

Anforderungsanalyse

Inhaltsverzeichnis

1	Introduction	2
1.1	Purpose of the System	2
1.2	Scope of the System	2
1.3	Objectives and Success Criteria	2
1.4	Definitions, Acronyms, and Abbreviations	2
1.5	References	3
1.6	Overview	3
2	Proposed System	4
2.1	Overview	4
2.2	Functional Requirements	4
2.3	Nonfunctional Requirements	9
2.3.1	User interface and human factors	9
2.3.2	Documentation	9
2.3.3	Hardware considerations	9
2.3.4	Performance characteristics	9
2.4	Pseudo Requirements	10
2.5	System Models	10
2.5.1	Use Case Model	10
2.5.2	Class Diagrams	11
2.5.3	User-Interface	11
3	Glossary	12

1 Introduction

1.1 Purpose of the System

Das Ziel dieses Projekts ist die Entwicklung eines 2D-Action-Platformers, der dem Spieler ein unterhaltsames, herausforderndes und atmosphärisches Spielerlebnis bietet. Dabei steht das Zusammenspiel aus Bewegungsfreiheit, Kampfmechaniken und Erkundungselementen im Vordergrund.

1.2 Scope of the System

Das System umfasst alle Kernfunktionen eines Einzelspieler-Actionspiels. Dazu gehören:

- Steuerung der Spielfigur mit Bewegungs-, Sprung- und Kampfmechaniken.
- Interaktion mit der Spielwelt durch Objekte, Gegner und Umgebungsbedingungen.
- Benutzeroberfläche zur Anzeige wichtiger Spielinformationen (Health, Stamina, Sünden, BloodVials).
- Verwaltung von Spielständen sowie grundlegende Konfigurationsmöglichkeiten.

1.3 Objectives and Success Criteria

Das Projekt gilt als erfolgreich, wenn folgende Punkte erreicht werden:

- Entwicklung einer stabilen, spielbaren Version mit vollständiger Implementierung der Bewegungs- und Kampfmechaniken.
- Integration von mindestens ein Spielabschnitten mit genau einem Endgegner.
- Übersichtliches und funktionales HUD zur Darstellung spielrelevanter Daten.
- Keine kritischen Fehler oder Abstürze bei regulärem Spielverlauf.
- Einhaltung der vorgegebenen Performance-Anforderungen.

1.4 Definitions, Acronyms, and Abbreviations

- TP – Trefferpunktzahl.
- DMG - Schaden.
- MAX_HEALTH - Maximale Trefferpunktzahl.
- MAX_STAMINA - Maximale Stamina.
- BVMaxUses - Maximale Anzahl an BloodVials.
- HEAL_AMOUNT - TP-Anzahl um die geheilt wird.
- HUD – Permanente Dateneinzeige für Spieler (HP, Stamina, Bloodvials, Sünden).
- FPS – Frames per Second.

- px - Pixel.
- Checkpoints - Ort, an dem der Spieler wiederbelebt wird.
- Spikes - Stacheln, die dem Spieler Schaden zufügen.
- Sünden – In-Game Währung, die durch Kämpfe gesammelt wird.
- Blood Vial – Heilitem, das Trefferpunkte wiederherstellt.
- Stamina – Ausdauer des Spielers.
- Fähigkeit - Eine Aktion, die bei Tastendruck ausgeführt wird.
- Dash - Schnelle Bewegungsfähigkeit des Spielers, um von einem Punkt zum anderen zu kommen.
- Achsen - x-Achse geht von links nach rechts, y-Achse geht von oben nach unten.
- Bewegungsrichtung - Ein 2D-Vektor, der normalisiert ist (Hat Betrag von 1) und die x und y Komponente können jeweils zwischen -1 und 1 liegen.
- Blickrichtung - Richtung in die der Spieler schaut, es ist entweder 0 oder 1 (Links oder Rechts).
- Velocity - Positionsänderung in x- und y-Richtung pro Sekunde.
- Detectionbereich - Bereich wo ein Gegner den Spieler erkennt.
- Armor - Rüstung des Gegners.
- Hitbox - Bereich für die Kollision eines Objektes

