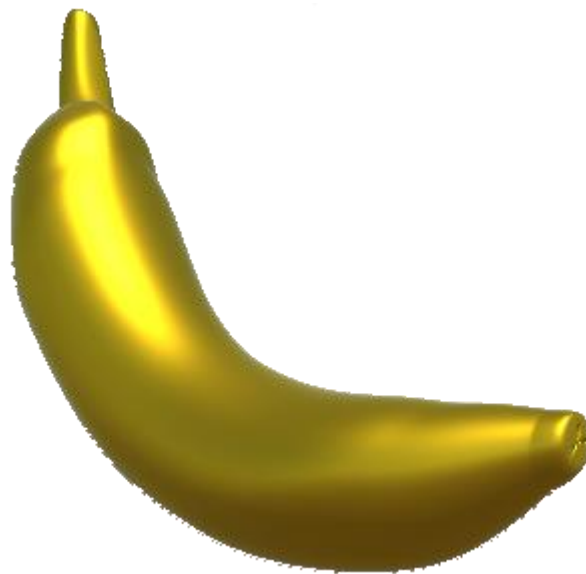


Game Design Document

Stand: 12.07.2022

Banana Hunting



Autor Christian Kitte, HS Emden-Leer

Modul Game Development – SS 2022 – Online Medieninformatik, Master

Inhaltsverzeichnis

Zusammenfassung	4
Zielplattform.....	4
Spielgenre.....	4
Zielgruppe	4
Unique Selling Proposition USP	4
Technologien	5
Game Engine	5
CRYENGINE.....	6
Godot	6
Unity.....	7
Unreal Engine	7
Ergebnis.....	8
Anmerkungen zur Umsetzung und verwendeten Technik Stack	8
Lizenzrecht	9
Dynamisch generiertes Playground	9
Eventbasierte Orchestrierung	9
Zentraler Soundmanager	9
Input Eingabesystem von Unity	9
UI Toolkit von Unity.....	10
Interaktionskonzept mit Spielobjekten	10
Spieldetails	10
Kurzbeschreibung – Die Legende von Hugo.....	10
Charaktere und Zielobjekte.....	11
Hugo	11
Aussehen.....	11
Eigenschaften	12
Aktionen und Animationen.....	12
Bewegungsmöglichkeiten	12
Geister	12
Aussehen.....	12
Eigenschaften	13
Aktionen und Animationen.....	13
Warnzeichen für den Spieler.....	13

Bewegungsmöglichkeiten	14
„Wandernde Wände“	14
Aussehen	14
Eigenschaften	14
Aktionen und Animationen	14
Warnzeichen für den Spieler	15
Bewegungsmöglichkeiten	15
Zwischenziele und finales Levelziel	15
Zwischenziel „Banane“	16
Zwischenziel „Halloween Kürbis“	17
Zwischenziel „Wandsegment“	18
Finales Levelziel „Goldene Banane“	18
Punkteberechnung	19
Levels	19
Startlevel	20
Beschreibung des Levels	20
Aufgabe des Spielers	21
Auftretende Charaktere/Spielfiguren/Elemente	21
Wie wird das Level beendet	22
Übersicht Spielverlauf	22
Skalierung der Folgelevel	22
Menü und Dialoge	23
Startmenü	23
Untermenü Spielanleitung	24
Untermenü Tastatur	24
Untermenü Einstellungen	25
Interaktions- und Navigationsdiagramme	26
InGame Menü	26
Leveldialog	27
Head Up Display	28
Wichtige Ereignisse	29
Steuerung	30
Gameplay	31
Ausblick	33

Credits	34
Abbildungen	35
Tabellen	36

Zusammenfassung

In diesem Abschnitt werden die wichtigsten Details zu dem Spiel und seiner Charakteristik in Kürze zusammengefasst.

Zielpattform

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 hier vorgestellte Spiel primär online und browserbasiert als WebGL Anwendung verfügbar gemacht werden.

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

Spielgenre

Bei dem Spiel handelt es sich um ein 3D Geschicklichkeitsspiel. Primär stehen Schnelligkeit beim Sammeln der Ziele und Erreichen des primären Levelziels im Vordergrund. Taktische Elemente sind nicht vorhanden.

Zielgruppe

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 verliert der Spieler keine Lebenspunkte und stirbt, sondern verliert „nur“ Bananen.

Unique Selling Proposition USP

Das Spiel verfolgt keine komplexen Strategien oder Ziele. Primäres Ziel ist das Erreichen einer möglichst hohen Zahl eingesammelter Bananen und das Erreichen eines möglichst hohen Levels. Perspektivisch ist ein möglicher, weiterer Ausbau die ursprünglich geplante Einrichtung einer Bestenliste, um einen zusätzlichen Anreiz zu bieten. Dies wurde aus Zeitgründen nicht umgesetzt. Zusammenfassend handelt es sich daher um ein typisches, einfaches Spiel, um sich „zwischendurch“ zu entspannen und abzuschalten.

Über die Möglichkeit, sich Vorteile zu verschaffen, siehe hierzu das Kapitel Charaktere und Zielobjekte, sowie der Einbeziehung des Zufalls werden Spielverläufe immer unterschiedlich, können jedoch vom Spieler beispielsweise durch Zwischenziele beeinflusst werden. Das Spiel ist nie verloren, da im schlechtesten Fall das vorherige Level¹ wiederholt wird. Andererseits existiert immer eine

¹ Befindet man sich im ersten Level, so bleibt man in diesem.

Möglichkeit, noch weitere Level zu meistern (im Prinzip ohne Ende), was zu einem guten Reply Wert führt.

Technologien

Im folgenden Kapitel erfolgt eine kurze Begründung zu der Wahl der Game Engine auf Basis einer Übersicht verschiedener Engines. Weiter wird auf einige Details der Umsetzung und des eingesetzten Technik Stack eingegangen.

Game Engine

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 hier angedachten Einsatz spielt die webbasierte Verfügbarkeit eine wichtige Rolle. Grundsätzlich ist jedoch 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 Kandidaten²:

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

² Es handelt sich hierbei ausdrücklich um eine subjektive erste Auswahl.

CRYENGINE

1. Bei CRYENGINE handelt sich um eine ausgereifte Game Engine. Mit ihr wurden bekannte Titel wie „Battle Cry“ oder „Prey“ umgesetzt. Aktuell³ ist CRYENGINE 5.7 mit Long Term Support (LTS) veröffentlicht. Trotz interessanter Fähigkeiten liegt der Fokus mehr im Spielbereich als auf der Industrie.
2. Wikipedia listet eine Reihe von Communities im Umfeld von CRYENGINE auf. Daneben existieren auf der Homepage Verweise auf weitere Communities. Die Seite bietet darüber hinaus ein großes Angebot an Informationen und videounterstützten 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. Die Angaben zu den Kosten sind 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 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. Daneben wird C# in einen sehr begrenzten Bereich 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 bei Godot die Organisation von Szenen in einer Baumstruktur sowie die Trennung in zwei getrennte Engines für 2D und 3D. Die neueste⁴ 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 qualitativ eher gemischt.
3. Über die Homepage ist eine „Asset Library“ mit diversen Erweiterungen und Assets verfügbar. Auch hier erscheint das Angebot eher gemischt.
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 sowie C++. Daneben ist visuelles Scripting möglich. Sprachen wie Rust, Nim oder D werden

³ Juni 2022

⁴ Juni 2022

durch die Community ermöglicht. Die Anzahl verfügbarer Sprachen ist somit sehr groß, verlangt jedoch teilweise manuelle 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, insbesondere im Bereich VR und AR.
2. Unity verfügt über eine umfangreiche Community. Im Internet und hier insbesondere auf YouTube, existiert eine kaum überschaubare Zahl an Informationen und Tutorien. Über die Website ist der kostenlose Zugang zu umfangreichen Dokumentationen, Tutorien sowie einer Vielzahl an geführten und gut ausgearbeiteten Lernpfaden möglich. Zu Unity existieren verschiedene Zertifizierungen wie „Certified User“, „Certified Associate“, „Certified Professional“ und „Certified Expert“.
3. Über die Homepage ist der Zugang zu einem „Asset Store“ mit einer umfangreichen Auswahl an Erweiterungen und Assets möglich. Viele davon sind zudem bereits kostenlos oder für einen geringen Betrag verfügbar. Über die kostenlosen Lernangebote wird der Zugang zu exklusiven Assets angeboten. Unity betreibt einen sehr großen Aufwand zu Integration und Bindung seiner Nutzer.
4. Die kostenlose 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, die Dokumentation sowie das verfügbare 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 mit eigener 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 OpenXR⁵ 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 somit

⁵ Bei OpenXR handelt es sich um einen offenen Standard im Umfeld von VR und AR (<https://www.khronos.org/openxr/>)

als Urgestein bezeichnet werden und ist insbesondere im Gamebereich stark, wird aber durchaus auch im industriellen Bereich eingesetzt. Insbesondere fokussiert auf deren Website der Automobilbereich sowie der Einsatz im Film. Aktuell⁶ ist Unreal Engine in der Version 5 verfügbar.

