

MIB14 – Softwareentwicklungsprojekt

WISE 2021/22

Prof. Dr. Carsten Lucke

Systemspezifikation

„Metroidvania“

von

Nada Savic (5316524)

Noah Fahrholz (5336038)

Azzedin Berrou (5321564)

Robin Prause (5323834)

Inhalt

Einleitung	3
<i>Zielsetzung</i>	<i>3</i>
<i>Spielkonzept</i>	<i>3</i>
<i>Referenzen</i>	<i>3</i>
<i>Teamorganisation</i>	<i>4</i>
Systemüberblick	5
<i>Verwendete Technologien</i>	<i>5</i>
<i>Kontextabgrenzung</i>	<i>5</i>
<i>Lösungsstrategie</i>	<i>5</i>
<i>Steuerung</i>	<i>6</i>
Grafiken	7
<i>Menüs</i>	<i>7</i>
<i>Leveldesign</i>	<i>7</i>
.....	<i>9</i>
.....	<i>11</i>
<i>Akteure</i>	<i>14</i>
Endboss	<i>14</i>
<i>Angriffswolke</i>	<i>15</i>
<i>Spieler</i>	<i>15</i>
Quellen:	15

Einleitung

Zielsetzung

Bei „Metroidvania“ handelt es sich um eine Kreuzung aus den beiden klassischen Spieleserien „Castlevania“ und „Metroid“. Die Spiele lassen sich an der 2-dimensionalen „Draufsicht“ erkennen und sind im Kern Plattformer.

Der Spieler bewegt sich klassischerweise durch eine große Welt, die nicht klar vorgibt, wohin dieser zunächst sich bewegen muss. Ziel ist es durch das Sammeln von Gegenständen den Endboss zu finden und zu besiegen.

Spielkonzept

Startet der Spieler das Spiel, so ist das grundsätzliche Ziel, den Endgegner zu finden und ihn zu besiegen. Um das zu erreichen, muss er zunächst Schlüsselteile finden, die in dem Level verstreut sind um damit die Tür zum Endgegner zu öffnen. Dabei muss der Spieler Hindernisse wie zb. Stacheln, Kanonenfeuer und einen stampfenden Schuh ausweichen. Im Vordergrund stehen somit das Bewegungssystem und das Zurechtkommen mit der Steuerung. Anschließend, wenn der Spieler die Tür zum Endgegner geöffnet hat, rückt auch das Kampfsystem in den Vordergrund.

Der Spieler besitzt eine bestimmte Anzahl von Leben. Der Gegner kann den Spieler Schaden zufügen, wodurch die Lebensanzeige sich verringert. Das gleiche gilt für Angriffe des Spielers and dem Gegner. Das Spiel ist beendet wenn der Gegner alle seine Lebenspunkte verloren hat.

Referenzen

Grundlage unseres Spiels „Lost Key“ ist, wie in der Zielsetzung bereits erklärt ein sogenannter „Metroidvania“ Plattformer. Die klassischen Spiele und Designelemente des 2D Spiels sind dabei an das Spiel „Ori and the Blind Forest“ angelehnt.



Teamorganisation

Unser 4-köpfiges Team setzt sich aus folgenden Rollen zusammen:

- Projektleiter: Robin Prause
- Chefdesigner: Azzedin Berrou
- Projektmanager: Noah Fahrholz
- Qualitätsbeauftragte: Nada Savic

Aufgrund der Größe des Teams beteiligen sich alle Rollen ebenfalls an der Entwicklung des Spiels. Mit „Trello“ definieren wir klare To-Dos, die bearbeitet und kommentiert werden können.