1.5 References

Im Rahmen der Entwicklung wurden folgende Technologien, Frameworks und Standards verwendet:

- **Godot Engine 4.4.1** – Open-Source-Spiel-Engine für 2D- und 3D-Spiele. Genutzt für die Implementierung der gesamten Spielmechanik, Grafik, Szenenstruktur und UI.
- **GUnit4 5.1.0** – Framework für automatisierte Unit-Tests innerhalb von Godot-Projekten. Dient der Absicherung der Funktionalität auf Codeebene.
- **GDMUT** – Erweiterung/Toolset zur Erstellung und Verwaltung von Mock-Objekten und Testfällen in GDScript zur Unterstützung von GUnit4.
- **Git** – Versionskontrollsystem zur Verwaltung des Projektfortschritts und der Zusammenarbeit im Team.

1.6 Overview

Dieses Dokument beschreibt alle funktionalen und nicht-funktionalen Anforderungen des Spiels. Dazu gehören Bewegungs-, Kampf- und Interaktionsmöglichkeiten, die Spielumgebung, Benutzeroberfläche, Performance und Entwicklungsstandards.

2 Proposed System

2.1 Overview

Alle Anforderungen sind nummeriert und enthalten eine quantitative Beschreibung.

2.2 Functional Requirements

2.2 Funktionale Anforderungen

2.2.1 Spieler

2.2.1.1 Movement - Der Spieler muss sich wie folgt bewegen können:

1. Der Spieler muss sich mit 100px/s nach rechts bewegen können, solange [Taste „D“] gedrückt wird
2. Der Spieler muss sich mit 100px/s nach links bewegen können, solange [Taste „A“] gedrückt wird
3. Der Spieler muss initial mit -300px/s nach oben springen können, wenn [Taste „W“] einmalig gedrückt wird
4. Der Spieler muss mit -300px/s doppelt springen können, wenn: [2x Taste „W“]
 - i) er sich in der Luft befindet
 - ii) er keine Fähigkeit währenddessen ausführt
 - iii) er mehr als 15 Stamina besitzt
5. Der Spieler muss mit 300px/s in die Bewegungsrichtung dashen können, wenn: [Taste „L-Shift“]
 - i) er mehr als 20 Stamina besitzt
 - ii) er nicht angreift
 - iii) die Bewegungsrichtung darf nicht (0,0) sein

2.2.1.2 Kampf - Der Spieler muss folgende kampfbezogene Aktionen ausführen können:

1. Der Spieler muss eine leichte Attacke in Blickrichtung durchführen können, wenn:
[Taste „Linke Maustaste“] einmalig gedrückt wird
 - i) er mehr als 10 Stamina besitzt
2. Der Spieler muss eine schwere Attacke in Blickrichtung durchführen können, wenn:
[Taste „Rechte Maustaste“] einmalig gedrückt wird
 - i) er mehr als 25 Stamina besitzt
3. Der Spieler muss ein Angriff blocken können, solange: [Taste „Leertaste“] gedrückt wird
4. Der Spieler muss sich mit einem Blood Vial heilen (+25TP) können [Taste „Q“]

2.2.1.3 Stamina – Der Spieler hat eine Ausdauer 0...MAX_STAMINA, diese regeneriert sich nach 3 Sekunden, wenn keine Stamina benötigt wurde. Folgende Aktionen reduzieren die Stamina:

1. Leichte Attacke [-10 Stamina]
2. Schwere Attacke [-25 Stamina]
3. Doppelsprung [-15 Stamina]
4. Dash [-20 Stamina]

5. Blocken stoppt Stamina Regeneration. Beim erfolgreichen Blocken wird von der Stamina der geblockte Damage (DMG) abgezogen. Falls es keine Stamina gibt, werden Lebenspunkte (TP) abgezogen.