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 in einem Game und Non Game Bereich mit unterschiedlichen Lizenzangeboten. Übersteigt ein „Livetime“ Gewinn die Millionengrenze, so wird auf die darüber liegenden Gewinne eine 5% Abgabe bei Spielen fällt. Für den Non Game Bereich 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.
6. Die Game Engine deckt das gesamte gängige Spektrum von Plattformen, mobilen Devices und Konsolen ab. 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 2022.1.7f1. Den Ausschlag gaben 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 zur Umsetzung und verwendeten 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. Der gesamte Code ist auf GitHub⁷ verfügbar.

⁶ Juni 2022

⁷ <https://github.com/ChristianKitte/GameDesignSemesterprojekt>

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, welche die Nutzung eines Minion Modells nach sich gezogen hätte.

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.

Dynamisch generiertes Playground

Der grundsätzlich Spielbereich besteht aus einer von Mauer umrandeten Fläche und bleibt über alle Level gleich. Alle weiteren Gameobjekte werden auf Basis von Vorgaben dynamisch und zufallsgesteuert generiert und animiert.

Hierzu wurden allen Komponenten als konfigurierbare Prototypen entworfen. Erst zur Laufzeit werden sie final erstellt und auf Basis des aktuellen Levels konfiguriert. Der hierfür benötigte Mehraufwand in der anfänglichen Implementierung rechnet sich durch die hierdurch eröffneten Freiheiten bei der Konfiguration.

Eventbasierte Orchestrierung

Ein zentraler Punkt des Spiels ist die Verwendung eines zentralen Event Managements, der alle Bereiche des Spiels beeinflusst. Hierdurch ist es möglich, alle Komponenten des Spiels lose zu koppeln und dadurch erweiterbar und flexibel zu gestalten. Eine zentrale Rolle spielt hierbei die Klasse EventManager. 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 drei Audioquellen, wobei zwei für die Wiedergabe von Hintergrundmusik und eine für das Abspielen von zu übergebenen Clips als Soundeffekten vorgesehen ist.

Eine Ausnahme hierzu bildet die Soundwiedergabe der Geister. Diese ist darin begründet, dass jeder Geist über eine eigene, an dessen Ort lokalisierte und in Ihrer Reichweite begrenzte Quelle verfügen muss. Versuche, dies an einen zentralen Ort umzusetzen haben sich als bei weitem zu umständlich gezeigt.

Input Eingabesystem von Unity

Für die Eingabe wird das neue Input System von Unity verwendet. Es ermöglicht, auf eine sehr

elegante und objektorientierte Art und Weise auf Ereignisse wie eine Mausbewegung in einer abstrakten Form reagieren zu können.

Hierbei ist es weder notwendig, das Eingabegerät zu kennen, noch die genaue Art der Eingabe. Sämtliche Eingaben werden vielmehr im Vorfeld deklarativ festgelegt und gemappt. Dies Mapping kann im Nachhinein erweitert oder angepasst werden, ohne dass der Code selbst hierbei überarbeitet werden muss.

UI Toolkit von Unity

Zur Gestaltung des Hauptmenüs wurde das neue UI Toolkit von Unity genutzt. Das UI Toolkit von Unity zeichnet sich dadurch aus, dass UI Elemente ähnlich dem Vorgehen bei der Gestaltung von HTML mit CSS, Dialoge erstellt und genutzt werden können. Somit muss nicht auf Panels und Canvas zurückgegriffen werden.

Elemente werden innerhalb von `UIDocument`s definiert und genutzt. Die eigenen Erfahrungen zeigen aktuell eine zwar einfache Verwendung, jedoch eher unhandliche Erstellung. Auch einfache Task wie die Schriftzuweisung scheint manchmal von Problemen betroffen zu sein.

Innerhalb des Spiels sorgt ein zentraler `PanelManager`, für die Anzeige und Wechsel innerhalb des Hauptmenüs. Hierfür werden ihm die genutzten `UIDocument`s als Komponenten zugeordnet.

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 Game Objects mit dem Spieler.

Hierfür wird bei jedem Update aus der aktuellen Position des Spielers ein Raycast in Vorwärtsrichtung erzeugt und ausgewertet. Wird ein Objekt vom Typ `Interactable` entdeckt, wird dessen definierte Funktionalität aufgerufen. Konkret wird diese Funktionalität zur Anzeige der Infotexte im Spiel genutzt.

Spieldetails

Im dritten Teil dieser Arbeit wird auf das Spiel und dessen Spielkonzept und -idee vertiefend eingegangen.

Kurzbeschreibung – Die Legende von Hugo

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 einem sicheren Ort versteckt und den Zugang durch ein fieses, lebendiges Labyrinth der „Wandernden

Wände“ versperrt. Bei jeder Kollision mit Ihnen verliert man Bananen. Aber nicht nur dass: Zusätzlich nutzt Gru seine neueste Erfindung: künstliche Geister, die einen verfolgen und alle gesammelten Bananen stehlen wollen. 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.

Hierfür muss er der Spur der goldenen Banane folgen. Nur wenn er diese rechtzeitig findet und mindestens eine Banane dabei hat, kann er die nächste Ebene erreichen. Aber der Weg ist schwer und gefährlich. Ebene um Ebene muss er durchqueren, um zu seinem Ziel zu gelangen und sein Vorrat an Bananen ist wie seine Zeit begrenzt.

Nur gut, dass genug Bananen rumliegen, die er aufsammeln kann. Kürbisse und kleine Wände bieten eine Zeitlang Schutz vor den „Wandernden Wänden“ und Geistern. Aber es ist nicht einfach, sie aufzusammeln. Sie Schweben und verschwinden im Gras um anschließend hoch aufzusteigen. Springe, um sie zu erreichen.

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

Die Hauptcharaktere des Spiels sind Hugo und die sich ihm in den Weg stellenden Geister und „Wandernden Wände“. Daneben gibt Zwischenziele, die von Hugo eingesammelt werden können und die „Goldene Banane“ als finales Ziel jedes Levels. In diesem Kapitel werden sie kurz vorgestellt.

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 dem ultimativen Bananenbaum.

Aussehen

Würde man Hugo sehen, sähe man einen typischen Minion in der allseits bekannten und beliebten Erscheinung: klein, gelb und mit Latzhose aus Jeans.



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

Eigenschaften

Um weiterzukommen und nicht im Labyrinth gefangen zu bleiben, muss er Bananen einsammeln. Leider verliert er immer wieder Bananen bei Zusammenstößen mit den „Wandernden Wänden“ des Labyrinths. Auch die überall herumschwebenden Geister versuchen ständig, seine Bananen zu klauen und verfolgen ihn, wenn er ihnen zu nahekommt. Schafft es Hugo in ein neues Level, so kann er seine Bananen mitnehmen.

Aktionen und Animationen

Hugo kann innerhalb eines Levels Bananen sammeln oder sich vor den Wänden und Geistern schützen, indem er die auf der Spielfläche verstreuten Zwischenzielen aufsammelt. Hierzu muss er nur auf sie zulaufen. Aber Achtung: Zwischenziele bewegen sich hoch und runter. Mal sind sie zu hoch, mal verschwinden sie in dem Grass. Manchmal reicht es, sie anzuspringen.

In zufällig generierten Intervallen und Ereignissen, gibt Hugo Audiosequenzen ab. Hierbei werden aktuell drei Ereignisse unterschieden:

- 1) Während der normalen Bewegung ist in zufälligen Abständen eine Art fröhliches „Summen“ zu hören.
- 2) Kann Hugo eine Banane einsammeln, so ertönt zufallsgesteuert ein freches „Lachen“.
- 3) Bei der Kollision mit einer Wand oder einem Geist, ertönt zufallsgesteuert ein unruhiges „Wispern“ von Hugo. Bei einer schnellen Abfolge von Kollisionen, beispielsweise wenn er von Wänden gefangen ist, kommt es zu einer deutlich wahrnehmbaren Kumulation dieser Audiosignale.

Bewegungsmöglichkeiten

Hugo kann sich frei im Spielfeld bewegen und wird über Tastatur und Maus gesteuert. Die Maus steuert hierbei seine Blick- und Bewegungsrichtung. Mit der Leertaste kann er springen, um hoch schwebende Gegenstände einzusammeln oder über Geister hinwegzuspringen. Hierbei ist nicht garantiert, dass Hugo keine Punkte verliert, jedoch kann es ihm ein Entkommen ermöglichen.

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. Haben sie ihn entdeckt, folgen sie ihm. Nur mit einem schnellen Sprint kann er ihnen entkommen und sie verlieren seine Spur.