Systemüberblick

Verwendete Technologien

Das Spiel wird in Unity entwickelt. Dabei handelt es sich um eine Entwicklungsplattform, die primär zum Entwickeln von Spielen, aber auch für das Erstellen interaktiver 3D Grafiken verwendet wird. VisualStudioCode dient dabei als CodeEditor, um die Skripte, die in C# geschrieben werden, zu bearbeiten. Die Grafiken die bei unserem Spiel zum Einsatz kommen, werden in ClipStudio Paint Pro designed. Für die Charaktere verwenden wir Maya zum Modellieren, Mixamo um diese zu animieren und anschließend Blender zum rendern. Als Schnittstelle zwischen den Anwendungen dient Trello zum organisieren. Letztlich verwenden wir Google Drive und Discord um Dateien auszutauschen.

Letztendlich wird ein sogenanntes Build des Spiels erzeugt und die resultierende .exe oder .dmg Datei kann heruntergeladen, installiert und gespielt werden.

Kontextabgrenzung

Wie bereits beschrieben, wird „Lost Key“ in Unity entwickelt. Die Systemabgrenzung beschränkt sich somit auf die Entwicklungsumgebung. Auch die eingesetzten Libraries stammen aus Unity selbst.

Mit der ausführbaren .exe oder .dmg Datei stellt der Endnutzer den Systemkontext durch die Installation des Spiels.

Lösungsstrategie

Um die Kreativität im Team zu fördern und somit die Qualität des Spiels zu verbessern, haben wir uns für ein agiles Vorgehensmodell entschieden, wobei zu Beginn ein Prototyp entwickelt wird. Aufgrund des einfachen Systemkontextes benötigen wir keine aufwendigen Schnittstellen, die das Prototyping durch starke Qualitätsanforderungen verhindern würden. Der Projektleiter definiert klar benötigte Informationen, die die Randbedingungen stellen, während die Entwickler mit der Grafikdesignerin erste Entwürfe machen. Somit können benötigte Dokumente gleichzeitig zum Entwicklungsfortschritt beschrieben werden. Ziel ist,

den Prototyp in der ersten Entwicklungswoche so weit zu bringen, dass bereits ein im Grundsatz funktionierendes Spiel entsteht, anhand dessen entschieden werden kann, was noch ergänzt werden muss oder was so in der Realität nicht umgesetzt werden kann.

Steuerung

A | laufen nach links

D | Laufen nach rechts

LEERTASTE | Springen

(2x) LEERTASTE | Doppelsprung (wenn in der Luft)

Linksklick | Schwertschlag bei aktueller Mausposition

Rechtsklick | an Wand festhalten

➔ Wenn der Spieler sich an der Wand festhält, kann von dort erneut gesprungen werden mit Leertaste

Shift | Dash (Geschwindigkeitsboost)

