

Game Design Dokument

Spiel: Agent 7 – Das Aufstreben von Dr. Silberblick



Version 1.0

Vorgelegt von
Maite-Aileen Brandt
Studiengang Medieninformatik
7008009

Leer, 11. Juli 2021

Betreut von
Hans-Georg Reimer

Inhalt

Abbildungsverzeichnis.....	3
1 Zusammenfassung.....	4
1.1 Zielplattform und Technologien	4
1.2 Genre.....	5
1.3 Zielgruppe	5
1.4 USP	5
2 Spieldetails	6
2.1 Kurzbeschreibung.....	6
2.2 Charaktere.....	6
2.2.1 Spielfigur.....	6
2.2.2 Partner.....	7
2.2.3 Gegner.....	7
2.3 Level	9
2.3.1 Level 1	9
2.4 Menü	16
2.4.1 Diagramm	18
2.4.2 Ingame-Menü	18
2.4.3 Ingame-GUI	20
2.5 Steuerung	23
2.6 Gameplay	23

Abbildungsverzeichnis

Abbildung 2.1: Charaktere Agent 7	6
Abbildung 2.2: Charaktere Sir K und SUS.....	7
Abbildung 2.3: Charaktere P.....	7
Abbildung 2.4: Charaktere Dr. Silberblick.....	7
Abbildung 2.5: Charaktere Dr. Silberblicks Roboter-Scherge	8
Abbildung 2.6: Skizze Levelaufbau	9
Abbildung 2.7: Skizze Gebäudeaufbau	10
Abbildung 2.8: Level isometrische Sicht	11
Abbildung 2.9: Level perspektivische Sicht.....	11
Abbildung 2.10: Gebäude isometrische Sicht.....	12
Abbildung 2.11: Gebäude erster Raum	12
Abbildung 2.12: Gebäude Gang.....	13
Abbildung 2.13: Gebäude großer Raum	13
Abbildung 2.14: Gebäude letzter Raum	14
Abbildung 2.15: Toon-Shader Shadergraf	14
Abbildung 2.16: Toon-Shader Ergebnis	15
Abbildung 2.17: Menü Hauptelemente.....	16
Abbildung 2.18: Menü Optionen	16
Abbildung 2.19: Menü Informationen.....	17
Abbildung 2.20: Menü Start	17
Abbildung 2.21: Menü Navigationsdiagramm	18
Abbildung 2.22: Menü Pause	19
Abbildung 2.23: Menü Pause Umsetzung	19
Abbildung 2.24: Menü Gewonnen Umsetzung	20
Abbildung 2.25: Ingame-GUI Elemente	20
Abbildung 2.26: Ingame-GUI Zeitlupe	21
Abbildung 2.27: Dialog Aufbau	21
Abbildung 2.28: Dialog Umsetzung	22
Abbildung 2.29: Inspizieren-Icon.....	22

1 Zusammenfassung

1.1 Zielplattform und Technologien

Das Spiel wird für den **PC** aufgrund der für die Steuerung notwendigen **Eingabegeräte**, welche sich durch das Genre und Gameplay ergeben – in diesem Fall Maus und Tastatur – entwickelt. Es wird als **Windows-Standalone-Version** exportiert, da auf dieser Plattform getestet werden kann. Das Ziel stellt eine **Executable** dar.

Die momentan mir bekannten **Spiele-Engines** für die **3D-Entwicklung** umfassen **Godot**, **Unreal**, **Lumberyard** und **Unity**. Für dieses Projekt dient als Technologie die Unity3D-Engine der **Version 2020.3.5f1 (LTS)** – dabei handelt es sich um die zu diesem Zeitpunkt aktuelle Long-Time-Support-Version und bietet so eine möglichst stabile Projektgrundlage.

Die Wahl fällt auf diese Engine vor allem aufgrund der integrierten **Physiksimulation**, welche für das Gameplay dieses Spiels entscheidend ist, siehe dazu den Abschnitt Gameplay. Ebenso aufgrund der Möglichkeit, **Skripte** für die Umsetzung von Funktionalitäten in **C#** zu schreiben, was über Unreal und Lumberyard nicht nativ möglich ist. Dies kommt mir aufgrund vorhandener Java-Kenntnisse und der Ähnlichkeit zu dieser Programmiersprache zugute, wie auch die große Menge an **Onlinetutorials**, **-hilfestellungen** und eine **aktive Community**. Ebenso richten sich die Beispiele der projektbegleitenden **Vorlesung** und **Unterlagen** an die Unity-Engine. Erzeugte **Skripte** können an **Gameobjekte der Szenen angehängt** werden, was eine interaktive und visuell unterstützte Entwicklung des Codes und für mich effizientere Entwicklungszeit und Debugging-Möglichkeiten bedeutet. Da Unity geläufige **Dateiformate** für Assets, wie beispielsweise das **FBX-Format** der 3D-Modelle und **PNG-Format** für Texturen, unterstützt, ist ein effektives Arbeiten mit externen Programmen für die Asset-Produktion möglich. Wie dem Gameplay-Abschnitt zu entnehmen, spielt die Zeitmanipulation in dem Spiel eine Rolle, wozu die **Echtzeit-Rendering-Funktionalität** von Unity bei der Implementierung dieses Elements dienlich ist. Zusätzlich können **Materials** und **Shader** für die 3D-Modelle mit der **Universal-Render-Pipeline** von Unity mit relativ geringem Aufwand erstellt werden und führen vor allem bei dem angestrebten Low-Poly-Stil zu visuell guten Ergebnissen, ohne dass für die Material-Erstellung externe Software – wie beispielsweise Substance Painter – eingesetzt werden muss. Unitys integrierte **Post-Processing**-Möglichkeiten unterstützen zudem gezielt die Entwicklung einer angestrebten visuellen Darstellung. Die **Shader** lassen sich über einen **visuellen Grafeditor** erstellen, der die Erstellung mit der visuellen Darstellung unterstützt. Ebenso spielt die **Dateiverwaltung** eine wichtige Rolle – in Unity ist diese meiner Meinung nach sehr übersichtlich, beispielsweise über selbst anzulegende Ordner, und wenig fehleranfällig. Auch die **Ladezeiten** für das **Kompilieren** und **Testen** des Spiels gestalten sich sehr kurz, verglichen mit den alternativen 3D-Engines, was somit ein sehr iteratives Arbeiten aus Entwicklung und Tests ermöglicht. Auch die Erstellung eines **Builds** für die Zielplattform ist über Unity ein sehr einfacher Prozess.

Die zusätzlichen Funktionen der **Unity-Pro-Version** umfassen einen änderbaren Splash Screen, Pro Editor UI Skin, Performance Reporting, Asset Bundle Streaming, Video Playback, Streaming und Multiplayer Networking mit RakNet von maximal 200 Clients. Die genannten Funktionen sind für dieses Projekt jedoch nicht vonnöten, weshalb die **Unity-Standard-Version** ausreicht.

Infolge der **Asset-Erstellung** wird **Photoshop** für die Texturen sowie **Maya** und **Blender** für die 3D-Modelle eingesetzt.

1.2 Genre

Es handelt sich bei dem Spiel um eine Mischung aus First-Person Parkour-Actionspiel, Shooter und Komödie, mit einem Schwerpunkt auf die Parkourelemente, also Bewegung des Charakters durch die Spielwelt.

1.3 Zielgruppe

Bei der Zielgruppe handelt es sich grundsätzlich um jugendliche und erwachsene Gelegenheitsspieler. Auch bietet es sich für Intensivspieler, wie beispielsweise Speedrunner, aufgrund der Möglichkeit, Level schnellstmöglich zu bewältigen und Rekorde aufzustellen, an.

1.4 USP

Spielprinzipien aus verschiedenen Parkourspielen werden zu einer intuitiven und flüssigen Charaktersteuerung vereint, um ein schnelles Gameplay zu ermöglichen. Zusammen mit einer komödiantischen Erzählweise, mit Anspielungen auf Agenten- und Detektivfilme, ist ein maximaler Spielspaß angestrebt.

2 Spieldetails

2.1 Kurzbeschreibung

Die Geschichte spielt in einer Welt, in der anthropomorphe Tiere, die wie Menschen leben, beheimatet sind.

