostengünstig verfügbar und einfach zu erreichen sein.
4. **Kosten:** Der auch kommerzielle Einsatz sollte mit möglichst geringen finanziellen Belastungen möglich sein und nicht zu laufenden Kosten zur Laufzeit führen.
5. **Programmierbarkeit:** Die Programmierbarkeit ist ein für die Arbeit essenzieller Punkt. Es sollten nach Möglichkeit bereits bekannte Sprachen unterstützt werden.
6. **Verfügbarkeit und Einsatzgebiet:** Insbesondere für den Einsatz spielt die webbasierte Verfügbarkeit eine wichtige Rolle. Grundsätzlich ist die Zahl verfügbarer Plattformen für den Editor und dem ausführbaren Produkt ein wichtiges Kriterium. Neben Konsolen als Zielplattform, sollten mindestens Windows, Linux sowie Web unterstützt werden.

Nach einer ersten Recherche ergeben sich in alphabetischer Reihenfolge als mögliche Kandidaten¹:

- CRYENGINE (<https://www.cryengine.com/>)
- Godot (<https://godotengine.org/>)
- Unity (<https://unity.com/>)
- Unreal Engine (<https://www.unrealengine.com/>)

CRYENGINE

1. Es handelt sich um eine ausgereifte Game Engine. Mit ihr wurden bekannte Titel wie "Battle Cry" oder "Prey" umgesetzt. Aktuell² ist CRYENGINE 5.7 mit Long Term Support (LTS) veröffentlicht. Trotz interessanter Fähigkeiten liegt der Fokus weniger auf der Industrie.
2. Wikipedia listet eine Reihe von Communities im Umfeld von CRYENGINE. Daneben existieren auf dessen Homepage Verweise auf weitere Communities. Die Seite bietet ein sehr großes Angebot an Informationen und videounterstützte Tutorials zu allen Aspekten der Engine. Es existieren Zertifizierungen zum "Technical Artist", "Designer" sowie "Programmer".
3. Über die Website gelangt man zur „Asset Database“ mit einem großen Angebot an Assets, Plugins sowie Skripten.
4. Leider sind die Angaben zu den Kosten eher unübersichtlich. Letztlich ist die Verwendung der Game Engine an sich kostenlos möglich. Der Quellcode ist verfügbar und die vollständigen Vermarktungsrechte stehen offen. Seit 2018 müssen 5% des Gewinns abgegeben werden, sofern Einnahmen von 5000 USD per Release und Game überschritten werden. Der Vertrieb

¹ Es handelt sich hierbei ausdrücklich um eine subjektive erste Auswahl.

² Juni 2022

über den Marketplace ist jedoch frei. Negativ wirkt der fast immer gegebene Zwang zur Registrierung von Spielen. Im Kommerziellen Bereich behält sich der Hersteller eine variable Abgabe vor.

5. Um alle Aspekte beim Skripten bedienen zu können, sind solide C++ Kenntnisse notwendig. vorhanden sein. Daneben wird C# in Teilen unterstützt.
6. Der Einsatzbereich umfasst das gesamte Spektrum an Plattformen und Anwendungsgebiete. Diese reicht von Spieleentwicklung bis hin zu Virtual Reality (VR) und Augmented Reality (AR) Anwendungen.

Godot

1. Trotz Open Source handelt es sich bei Godot um eine ausgereifte Game Engine. Bemerkenswert ist die Organisation von Szenen in einer Baumstruktur sowie die Trennung in zwei getrennte Engines für 2D und 3D. Die neueste³ Version ist 3.5 RC 2. Spiele wie "Kingdom of the Dump" oder "Dungeondraft" wurden mit ihr umgesetzt. Die Engine existiert erst seit 2014 und ist im industriellen Umfeld fast nicht existent.
2. Als Open Source wird die Engine zu einem Großteil von seiner Community getragen. Einen übersichtlichen Zugang zu ihr findet man auf der Homepage. Das Angebot erscheint jedoch typisch Open Source eher gemischt.
3. Über die Homepage ist eine „Asset Library“ mit diversen Erweiterungen und Assets verfügbar.
4. Bei der Engine handelt es sich um Open Source (MIT License). Somit werden keine Abgaben oder Lizenzgebühren fällig. Erstellte Ergebnisse können jedoch ausdrücklich monetarisiert werden.
5. Godot ist programmierbar. Verwendet werden GD Script (Python ähnlich), C# 8.0 Mono, C++. Daneben ist visuelles Scripting möglich. Sprachen wie Rust, Nim oder D werden durch die Community ermöglicht. Die Anzahl verfügbarer Sprachen ist somit sehr groß, verlangt jedoch teilweise extra Einstellungen.
6. Der Einsatzbereich umfasst alle grundlegenden Anwendungsgebiete einer Game Engine bis hin zur Unterstützung von VR und AR. Teamwork wird nativ unterstützt. Zielplattformen sind Windows, macOS, Linux sowie HTML5 WebAssembly (Editor und Projekte) und zusätzlich Android, iOS sowie Konsole (Projekte).

Unity

1. Unity wurde im Jahr 2005 gegründet und hat es geschafft, einen bemerkenswert hohen Bekanntheits- und Verbreitungsgrad zu erreichen. Nicht selten werden Kenntnisse im Rahmen von Stellenangeboten nachgefragt. Spiele wie „D.R.O.N.E.“ oder „Pray for the Gods“ wurden mit Unity erstellt. Neben der Spieleentwicklung hat Unity einen starken Bezug zur Industrie und VR sowie AR.
2. Unity verfügt über eine umfangreiche Community. Im Internet und insbesondere auf YouTube existiert zudem eine kaum überschaubare Zahl an Informationen und Tutorien. Über die Website ist der kostenlose Zugang zu umfangreichen Dokumentationen, sowie einer Vielzahl an geführten und gut ausgearbeiteten Lernpfaden möglich. Es werden Lizenzierungen zum „Certified User“, „Certified Associate“, „Certified Professional“ sowie

³ Juni 2022

„Certified Expert“ angeboten.

3. Über die Homepage ist der Zugang zu einem „Assetstore“ mit einer umfangreichen Auswahl an Erweiterungen und Assets möglich. Viele davon sind zudem bereits kostenlos verfügbar. Über die kostenlosen Lernangebote wird zudem der Zugang zu exklusiven Assets angeboten. Unity betreibt sehr viel Aufwand zu Integration und Bindung seiner Nutzer.
4. Die Nutzung von Unity ist auch kommerziell bis zu einem Einkommen 100k USD in den letzten 12 Monaten möglich. Für höhere Einkommen greift ein Abo Modell. Grundsätzlich erhält man Zugriff auf das vollständige Programm und die Dokumentation und Lehrmaterial. Unterschiede zeigen sich im Support und zusätzlichen Funktionen, welche für einzelne Entwickler und kleine Teams zunächst eher weniger wichtig sind. Im Gegenzug werden keine Laufzeitabgaben fällig.
5. Unity selbst wurde in C++ geschrieben, jedoch wird für das Scripting auf C# gesetzt. Bekannte IDEs von Microsoft und JetBrains unterstützen dies auch mit Dokumentation nachhaltig. Sogenannte „Prefabs“ dienen als Templates für komplexere Game Objekte und stellen ein mächtiges Werkzeug dar. Für sehr tiefe Anpassungen kann zudem C++ verwendet werden.
6. Unity ist für Windows, macOS und Linux (Entwicklungsumgebung und Laufzeit) verfügbar. Weitere mögliche Laufzeiten sind fast alle gängigen Konsolen, Android, Web mit WebGL, tvOS, iOS, HoloLens. Neben der nativen Unterstützung einiger HMDs wie Oculus, wird via OpenXR⁴ eine Vielzahl weiterer Head Mounted Displays (HMD) unterstützt.