Grafiken

Menüs

Leveldesign

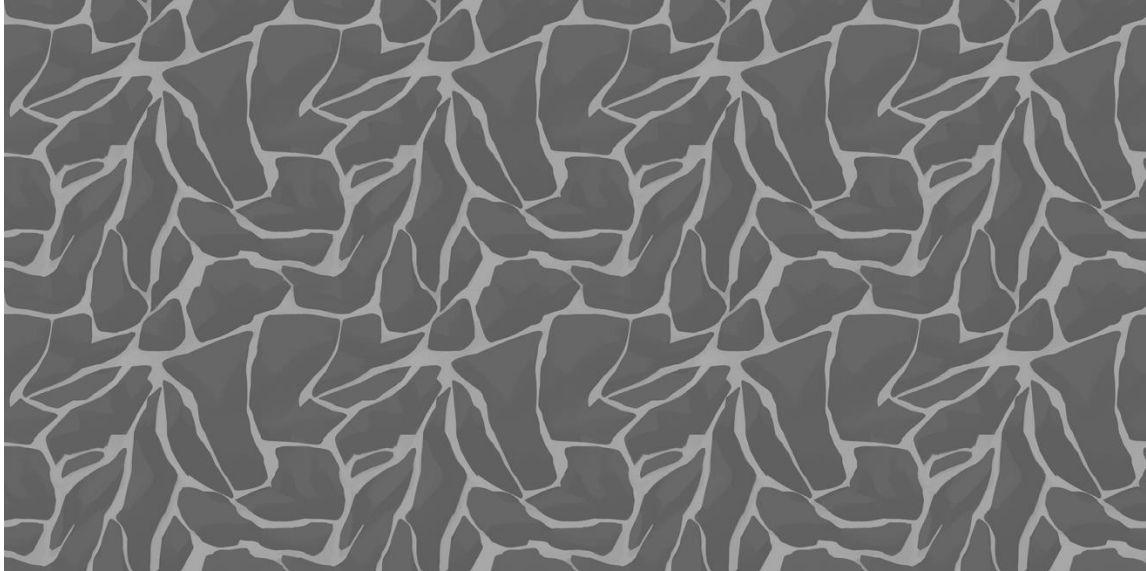


Abbildung 1 Hintergrund - risse im Spiel leuchtend

*Abbildung 2 Wand –
ende eines Raumes*



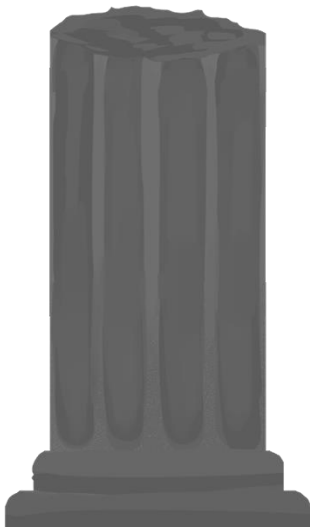


Abbildung 2 kaputte Säule - klein

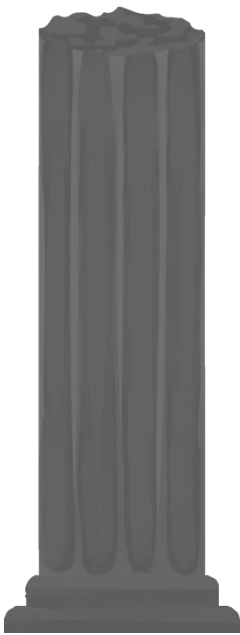


Abbildung 3 kaputte Säule

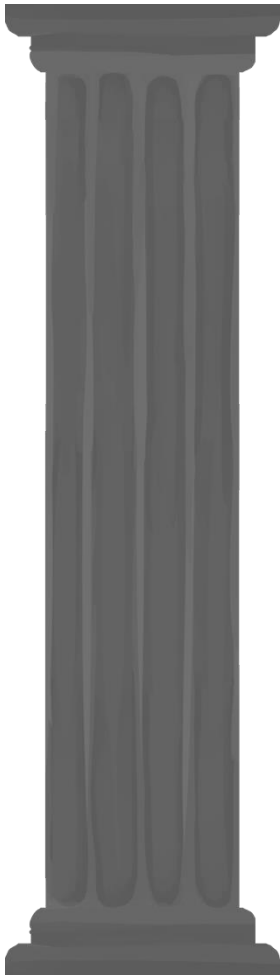


Abbildung 4 ganze Säule

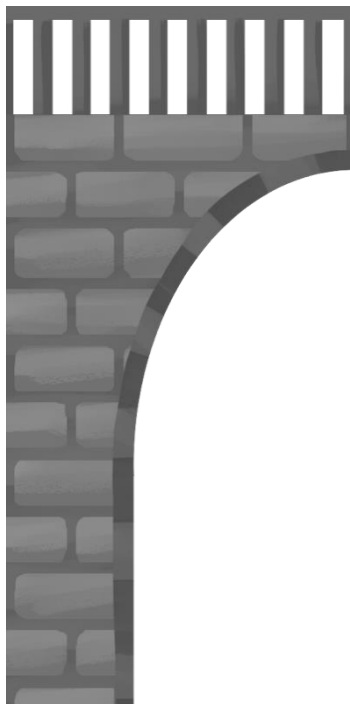


Abbildung 5 Halber Torbogen

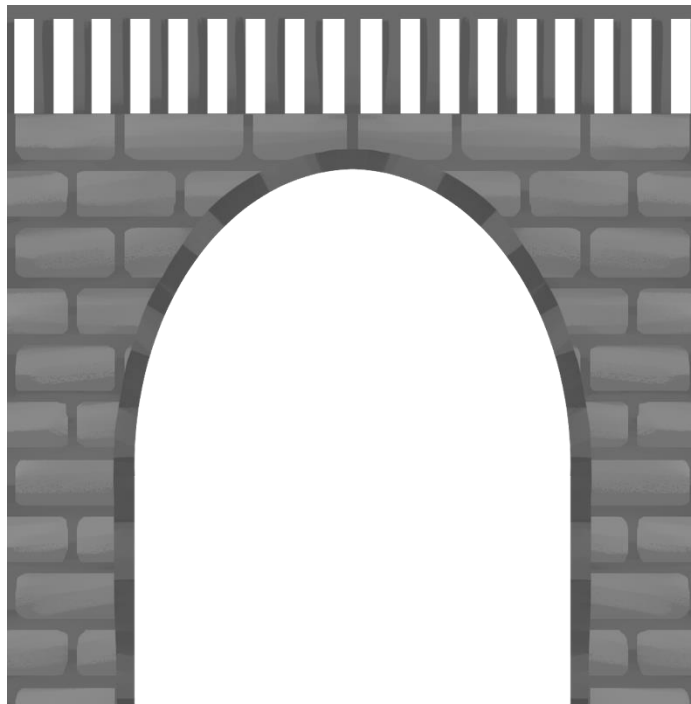


Abbildung 6 Torbogen

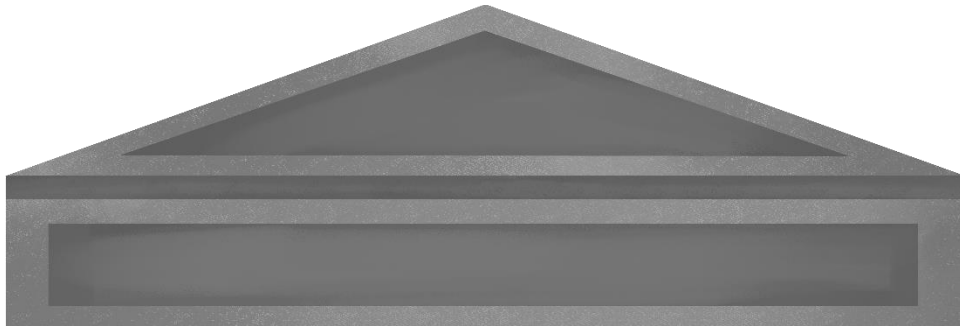


Abbildung 7 Temepl giebel



*Abbildung 8 orientalischer
Bogen*



Abbildung 9 hängende Pflanzendekoration

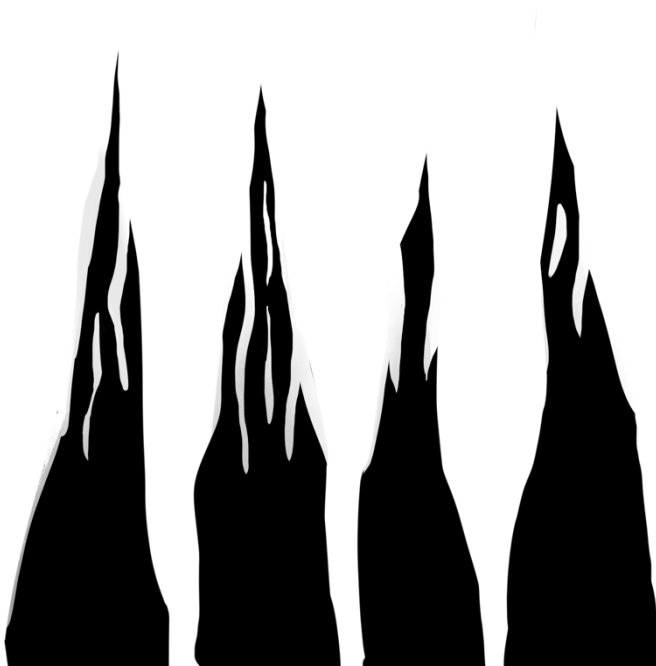


Abbildung 10 - Spikes | führt bei Kollision mit Spieler zu „instant-death“