Der spielbare Charakter, eine Katze mit Namen Agent 7, wird per Telefon kontaktiert und von seinem Vorgesetzten Sir K darüber informiert, dass Silberblick, der Hauptantagonist des Spiels, die gesamte Katzenminze der Stadt gestohlen hat. Dies ist sein erster Schritt, um für sich ein weltweites Monopol zu schaffen.

Silberblick hat zu diesem Zweck seine Schergen, Roboterhunde, entsendet, um die Schuld auf die Hunde zu projizieren, um so zusätzlich den Frieden zwischen den Hunden und Katzen zu stören.

Agent 7 erhält die geheime Mission, die Minze zurückzuerobern, und möglichst die Roboter zu zerstören.

2.2 Charaktere

Nachfolgend werden die Charaktere des Spiels benannt und beschrieben. Dabei werden sie aufgeteilt in Spielfigur, Partner sowie Gegner.

2.2.1 Spielfigur

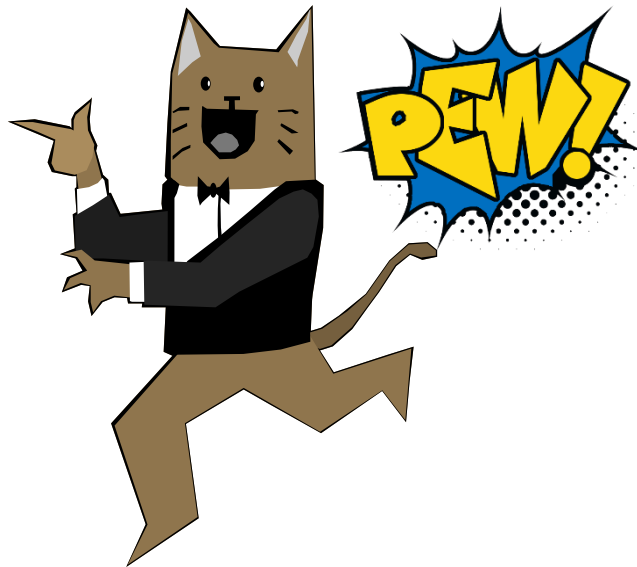


Abbildung 2.1: Charaktere Agent 7

Agent 7 ist ein angehender Geheimagent, welcher jüngst seine Ausbildung abgeschlossen hat. Er ist ein Kater von enthusiastischer, begeisterungsfähiger Natur und strebt nach dem maximalen Adrenalinrausch. Über die First-Person-Steuerung kann er sich in der 3D-Welt frei bewegen, springen, an Wänden entlanglaufen und über den Boden rutschen. Ein Greifhaken unterstützt seine Bewegungsfreiheit zusätzlich, um größere Distanzen zu überwinden. Parallel zu den Bewegungen kann er mithilfe seiner Fingerpistole feuern und damit Gegner besiegen. Er besitzt sieben Adrenalinkerne. Das Adrenalin, je Kern, kann per Knopfdruck verbraucht werden und damit die Zeit verlangsamen, um so das Navigieren und Zielen zu unterstützen. Die Basisanzahl der Adrenalinkerne erhöht sich durch Sammeln des Items Katzenminze. Die Kerne werden durch bestimmte Adrenalin-Aktionen aufgefüllt, wie beispielsweise das Treffen von Gegnern ohne dass Bodenkontakt besteht und Erreichen von bestimmten Geschwindigkeiten.

2.2.2 Partner

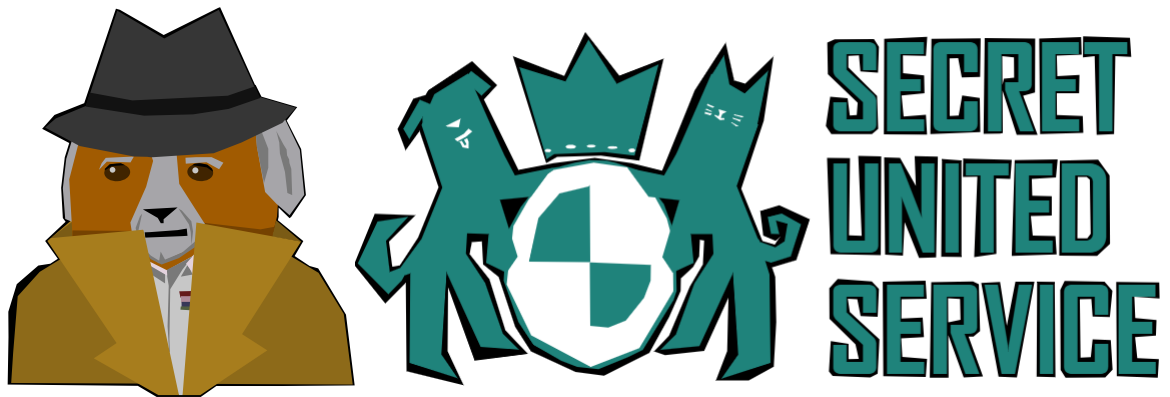


Abbildung 2.2: Charaktere Sir K und SUS

Sir K (Kat) ist der Vorgesetzte von Agent 7 und kontaktiert ihn für den Auftrag. Er ist der Leiter des Geheimdienstes SUS (Secret United Service), der im Geheimen agiert, um den Frieden und das Gleichgewicht des Landes US (United Species) zu wahren.



Abbildung 2.3: Charaktere P

P (Peter) ist ein Wolf und Erfinder des SUS. Er ist verantwortlich für die Entwicklung der Werkzeuge von Agent 7, wie zum Beispiel seinen Greifhaken.

2.2.3 Gegner



Abbildung 2.4: Charaktere Dr. Silberblick

Dr. Silberblick ist der Hauptantagonist des Spiels. Er ist eine weiße Perserkatze mit Narbe über dem rechten Auge, das durch ein silbernes, namensgebende ersetzt wurde. Außerdem hat er einen russischen Akzent – in der Spielwelt als schmusischer Akzent bekannt. Dadurch werden einige Buchstaben seines Textes im Dialogfenster durch russische ersetzt.

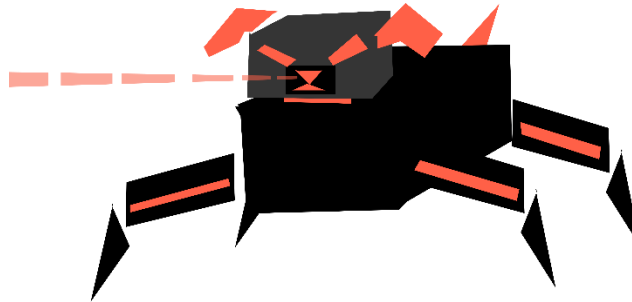


Abbildung 2.5: Charaktere Dr. Silberblicks Roboter-Schergen

Silberblicks Schergen sind Roboter mit Hundeattributen. Sie befinden sich im Level und stellen die bekämpfbaren Gegner dar. Sie zielen und feuern auf den Spieler und sind von diesem besiegbare. Über einen Kopftreffer werden sie direkt zerstört. Die Beine können einzeln abgeschossen werden, wodurch sie manövrierunfähig werden. Sind sie zerstört, explodieren sie.

2.3 Level

Für dieses Projekt wurde ein Spielbares Level entworfen und umgesetzt. Näheres dazu im folgenden Unterkapitel.

2.3.1 Level 1

Die nachfolgende Skizze repräsentiert im Groben den Aufbau des Levels. Für die erwähnten Spieleraktionen siehe Kapitel Gameplay.