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 Hugo verfolgen, verlieren aber den Kontakt, wenn er sich permanent von ihnen entfernt. Das Sammeln von Kürbissen verbirgt Hugo eine Zeitlang vor den Geistern.

Mit zunehmendem Level werden der Schaden und notwendige Abstand, um nicht verfolgt zu werden, größer und die Zeitspanne, in der man sich vor ihnen verbergen kann, kleiner. Geister können von Hugo übersprungen werden, was jedoch keine Garantie bietet, nicht doch Punkte zu verlieren.

Aktionen und Animationen

Geister attackieren den Spieler, wenn er ihnen zu nah kommt, indem sie sich ihm nähern und wiederholt kollidieren⁸. Mit jeder Kollision verliert Hugo mehr Bananen. Das Annähern der Geister wird dem Spieler durch einen zunehmend lauter werdenden „Geistersound“ signalisiert. Eine Kollision und der Verlust von Bananen werden ebenfalls durch ein auditives Feedback angezeigt. Die Zahl über den Geistern zeigt an, wie hoch der Verlust an Bananen bei jeder Kollision ist.

Warnzeichen für den Spieler

Bei der Kollision kommt es zu einer Wechselwirkung zwischen zwei physikalischen Körpern (Geist und Spieler). Hierdurch kommt es zu einem „Zittern“ im Bild, welches deutlich wahrzunehmen ist.

Zudem wird ein dumpfer „Kollisionston“ als Soundeffekt abgespielt und Hugo gibt manchmal einen Ton der „Unruhe“ ab. Je kürzer die Intervalle hierbei sind, umso hörbarer wird dies Signal. Im Head Up Displays des Spiels erscheint zudem eine Mitteilung mit der Höhe der verlorenen Bananen in roter Schrift.

⁸ Die wiederholte Kollision ist in dieser Version deaktiviert und würde erst in höheren Leveln greifen.



Abbildung 3: Anzeige eines Verlustes an Bananen.

Geister spielen zudem ein dauerndes „Geisterhaftes Stöhnen“ ab. Dieser Ton verblasst mit zunehmendem Abstand und wird umgekehrt lauter. Das Hören des Tons ist ein Warnsignal.

Da die Kollision mit einem Verlust an Bananen verbunden ist, wird die Anzeige der vorhandenen Bananen um den erlittenen Verlust reduziert.

Bewegungsmöglichkeiten

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

„Wandernde Wände“

Bei den „Wandernden Wänden“ handelt es sich um ein Teil des Spielfeldes, welcher jedoch eine eigene Dynamik besitzt. Es handelt sich somit nicht um ein Non Player Character im eigentlichen Sinn.

Die „Wandernden Wände“ kommen aus den das Spielfeld begrenzenden Wänden und wandern zu der jeweils entgegengesetzten Wand. Sie können Zwischenziele und andere Wände kreuzen, ohne diese zu beeinflussen. Die Anzahl, der Ursprung sowie die Geschwindigkeit werden zufällig und abhängig vom Level festgelegt und können von Wand zu Wand variieren.

Aussehen

Das Aussehen der „Wandernden Wände“ ist eher schlicht und einfach. Sie stellen keine besonderen Objekte dar, sondern können eher als ein dynamischer Teil der Spielfläche gesehen werden.

Eigenschaften

„Wandernde Wände“ haben die Besonderheit, dass sie mit keinem der Spielobjekte interagieren oder kollidieren. Nur gegenüber dem Spieler zeigen sie sich als konkrete Wände, welche nicht überwunden werden können. Daher endet eine Begegnung mit einer Kollision.

Aktionen und Animationen

„Wandernde Wände“ sind rein passive Objekte im Spiel. Sie kommen aus den das Spielfeld begrenzenden Wänden und wandern in gerader Linie zu der jeweils entgegengesetzten Wand. Als Teil der Spielfläche wird hier bewusst auf ein auditives Feedback verzichtet.

Sie interagieren mit keinem Spielobjekt bis auf dem Spieler selbst. Kommt es zu einer Kollision, so wird ein dumpfer Ton als Effekt abgespielt. Hierbei verliert der Spieler Bananen. Es ist hierbei für den Spieler nicht vorhersehbar, wie hoch der Verlust sein wird.

Warnzeichen für den Spieler

Bei der Kollision kommt es zu einer Wechselwirkung zwischen zwei physikalischen Körpern (Wand und Spieler). Hierdurch kommt es zu einem „Zittern“ im Bild, welches deutlich wahrzunehmen ist.

Zudem wird ein dumpfer „Kollisionston“ als Soundeffekt abgespielt und Im Head Up Displays des Spiels erscheint eine Mitteilung mit der Höhe der verlorenen Bananen in roter Schrift.

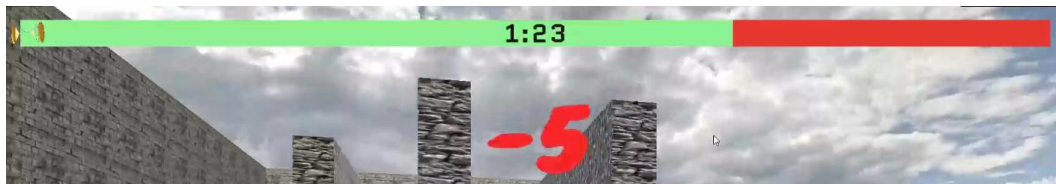


Abbildung 4: Anzeige eines Verlustes an Bananen.

Da die Kollision mit einem Verlust an Bananen verbunden ist, wird die Anzeige der vorhandenen Bananen um den erlittenen Verlust reduziert und Hugo gibt manchmal einen Ton der „Unruhe“ ab. Je kürzer die Intervalle hierbei sind, umso hörbarer wird dies Signal.

Bewegungsmöglichkeiten

Die Bewegungsmöglichkeit der „Wandernden Wände“ ist sehr begrenzt. Sie bewegen sich lediglich in gerade Linie von ihrem Ursprung zur gegenüberliegenden Wand, um durch diese aus dem Spiel zu gehen.

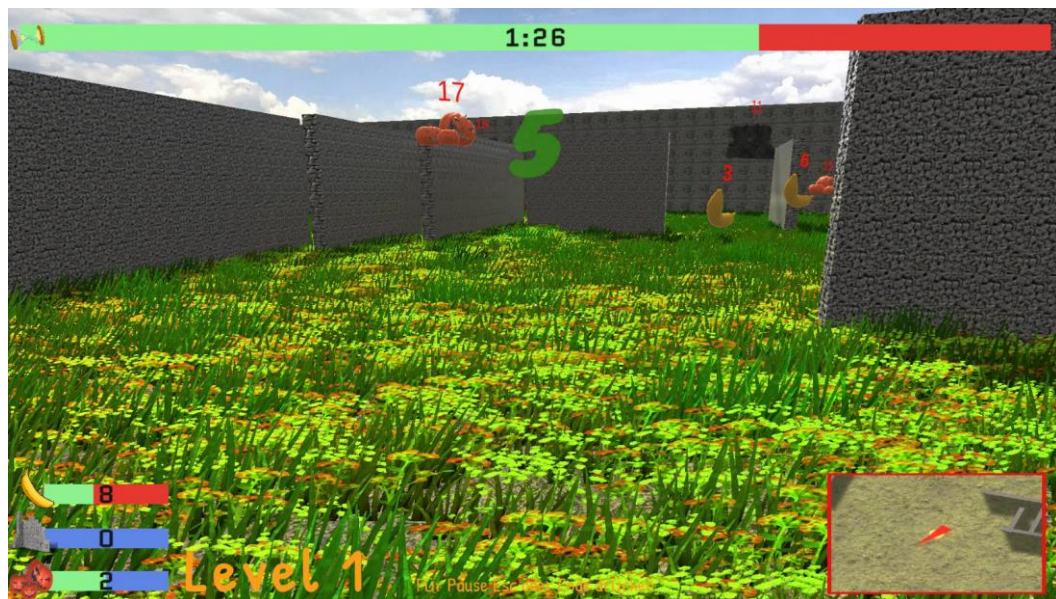


Abbildung 5: „Wandernde Wände“ können mit Hugo kollidieren.

Zwischenziele und finales Levelziel

Das Spiel kennt drei verschiedene Zwischenziele und ein finales Levelziel in Form einer „Goldenen Banane“. Zwischenziele sind alle innerhalb der Begrenzung befindlichen Gegenstände, welche durch den Spieler im Spielverlauf aufgesammelt werden können und ihm Vorteile und Bananen verschaffen.

Erschwert wird das Aufsammeln durch eine ständige Auf- und Abwärtsbewegung der Ziele. Hierdurch schweben sie abwechselnd in einer nicht erreichbaren Höhe oder verschwinden im Rasen der Spielfläche. Nur dazwischen können sie eingesammelt werden. Hierfür müssen sie überlaufen oder angesprungen werden.