Unreal Engine

1. Unreal Engine wurde 1998 gegründet und kann auf eine sehr lange Liste bekannter Spiele zurückblicken. Hierzu gehören „Rune“, „Deus Ex“, „Fortnite“, „Tomb Raider“. Es kann als Urgestein bezeichnet werden. Unreal Engine ist besonders im Gamebereich stark, wird aber durchaus auch im industriellen Bereich eingesetzt. Insbesondere rückt hier auf dessen Website der Automobilbereich sowie der Einsatz bei Filmen in den Fokus. Aktuell⁵ ist Unreal Engine in der Version 5 aktuell.
2. Auch auf Grund des langen Bestehens existiert eine gewachsene Community außerhalb des Anbieters. Auf dessen Homepage selbst existiert ein eigener Community Bereich. Zudem werden diverse Lernmöglichkeiten und Videos angeboten. Nachteilig wirkt sich deren etwas unklare Anordnung aus.
3. Auch Unreal Engine bietet auf seiner Website einen Zugang zu einem umfangreichen Marketplace mit Assets und Plugins an.
4. Bei Lizenzen wird zwischen einer Standardlizenz für kleiner Teams und Studenten (frei ohne Support) sowie den Enterprise Bereich unterschieden. Dieser teilt sich zudem in einem Game und Non Game Bereich mit unterschiedlichen Lizenzangeboten. Übersteigt ein „Lifetime“ Gewinn die Millionengrenze, so wird auf die darüber liegenden Gewinne eine 5% Abgabe bei Spielen fällt. Für den Non Game müssen Einzelvereinbarungen getroffen werden. Ausgenommen sind Kundenprojekte, nicht lineare Produkte sowie interne Projekte.
5. Unreal Engine bietet vollen Zugriff auf den C++ Sourcecode. Somit können alle Aspekte angepasst oder erweitert werden. Hierzu ist jedoch ein solides Wissen in C++ notwendig. Dies gilt ebenso für die Implementierung dynamischen Verhaltens. Die Möglichkeit des Visual Scripting bietet hierfür eher wenig Möglichkeiten keinen Platz.

⁴ Bei OpenXR handelt es sich um einen offenen Standard im Umfeld von VR und AR (<https://www.khronos.org/openxr/>)

⁵ Juni 2022

6. Die Game Engine das gesamte gängige Spektrum von Plattformen, mobilen Devices und Konsolen. Im Bereich VR und AR werden, auch durch die Berücksichtigung von OpenXR, eine Vielzahl an Devices unterstützt. Die Engine selbst steht hierbei in einer Version für Windows, macOS und Linux zur Verfügung.

Ergebnis

In den vorhergehenden Abschnitten wurde ein kurzer und subjektiver Überblick über vier der bekanntesten Game Engines gegeben. Es zeigte sich, dass die drei kommerziellen Anbieter alle relevanten Bereiche abdecken und sich auch in der Industrie behaupten können. Bezogen auf den Umfang und Mächtigkeit muss sich auch die Open Source Engine Godot nicht verstecken.

Jede dieser Umgebungen könnte grundsätzlich für die Umsetzung des hier vorgestellten Projektes genutzt werden. Final fällt die Wahl jedoch auf Unity in der aktuellen Version 2021.2.5f1. Ausschlag gebend waren hierfür vor allem deren Verfügbarkeit, Lernressourcen und Programmierbarkeit mit C#. Dies kommt mir als C# Programmierer entgegen. Nicht weniger wichtig ist jedoch auch dessen Lizenzsystem, das ein Abo Modell vor Laufzeitabgaben stellt.

Anmerkungen zum Technik Stack

An dieser Stelle sollen noch ein paar wenige Anmerkungen zur Umsetzung ausgeführt werden. Für eine detaillierte Beschäftigung mit dem Code sei auf die codeseitig durchgeführte Dokumentation verwiesen. Technisch gesehen handelt es sich bei dem hier vorgestellten Spiel um ein webbasiertes Single Player Spiel aus der First Person Perspektive. Diese Entscheidung ist primär durch lizenzrechtliche Fragen und der hierdurch entstehenden Problematik gefallen.

Lizenzrecht

Bei den Minions handelt es sich um in Bereichen geschützte Inhalte. Es hat sich in der Praxis gezeigt, dass gut durchdachte und umgesetzte Modelle von Minions gerade nicht für den Spielbereich lizenziert sind oder aber zusätzliche Genehmigungen benötigen. Diese Problematik wird durch Verwendung einer First Person Perspektive mit entsprechendem Controller umgangen und sollte für die hier gezeigte, nicht kommerzielle Verwendung ausreichen.

Eventbasierte Orchestrierung

Ein zentraler Punkt des Spiels ist die Verwendung eines zentralen Event Managements, der alle Bereiche des Spiels beeinflusst. Hierdurch ist es möglich, dass Spiel sehr lose zu koppeln und dadurch erweiterbar und flexibel zu gestalten. Die zentrale Rolle spielt hierbei die Eventmanager Klasse. Durch das Anlegen als Singleton ist sie in allen Bereichen des Spiels und innerhalb von Skripten einfach verfügbar.

Zentraler Soundmanager

Alle Audioausgaben erfolgen über eine zentrale Instanz (Singleton) eines Audiomanagers, welcher hierdurch ebenfalls aus dem gesamten Spiel erreichbar ist. Er verfügt insgesamt über vier

Audioquellen, wobei zwei für die Wiedergabe von Hintergrundmusik und zwei für das Abspielen von zu übergebenen Clips als Soundeffekten vorgesehen ist.

Input Eingabesystem von Unity

Für die Eingabe wird das neue Input System von Unity verwendet. Es ermöglicht es auf eine sehr elegante und objektorientierte Art und Weise auf die Ereignisse in einer abstrakten Form zugreifen zu können. Hierbei ist es aus dem Code weder notwendig, das Eingabedevise zu kennen noch die genaue Eingabe. Sämtliche Eingaben werden vielmehr im Vorfeld deklarativ festgelegt und gemappt. Dies Mapping kann erweitert werden, ohne dass der Code hierbei angepasst werden muss.