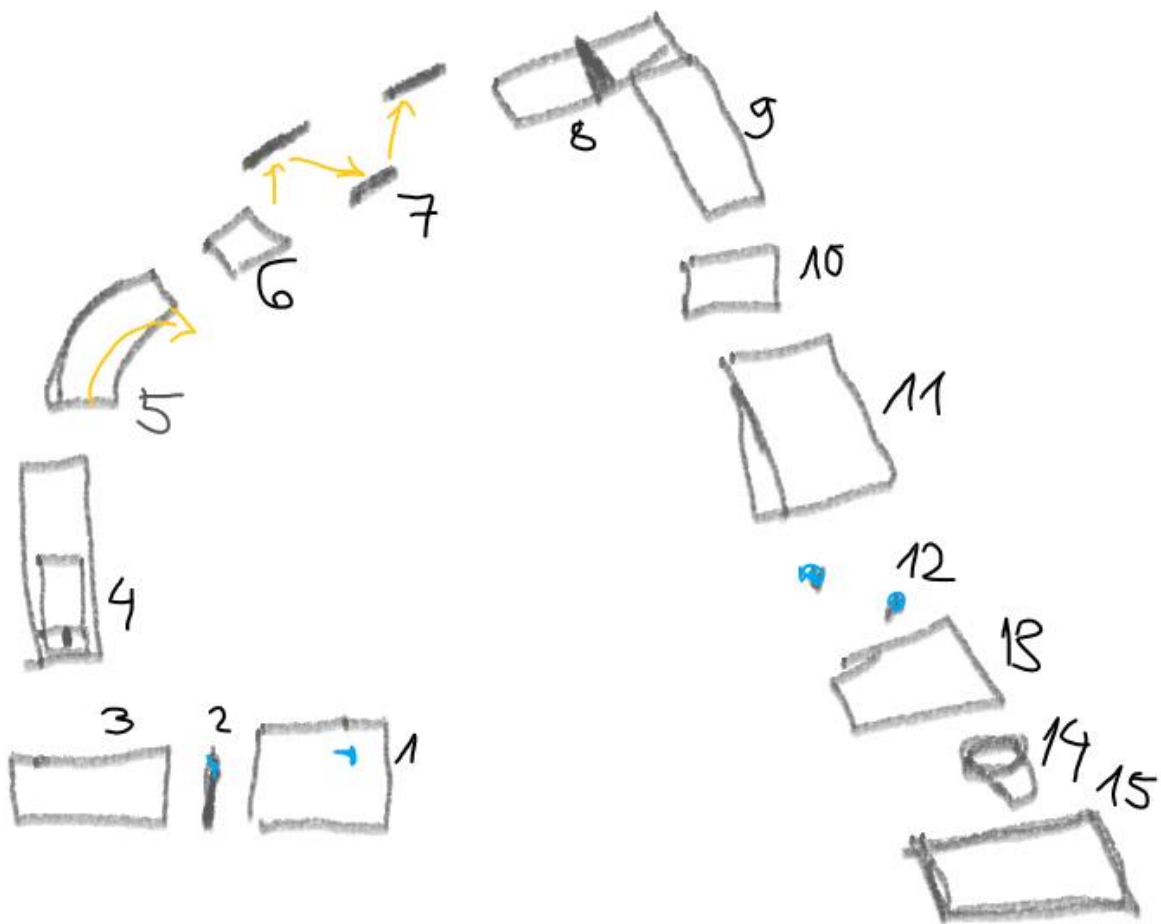


Abbildung 2.6: Skizze Levelaufbau

Bei 1 startet der Spieler – es handelt sich hierbei um die Wohnung des Charakters in einem Hochhaus, von dem aus die Stadt, in der sich das Level befindet, überblickbar ist. Hier erhält der Spieler den Greifhaken und die Mission, also das Ziel des Levels. Zum Fortfahren muss der Spieler zunächst eine Tür per Sliden zerstören, durch das Fenster aus der Wohnung kommen und den Greifhaken an einem Baukran (2) einsetzen, um auf das nächste Hochhausdach (3) zu gelangen. Hier lernt der Spieler einige weitere Bewegungsmöglichkeiten kennen – er muss durch eine Lücke hindurchsliden und sich mithilfe eines Walljumps auf eine höhere Ebene des Dachs begeben. Mit einem Wallrun und Walljump gelangt der Spieler auf die nächste Plattform durch ein Fenster, welches über den Einsatz der Pistole zerstörbar ist, zum nächsten Gebäude (4). Die Hindernisse dieses Gebäude kann der Spieler mit bereits bekannten Bewegungen überqueren. Bei 5 bewegt der Spieler sich per Wallrun vertikal nach unten und erreicht damit die Plattform 6. Darauf muss sich der Spieler zwischen drei Gebäuden mit dem Wallrun und -jump bewegen (7) und erreicht die Plattform (8), in der er unter ein Hindernis durchsliden muss. 9 ist eine Plattform mit Gefälle. Slidet der Spieler herunter, wird Geschwindigkeit

aufgebaut. Landet der Spieler mit ausreichender Geschwindigkeit auf das federnde Feld von Dach 10, erreicht er mit dem Schwung Gebäudedach 11. Auf dem Dach befindet sich ein Gegner und der Spieler lernt die Grundlagen, wie dieser zu besiegen ist. Der folgende Abgrund ist mit dem Greifhaken bei 12 überwindbar. Das Dach von Gebäude 13 ist über unterschiedliche Weisen erklimmbar. Der Spieler kann das untere Dach direkt über den Schwung von 12 erreichen, dann entweder über eine federnde Plattform nachhelfen oder über eine Treppe hochklettern. Das obere Dach ist zum einen über eine Walljump- und Sprungpassage erreichbar und zum anderen kann der Spieler sich mit Sprengfässern auf das Dach hinaufmanövrieren. Auf dem oberen Dach ist ein Gegner platziert, der Spieler kann ihn entweder mit der Pistole oder mit einem Sprengfass besiegen. Von dem Gebäude aus gelangt der Spieler über eine Wallrun- und -jumppassage durch eine Rutschbahn (14) und in das Gebäude 15. Der Aufbau von Gebäude 15 ist in folgender Abbildung dargestellt.

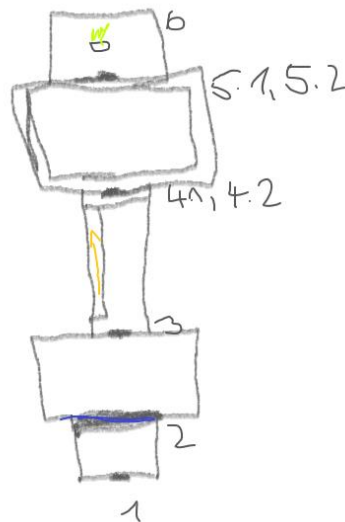


Abbildung 2.7: Skizze Gebäudeaufbau

Durch den Eingang (1) gelangt der Spieler hinein, dabei kann er entweder die Tür zerstören oder sie mit einem Wallrun umgehen. Hier lernt der Spieler, wie er Objekte aufheben und das folgende Fenster auch mit aufgehobenen Objekten zerstören kann. Durch das Fenster gelangt der Spieler in den nächsten Abschnitt, in welchem zwei Gegner bereitstehen. Über eine Tür (3) gelangt der Spieler in einen Gang. Hier kann entweder der untere (4.1) oder obere Pfad (4.2) genommen werden. Beide führen in einen größeren Raum, in dem sich drei Gegner befinden. Je nachdem welchen Pfad der Spieler genommen hat, tritt er ihnen direkt gegenüber oder kann den oberen Pfad zu seinem Vorteil nutzen, da der Spieler den Gegnern nicht frontal gegenübersteht. Hinter der letzten Tür befindet sich das Zielitem des Levels. Sobald das Item eingesammelt wird, ist das Level abgeschlossen.

Gameplaytechnisch wird der Spieler unterstützt, indem das Level entsprechend der geforderten Aktionen Assoziationen über Farben herstellt. Wenn die Aktionen Wallrun oder -jump eingesetzt werden sollen, wird dies über ein Orange verdeutlicht. Der Greifhaken wird an blauen Objekten eingesetzt und bei gelben Objekten soll das Sliden angewendet werden. Rosarote Objekte haben federnde Eigenschaften und unterstützen das Springen des Spielers. Explosionsfässer und Gegner sind Rot kodiert.

Der folgende Screenshot zeigt aus isometrischer Perspektive den umgesetzten Levelaufbau in der Engine.



Abbildung 2.8: Level isometrische Sicht

Aus perspektivischer Sicht sieht der Abschnitt wie folgt aus.