Die Anzahl der Zwischenziele, ihre Position und Wertigkeit werden zufällig in Abhängigkeit vom erreichten Level konfiguriert. Ebenso wird das finale Levelziel zufällig positioniert. Jedes Ziel kann nur einmal eingesammelt werden, anschließend verschwindet es vom Spielfeld. Das Erreichen des finalen Zwischenziels beendet das Level.

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 bei Erreichen der „Goldenen Banane“ 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 Besitz an Bananen erhöhen. Muss Hugo ein Level zurück gehen, so wird sein Bestand an Bananen als Trost wieder auf null gesetzt.



Abbildung 6: Das Zwischenziel „Banane“ gibt Hugo die Kraft, weiterzumachen.

Auditives und visuelles Feedback

Das erfolgreiche Aufsammeln einer Banane wird mit einem kurzen Audiosignal sowie einer zentralen Benachrichtigung im Head Up Display quittiert:

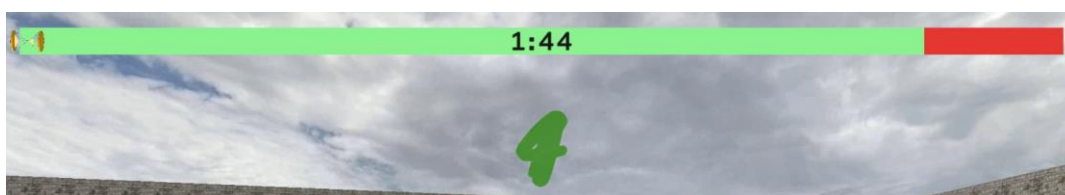


Abbildung 7: Anzeige eines Gewinns an Bananen.

Zudem wird die Anzeige für eingesammelte Bananen um den angezeigten Wert erhöht. Hin und wieder nimmt Hugo das Einsammeln einer Banane als Anlass für ein „herzhaftes Lachen“.

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. Auch eine Kollision ist ohne Wirkung.

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 „Halloween Kürbis“ 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 8: Das Zwischenziel „Halloween Kürbis“ schützt eine Zeitlang vor Geister.

Auditives und visuelles Feedback

Das erfolgreiche Aufsammeln eines „Halloween Kürbis“ wird mit einem kurzen Audiosignal sowie einer zentralen Benachrichtigung im Head Up Display quittiert. Neben der Anzeige der tatsächlich gewonnenen Zeit erfolgt ein Hinweis auf den Zweck selbst:



Abbildung 9: Anzeige eines Zeitgewinns gegen Geister.

Zudem wird die Anzeige für eingesammelte „Halloween Kürbisse“ auf den angezeigten Sekundenwert erhöht, sofern der aktuelle Wert kleiner ist.

Zwischenziel „Wandsegment“

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

Wie schon zuvor bei Zwischenzielen vom Typ „Halloween Kürbis“, 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 „Wandsegment“ wirken nur, sofern die restliche Zeit kleiner als die der neu hinzugewonnenen Zeit ist.

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



Abbildung 10: Das Zwischenziel „Wandsegment“ schützt eine Zeitlang vor den Wänden.

Auditives und visuelles Feedback

Das erfolgreiche Aufsammeln eines „Wandsegments“ wird mit einem kurzen Audiosignal sowie einer zentralen Benachrichtigung im Head Up Display quittiert. Neben der Anzeige der tatsächlich gewonnenen Zeit erfolgt ein Hinweis auf den Zweck selbst:

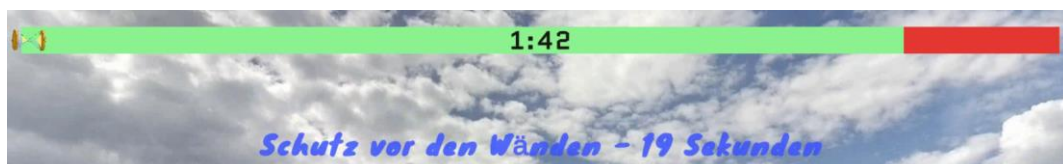


Abbildung 11: Anzeige eines Zeitgewinns gegen "Wandernde Wände".

Zudem wird die Anzeige für eingesammelte „Wandsegmente“ auf den angezeigten Sekundenwert erhöht, sofern der aktuelle Wert kleiner ist.

Finales Levelziel „Goldene Banane“

In jedem Level gibt es genau ein finales Levelziel, die „Goldene Banane“. 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 jedoch der Bananenbestand wieder auf null gesetzt.



Abbildung 12: Das finale Levelziel „Goldene Banane“ beendet das Level.

Auditives Feedback

Das erfolgreiche Erreichen der „Goldenen Banane“ wird mit einer kurzen Fanfare und dem Einblenden des Leveldialogs quittiert.

Punkteberechnung

Für die Punkteberechnung kann somit generell gesagt werden, dass die Berechnung der Punkte bei Erreichen eines Zwischenziels sich grundsätzlich nach dem Typ des Ziels richtet. Im Folgenden ist hier nochmals die Methodik zur Verdeutlichung zusammengefasst:

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

Levels

In diesem Abschnitt wird auf den Aufbau der Levelsystems im Spiel eingegangen. Charakteristisch hierfür ist die Verwendung einer immer gleichbleibenden Basisszene und gleichartiger Spielobjekte in allen Levels. Lediglich die Konfiguration der Spielobjekte selbst wird zu Beginn eines neuen Levels angepasst.

Die Anpassung erfolgt hierbei auf Basis globaler und fest vorgegebener Wertebereiche auf Programmebene. Diese werden durch Algorithmen unter Einbeziehung des jeweils aktuellen Levels angepasst⁹ und die konkreten Werte zum Zeitpunkt der Erzeugung zufallsbasiert festgelegt. Als direkte Folge ist jedes Level anders als das zuvor generierte.

In der Umsetzung resultieren daraus praktisch unendlich viele Möglichkeiten des Ablaufs und des Schweregrades. Zudem gibt es keinen fest vorgegebenen besten Lösungsweg. Alle Gegenstände

⁹ In der aktuellen Version des Spiels erfolgt noch keine Anpassung der Eigenschaften.

werden zufallsgesteuert immer wieder neu generiert. Die folgende Abbildung zeigt einen Level aus Entwicklersicht kurz nach der Erstellung:



Abbildung 13: Basislevel nach etwa 30 Sekunden aus der Entwicklersicht

Die zentrale Basisszene selbst besteht hierbei aus der Rasenfläche sowie der umgebenen Mauer. Bei den auf der Rasenfläche zu sehenden Objekten handelt es sich um im Kapitel Charaktere und Zielobjekte näher ausgeführten Spielobjekte, mit denen der Spieler (Hugo) interagieren kann.

Zu sehen ist mittig rechts auch ein NPC Geist, sowie eine bereits größere Zahl „Wandernder Mauern“. Ganz hinten und etwa mittig ist das finale Levelziel, die „Goldene Banane“ zu sehen.

Startlevel

Wie bereits zuvor ausgeführt, unterscheiden sich einzelne Level lediglich durch ihre Konfiguration. Der erste Level ist das Startlevel. Wird dieser Level gewonnen, so wird das Spiel mit dem nächsthöheren Level fortgeführt.

Es unterscheidet sich von den nachfolgenden Leveln durch die Behandlung im Fall des Verlierens. Während man in höheren Leveln um ein Level zurückgestuft wird, wiederholt man im Startlevel dies nach jeder verlorenen Runde erneut. Somit endet das Spiel streng genommen nicht.

Beschreibung des Levels

Der Spieler erscheint an einen zufällig bestimmten Punkt eines großen, durch Mauern umrandeten Spielfeldes. Er sieht verschiedene Gegenstände auf dem Feld verteilt. Vom Aussehen lassen sich Rückschlüsse auf deren Nutzen ziehen¹⁰.

Besonders sticht das finale Levelziel, die „Goldene Banane“ visuell hervor. Diese kann entweder direkt gesehen werden, oder aber deren Richtung anhand der Richtungsanzeige in der MiniMap¹¹ bestimmt werden. Im Startlevel verfügt der Spieler über keine Bananen.

¹⁰ Siehe hierzu den Abschnitt Charaktere und Zielobjekte

¹¹ Siehe hierzu den Abschnitt Head Up Display

Die folgende Abbildung zeigt eine typische Spielszene kurz nach Start des ersten Levels:



Abbildung 14: Spieleindruck zu Beginn des ersten Levels

Aufgabe des Spielers

Aufgabe des Spielers (Hugo) ist das rechtzeitige Erreichen des finalen Levelziels innerhalb der Spielzeit mit mindestens einer Banane. Hierfür hat er im Startlevel zwei Minuten Zeit. Wird dies nicht geschafft, so hat er das Level verloren.