2.2.1.4 Health – Der Spieler hat eine Trefferpunktanzahl (TP) von 0...MAX_HEALTH.

1. Der Spieler stirbt wenn $TP \leq 0$
2. Der Spieler heilt sich wenn er ein Blood Vial trinkt [+25 TP]
3. Beim Checkpoint wird der Spieler komplett geheilt [$TP = MAX_HEALTH$]
4. Beim erleiden des Schadens verliert der Spieler den Schadenswert an TP [$TP -= DMG$]
5. Health regeneriert sich NICHT über Zeit

2.2.1.5 Leveling & Sünden – Der Spieler kann durch Gegner und Umgebung Sünden erlangen, mit diesen kann er folgende Attribute bei einem Checkpoint verbessern:

1. MAX_HEALTH + 25TP [100 Sünden]
2. MAX_STAMINA + 25 Stamina [100 Sünden]
3. Waffenangriff +5 DMG [100 Sünden]
4. Blood Vials: Heilkraft +5 TP [100 Sünden]

Glasscherben können verwendet werden, um ein extra Blood Vial zu erlangen

2.2.1.6 Tod – Der Spieler verliert alle Sünden beim Sterben.

2.2.1.7 Kollision – Der Spieler muss mit folgenden Objekten kollidieren können:

1. Boden und Wände
2. Gegner

2.2.2 HUD

2.2.2.1 Healthbar – Im Display muss die Healthbar oben links angezeigt werden. Diese muss die momentane TP des Spielers abbilden.

2.2.2.2 Staminabar – Im Display muss die Staminabar oben links unter der Healthbar angezeigt werden. Diese muss die momentane Stamina des Spielers abbilden.

2.2.2.3 Bloodvials – Im Display muss die Anzahl der Blood Vials unten links des HUDs angezeigt werden. Diese muss die momentane Anzahl der Blood Vials abbilden.

2.2.2.4 Sünden – Im Display muss die Anzahl der Sünden unten rechts des HUDs angezeigt werden. Diese muss die momentane Anzahl der Sünden abbilden.

2.2.3 Gegner & Boss

2.2.3.1 Movement:

1. Der Gegner muss den Spieler anfangen zu verfolgen, wenn dieser sich in seinem Detectionbereich (Kreis mit Radius von 70px) befindet.
2. Der Gegner muss aufhören den Spieler zu verfolgen, wenn dieser den Pursuingbereich (Kreis mit Radius von 179px) verlässt.
3. Jeder Gegner hat eine eigene Startposition [x,y – koordinate im jeweiligen Raum]
4. Der Gegner kehrt nach 5 Sekunden zur Startposition zurück, solange dieser keinen Spieler verfolgt.

2.2.3.2 Angriff:

1. Der Gegner muss eine leichte Attacke durchführen, wenn der Spieler weniger als 45px entfernt ist.

2.2.3.3 Health – Der Gegner hat eine Trefferpunktanzahl (TP) 0....MAX_HEALTH.

1. Der Gegner stirbt, wenn Health \leq 0.
2. Die TP des Gegners muss 5px über den Gegnersprite sichtbar sein

2.2.3.4 Tod – Wenn der Gegner stirbt, bekommt der Spieler Sünden

[Anzahl Sünden = SIN_AMOUNT]

2.2.3.5 Kollision – Der Gegner muss mit folgenden Objekten kollidieren können:

1. Boden und Wände
2. Spieler

2.2.3.6 Bei 50% Leben des Bosses –

1. Der Boss wird goldfarben RGBA(1.0f, 0.84f, 0.0f, 1.0f) , sobald 50 % seiner TP erreicht ist.
2. Der Boss erhält doppelte Rüstung [2x Armor] unter 50 % TP.
3. Der Boss belebt alle gestorbenen Gegner im Raum wieder, sobald er unter 50 % TP fällt.

2.2.4 Environment

2.2.4.1 Brunnen – Diese dienen als Checkpoint für den Spieler. Wenn dieser die Hitbox (200px × 200px) des Brunnen betritt, passiert folgendes:

1. Alle Blood Vials werden aufgefüllt [Anzahl BloodVials = BVMaxUses]
2. Der Spieler heilt sich zu 100%. [TP = MAX_HEALTH]
3. Stamina füllt sich zu 100%. [Stamina = MAX_STAMINA]

2.2.4.2 Kiste – Der Spieler kann mit Kisten interagieren. Kann Glasscherben oder Sünden enthalten.

2.2.4.3 Plattform - Der Spieler kann sich auf einer Plattform mit der Form 45px × 5px bewegen und wird bei Kontakt aus allen Richtungen (oben, unten, seitlich) durch eine CollisionShape2D zuverlässig gestoppt.