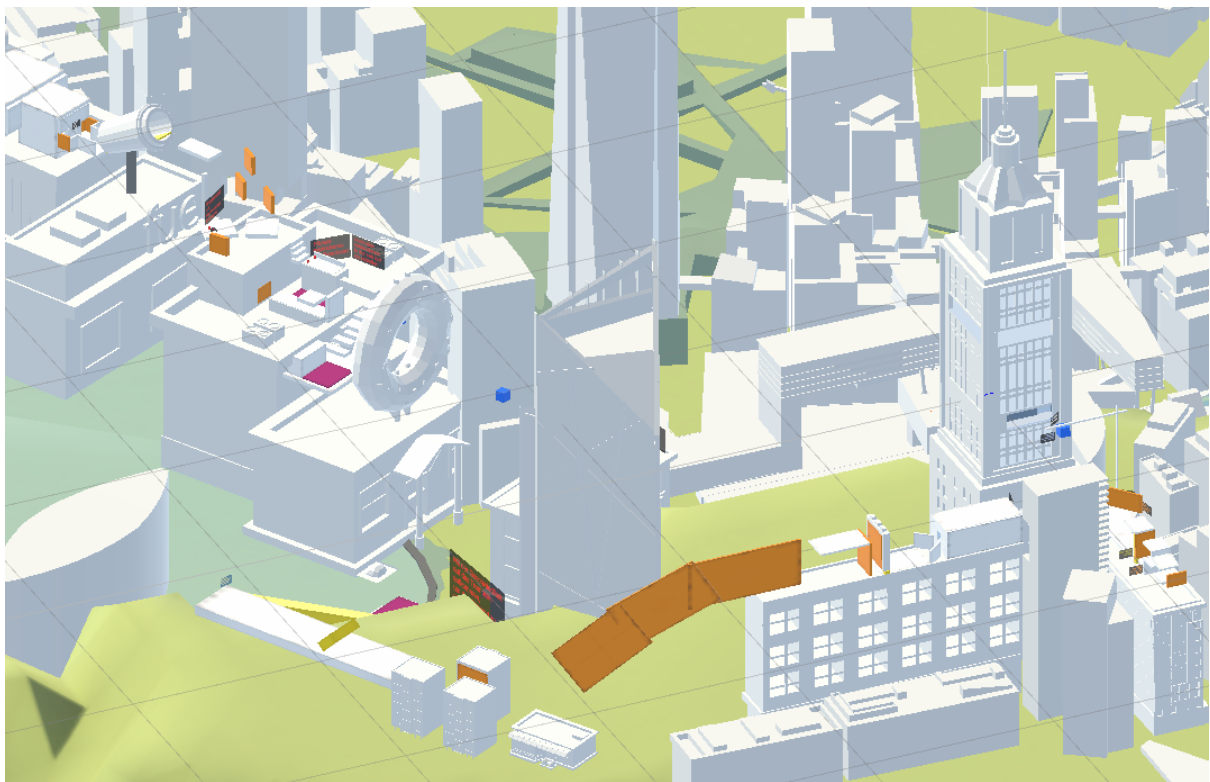


Abbildung 2.9: Level perspektivische Sicht

Das Gebäude von Punkt 15 ist von oben im nachfolgenden Screenshot dargestellt.

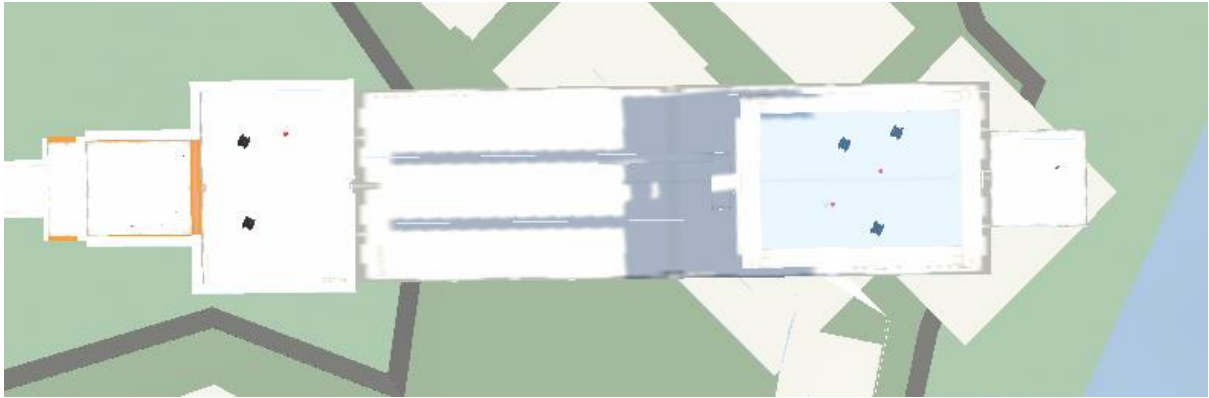


Abbildung 2.10: Gebäude isometrische Sicht

Dies ist der erste Raum des Gebäudes.

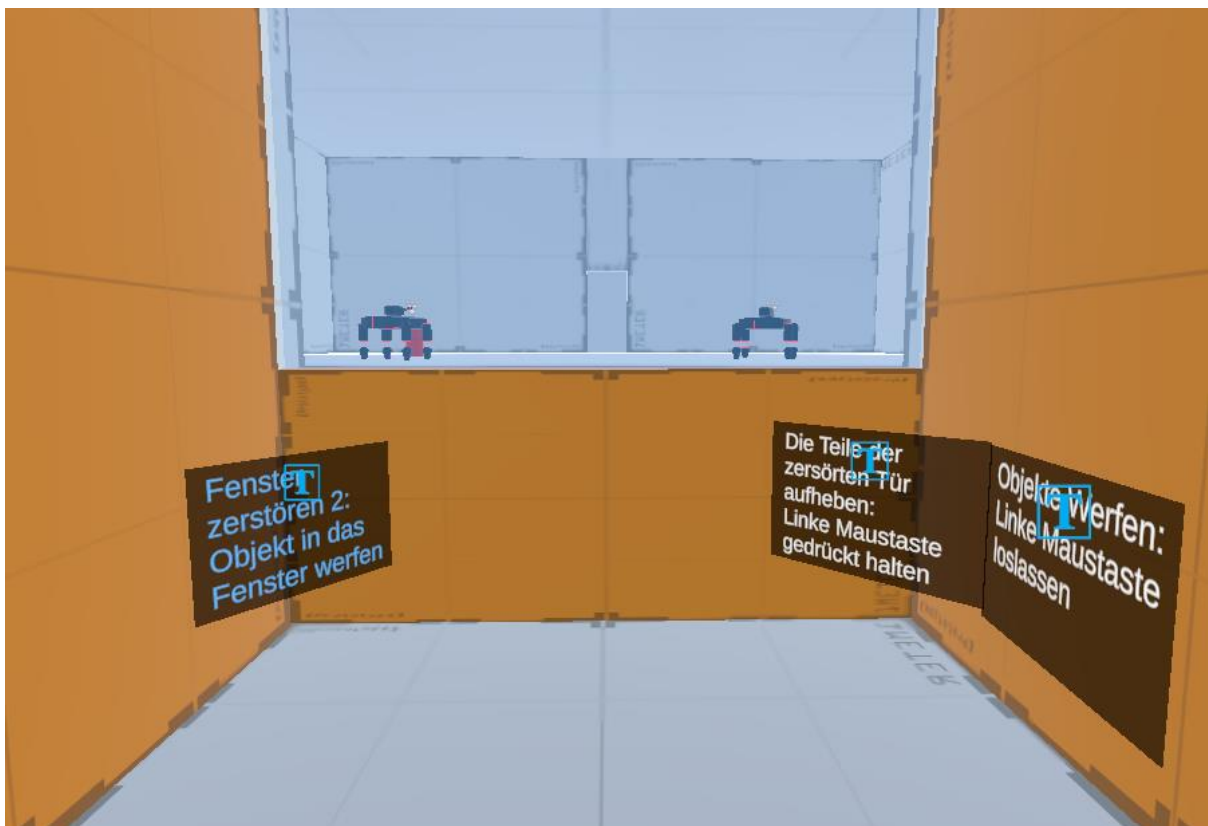


Abbildung 2.11: Gebäude erster Raum

Darauf folgt der, in der Skizze beschriebene, Gang.