Auf dem Spielfeld vorhandene Gegenstände, insbesondere Bananen, können, müssen – bis auf eine - aber nicht eingesammelt oder angesteuert werden. Insbesondere darf der Bananenbestand zwischen dem Levelstart und -ende negativ sein. Gerade zu Anfang kann es jedoch sinnvoll sein, Bananen auf Vorrat zu sammeln, da diese in ein neues Level mitgenommen werden dürfen.

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. Hiervor kann er sich für eine Zeit schützen, indem er die im Spiel verteilten Wandstücke einsammelt.

Gefahr droht dem Spieler zudem von Geistern, welche sich ab einer gewissen Distanz direkt auf den Spieler zubewegen und ihn verfolgen. Kommen sie ihn zu nahe, verliert er kontinuierlich Bananen¹². Nur wenn er sich von den Geistern über eine gewisse Distanz entfernt, verlieren sie ihn. Durch das Sammeln der im Spiel verteilten „Halloween Kürbisse“ kann er sich vor den Geistern für eine Zeit verbergen. Weder klauen sie seine Bananen, noch verfolgen sie ihn.

¹² In dieser Version wurde der kontinuierliche Faktor deaktiviert und würde in einen höheren Level greifen.

Wie wird das Level beendet

Das Startlevel endet analog zu den nachfolgenden Levels. Wird das finale Levelziel, die „Goldene Banane“, innerhalb der vorgegebenen Zeit erreicht und besitzt der Spieler (Hugo) mindestens eine Banane, so ist das Level gewonnen. 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 Startlevel verloren, so wiederholt er dies. In jedem Fall fängt er in einem neuen Level mit mindestens null Bananen an, auch wenn er zuvor weniger hatte.

Übersicht Spielverlauf

Wie bereits zuvor ausgeführt, existieren keine im Detail festgelegte Level oder Reihenfolgen. Durch die algorithmisch gesteuerte Konfiguration neuer Level nach oben und Wiederholung des Startlevels bei einem verlorenen Startlevel, kann theoretisch eine beliebige Anzahl an Leveln erreicht und ohne Ende gespielt werden.

Grundsätzlich handelt es sich jedoch um einen recht einfachen Verlauf, dessen Aufbau in dem nachfolgenden Diagramm des Spielverlaufs auf Ebene der Level verdeutlicht wird:

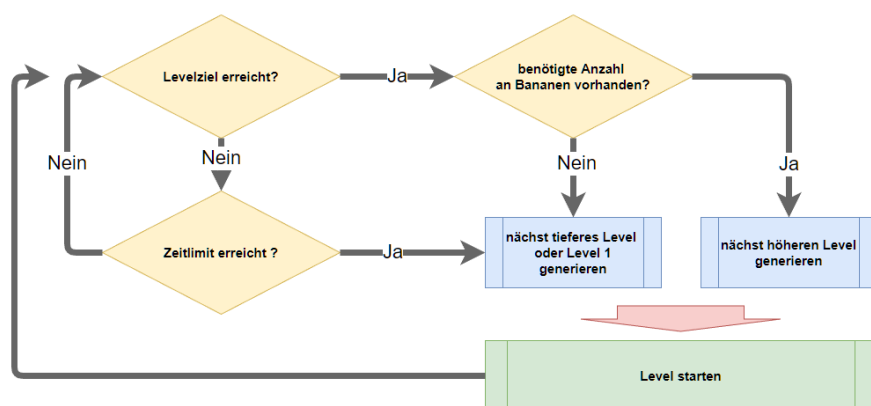


Abbildung 15: Spielverlauf auf Ebene der Levels

Im Kern wird eine ständige Schleife durchlaufen. Wurde das Levelziel erreicht? Wenn nein, ist noch Spielzeit vorhanden? Wenn eine der Bedingungen wahr ist, so endet das Level.

Für den Fall, dass das Levelziel erreicht wurde, entscheidet die Anzahl der Bananen. Ist mindestens eine Banane vorhanden, so ist das Spiel gewonnen. In jedem Fall wird vor dem Start des neuen Levels (grün) dieses generiert (blau).

Skalierung der Folgelevel

Alle folgenden Level basieren auf den anfänglichen Basislevel, werden aber durch Anpassung der

vorhandene Parameter zunehmend komplexer¹³. Dies wirkt sich insbesondere auf vier Kernelement aus:

- **NPC „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 sie den Spieler (Hugo) früher erkennen können.
- **Zwischenziel „Wandsegmente“**: Die Anzahl dieser Ziele wird weniger und der maximale Zeitraum, in dem sie Schutz bieten, wird kürzer.
- **Zwischenziel „Halloween Kürbisse“**: Die Anzahl dieser Ziele wird weniger und der maximale Zeitraum, in dem sie Schutz bieten, wird kürzer.

Menü und Dialoge

Das Programm verfügt über ein zentrales Startmenü, das den Einstiegspunkt in das Spiel darstellt, sowie einen Leveldialog, der den Übergang zwischen zwei Leveln begleitet.

Startmenü

Das Startmenü stellt den Einstieg in das Spiels dar, an dem immer wieder aus dem Spiel durch Betätigen der Taste ESC oder ENDE zurückgekehrt werden kann. Hierbei muss angemerkt werden, dass die Taste ESC im Vollbildmodus bereits für die Rückkehr in den Normalmodus reserviert ist. Aus Gründen der Konsistenz zu allgemein verbreiteten Standards wurde dies belassen. Für den Fall der Vollbilddarstellung bewirkt daher das Betätigen der Taste ENDE einen Aufruf des Startmenüs.

Aus dem Startmenü lassen sich alle weiteren Untermenüs des Spiels über Buttons einfach erreichen. Im Folgenden wird zunächst auf das Startmenü, dann auf die einzelnen Untermenüs eingegangen.

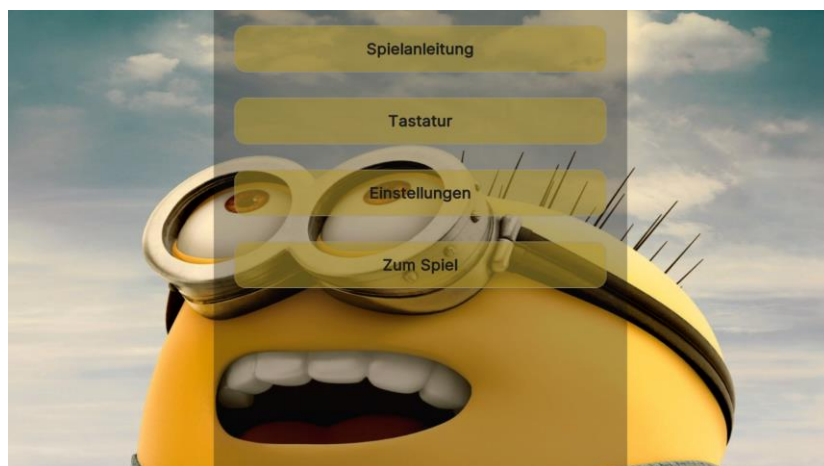


Abbildung 16: Übersicht Startmenü

¹³ Die Skalierung der Folgelevel ist in dieser Version vorbereitet, jedoch noch nicht implementiert.

Button	Anzeige/Eingabe	Aktion
Spielanleitung		Sprung ins Untermenü Spielanleitung
Tastatur		Sprung ins Untermenü Tastatur
Einstellungen		Sprung ins Untermenü Einstellungen
Zum Spiel		Wechselt zum Spiel

Tabelle 1: Übersicht Interaktion Startmenü

Untermenü Spielanleitung

Im Untermenü Spielanleitung wird der Spieler in Form einer kurzen Geschichte abgeholt und in die Welt des Spiels versetzt. Warum ist er im Labyrinth? Wie kam es dazu und was ist seine Aufgabe?

Aufgabe der Spielanleitung ist somit eher die emotionale Ansprache als eine kurze und formale Beschreibung der Interaktionsmöglichkeiten. Letzteres kann dem Untermenü Tastatur entnommen werden.

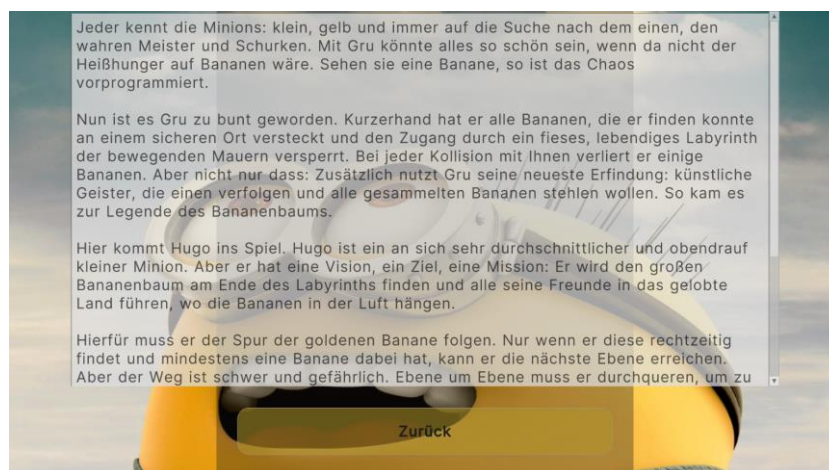


Abbildung 17: Untermenü Spielanleitung

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

Tabelle 2: Übersicht Interaktion Spielanleitung

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.