2.2.4.4 Spike – Der Spieler bekommt DMG bei betreten der Hitbox 30px × 7px. [TP == DMG]

2.2.4.5 Tür - Diese dienen zum Wechseln der Räume. Sie teleportiert den Spieler zur eine anderen festgelegten Tür, sobald der Spieler die Hitbox 8px × 50px betritt.

2.2.5 Spielmanagement

2.2.5.1 – Der Benutzer muss das Spiel über einen Button im Hauptmenü mit der Beschriftung „Continue“ starten und über den Button „Quit“ zuverlässig schließen können; beide Buttons müssen innerhalb von 1 Sekunde auf Eingaben reagieren.

2.2.5.2 Spielstand:

1. Der Benutzer muss in der Lage sein, den Spielstand zu speichern, über den Knopf „Save Game“
2. Der Benutzer muss in der Lage sein, den Spielstand zu löschen, über den Knopf „Delete“
3. Der Benutzer muss in der Lage sein, den Spielstand laden zu können, über den Knopf „Load Game“

2.2.5.3 Optionen – Diese Optionen dienen für die Konfiguration des Spiels.

1. Der Benutzer muss die Steuerung beliebig ändern können [Tastebelegung]
2. Der Benutzer muss die Grafik ändern können [Auflösung]
3. Der Benutzer muss die Sounds ändern können, zwischen 0% – 100% [Lautstärke]

2.3 Nonfunctional Requirements

2.3.1 User interface and human factors

- **UI-Reaktionszeit:** Alle Benutzeroberflächenelemente (Buttons, Menüpunkte, Interaktionen) müssen auf Benutzereingaben innerhalb von maximal **1 Sekunde** reagieren.
- **Button-Größe:** Interaktive Schaltflächen müssen eine Mindestgröße von **64 × 64 Pixel** besitzen, um auch bei niedriger Auflösung gut bedienbar zu sein.
- **HUD-Anzeige:** Die wichtigsten Werte (Health, Stamina, Sünden, Blood Vials) müssen dauerhaft sichtbar und innerhalb von **100 Millisekunden** nach einer Veränderung (z.B. Schaden, Heilung) aktualisiert werden.
- **Menüführung:** Alle Spielmenüs müssen logisch gegliedert und mit maximal **3 Klicks** erreichbar sein.

2.3.2 Documentation

- **Nachvollziehbarkeit:** Der Quellcode muss so strukturiert und kommentiert sein, dass neue Entwickler die Funktionalität einzelner Module innerhalb von **30 Minuten** nachvollziehen können.
- **Dokumentationstool:** Der gesamte Code muss mit **Doxygen**-kompatiblen Kommentaren versehen werden, sodass bei jeder Hauptversion eine HTML-Dokumentation automatisch generiert werden kann.
- **Abdeckung:** Mindestens **90 %** aller Klassen, Methoden und Konstanten müssen mit beschreibenden Doxygen-Kommentaren versehen sein (/// oder /** */).
- **Aktualität:** Die Dokumentation muss stets mit der aktuellen Codebasis übereinstimmen und bei Änderungen am Code spätestens im nächsten Commit angepasst werden.

2.3.3 Hardware considerations

- **Eingabegeräte:** Das Spiel muss vollständig mit **Tastatur und Maus** spielbar sein, ohne die Notwendigkeit zusätzlicher Peripheriegeräte wie Gamepads oder Touchscreens.