Abbildung 2.12: Gebäude Gang

Auf den Gang folgt der Raum, welcher sowohl von unten als auch von oben betretbar ist – im folgenden Screenshot ist er von oben dargestellt.

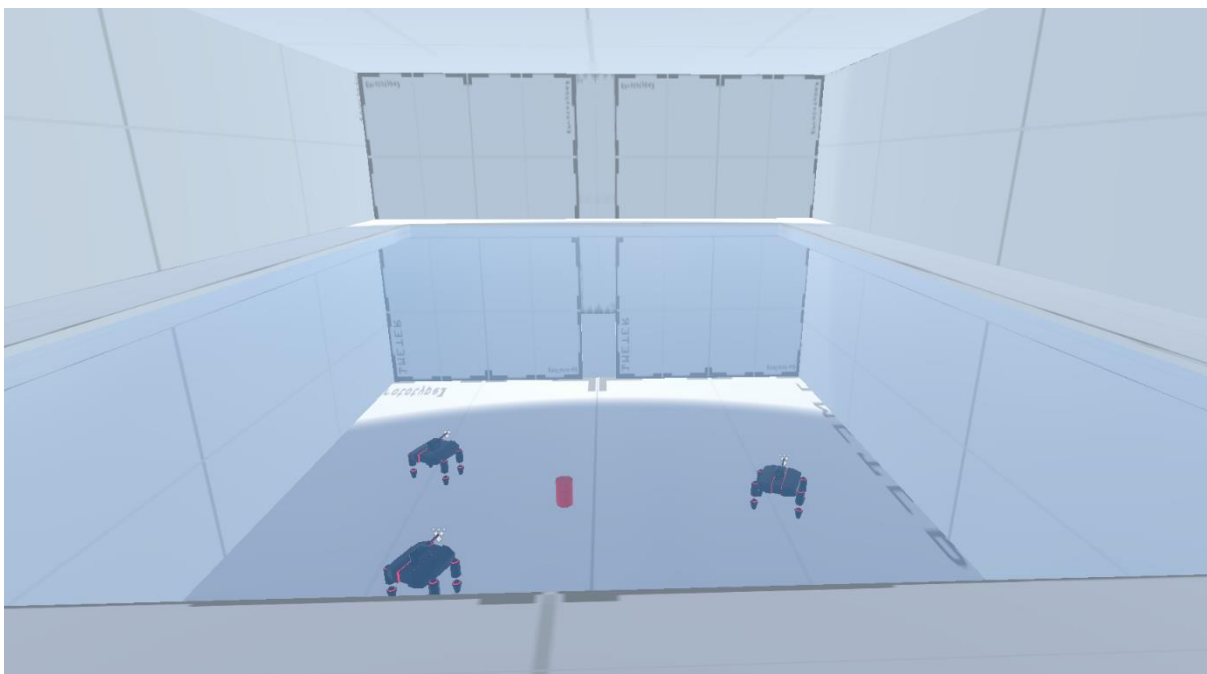


Abbildung 2.13: Gebäude großer Raum

Der letzte Raum mit dem Ziel-Item wurde wie folgt umgesetzt.



Abbildung 2.14: Gebäude letzter Raum

Um einen visuellen Comic-Effekt bei den Objekten des Levels zu erzeugen, wurde ein Toon-Shader eingesetzt. Der zugehörige Shadergraph sieht wie folgt aus.

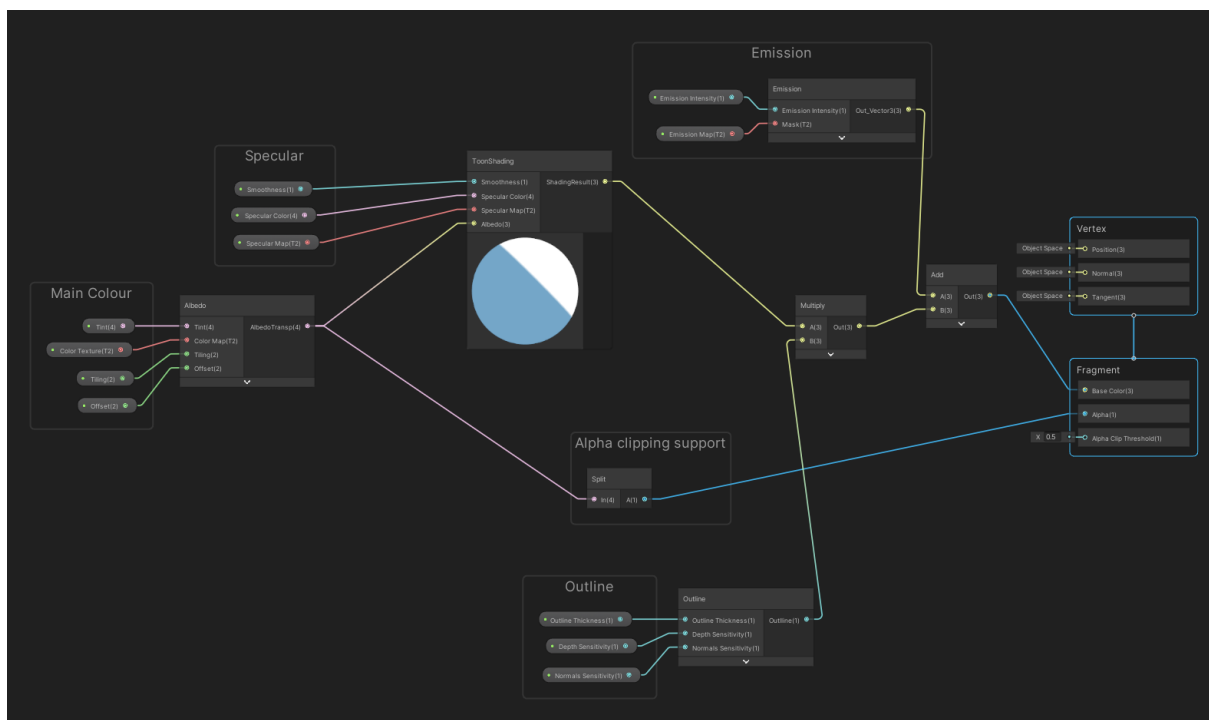


Abbildung 2.15: Toon-Shader Shadergraph

Näheres zur Erstellung des Toon-Shaders findet sich auf der folgenden GitHub-Seite von Unity: <https://github.com/UnityTechnologies/open-project-1/tree/devlogs/1-toon-shading>.

Über den Shader wurden Materialien erzeugt und auf Objekte in der Szene gelegt. Ein Beispiel des Ergebnisses ist in folgender Abbildung zu sehen.



Abbildung 2.16: Toon-Shader Ergebnis

Es wurden alle Objekte damit versehen, auf denen die Spielfigur sich bewegen beziehungsweise mit denen sie interagieren kann. Dadurch soll der Spielfluss verbessert werden, da der Spieler einfacher erkennen kann, wohin er gehen muss, beziehungsweise womit er interagieren kann.

2.4 Menü

Das Hauptmenü bietet dem Benutzer die Möglichkeit, das Spiel zu starten, auf Optionen zuzugreifen und Informationen zur Steuerung sowie des Entwicklers einzusehen. Es ist wie folgt aufgebaut.

Start
Optionen
Informationen

Beenden

Abbildung 2.17: Menü Hauptelemente

Über einen Klick auf den Optionen-Button gelangt der Spieler in ein wie folgt skizziertes Optionsmenü.