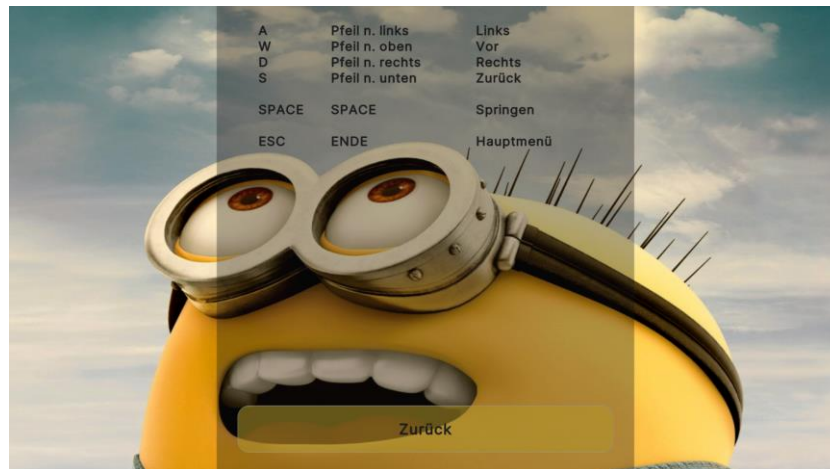


Abbildung 18: Untermenü Tastatur

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

Tabelle 3: Übersicht Interaktion Tastatur

Untermenü Einstellungen

Alle für das Spiel zu machenden Einstellungen finden sich im Untermenü Einstellungen. Bei einer Übernahme werden sie für das aktuelle Spiel angewendet¹⁴. Eine darüberhinausgehende Speicherung ist nicht vorgesehen.

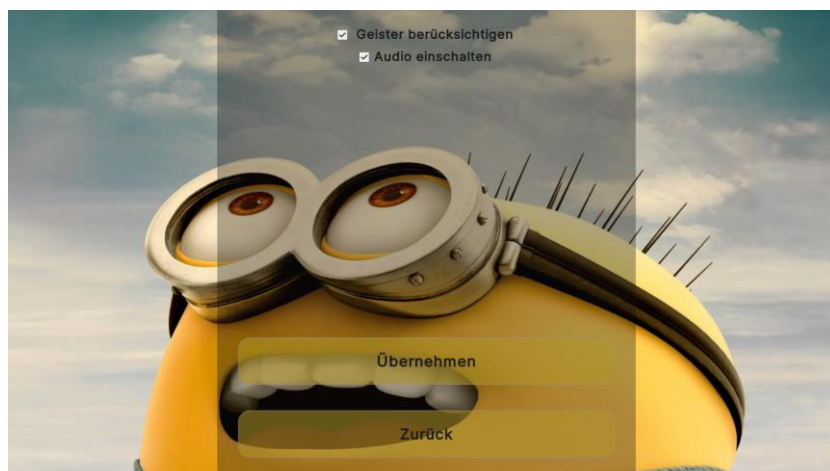


Abbildung 19: Untermenü Einstellungen

¹⁴ In dieser Version werden die hier gemachten Einstellungen nicht berücksichtigt.

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 4: Übersicht Interaktion Einstellungen

Interaktions- und Navigationsdiagramme

In der folgenden Abbildung sind die Interaktionsmöglichkeiten innerhalb des Spiels nochmals in Form eines Diagramms aufgeführt:

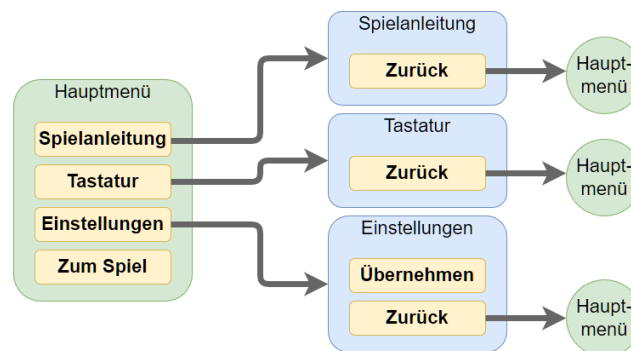


Abbildung 20: Navigationsübersicht des Hauptmenüs

Wie bereits zu Anfang ausgeführt, kann der Spieler jederzeit mit Hilfe der Taste ESC oder ENDE das Startmenü aufrufen. Während das Menü aktiv ist, pausiert das Spiel.

InGame Menü

Auf die Konzeption eines eigenständigen InGame Menüs wurde verzichtet. Wie zuvor bereits erörtert, kann der Spieler jederzeit mit Hilfe der ESC oder ENDE Taste zum Hauptmenü des Spiels wechseln und wieder zurückkehren.

Einschränkend ist anzumerken, dass im Vollbildmodus nur die ENDE Taste effektiv ist, da die andere Option allgemein für die Rückkehr aus dem Vollbildmodus reserviert, und vielen Internetnutzern in

dieser Funktion bekannt ist.

Auf Grund des Zielplattform wurde auf eine explizite Beendigung des Spiels verzichtet. Sobald ein Spieler die Webseite verlässt, endet das Spiel. Hierdurch ist auch ein kompletter Neuanfang einfach zu realisieren.

Leveldialog

Die Aufgabe des Leveldialogs ist die Begleitung des Spielers aus dem vorangegangenen Level in den nächstfolgende. Hierbei passt sich der Dialog der aktuellen Spielsituation an, wie in den folgenden zwei Abbildungen zu sehen ist:



Abbildung 21: Leveldialog zwischen zwei Levels im Kontext „Level gewonnen“



Abbildung 22: Leveldialog zwischen zwei Levels im Kontext „Level verloren“

Hierbei existieren innerhalb des Dialogs grundsätzlich immer zwei Optionen. Zum einen die immer links angeordnete positiv besetzte Position, welche in das nächste Level führt, zum anderen die immer rechts angeordnete negativ besetzte Position, die in das Hauptmenü verzweigt und einer Beendigung des Spiels entspricht. Hierbei wurde auf eine positive Ansprache geachtet, mit der der Spieler zu einem Weiterspielen animiert werden soll.

Head Up Display

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

Die folgende Abbildung zeigt einen beispielhaften Screenshot eines Spiels mit allen relevanten und kontinuierlich angezeigten Elementen des aktuellen Head Up Displays:

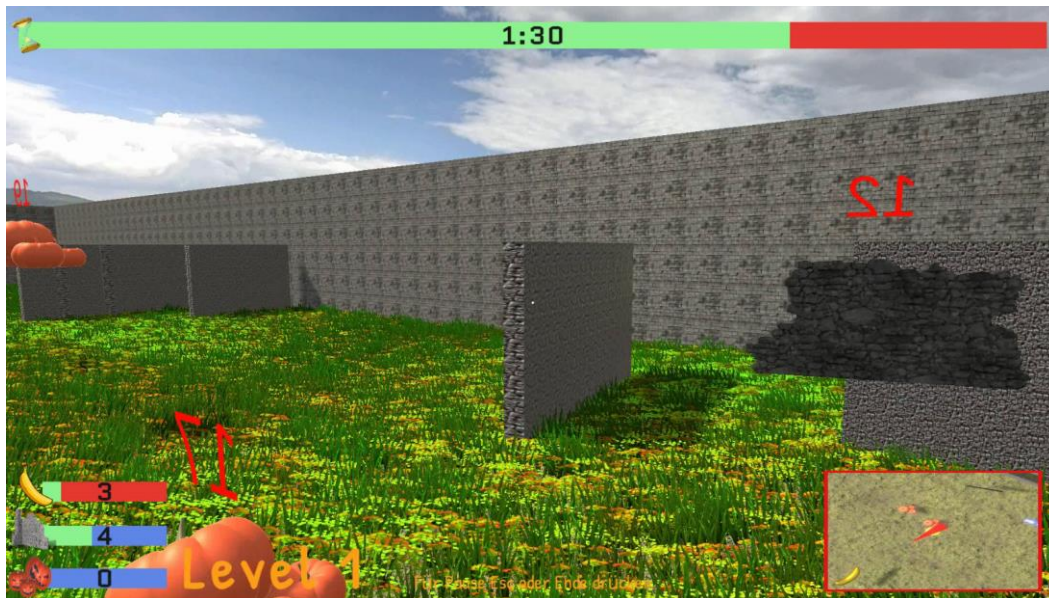


Abbildung 23: Spielszene zur Demonstration des In Game Menüs

Im oberen Bereich wird ein Fortschrittsbalken zur Anzeige der noch verbleibenden Zeit über die gesamte Breite des Bildschirms angezeigt:

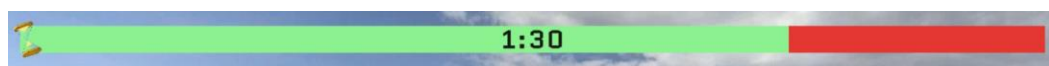


Abbildung 24: Oberer Bereich des In Game Menüs

Hierbei wurde der Hintergrund in Rot gehalten, um den negativen Aspekt bereits vergangener Zeit anzuzeigen. Die noch verbleibende Zeit wird hingegen in einem eher positiven grün gezeichnet.