Interaktionskonzept mit Spielobjekten

Das Spiel verfügt über ein grundsätzliches Interaktionssystem zwischen Spieler und Umwelt. Hierfür wurde die abstrakte Klasse Interactable als Basis eingeführt. Das Hinzufügen einer von Interactable abgeleiteten Komponente ermöglicht die Interaction eines GameObjects mit dem Spieler. Hierfür scannt der Spieler bei jedem Update in Vorwärtsrichtung. Wird eine Objekt vom Typ Interactable entdeckt, wird dessen definierte Funktionalität aufgerufen. Konkret wird diese Funktionalität zur Anzeige der Infotexte im Spiel genutzt.

Spieldetails

Kurzbeschreibung (Version 2)

Jeder kennt die Minions: klein, gelb und immer auf die Suche nach dem einen, den wahren Meister und Schurken. Mit Gru könnte alles so schön sein, wenn da nicht der Heißhunger auf Bananen wäre. Sehen sie eine Banane, so ist das Chaos vorprogrammiert.

Nun ist es Gru zu bunt geworden. Kurzerhand hat er alle Bananen, die er finden konnte an einen sicheren Ort versteckt und den Zugang durch ein fieses lebendes Labyrinth der bewegenden Mauern versperrt. Aber nicht nur dass: Zusätzlich nutzt er seine neueste Erfindung: künstliche Geister, die einen verfolgen und kostbare Lebenspunkte stehlen. So kam es zur Legende des Bananenbaums.

Hier kommt Hugo ins Spiel. Hugo ist ein an sich sehr durchschnittlicher und obendrauf kleiner Minion. Aber er hat eine Vision, ein Ziel, eine Mission: Er wird den großen Bananenbaum am Ende des Labyrinths finden und alle seine Freunde in das gelobte Land führen, wo die Bananen in der Luft hängen.

Aber der Weg ist schwer und gefährlich. Level um Level muss er durchqueren, um zu seinem Ziel zu gelangen und sein Vorrat an Lebenspunkten ist wie seine Zeit begrenzt. Nicht nur droht ihm hierbei ständig Gefahr durch die Kollision mit umherschwebenden Hindernissen, sondern auch durch Geister. Diese Plagegeister heften sich an Ihm, sobald er in ihre Nähe kommt, und verfolgen ihn. Nur wenn er Abstand zu ihnen hält, kann er seine Lebenspunkte retten und nicht einschlafen. Nur gut, dass er seinen Vorrat auffrischen und Hilfe finden kann.

Zu Anfang scheint der Weg einfach, aber je näher er sein Ziel kommt, umso größere Probleme muss er überwinden. Sehen wir, wie weit er kommt.

Charaktere und Zielobjekte (Version 4)

Die Hauptcharaktere des Spiels sind der vom Spieler kontrollierte Minion Hugo und die sich im Weg stellenden Geister, welche eigenständig agieren und versuchen, Hugo Lebenspunkt zu stehlen.

Hugo

Auch wenn Hugo nicht selbst in Erscheinung tritt, ist er der Held des Spiels und wird vom Spieler gesteuert. Hugo ist der eine Minion auf der Suche nach den ultimativen Bananenbaum.

Aussehen

Hugo kann man sich als einen typischen Minion in der allseits bekannten und beliebten Erscheinung vorstellen: klein, gelb und mit Latzhose aus Jeans.



Abbildung 1: "Hugo" ist der Held des Spiels und Spielercharakter.

Eigenschaften

Er verfügt über Lebenspunkt, welche jedoch durch den Zusammenstoß mit Hindernissen oder Geister viel zu schnell aufgebraucht werden. Diese kann er auch in ein neues Level mitnehmen.

Aktionen und Animationen

Hugo kann innerhalb eines Levels Lebenspunkte sammeln oder seine Eigenschaften verbessern, indem er sich zu den Zwischenzielen bewegt und sich ihnen bis auf einen bestimmten Abstand annähert. Weitere Aktionen sind hier nicht notwendig. Eingesammelte Punkte und Fähigkeiten werden durch ein auditives als Feedback rückgemeldet.

Bewegungsmöglichkeiten

Hugo kann sich frei im Spielfeld bewegen und wird über Tastatur und Maus gesteuert. Die Maus steuert hierbei seine Blick- und Bewegungsrichtung.

Geister

Geister sind namenlose Gestalten, die Hugo attackieren und ihm seine Bananen stehlen, sofern er ihnen zu nah kommt. Den Geistern entkommt Hugo, indem er Abstand von ihnen hält oder Halloween Kürbisse sammelt.

Aussehen

Geister werden eher naiv und kindgerecht als harmlose Gespenster ähnlich dem bekannten „Hui-Buh“ visualisiert. Die Zahl über dem Geist gibt die Anzahl an Bananen an, die jeder Kontakt zu dem Geist kostet.



Abbildung 2: Der NPC „Geist“ verfolgt Hugo und klaut seine Bananen.

Eigenschaften

Geister können Spieler ab einer bestimmten Entfernung lokalisieren. Sie können einen Spieler verfolgen, verlieren aber den Kontakt, wenn er sich permanent von ihnen entfernt oder sie durch das Sammeln von Halloween Kürbissen geschützt sind. Mit zunehmendem Level werden der Schaden und notwendige Abstand, um nicht verfolgt zu werden, größer.

Aktionen und Animationen

Geister attackieren den Spieler, wenn er ihnen zu nah kommt, und stehlen seine Bananen. Dies gilt auch, wenn Sie überlaufen werden. Das Stehlen wird durch ein visuelles Feedback angezeigt. Mit zunehmendem Level wird die Geschwindigkeit, mit der Lebenspunkte gestohlen werden, größer. Die Zahl über den Geistern zeigt an, wie hoch der Verlust bei jedem Zusammentreffen ist.

Bewegungsmöglichkeiten

Geister bewegen sich in gerade Linie auf den Spieler zu und bewegen sich anschließend mit ihm, jedoch etwas langsamer. Hierbei werden Sie von keinem Gegenstand des Spiels aufgehalten.

Zwischenziele und finales Levelziel

Das Spiel kennt drei verschiedene Zwischenziele und ein finales Levelziel. Zwischenziele sind alle innerhalb der Begrenzung befindlichen Gegenstände, welche durch den Spieler im Spielverlauf aufgesammelt werden können. Das Überlaufen oder Anspringen dieser Zwischenziele bringt dem Spieler (Hugo) Vorteile und Punkte. Das Erreichen des finalen Levelziel beendet das aktuelle Level.

Alle Zwischenziele sowie das finale Levelziel werden zu Beginn eines neuen Levels zufällig über der gesamten Spielfläche verteilt. Die Anzahl der Zwischenziele sowie ihre Wertigkeit wird zufällig in Abhängigkeit vom erreichten Level konfiguriert. Jedes Ziel kann nur einmal eingesammelt werden, anschließend verschwindet es vom Spielfeld.

Zwischenziel „Banane“

Durch die Kollision mit Wänden oder Geister verliert der Spieler (Hugo) Bananen. Durch Aufsammeln von Zwischenzielen des Typs „Banane“ erhält Hugo die über dem Symbol angegebene Zahl an „Bananen“ wieder gutgeschrieben.

Für das erfolgreiche Absolvieren eines Levels muss am Ende des Levels mindestens eine Banane im Besitz von Hugo sein. Hierbei darf in der Zwischenzeit der Bananenbestand auch im negativen Bereich sein.