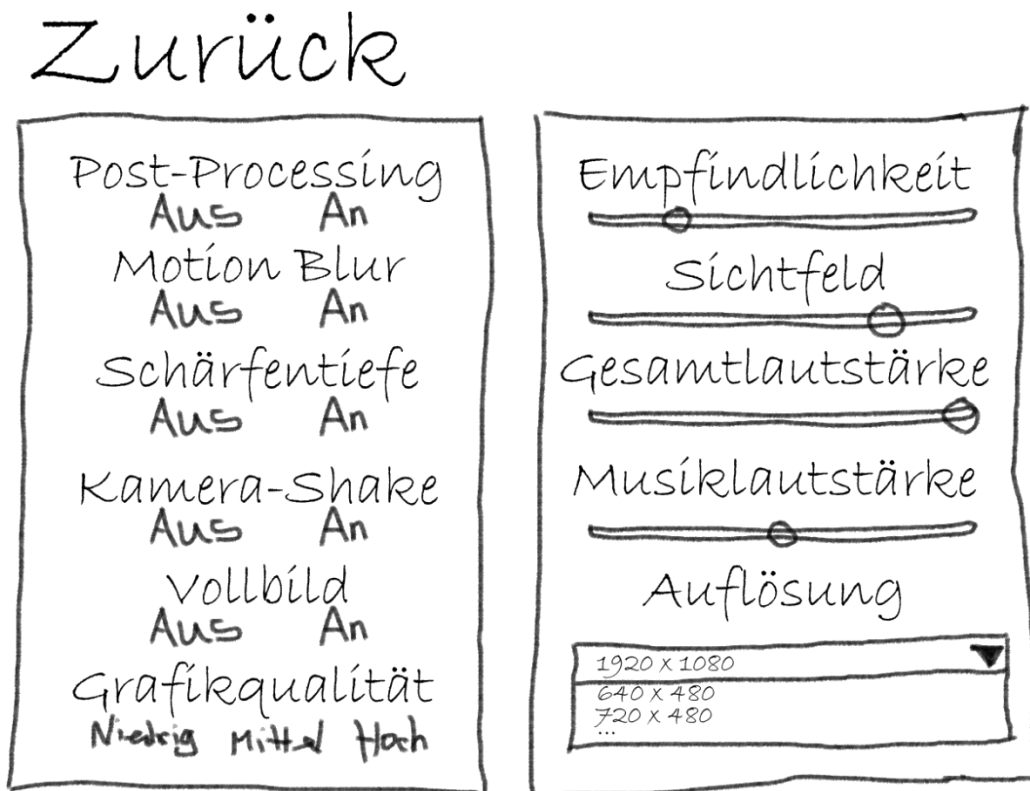


Abbildung 2.18: Menü Optionen

Hier werden alle Einstellungen zur Verfügung gestellt, bestehend aus den hier skizzierten Möglichkeiten. Auf der linken Hälfte werden alle Optionen aufgelistet, welche über Buttons eingestellt

werden. Rechts sind die Slider und ein Dropdown-Menü mit allen Verfügbaren Auflösungen, die zuvor dynamisch ermittelt werden, aufgeführt.

Über den Button Informationen gelangt der Nutzer zu einer Darstellung der Entwicklerinformationen.

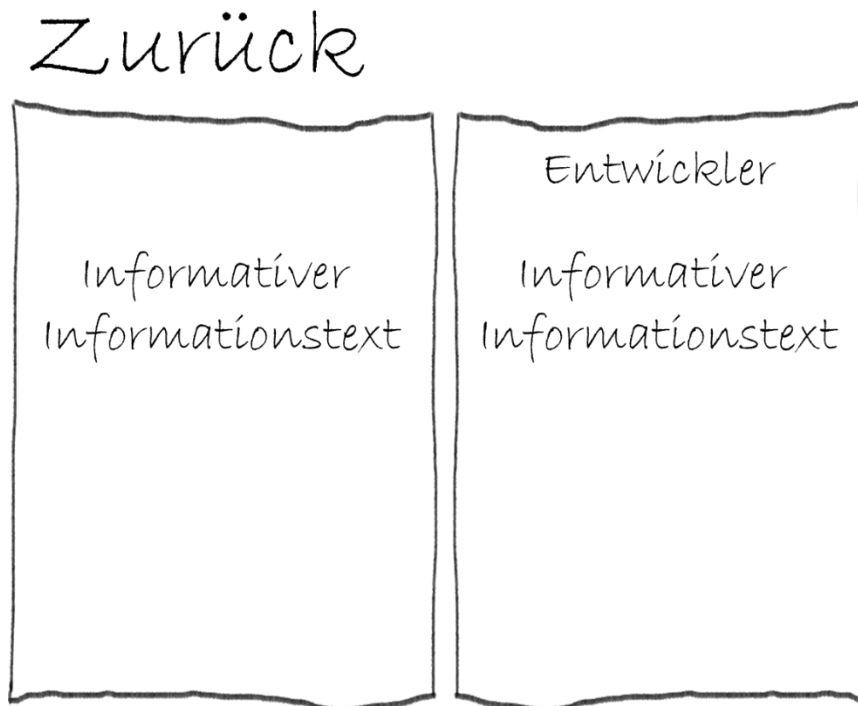


Abbildung 2.19: Menü Informationen

Der Start-Button führt zu dem Levelauswahlmenü.

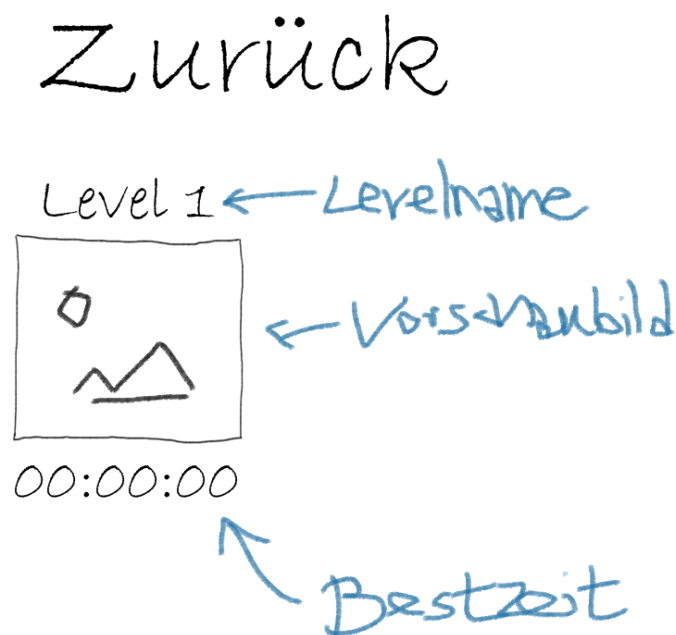


Abbildung 2.20: Menü Start

Hier werden alle zur Verfügung stehenden Level dargestellt – es handelt sich im Rahmen dieses Projektes um ein einziges Level. Es werden jeweils Levelname, ein Vorschaubild und die zugehörige Bestzeit dargestellt. Mit einem Klick auf das Bild, wird das Level gestartet – nähere Informationen zum Level im Kapitel Level.

Alle Untermenüs führen über einen Zurück-Button wieder in das Hauptmenü. Die Menüs, beziehungsweise die Buttons, befinden sich dabei direkt in der 3D-Szene des ersten Levels, also der Wohnung von Agent 7. Der Spieler befindet sich in der Egoperspektive und liegend auf dem Bett in der Wohnung. Sobald einer der Menüpunkte geklickt wird, dreht sich die Kamera in die Richtung des von dem Spieler angestrebten Untermenüs.

Der folgende Screenshot zeigt als Beispiel die Umsetzung der Levelauswahl, wobei sich die Kamera zuvor – nach Drücken des Start-Buttons – nach rechts zum Levelauswahlmenü gedreht hat.

2.4.1 Diagramm

Die nachfolgende Skizze verdeutlicht die Navigation unter den zuvor genannten Untermenüs.