2.3.4 Performance characteristics

- **Bildrate:** Das Spiel muss bei normaler Nutzung eine stabile Bildrate von mindestens **30 FPS** erreichen, gemessen auf einem durchschnittlichen System (z.B. Mittelklasse-PC ohne dedizierte Grafikkarte).
- **Stabilität:** Während einer Spielsitzung von mindestens **60 Minuten** darf das Spiel zu keinem Zeitpunkt abstürzen oder ungewollt beendet werden.
- **Ladezeiten:** Das Laden eines Levels oder gespeicherten Spielstands darf nicht länger als **5 Sekunden** dauern.

2.4 Pseudo Requirements

- **Entwicklungsumgebung:** Die Projektstruktur ist kompatibel mit Versionierungssystemen wie Git und kann problemlos über Plattformen wie GitHub verwaltet werden.
- **Betriebssystem:** Die Entwicklung und der Ziel-Build richten sich primär an Windows-Systeme, sollen jedoch grundsätzlich auch unter Linux oder MacOS lauffähig sein.

2.5 System Models

2.5.1 Use Case Model

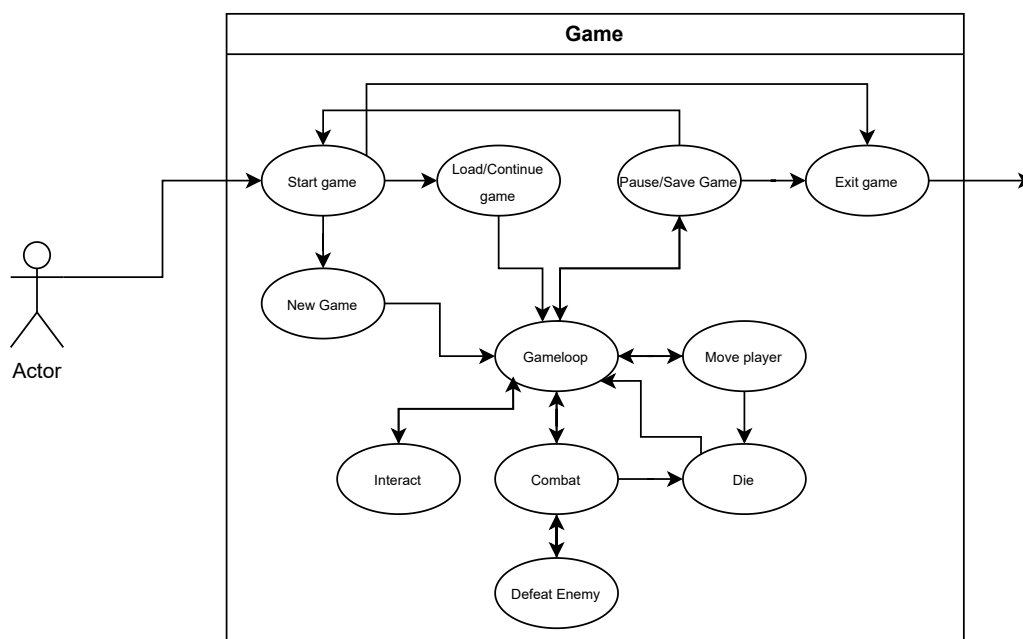


Abbildung 1: Finales Use-Case-Diagramm

2.5.2 Class Diagrams