Bei Betreten eines neuen und höheren Levels bleibt der bis dahin gesammelte Bestand an Bananen erhalten. Hierdurch kann sich im Laufe des Spiels der Punktestand stark erhöhen. Muss Hugo ein Level zurück gehen, so wird sein Bestand als Trost wieder auf null gesetzt.

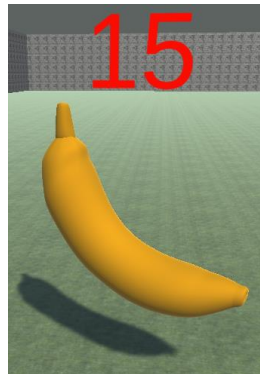


Abbildung 3: Das Zwischenziel „Banane“ gibt Hugo die Kraft, weiter zu machen.

Zwischenziel „Halloween Kürbis“

Geister stehlen Hugo wertvolle Bananen, um ihn daran zu hindern, weiterzukommen. Gelingt es Hugo, ein Zwischenziel vom Typ „Halloween Kürbis“ einzusammeln, so kann sich Hugo unbemerkt für die angegebene Zeit in Sekunden bewegen.

Die gewonnenen Sekunden können nicht angesammelt werden. Sobald eine Anzahl an Sekunden gewonnen wurde, wird diese je Sekunde um eins verkleinert. In der Zwischenzeit eingesammelte Ziele vom Typ „Geist“ wirken nur, sofern die restliche Zeit kleiner als die der neu hinzugewonnenen Zeit ist.

Zwischenziele vom Typ „Halloween Kürbis“ können nicht in ein neues Level mitgenommen werden. Mit dem Erreichen eines neuen Levels verfallen noch vorhandene Zeitguthaben.



Abbildung 4: Das Zwischenziel „Halloween Kürbis“ schützt eine Zeitlang vor Geister.

Zwischenziel „Wand“

Bei der Kollision mit Wänden verliert Hugo immer wertvolle Bananen. Das Sammeln von Zwischenzielen des Typs „Wand“ schützt ihn für die angegebene Zeit an Sekunden vor der schädlichen Wirkung der Wände.

Wie zuvor bei dem Zwischenziel des Typen „Geist“, können Sekunden nicht angesammelt werden. Sobald eine Anzahl an Sekunden gewonnen wurde, wird diese je Sekunde um eins verkleinert. In der

Zwischenzeit eingesammelte Ziele vom Typ „Wand“ wirken nur, sofern die restliche Zeit kleiner als die der neu hinzugewonnenen Zeit ist.

Ebenfalls wie zuvor, so ist auch dies Zwischenziel nur für das aktuelle Level verfügbar. Wird das aktuelle Level beendet, so verfallen noch vorhandene Zeitguthaben.



Abbildung 5: Das Zwischenziel „Wand“ schützt eine Zeitlang vor den Wänden.

Finales Levelziel

In jedem Level gibt es genau ein finales Levelziel. Wird das finale Levelziel erreicht, so endet das aktuelle Level unabhängig von der bis dahin vergangenen Zeit. Auf der anderen Seite muss das finale Zwischenziel innerhalb der vorgegebenen Zeit erreicht werden.

Wird das finale Levelziel erreicht, kann das nächsthöhere Level betreten werden, sofern der Spieler (Hugo) mindestens eine Banane in seinen Besitz hat. Sind keine Bananen vorhanden oder aber sogar weniger als null, so wird man ein Level zurückgestuft. Als Trost wird der Bananenbestand auf null gesetzt.



Abbildung 6: Das finale Levelziel „Goldene Banane“ beendet das Level.

Punkteberechnung

Die Berechnung der Punkte bei Erreichen eines Zwischenziels richtet sich nach dem Typ des Ziels. Hierbei wird die folgende Methodik verwendet:

- **Zwischenziel „Banane“:** Es wird die über dem Ziel vermerkte Menge an Bananen für den Spieler gutgeschrieben.
- **Zwischenziel „Geist“:** Existiert zum Zeitpunkt des Aufsammelns bereits eine aktuell laufende Zeit, die höher ist als der angezeigte Wert des Geistes, so wird diese Summe nicht erhöht. Ist die aktuell laufende Zeit kleiner oder null, so wird der Wert des Geistes zum neuen Wert.
- **Zwischenziel „Wand“:** Existiert zum Zeitpunkt des Aufsammelns bereits eine aktuell laufende Zeit, die höher ist als der angezeigte Wert der Wand, so wird diese Summe nicht erhöht. Ist

die aktuell laufende Zeit kleiner oder null, so wird der Wert der Wand zum neuen Wert.

Level (Version 3)

In dem Spiel existieren keine fest definierten Level oder Spielszenen. Die Spielszene ist über alle Level grundlegend identisch und wird bei der Entwicklung als Basislevel definiert und bezeichnet. Die folgende Abbildung zeigt eine beispielhafte Darstellung des Basislevels aus Entwicklersicht in Unity:

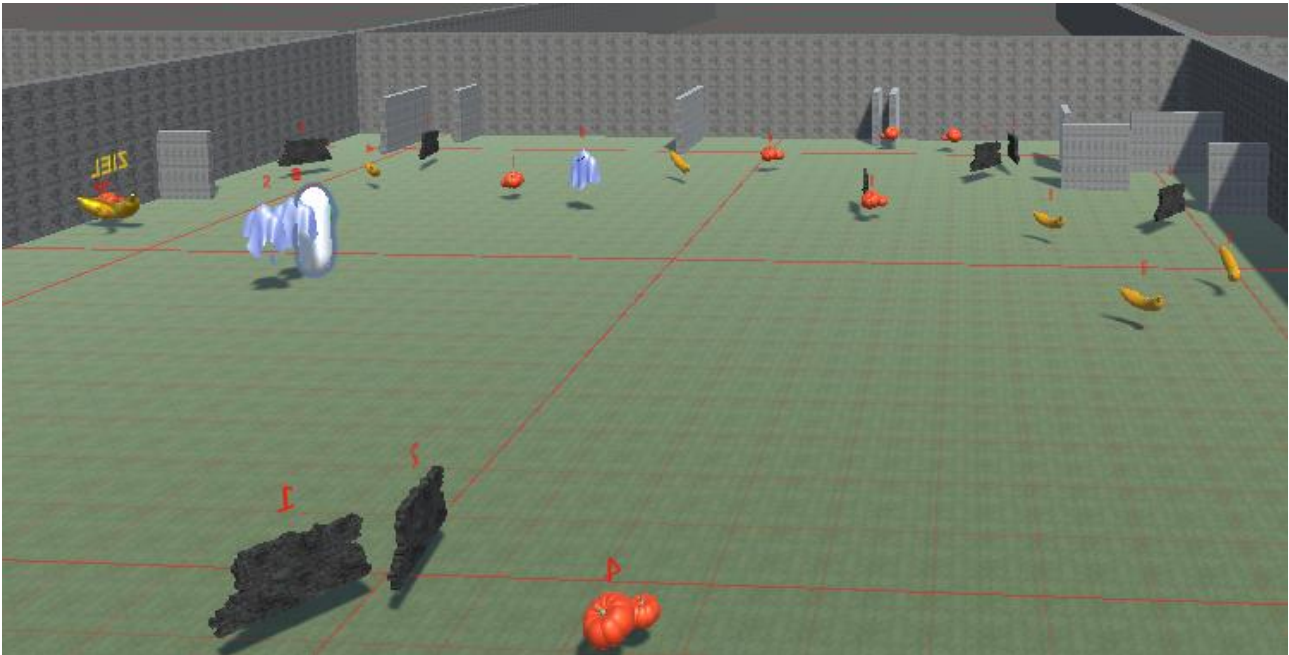


Abbildung 7: Basislevel kurz nach dem Start aus der Entwicklersicht

Mittig ganz auf der linken Seite befindet sich das aktuelle Zwischenziel („Goldene Banane“ mit dem Schriftzug „Ziel“). Dicht daneben ist der aktuelle Avatar des Spielers (Sphere) zu sehen, der von zwei „Geistern“ attackiert wird.

Alle Hindernisse, Zwischen- und finale Levelziele sowie die Non-Player Charaktere (NPCs) werden auf Basis von Zufallszahlen und dem aktuellen Spielverlauf *on-the-fly* generiert und gesteuert. Hierbei kann auf alle wichtigen Parameter über eine Vielzahl an Parametern Einfluss genommen werden.

Bei Abschluss eines Levels wird der neue Level anhand eines Algorithmus und unter Zuhilfenahme der Parameter erstellt und skaliert. Dies ermöglicht zudem eine breite Anpassung ohne weiteren Zugriff auf den Sourcecode.

In der Umsetzung resultieren daraus praktisch unendlich viele Möglichkeiten des Ablaufs und des Schweregrades. Zudem gibt es keinen fest vorgegebenen besten Lösungsweg. Alle Gegenstände werden zufallsgesteuert immer wieder neu generiert. Ebenfalls werden die Hindernisse zufällig konfiguriert.

Über das Sammeln von Gegenständen kann man seine Chancen beeinflussen, indem beispielsweise

zunächst Lebenspunkte gesammelt werden. Gesammelte Fähigkeiten bleiben im nächsten Level erhalten.

Startlevel

Der erste Level ist das Startlevel. Wird dieser Level gewonnen, so wird das Spiel fortgeführt, indem in das nächsthöhere Level gewechselt wird. Ist das Level hingegen verloren, beginnt es von neuem. Somit endet das Spiel streng genommen nicht.

Beschreibung des Levels

Der Spieler erscheint an einen zufällig bestimmten Punkt eines großen, durch Mauern umrandeten Spielfeld. Er sieht verschiedene Gegenstände auf dem Feld verteilt. Vom Aussehen lassen sich Rückschlüsse auf deren Nutzen ziehen. Besonders sticht das finale Levelziel, die „Goldene Banane“ visuell hervor. Im ersten Level verfügt der Spieler über keine Bananen.

Die folgende Abbildung zeigt die aktuelle Sicht aus einer Third-Person Perspektive:

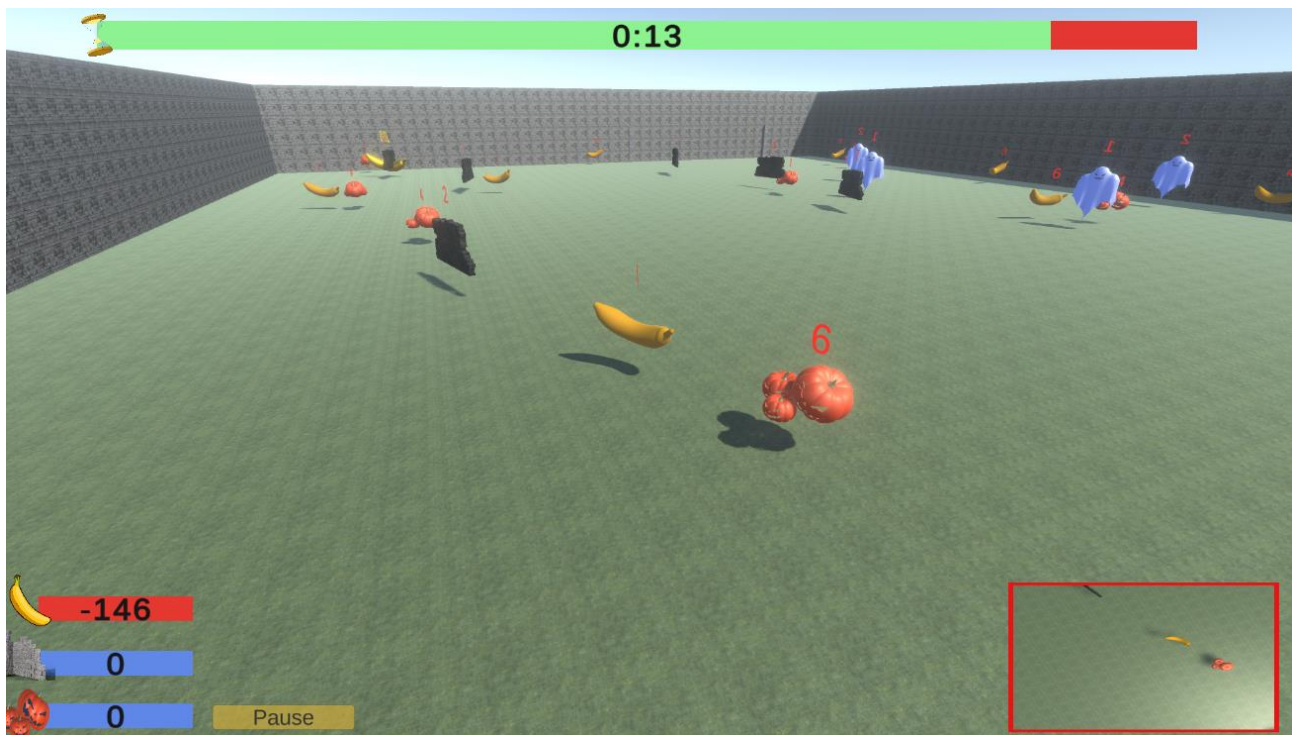


Abbildung 8: Spieleindruck aus Third-Person Perspektive

Aufgabe des Spielers

Aufgabe des Spielers ist das rechtzeitige Erreichen des finalen Levelziels innerhalb der Spielzeit mit mindestens einer Banane. Weitere vorhandene Gegenstände, insbesondere Bananen, können, müssen aber nicht eingesammelt oder angesteuert werden. Insbesondere darf der Bananenbestand zwischen dem Levelstart und -ende negativ sein.

Auftretende Charaktere/Spielfiguren/Elemente

Sobald das Spiel startet, erscheinen aus allen vier Wänden in zufälliger Anzahl, Zeit, Höhe und Länge Hindernisse. Diese bewegen sich in Richtung der gegenüberliegenden Wand, um dort wieder zu verschwinden.