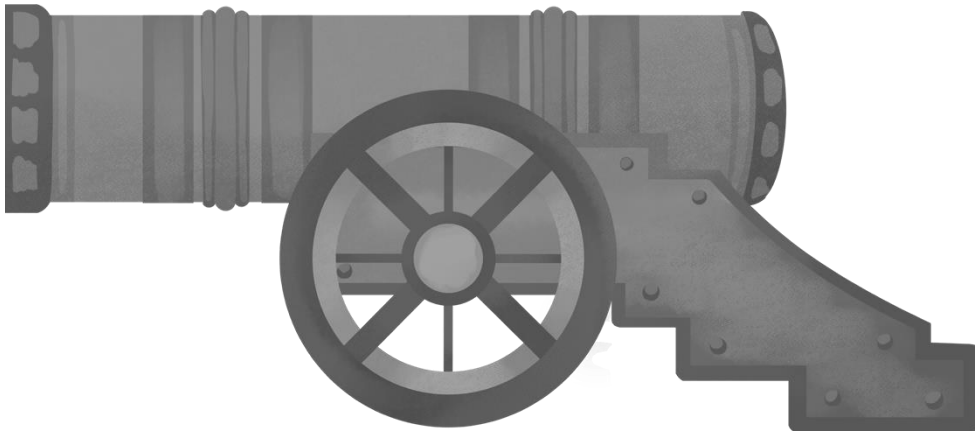


Abbildung 11 - Kanone | führt bei Kollision mit Spieler zu „instant-death“



Abbildung 12 großer Baum

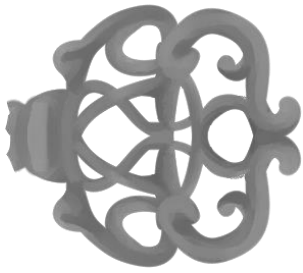


Abbildung 17 Schlüsselteil 1



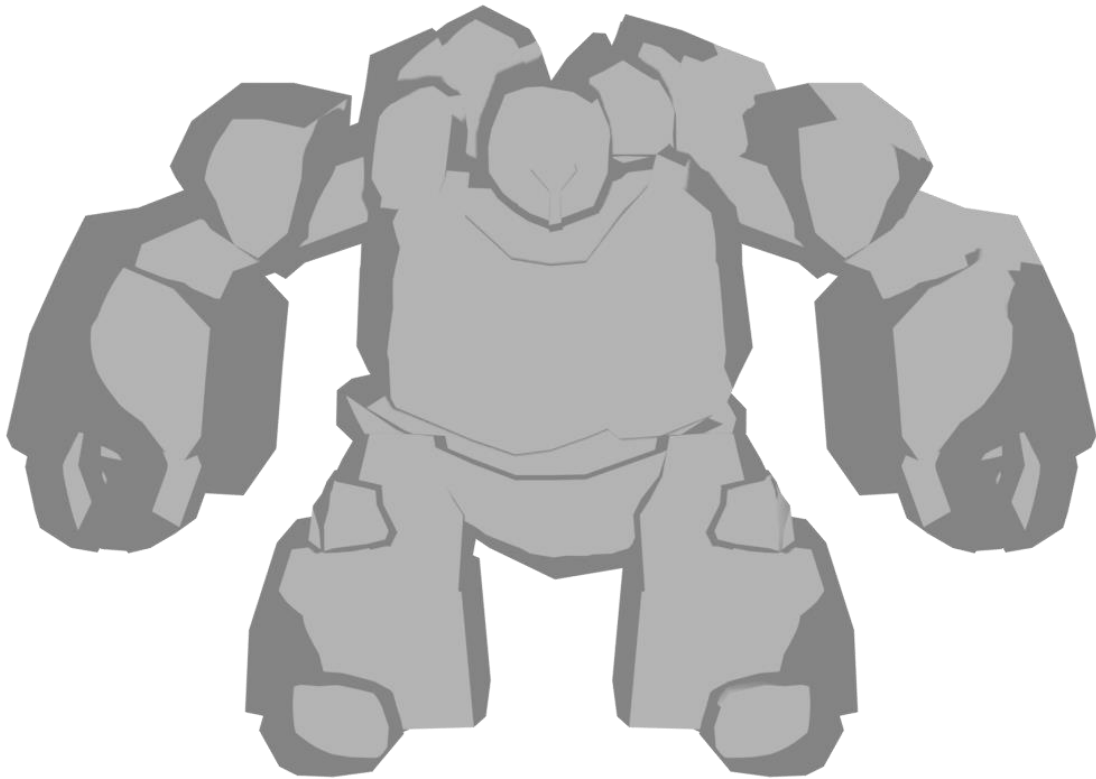
Abbildung 18 Schlüsselteil 2



Abbildung 19 Schlüsselteil 3

Akteure

Endboss



Erklärung

Der Golem Kampf besteht aus zwei Phasen. Die erste Phase besteht aus zwei Attacken und einer zwischen Animation. Bei der ersten Attacke rennt der Golem auf den Spieler zu und führt einen Schlag aus. Die zweite Attacke ist ein Steinwurf auf den Spieler. Die zwischen Animation wird nach dem Schlag ausgeführt, so dass der Spieler Zeit hat sich auf den nächsten Angriff vor zu bereiten.