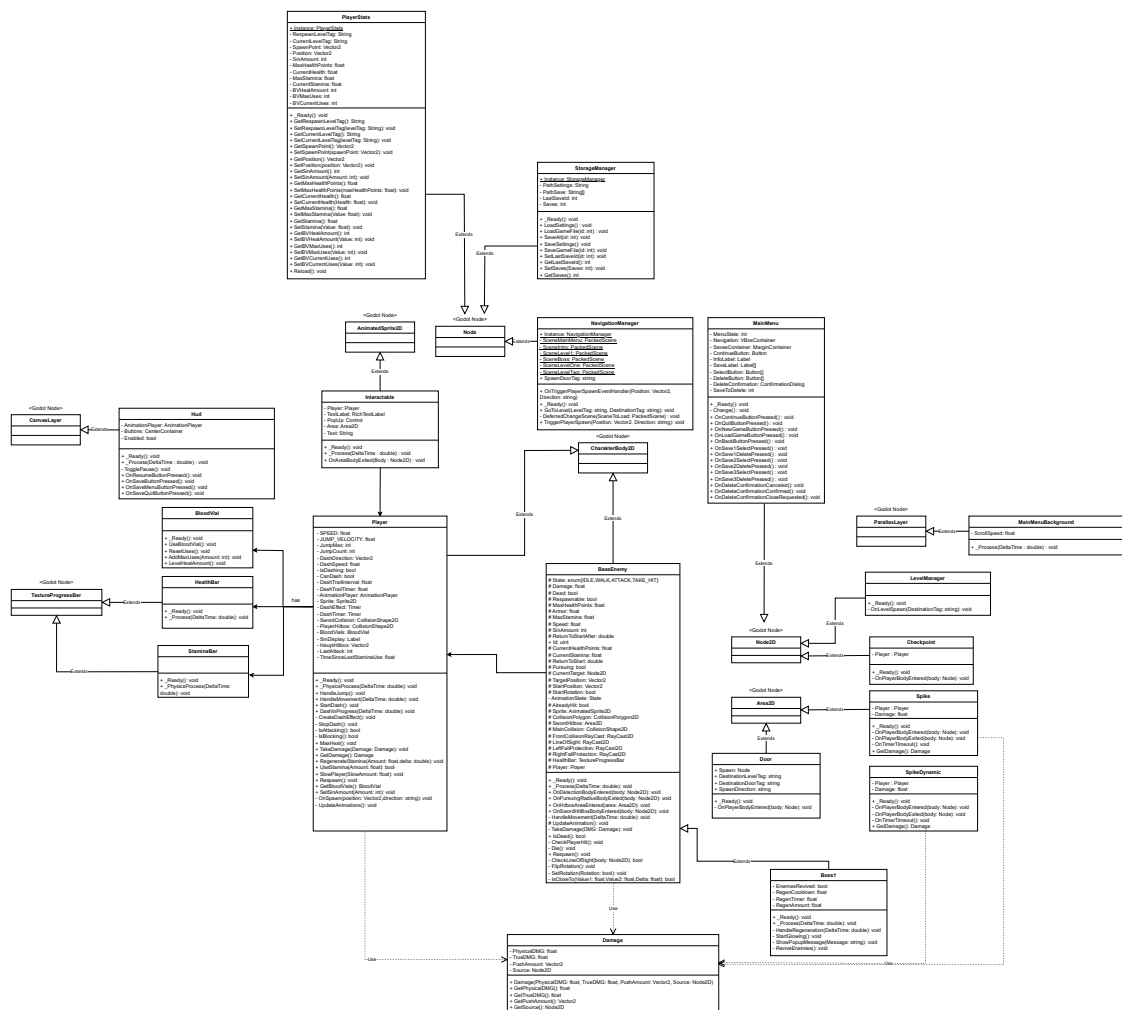


Abbildung 2: Finales UML-Klassendiagramm

2.5.3 User-Interface

Die Benutzeroberfläche ist in verschiedene, klar strukturierte Bereiche gegliedert und orientiert sich an intuitiver Bedienung.

Hauptmenü: Nach dem Start des Spiels erscheint das Hauptmenü mit folgenden Optionen:

- **Continue** – lädt den zuletzt automatisch gespeicherten Spielstand, sofern vorhanden.
- **New Game** – startet ein neues Spiel von Anfang an.
- **Load Game** – öffnet ein Untermenü mit drei Speicherplätzen (Slots), aus denen ein Spielstand geladen oder gelöscht werden kann.
- **Quit** – beendet das Spiel vollständig.

Pause-Menü: Wenn das Spiel pausiert wird über die ESC-Taste, erscheint ein Menü mit folgenden Optionen:

- **Resume** – kehrt zurück ins laufende Spiel.
- **Save Game** – speichert den aktuellen Fortschritt im gewählten Slot.
- **Save & Return to Title** – speichert den Fortschritt und kehrt ins Hauptmenü zurück.
- **Save & Quit Game** – speichert den Fortschritt und beendet das Spiel.

3 Glossary