Wird der Spieler von einem dieser Hindernisse getroffen oder läuft davor, so verliert er eine festgelegte Zahl an Bananen.

Gefahr droht dem Spieler zudem von Geistern, welche sich direkt auf den Spieler zubewegen und ihn verfolgen. Kommen sie ihn zu nahe, verliert er kontinuierlich Bananen. Nur durch Abstand von den Geistern oder dem Einsammeln von Halloween Kürbissen kann er eine Zeitlang entkommen.

Durch das Einsammeln der im Level verteilten „Wände“ erhält der Spieler für eine Zeitspanne Schutz vor den sich über das Spielfeld bewegend großen Wänden.

Wie wird das Level beendet

Ein Level wird beendet und gewonnen, wenn das finale Levelziel (die „Goldene Banane“) innerhalb einer bestimmten Zeit erreicht wurde und der Spieler mindestens eine Banane besitzt. In diesem Fall betritt er das nächsthöhere Level und nimmt seine aktuell vorhandenen Bananen mit.

Ein Level wird verloren, wenn der Spieler das finale Levelziel nicht in einer bestimmten Zeit erreicht, oder er beim Erreichen nicht über mindestens eine Banane verfügt. In diesen Fall geht er ein Level zurück und startet mit null Bananen. Wird das erste Level verloren, so ist das Spiel als solches verloren.

Skalierung der Folgelevel

Alle folgenden Level basieren auf den anfänglichen Basislevel, werden aber durch Anpassung der vorhandene Parameter zunehmend komplexer. Dies wirkt sich insbesondere auf vier Kernelement aus:

- **Spielement „bewegende Wände“:** Anzahl, Geschwindigkeit, Länge und möglicher Schaden nehmen zu
- **NPC „Geister“:** Anzahl, Geschwindigkeit und möglicher Schaden nehmen zu. Zudem wird es schwerer ihnen zu entkommen, da deren Reichweite größer wird, in der sie den Spieler erkennen können
- **Zwischenziel „Wände“:** Die Anzahl dieser Ziele wird weniger und der maximale Zeitraum, in dem sie vor den sich bewegend Wänden schützen, kleiner.
- **Zwischenziel „Halloween Kürbisse“:** Die Anzahl dieser Ziele wird weniger und der maximale Zeitraum, in dem vor Geister sie schützen, kleiner.

Überarbeiten Menü (Version 2)

Das Programm startet mit einem Eingangsmenü, welches gleichzeitig auch das Hauptmenü darstellt. Insbesondere das Hintergrundbild bedarf aus lizenzrechtlichen Gründen noch einer Änderung. Zusätzlich zum Hauptmenü existiert ein Menü in Form eines Dialoges zwischen den einzelnen Level.

Eingangsmenü

Die Folgenden Unterkapitel geben einen Überblick über die Menüstruktur des Hauptmenüs und deren Elemente. Am Ende zeigt ein Interaktionsdiagramm die möglichen Entscheidungspfade zusammenfassend an.

Hauptmenü

Das Hauptmenü stellt den Startpunkt des Spiels und zugleich den zentralen Menüpunkt dar, an der immer wieder aus dem Spiel zurückgekehrt werden kann. Der Punkt Spielanleitung ist aktuell nicht implementiert.

Button	Anzeige/Eingabe	Aktion
Spielanleitung		Zeigt eine Spielanleitung
Tastatur		Sprung ins Untermenü Tastatur
Einstellungen		Sprung ins Untermenü Einstellungen
Starten		Startet das Spiel

Tabelle 1: Übersicht Hauptmenü



Abbildung 9: Übersicht Startmenü

Untermenü Tastatur

Im Untermenü Tastatur wird eine Übersicht über alle verwendbaren Tastaturbefehle gegeben. Eine individuelle Einstellung ist im Rahmen der Semesteraufgabe nicht vorgesehen, würde aber hier seinen Platz finden.

Button	Anzeige/Eingabe	Aktion
	Anzeige der aktuellen Tastaturbelegung	
Zurück		Rücksprung zum Hauptmenü

Tabelle 2: Übersicht Untermenü Tastatur

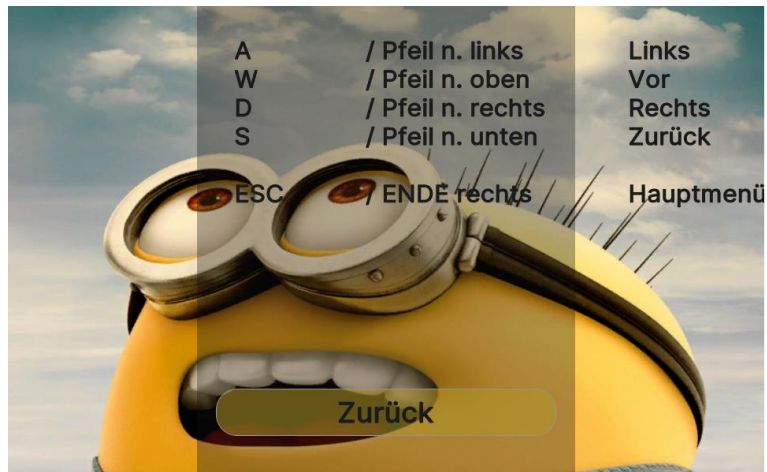


Abbildung 10: Untermenü Tastatur

Untermenü Einstellungen

Alle für das Spiel zu machenden Einstellungen finden sich im Untermenü Einstellungen. Bei einer Übernahme erfolgt die Anwendung für das aktuelle Spiel. Eine darüberhinausgehende Speicherung ist in dieser Semesterarbeit nicht vorgesehen.

Button	Anzeige/Eingabe	Aktion
	Checkbox Geister berücksichtigen	Ist die Checkbox gesetzt, werden im Spiel Geister erscheinen. Die Einstellung wird standardmäßig bei Spielstart aktiviert.
	Checkbox Audio einschalten	Ist die Checkbox gesetzt, erfolgt eine Audioausgabe im Spiel. Die Einstellung wird standardmäßig bei Spielstart aktiviert.
Übernehmen		Übernimmt die gemachten Einstellungen für das Spiel
Zurück		Rücksprung zum Hauptmenü