Im unteren Bereich bilden eine Gruppe von drei Balkenanzeigen auf der linken Seite, sowie eine MiniMap auf der rechten Seite die Hauptbestandteile des Head Up Displays. Dazwischen erscheinen Angaben zum aktuellen Level sowie ein Hinweis zum Aufruf des Startmenüs.



Abbildung 25: Unterer Bereich des In Game Menüs

Auch hier wurde bei der Umsetzung der Anzeigeelemente verstärkt auf die Visualisierung geachtet,

um den Spieler einfach zu informieren. Hierfür wurde eine Übersicht auf Basis von Fortschrittsbalken, hier links im Bild, implementiert.

Wie bei der Zeitanzeige erscheinen negative Aspekte hierbei einheitlich in roter, neutrale in blauer und positive in grüner Farbe. Die folgende Abbildung verdeutlicht dies:



Abbildung 26: Balkenanzeige zur Anzeige wichtiger Werte

Das Fehlen von Bananen verhindern das Weiterkommen in einen höheren Level. Daher ist der Balken zu Anzeige des Vorrats an Bananen rot hinterlegt. Kein Zeitguthaben für den Schutz vor Geistern oder den „Wandernden Wänden“ ist zwar nicht vorteilhaft, behindern aber nicht ein Weiterkommen. Daher ist deren Hintergrund in blau gehalten.

Der Grünanteil der Balken steht für die Anzahl der Bananen bzw. eines Zeitguthabens für den Schutz vor Geistern oder den „Wandernden Wänden“.

Die MiniMap auf der rechten Seite folgt den Spieler und zeigt das Geschehen von oben. Hierbei ist der Spieler (Hugo) immer mittig als gelber Punkt visualisiert.



Abbildung 27: MiniMap zur Anzeige des Spielgeschehens von oben

Über dem gelben Punkt befindet sich ein Pfeil, dessen Spitze stets in Richtung des finalen Levelziels, der „Goldenen Banane“ (hier unten links) ausgerichtet ist. Hierdurch erhält der Spieler einen zusätzlichen Punkt zur Orientierung.

Wichtige Ereignisse

Für den Spieler wichtige Ereignisse werden zudem als Pop Up Nachrichten zur Verfügung gestellt, welche in kurzer Zeit verblassen. Hierbei wurde wieder mit der im Spiel üblichen Farbcodierung der Anzeigebalkens gearbeitet.

Rote Schrift steht für negative (Verlust von Bananen), grün für positive (Gewinn von Bananen) Ereignisse. Gewinne von Schutzzeiten werden in einem neutralen Blau wiedergegeben:

Es werden vier unterschiedliche Nachrichten ausgegeben. Details hierzu finden sich im Abschnitt Charaktere und Zielobjekte:

- Das Gewinnen von Bananen:

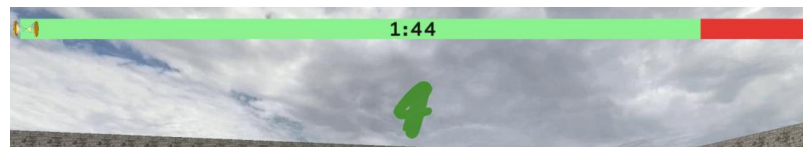


Abbildung 28: Gewinn von Bananen

- Der Verlust von Bananen durch Geister oder Wände:

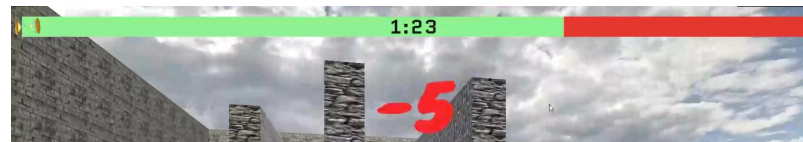


Abbildung 29: Verlust von Bananen

- Das Gewinnen von Schutzzeiten für „Wandernde Wände“ in blau



Abbildung 30: Gewinn von Schutzzeit vor „Wandernden Wänden“.

- Das Gewinnen von Schutzzeiten für Geister in blau



Abbildung 31: Gewinn von Schutzzeit vor Geister.

Steuerung

Die Spielsteuerung des Spiels ist für Desktop, Tastatur und Maus optimiert. Insofern wird hier bewusst kein mobil-first Ansatz verfolgt.

Auf Grund der Verwendung des neuen Input Systems in Unity kann die Steuerung bei Bedarf für weitere Devices wie Gamepad ohne Modifizierung der Codebasis erweitert oder die vorhandene modifiziert werden. Die aktuelle Belegung der Tasten richtet sich nach der gängigen Belegung in bekannten Spielen:

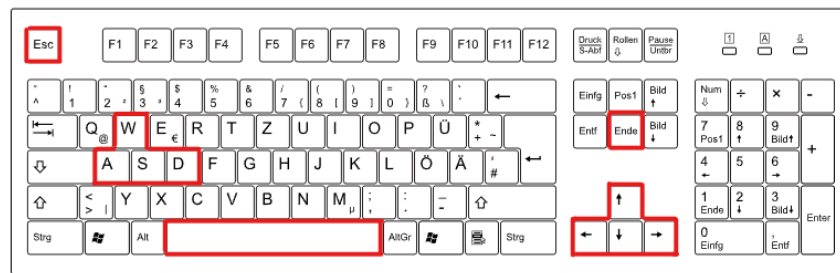


Abbildung 32: Belegung der Tastatur

In der nachfolgenden Tabelle sind alle Möglichkeiten zur Interaktion im Detail aufgelistet. Insbesondere wurde Wert auf eine alternative Belegung der Steuerung gelegt, welche Linkshänder entgegenkommt:

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

Tabelle 5: Belegung der Tastatur und Maus

Gameplay

Auf einem grundsätzlich gleichbleibenden, durch hohe Mauern umrandeten Spielfeld muss der Spieler von einem willkürlichen Startpunkt zu einem nicht zwingend sofort sichtbaren Zielpunkt (finale Levelziel, „Goldene Banane“) 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, behält jedoch seine gesammelten Bananen. Wird eine der Vorgaben nicht erfüllt, so wird das Spiel im nächsttieferen Level fortgesetzt. Hierbei wird sein Bestand an Bananen auf null gesetzt. Wird der erste und einfachste Level verloren, so wird dieser wiederholt. Insofern kann das Spiel an sich nicht verloren werden.

Zu Anfang des Spieles (erster Level, Startlevel) verfügt der Spieler über keine Bananen. Durch Überlaufen oder Anspringen der auf dem Spielfeld verteilten Bananen ist er jedoch in der Lage, seinen Vorrat zu vergrößern. Jede Banane kann nur einmal eingesammelt werden und verschwindet hierdurch.

Neben Bananen existieren weitere Objekte (Zwischenziele). Zwischenziele vom Typ „Wandsegment“ bieten Schutz vor den „Wandernden Wänden“, „Halloween Kürbisse“ vor Geister. Dieser Schutz gilt jedoch nur für eine gewisse Anzahl an Sekunden. Während dieser Zeit ist der Spieler für Geister unsichtbar und eine Kollision mit ihnen oder einer „Wandernden Wand“ bleibt ohne Folgen. Zwischenziele können analog zu Bananen durch Überrennen oder Anspringen einmalig eingesammelt werden.

Sowohl Bananen als auch Zwischenziele verharren während eines Levels an ihrer Position, ändern jedoch beständig ihre Höhe. Hierbei verschwinden sie abwechselnd im Boden und sind für den Spieler nicht oder nur schwer sichtbar, oder schweben in eine für den Spieler nicht erreichbaren Höhe.

Im Verlauf des Spiels wirken der Erfüllung der Aufgaben zwei Hindernisse entgegen. Bei den „Wandernden Wänden“ handelt es sich um Wandstücke, die aus den umgebenen Wänden des Spielfeldes erscheinen und sich gradlinig zur gegenüberliegenden Wand bewegen und dort verschwinden. Bei der Kollision des Spielers mit den Wänden, verliert dieser einen Teil der eingesammelten Bananen. Verfügt er über keine Bananen, so wird dessen Bestand trotzdem um die Zahl reduziert. Um das Level erfolgreich zu beenden, muss dieser Bestand zunächst ausgeglichen werden.