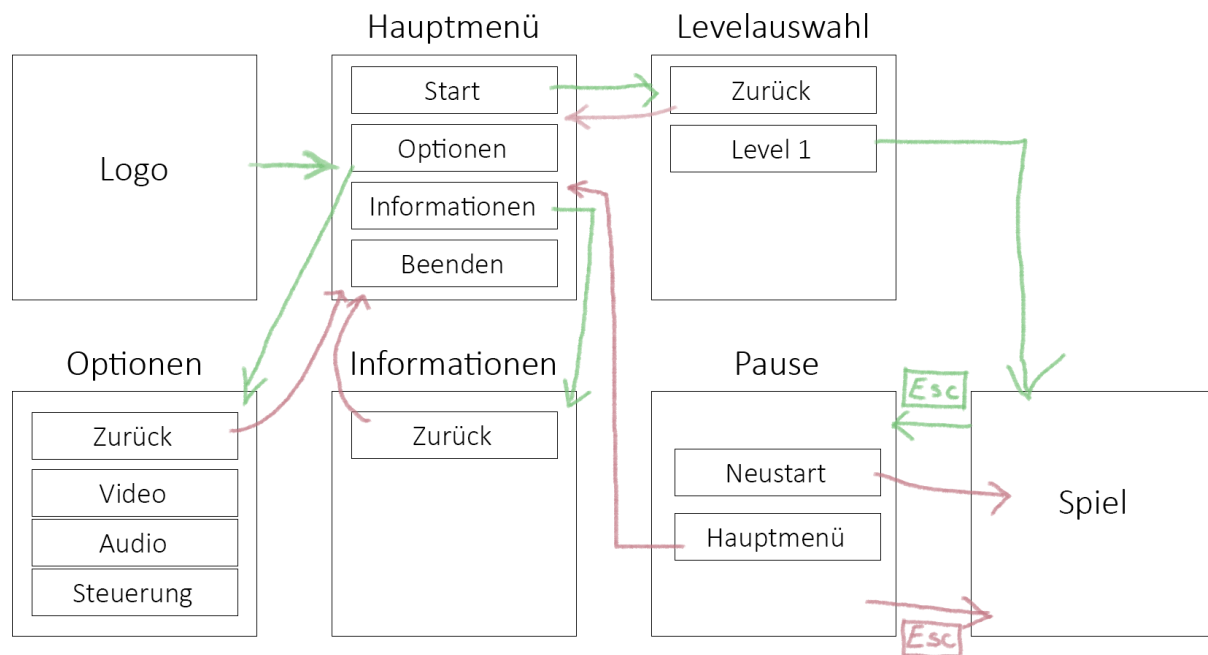


Abbildung 2.21: Menü Navigationsdiagramm

Es ist ebenso die Navigation über das Ingame-Menü aufgeführt, welches nachfolgend näher erläutert wird.

2.4.2 Ingame-Menü

Das Ingame-Menü wird über die Esc-Taste aufgerufen und bietet die folglich skizzierten Möglichkeiten das Level neu zu starten, in das Hauptmenü zurückzukehren oder – über das erneute Drücken der Esc-Taste ins Spiel zurückzugelangen.

00:13:37

Neustart

Hauptmenü

Abbildung 2.22: Menü Pause

Während dieses Menü aufgerufen ist, wird sowohl das Spiel als auch der Counter pausiert. Die aktuelle Zeit ist pausiert dargestellt.

Das Spiel ist weiterhin, hinter den Buttons des Pause-Menüs, erkennbar – eine Vignette verdeutlicht, dass das Menü aufgerufen, beziehungsweise das Spiel pausiert, ist. Die Umsetzung ist im folgenden Screenshot zu sehen.



Abbildung 2.23: Menü Pause Umsetzung

Stirbt der Spieler durch einen Gegner oder hat er das Level beendet, sprich gewonnen, erscheint das gleiche Menü, jedoch ist das Fortfahren des Spiels über die Esc-Taste nicht möglich.

Die Gewonnen-UI stellt zusätzlich die ausgewerteten Spielergebnisse dar. Sie ist der folgenden Abbildung zu entnehmen.

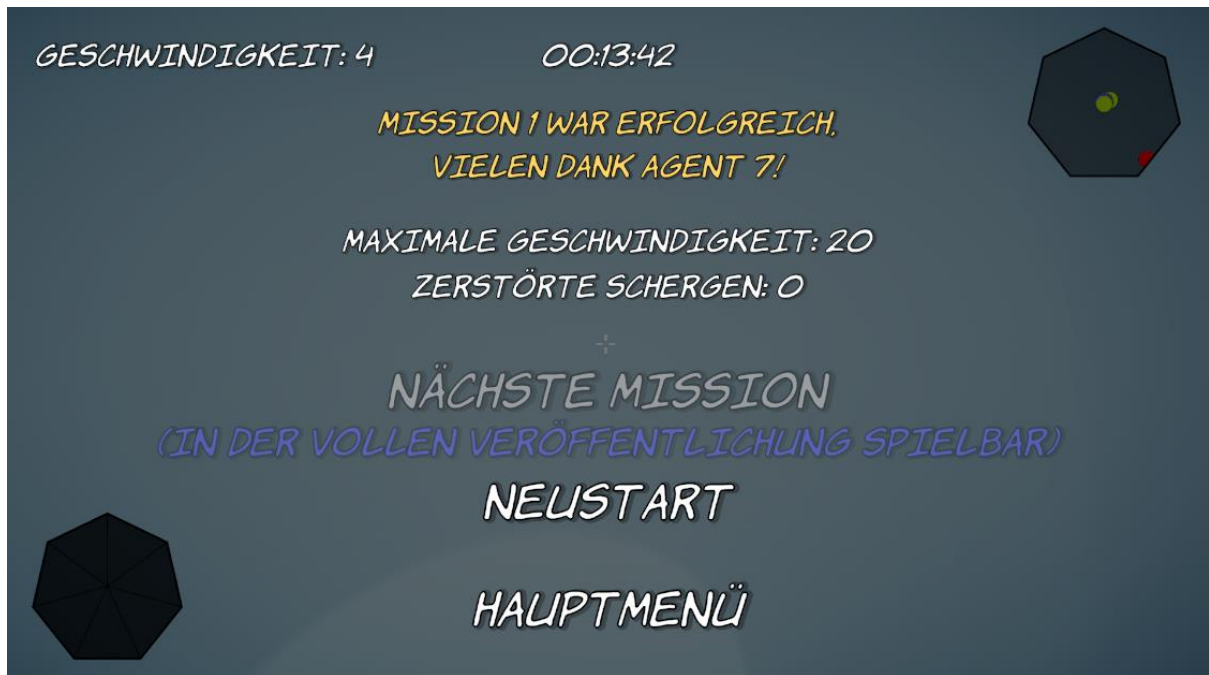


Abbildung 2.24: Menü Gewonnen Umsetzung

2.4.3 Ingame-GUI

Das Ingame-GUI besteht aus der Darstellung des Adrenalinkerns – näheres dazu unter dem Kapitel Charaktere beziehungsweise Gameplay – einem Fadenkreuz, welches das Zielen des Spielers unterstützt und einer Minimap, welche vorrangig dazu dient, die Positionen der Gegner und des Level-Zielitems zu verdeutlichen. Ebenso werden der Timer und eine Geschwindigkeits-Anzeige dargestellt. Nachfolgend ist das Spiel-GUI skizziert.