Tabelle 3: Übersicht Untermenü Einstellungen

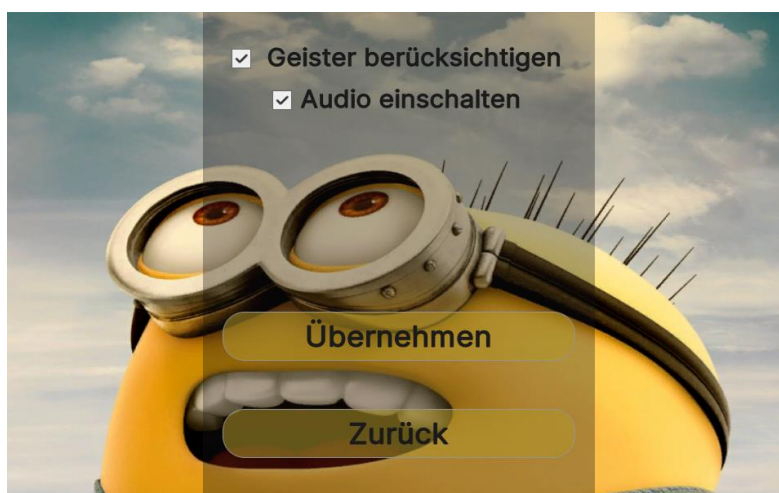
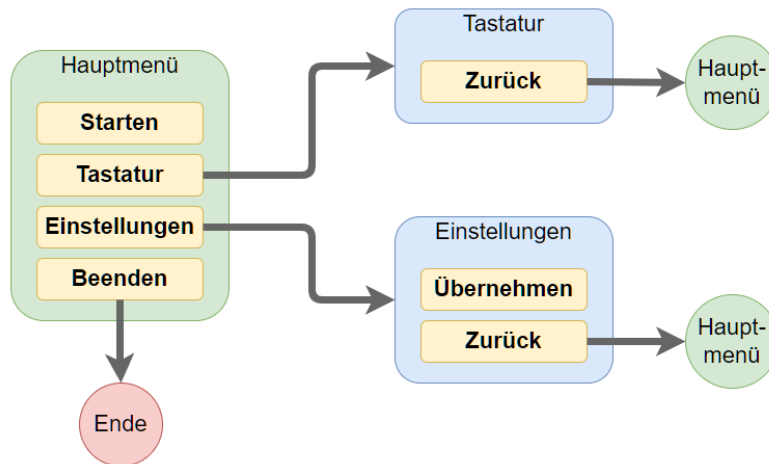


Abbildung 11: Untermenü Einstellungen

Interaktions- und Navigationsdiagramme



Innerhalb des Spiels kann der Spieler jederzeit mit Hilfe der ESC Taste das Hauptmenü aufrufen und alle dort angebotenen Menüpunkte auswählen. In dieser Zeitspanne pausiert das Spiel.

Leveldialog

Die Aufgabe des Leveldialogs ist die Begleitung des Spielers aus dem vorangegangenen Level in den nächstfolgenden Level.

Innerhalb des Dialogs existieren grundsätzlich immer zwei Optionen, welche jedoch auf die Situation angepasst textlich unterschiedlich dargestellt werden.

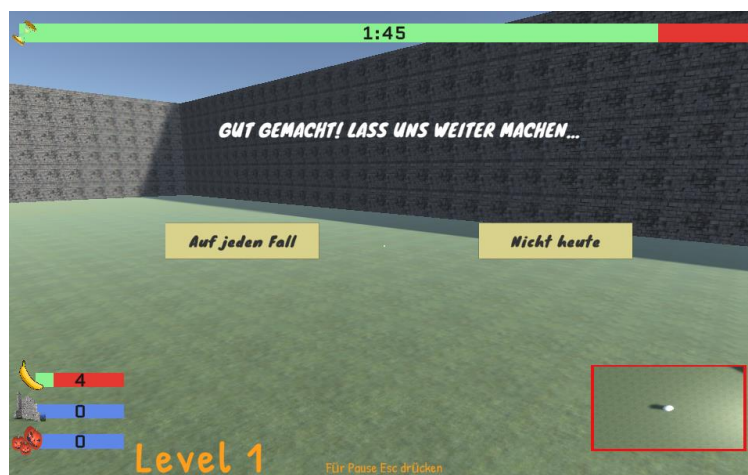


Abbildung 12: Leveldialog zwischen zwei Levels im Kontext „Level gewonnen“



Abbildung 13: Leveldialog zwischen zwei Levels im Kontext „Level verloren“

In den obigen Abbildungen ist ganz oben ein Leveldialog im Kontext „Level gewonnen“, darunter im Kontext „Level verloren“ zu sehen. Innerhalb der Dialoge wurde auf eine positive Ansprache geachtet, mit der der Spieler zu einem Weiterspielen animiert werden soll.

Grundsätzlich existieren immer zwei Optionen, welche auf die Situation angepasst textliche unterschiedliche aufweisen. Während die immer links dargestellte Auswahlmöglichkeit in den nächsten Level führt, führt die immer rechts eingeblendete Option stets in das Hauptmenü.

Von hier kann das Spiel ganz verlassen, oder aber nochmals zurück gegangen werden. In diesem Fall wird mit dem nächsten Level fortgefahren.

Head Up Display

Innerhalb des Spiels erfolgt die Anzeige relevanter Informationen auf Basis eingeblendeter GUI Elemente. Hierdurch erhält der Spieler alle wesentlichen Informationen, ohne jedoch zu sehr abgelenkt zu werden.

Bei der Umsetzung wurde verstärkt auf die Visualisierung geachtet, um den Spieler einfach zu informieren. Hierfür wurde eine Übersicht auf Basis von „Balkenanzeigen“ implementiert. Negative Aspekte erscheinen hierbei einheitlich in roter, neutrale in blau und positive in grüner Farbe. Die folgende Abbildung verdeutlicht dies:

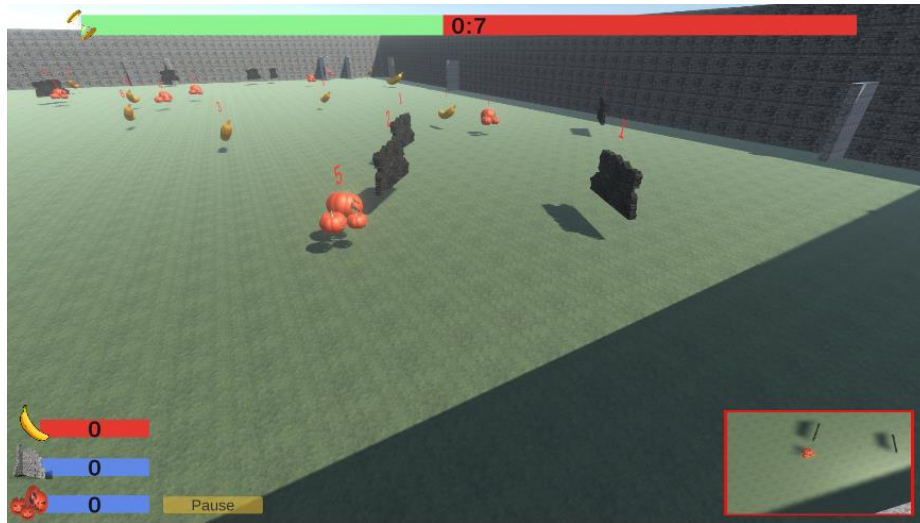


Abbildung 14: Spielansicht mit allen HUD Elementen.

Zu sehen ist, wie oben am Bildschirm die Spielzeit primär visuell anhand des sich verkleinernden, grünen Balkens und der sich drehenden Sanduhr ausgedrückt wird. Daneben ist die verbliebene Zeit, hier sieben Sekunden, als Text innerhalb der Anzeige zu sehen.

Unten links erscheint eine Übersicht der wichtigsten Spielstände ebenfalls wieder durch Symbole und Farbe mehrfach codiert. Die folgende Abbildung zeigt dies im Detail:



Abbildung 15: Detailansicht des primären HUP.