Bei der zweiten Phase ändern sich eine Attacke und eine weitere kommt dazu. Anstatt der Schlag Attacke wird nun eine Sprung Attacke ausgeführt, welche ebenso nach dem Rennen stattfindet. Der Steinwurf bleibt bei dieser Phase unverändert. Bei der neuen Attacke handelt es sich um einen Steinhagel. Hier führt der Golem eine „Brüllen“ Animation durch. Die zweite Phase wird durch den Lebensverlust des Golems ausgelöst, und zwar genau nach dem er 50% seines Lebens verloren hat.

Angriffswolke



Die Angriffswolke ist erst freigeschaltet, wenn der Spieler das Level mit dem Endkampf erreicht. Diese Wolke dient als Waffe um gegen den Gegner zu attackieren. Es folgt den Bewegungen des Spielers.

Spieler



Mit dem Spieler kann der Nutzer sich durch die Level bewegen und mit der Umwelt interagieren. Durch die flinken Bewegungen soll der Nutzer mit dem Spieler seine Geschicklichkeit unter Beweis stellen. Der Spieler befindet sich in zwei Zuständen. Im ersten Level (wenn die Schlüsselfragmente gesucht werden) kann der Spieler nicht angreifen und stirbt nach einer Kollision mit bestimmten Objekten.

Quellen:

Bild1 : <https://tse2.mm.bing.net/th?id=OIP.kmHFFyrMhXMZTOSDUphsfAHaEK&pid=Api>

Bild2: [http://www.fprq.net/wp-](http://www.fprq.net/wp-content/uploads/2015/05/wallpaper-ori-and-the-blind-forest-01-1920x1080.jpg)

[content/uploads/2015/05/wallpaper ori and the blind forest 01 1920x1080.jpg](http://www.fprq.net/wp-content/uploads/2015/05/wallpaper-ori-and-the-blind-forest-01-1920x1080.jpg)