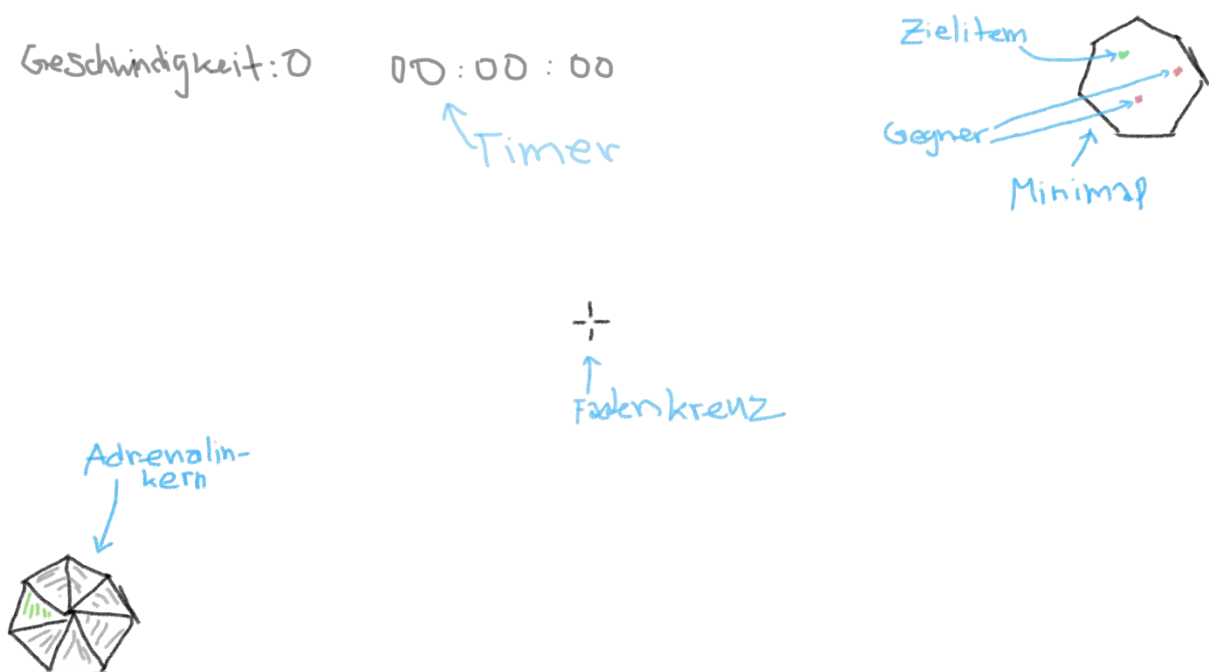


Abbildung 2.25: Ingame-GUI Elemente

Die Anzahl der zur Verfügung stehenden Adrenalinkerne wird über die Farben visualisiert. Die freigeschalteten Kerne sind grün, die übrigen grau dargestellt. Wird die Zeitlupenfähigkeit eingesetzt,

leeren sich die, zur Verfügung stehenden, Kerne und füllen sich wieder, wie unter Gameplay beschrieben. Die Anzeige zur Geschwindigkeit dient der Motivation hohe Geschwindigkeiten zu erreichen, vor allem da die höchste beim Levelende ausgewertet wird. Ebenso füllt sich Adrenalin ab bestimmten Werten wieder auf.

Durch eine Vignette wird verdeutlicht, wenn die Zeitlupe aktiv ist, wie im nachfolgenden Screenshot zu sehen.



Abbildung 2.26: Ingame-GUI Zeitlupe

Wird ein Dialog im Spiel gestartet, erscheint das Bild des sprechenden Charakters mit seinem Namen und dem Dialogtext in einer Sprechblase.

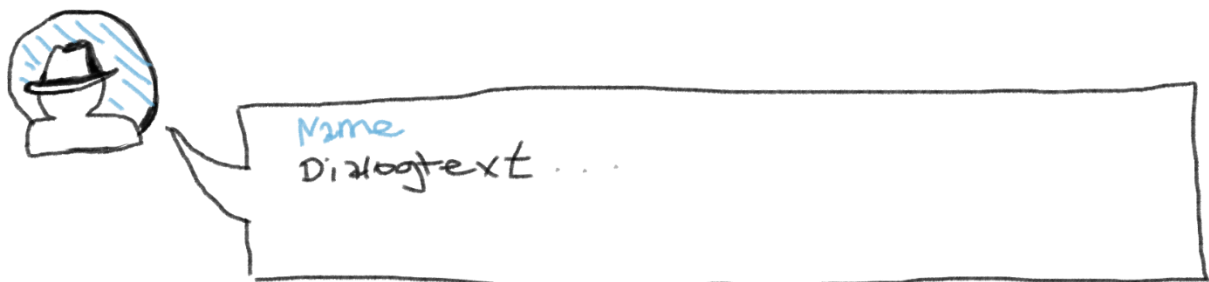


Abbildung 2.27: Dialog Aufbau

Der Hintergrund und Name des Sprechers sind dabei an diesen jeweils angepasst. Die Umsetzung ist im folgenden Screenshot zu sehen.



Abbildung 2.28: Dialog Umsetzung

Während eines Dialoges wird das Spiel pausiert, damit die Zeit nicht weiter hochzählt. Ein Dialog wird mit der Taste E jeweils fortgesetzt und kann mit der Esc-Taste abgebrochen werden.

Im Level sind einige alternative Dialoge verteilt. Diese können mit der Taste I aufgerufen werden, sobald das folgende Icon in der Spielwelt zu sehen ist.



Abbildung 2.29: Inspizieren-Icon

2.5 Steuerung

Die Bewegungsmöglichkeiten der Spielfigur sind unter dem Kapitel Charaktere beschrieben. Unter Zielplattform und Technologien wurde spezifiziert, dass die Steuerung über Maus und Tastatur erfolgt. Mit den Tasten W und A bewegt sich der Spieler nach vorne und hinten sowie mit S und D zur Seite. Die Maus dient der vertikalen und horizontalen Rotation der Kamera, wobei sie vertikal in einem 90° und -90° Winkel gelockt ist, um zu verhindern, dass der Spieler (mehrfach) um die eigene vertikale Achse rotiert. Sobald der Charakter Bodenkontakt hat, ermöglicht die Leertaste das Springen. Mit der Strg-Taste duckt er sich – wird sie während des Laufens eingesetzt, slidet er über den Boden. Mit der Taste E sind Objekte – in diesem Fall der Greifhaken – einsammelbar und mit F wird die Pistole bereitgehalten oder weggesteckt. Mit der Rechten Maustaste wird die sekundäre Hand, in der sich der Greifhaken befinden, eingesetzt und mit der linken Maustaste die primäre. Ist die Pistole bereit, wird diese demnach abgeschossen. Wenn sie weggesteckt ist, können aufhebbare Objekte mit Gedrückthalten der linken Maustaste aufgehoben und durch loslassen geworfen werden. Die Shift-Taste dient der Fähigkeit der Zeitverlangsamung des Charakters.

Das normale Spiel enthält, wie bereits im Kapitel Genre und Charaktere angesprochen, eine First-Person-Kamera aus der Sicht des steuerbaren Charakters. Diese Sicht setzt sich aus zwei Kameras zusammen – eine, die alle Elemente der Spielwelt rendert, die andere, die als Overlay-Kamera agiert und die Arme des Charakters überlagert, um sie unabhängig von der – beziehungsweise über die – Spielwelt zu rendern. Diese Vorgehensweise verhindert zum Beispiel, dass die Arme des Charakters durch Objekte der Spielwelt clippen, wenn der Spieler direkt vor ihnen steht.

Das Bild einer weiteren Kamera wird als eine Textur auf eine Fläche gelegt, um eine Minimap zu erstellen, wie unter dem vorherigen Kapitel skizziert. Sie dient der Orientierung im Level und hebt Gegner sowie das Sammel-Item – siehe dazu das Kapitel Level – hervor. Der Spieler hat dabei keinen direkten Einfluss auf die Alternativ-Kamera, indirekt aber durch seine Bewegung.

2.6 Gameplay

Das Gameplay wird überwiegend über die Unity-Physikengine mit den Rigidbody-Komponenten realisiert. Es zielt auf Bewegungsschnelligkeit und -genauigkeit ab. Die Bewegungsfreiheit wird mithilfe des Greifhakens unterstützt. Ebenso kann der Spieler über den Boden rutschen und an Wänden entlanglaufen.

Das Schießen erfolgt über eine Fingerpistole und unterstützt damit den Humor des Spiels. Die Zeit lässt sich per Knopfdruck verlangsamen und ist über Adrenalinkeine begrenzt. Diese lassen sich über bestimmte Aktionen wieder auffüllen.

Sowohl der Charakter als auch die Gegner – bei einem Kopfschuss – besitzen nur ein Leben. Das bedeutet, dass ein Gegner nach einem Treffer besiegt ist. Der Spieler startet nach einem Gegnertreffer, ebenso wenn er in einen Abgrund fällt, vom letzten Kontrollpunkt neu. Damit, und dass der Charakter keinen Fall- und Explosionsschaden erleidet, wird der Spielfluss und die Schnelligkeit des Spiels unterstützt.

Während des Levels zählt ein Timer hoch und ist in dem UI dargestellt.

Ein Level ist abgeschlossen, sobald der Spieler das Item Katzenminze am Levelende einsammelt. Dieses Item erhöht zusätzlich die Basisanzahl der Adrenalinpunkte. Darauf werden dem Spieler die Levelstatistiken, sprich die Zeit der Levelbewältigung, die Anzahl der besiegten Gegner und die maximal erreichte Geschwindigkeit, angezeigt.