In dem Bild ist zu sehen, dass genaue Angaben textlich eingeblendet werden, jedoch der grundlegende Zustand einfach zu erfassen ist. Die Verwendung von Symbolen ermöglicht hierbei die einfache Zuordnung zu den Spielelementen.

Ergänzt wird das Informationssystem durch das Einblenden einer „Mini Map“ auf der rechten, unteren Seite. Diese zeigt von oben auf den Spieler und ermöglicht so einen schnellen Überblick über die aktuelle Spielsituation:

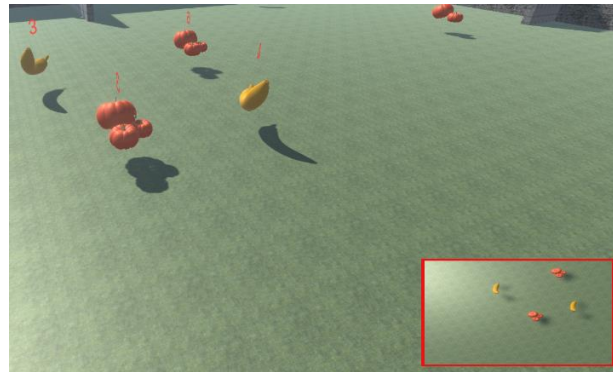


Abbildung 16: Detailansicht der „Min Map“ mit einem Teil des Spielfeldes.

Steuerung (in Bearbeitung)

Die Spielsteuerung erfolgt über die Maus und der Tastatur. Auf Grund der Verwendung des neuen Input Systems in Unity kann die Steuerung für weitere Devices wie Gamepad ohne Modifizierung der Codebasis erweitert werden. Die Tastenbelegung richtet sich nach der gängigen Belegung in Spielen:

Tasten Belegung	Alternative Belegung	Aktion
W	Pfeil nach oben	Vorwärts
S	Pfeil nach unten	Rückwärts
A	Pfeil nach Links	Links
D	Pfeil nach Rechts	Rechts
SPACE	ENTER rechts	Springen
MAUS	MAUS	Blick Links/Rechts/Oben/Unten

Gameplay (Version 2)

Auf einem grundsätzlich gleichbleibenden Spielfeld muss der Spieler von einem willkürlichen Startpunkt zu einem sichtbaren Zielpunkt (Finale Levelziel) gelangen. Hierfür hat er nur eine gewisse Zeitspanne zur Verfügung. Erreicht er das Ziel in dieser Zeitspanne mit mindestens einer Banane, fängt das nächste und schwerere Level an. Der Spieler startet an einer neuen, zufälligen Position.

Zu Anfang des Spieles verfügt der Spieler über keine Bananen. Durch bestimmte Aktionen können weitere Bananen gewonnen oder verloren werden. Bei Erreichen eines höheren Levels werden die aktuell gesammelten Bananen mitgenommen, bei Beginn eines niedrigeren Levels werden diese wieder auf einen Anfangsstand gesetzt.

Wird der erste und einfachste Level verloren, so hat er das Spiel als Ganzes verloren. Gewinnt der Spieler das aktuelle Level, indem er mit mindestens einer Banane und innerhalb der vorgegebenen Zeit das finale Levelziel erreicht, so betritt er an einen zufälligen Ort mit seinen bis dahin erreichten Punktestand das nächsthöhere Level.

Auf dem Weg dorthin kann er visuell sichtbare Zwischenziele anlaufen. Das Erreichen eines Zwischenziels wird mit der angezeigten Menge an Bananen oder einem Schutz entweder vor Geistern oder Mauern über die jeweils angegebene Zeit in Sekunden belohnt. Die Ziele können

anhand ihrer Erscheinung unterschieden und nur einmalig genutzt werden. Ihre Anzahl und die maximal mögliche Menge an Lebenspunkten oder Zeit ist antiproportional zum erreichten Level.

Bei der Erfüllung der Anforderung stehen dem Spieler sich geradlinig bewegend Hindernisse in Form von Wänden im Weg. Diese kommen aus zufälligen Richtungen aus den umliegenden Wänden und bewegen sich orthogonal zur gegenüberliegenden Wand der Spielfeldbegrenzung. Hierbei ist deren Anzahl, Geschwindigkeit, Höhe und Breite zufallsbasiert gesteuert. Die maximale Anzahl und Geschwindigkeit sind hierbei proportional zum erreichten Level.

Der Spieler (Hugo) kann diese Hindernisse umlaufen. Kollidiert der Spieler jedoch mit einem Hindernis, so verliert er Bananen. Die Anzahl der hierbei verlorenen Bananen ist ebenfalls zufallsgesteuert und proportional zum erreichten Level.

Ein weiterer Widerstand ist in Gestalt von Geistern gegeben. Geister sind zufällig positionierte NLPs mit einer gewissen „Intelligenz“. Kommt ein Spieler in die Nähe eines Geistes, so wird der Geist sich in Richtung des Spielers bewegen und ihn verfolgen. Hierbei wird seine Wahrnehmung und Fortbewegung durch alle Hindernisse behindert.

Erreicht ein Geist den Spieler, so verliert dieser in konstanter Folge Bananen. Dem Geist kann entkommen werden, sobald eine gewisse Distanz zwischen Geist und Spieler geschaffen wird. Mit zunehmendem Level wird die dazu notwendige Distanz größer, die Anzahl der Geister höher und der von ihnen angerichtete Schaden größer.

Beendet ein Spieler ein Level und will nicht weiterspielen, so kann es sich in einer Bestenliste mit einem frei wählbaren Namen eintragen, sofern er zu den 15 besten Spielern gehört.

Abbildungen

Abbildung 1: "Hugo" ist der Held des Spiels und Spielercharakter.	7
Abbildung 2: Der NPC „Geist" verfolgt Hugo und klaut seine Bananen.	8
Abbildung 3: Das Zwischenziel „Banane" gibt Hugo die Kraft, weiter zu machen.....	9
Abbildung 4: Das Zwischenziel „Halloween Kürbis" schützt eine Zeitlang vor Geister.....	9
Abbildung 5: Das Zwischenziel „Wand" schützt eine Zeitlang vor den Wänden.	10
Abbildung 6: Das finale Levelziel „Goldene Banane" beendet das Level.	10
Abbildung 7: Basislevel kurz nach dem Start aus der Entwicklersicht	11
Abbildung 8: Spieleindruck aus Third-Person Perspektive	12
Abbildung 9: Übersicht Startmenü.....	14
Abbildung 10: Untermenü Tastatur	15
Abbildung 11: Untermenü Einstellungen	16
Abbildung 12: Spielansicht mit allen HUD Elementen.	18
Abbildung 13: Detailansicht des primären HUP.	18
Abbildung 14: Detailansicht der „Min Map“ mit einem Teil des Spielfeldes.	19

Tabellen

Tabelle 1: Übersicht Hauptmenü14

Tabelle 2: Übersicht Untermenü Tastatur14

Tabelle 3: Übersicht Untermenü Einstellungen15