Ein weiteres Hindernis bilden Geister. 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 keine Hindernisse behindert. Erreicht ein Geist den Spieler, so verliert dieser in konstanter Folge Bananen. Dem Geist kann entkommen werden, wenn eine gewisse Distanz zwischen Geist und Spieler geschaffen wird.

Mit jedem erfolgreich absolvierten Level ist es schwieriger, das Ziel zu erreichen, da tendenziell immer weniger Objekte vom Typ „Banane“ mit immer kleineren Werten vorhanden sind. Gleichzeitig wird sowohl die Anzahl an „Halloween Kürbissen“ und „Wandfragmenten“ sowie deren Schutzwirkungen immer kleiner. Eine weitere Schwierigkeit stellen die zunehmend größeren und schädlicheren „Wandernden Wände“ und sensiblere Geister da.

Ausblick

Bei der Umsetzung entstanden einige weitere Ideen für das Spiel, welche jedoch aufgrund der Zeit nicht umgesetzt wurden. Trotzdem sollen sie hier kurz erwähnt werden:

1. Aktuell verharren alle Zielobjekte lokal an einem Ort. Es könnte für mehr Spannung sorgen, wenn sich die Zielobjekte stattdessen innerhalb des Spielfeldes in eine Art Schweben frei bewegen würden.
2. Aktuell ist das Levelziel darin konstant, dass die „Goldenen Banane“ in einer definierten Zeit mit mindestens einer Banane erreicht werden muss. Es könnte interessant sein, die Anzahl von Level zu Level zu erhöhen. Auch die Einbeziehung der gebrauchten Zeit als Ersatz für die Vorgabe einer festen Zeitspanne könnte neue Aspekte öffnen. So könnte die Anzahl an benötigten Bananen proportional zur gebrauchten Zeit sein.
3. Die „Wandernden Wände“ könnten dahingehend verfeinert werden, dass sie ein richtiges und eventuell konstantes Labyrinth mit Gängen bilden, durch denen nach wie vor ein Weg gefunden werden kann. Hierfür müsste das Handling bei einer Kollision in Gänze überarbeitet werden.
4. Die Gestaltung der das Spielfeld umgebenen Begrenzung sowie der „Wandernden Wände“ könnte verbessert werden bis hin zu einem Wechsel von einer konkreten Wand zu einer Art realistischen, „lebenden“ Hecke.
5. Das Spiel könnte realistischer und interessanter gestaltet werden, wenn zwischen First- und Third Person gewechselt werden könnte. Hierzu bedarf es jedoch eines – nutzbaren – Modells.
6. Die Koordinierung der von Hugo wiedergegebenen Audiosequenzen könnte verbessert werden. Insbesondere werden sie im Moment ereignis- und zufallsgesteuert abgespielt. Der aktuelle Spielkontext spielt hierbei jedoch keine Rolle. Beispielsweise ist der Verlust der einzigen Banane schlimmer als der Verlust einer von vielen. Geht man nach den Filmen, wäre ein Minion im ersten Fall wahrscheinlich traurig, im zweiten eher frech. Dies könnte ein wirklicher Mehrwert sein.
7. Das Anbieten einer Bestenliste kann ein Anreiz zum Spielen bieten und war in den ersten Entwürfen noch Bestandteil. In der Praxis zeigten sich die zwei ersten Ideen problembehaftet: Die Interaktion von Unity mit der umgebenen Website ist lösbar, aber nicht ganz trivial. Die Einarbeitung würde in diesen Rahmen zu viel Zeit kosten. Eine Lösung rein in PHP stellt ebenfalls keine Lösung dar, da PHP rein auf der Website basiert und daher ebenso wenig mit Unity interagiert.

Credits

Soweit nicht anders angegebenen, stammen alle verwendeten Assets aus dem Unity Asset Store oder von mir.

- **Hintergründe und Abbildungen**

Abbildung 1: <https://www.pngwing.com/en/free-png-pbamc>
Frei für die nicht Kommerzielle Nutzung

Hintergrund Hauptmenü: <https://wallpaperaccess.com/despicable-me#related>
Frei für die nicht Kommerzielle Nutzung

- **Musik und Sound**

Hintergrundmusik und Soundeffekte aus dem Unity Asset Paket **Casual Game Sound**
Lizenz: CC zero (Creative Commons zero)

Sound der Geister: **ghost_whispering01a-forward** von **toam** auf Freesound. Frei mit Bitte um Quellenangabe: <http://www.freesound.org/people/toam/>

HeHeHe Feedback: <https://www.soundboard.com/sb/sound/884073>

Hmhm Feedback: <https://www.soundboard.com/sb/minions>

Restless Feedback: <https://www.soundboard.com/sb/sound/884081>

Frei für die nicht Kommerzielle Nutzung

- **Fonts**

Fonts verfügbar durch GoogleFonts: <https://fonts.google.com/>

Schoolbell: <https://fonts.google.com/?query=schoolbell>

Quantico: <https://fonts.google.com/?query=quantico>

Patrick Hand: <https://fonts.google.com/?query=patrick+hand>

Knewave: <https://fonts.google.com/?query=knewave>

Zur Lizenzierung finden sich auf der Seite der Schriftarten detaillierte Informationen.

- **Modelle**

Alle Modelle stammen von Turbosquid (<https://www.turbosquid.com>) und sind frei nach Anmeldung.

Alle Bananen:

<https://www.turbosquid.com/de/3d-models/3d-model-banana-fruits-1581307>

„Wandsegmente“:

<https://www.turbosquid.com/de/3d-models/ruins-stone-obj-free/1125163>

Geister und „Halloween Kürbisse“:

<https://www.turbosquid.com/de/3d-models/halloween-pumpkin-and-ghost-free-pack-3d-1807791>

Abbildungen

Abbildung 1: "Hugo" ist der Held des Spiels und Spielercharakter.	11
Abbildung 2: Der NPC „Geist" verfolgt Hugo und klaut seine Bananen.	13
Abbildung 3: Anzeige eines Verlustes an Bananen.	14
Abbildung 4: Anzeige eines Verlustes an Bananen.	15
Abbildung 5: „Wandernde Wände" können mit Hugo kollidieren.	15
Abbildung 6: Das Zwischenziel „Banane" gibt Hugo die Kraft, weiterzumachen.	16
Abbildung 7: Anzeige eines Gewinns an Bananen.	16
Abbildung 8: Das Zwischenziel „Halloween Kürbis" schützt eine Zeitlang vor Geister.	17
Abbildung 9: Anzeige eines Zeitgewinns gegen Geister.	17
Abbildung 10: Das Zwischenziel „Wandsegment" schützt eine Zeitlang vor den Wänden.	18
Abbildung 11: Anzeige eines Zeitgewinns gegen "Wandernde Wände".	18
Abbildung 12: Das finale Levelziel „Goldene Banane" beendet das Level.	19
Abbildung 13: Basislevel nach etwa 30 Sekunden aus der Entwicklersicht.	20
Abbildung 14: Spieleindruck zu Beginn des ersten Levels	21
Abbildung 15: Spielverlauf auf Ebene der Levels.	22
Abbildung 16: Übersicht Startmenü.	23
Abbildung 17: Untermenü Spielanleitung.	24
Abbildung 18: Untermenü Tastatur	25
Abbildung 19: Untermenü Einstellungen	25
Abbildung 20: Navigationsübersicht des Hauptmenüs.	26
Abbildung 21: Leveldialog zwischen zwei Levels im Kontext „Level gewonnen“	27
Abbildung 22: Leveldialog zwischen zwei Levels im Kontext „Level verloren"	27
Abbildung 23: Spielszene zur Demonstration des In Game Menüs.	28
Abbildung 24: Oberer Bereich des In Game Menüs	28
Abbildung 25: Unterer Bereich des In Game Menüs	28
Abbildung 26: Balkenanzeige zur Anzeige wichtiger Werte.	29
Abbildung 27: MiniMap zur Anzeige des Spielgeschehens von oben.	29
Abbildung 28: Gewinn von Bananen.	30
Abbildung 29: Verlust von Bananen.	30
Abbildung 30: Gewinn von Schutzzeit vor „Wandernden Wänden".	30
Abbildung 31: Gewinn von Schutzzeit vor Geister.	30
Abbildung 32: Belegung der Tastatur.	31

Tabellen

Tabelle 1: Übersicht Interaktion Startmenü	24
Tabelle 2: Übersicht Interaktion Spielanleitung	24
Tabelle 3: Übersicht Interaktion Tastatur	25
Tabelle 4: Übersicht Interaktion Einstellungen	26
Tabelle 5: Belegung der Tastatur und Maus.....	31