

INF 303 Software Engineering Projekt  
Wintersemester 2023/2024

## Projektdokumentation

# Infinite Maze

05. Januar 2024

Autoren:

Topal, Oğuzhan,  
190503001@stud.tau.edu.tr  
İlhandağ, Mahmutcan,  
e190503018@stud.tau.edu.tr  
Özmen, İdil Dilara,  
e190503035@stud.tau.edu.tr  
Çek, Barkan,  
e190503034@stud.tau.edu.tr  
Salman, Ramadan,  
e190503007@stud.tau.edu.tr  
Serbest, Dilara,  
e190503037@stud.tau.edu.tr

Betreuer:

Dr.-Ing. Thomas Wilde  
M. Sc. Fulya Yenilmez

# Dokumentenverwaltung

<b>Titel:</b>	Projektdokumentation - Infinite Maze		
<b>Verantwortlicher:</b>			
<b>Datum:</b>	22 . 10 . 2023	<b>Version:</b>	0.04
<b>Seiten:</b>	24	<b>Status:</b>	gültig ▾

## Historie

Version:	Status:	Datum:	Verantwortlicher:	Änderungsgrund:
0.01	gültig ▾	17. 10. 2023	-Mahmutcan İlhandag -Ramadan Salman -Oğuzhan Topal	Meilenstein 1
0.02	gültig ▾	11. 11. 2023	-Mahmutcan İlhandag -Ramadan Salman -Oğuzhan Topal -Barkan Çek	Meilenstein 2
0.03		24.12.2023	-Mahmutcan İlhandag -Dilara Serbest -İdil Dilara Özmen -Ramadan Salman -Barkan Çek	Meilenstein 3
0.04		05.01.2024	-İdil Dilara Özmen -Mahmutcan İlhandag -Barkan Çek -Ramadan Salman -Oğuzhan Topal -Dilara Serbest	Meilenstein 4

## Verwende Hilfsmittel

Dieses Dokument wurde mit folgenden Hilfsmitteln erstellt

- Google Docs
- Stable Diffusion
- Aseprite

# Inhaltsverzeichnis

<b>Dokumentenverwaltung.....</b>	<b>1</b>
Historie.....	1
Verwende Hilfsmittel.....	2
<b>Inhaltsverzeichnis.....</b>	<b>2</b>
<b>Einleitung.....</b>	<b>3</b>
Motivation.....	3
Zweck des Dokuments.....	4
Projektmanagement.....	5
MPM-Netzplan.....	5
Product Backlog.....	6
Sprint Backlog.....	7
Produktumfang.....	8
Technische Infrastruktur.....	8
<b>Komponentendiagramm.....</b>	<b>9</b>
Anforderungsanalyse.....	11
Funktionale Anforderungen.....	11
Nicht-funktionale Anforderungen.....	11
Use Cases.....	12
Risiken.....	15
Technischer Entwurf als OOM.....	16
Qualitätssicherung – Testing.....	17
A. Testplanung.....	17
<b>Ergebnisse.....</b>	<b>19</b>
<b>Zusammenfassung.....</b>	<b>21</b>

# Einleitung

## Motivation

Für Menschen, die Indie-Spiele und Rätsel lieben, treibt uns der Wunsch an, ein eindringliches und fesselndes Erlebnis für Spieler zu schaffen. Unser Projekt zielt darauf ab, dem Gaming-Universum eine neue Dimension zu verleihen, indem es ein Puzzle-Spiel bietet, das die Spieler herausfordert, unterhält und befähigt, komplexe Labyrinth zu erkunden und zu erobern.

Unsere Vision besteht darin, ein einzigartiges Spielerlebnis zu entwickeln, das den Nervenkitzel des Plattformspiels mit den logischen und problemlösenden Elementen eines Rätselspiels verbindet. Die Spieler werden sich in einer ständig verändernden Welt von Labyrinthen wiederfinden, die dynamisch von einem ausgeklügelten Algorithmus generiert werden. Kein Labyrinth gleicht dem anderen, so dass jede Reise frisch und unvorhersehbar ist.

Das zentrale Ziel unseres Spiels ist es, sich durch diese komplexen Labyrinth zu navigieren und den Ausgang zu finden. Der Weg ist nicht geradlinig, da die Spieler auf dem Weg auf zahlreiche Hindernisse und Herausforderungen stoßen werden. Interaktionen mit nicht-spielbaren Charakteren (NPCs), Begegnungen mit Mobs und die Verlockung versteckter Truhen werden die Spieler engagieren und jede Entscheidung entscheidend machen.

Wir verstehen, dass ein Teil der Freude am Spielen aus der Charakterentwicklung kommt. In unserem Spiel können die Spieler ihre Charaktere durch verschiedene Level weiterentwickeln, neue Fähigkeiten erwerben und ihre Werte verbessern. Diese Entwicklung verleiht dem Gameplay Tiefe und motiviert die Spieler, selbst die anspruchsvollsten Labyrinth zu meistern.

Unser Projekt ist von dem Wunsch motiviert, den Spielern ein wirklich einzigartiges und fesselndes Erlebnis zu bieten. Wir glauben an die Kraft von Spielen, den Verstand herauszufordern, Kreativität zu fördern und eine Flucht aus dem Alltag zu bieten.



*Abb.1: Das Bild unseres Spiels in unseren Träumen wird diesem ähnlich sein, obwohl sich das Endprodukt im Rahmen der Möglichkeiten ändern kann, die die Software bietet, die wir bei der Entwicklung verwenden*

## Zweck des Dokuments

Diese Dokumentation enthält Beschreibungen der Funktionalitäten des Projekts, der Methoden, die zur Umsetzung der Projektidee verwendet werden, der Technologien und Werkzeuge und des Verhaltens der Anwendung sowie eine Anleitung für die Funktionsweise der Anwendung.

# Projektmanagement

Im folgenden Kapitel greifen wir die Elemente des agilen Projektmanagements mit Scrum auf. Scrum ist eine bewährte Methode, die es unserem Team ermöglicht, sich flexibel den Herausforderungen des Softwareentwicklungsprozesses zu stellen. Dabei werden wir uns auf drei Schlüsselemente konzentrieren, die das Rückgrat unserer Entwicklung bilden: einen Netzplan, der mit der Metra Potential Methode erstellt wurde, das Scrum Product Backlog und das Scrum Sprint Backlog.

## MPM-Netzplan

Der folgende Netzplan wurde mit der Metra Potential Methode (MPM) erstellt. Der MPM-Netzplan bildet die verschiedenen Vorgänge der Entwicklung ab. Er hilft, die zeitlichen Abhängigkeiten zwischen verschiedenen Aufgaben zu visualisieren und die Wahrscheinlichkeit ihres rechtzeitigen Abschlusses zu analysieren. Dieser Ansatz ermöglicht es unserem Team, Ressourcen optimal zu nutzen und Engpässe frühzeitig zu erkennen, um den Projektablauf zu optimieren.

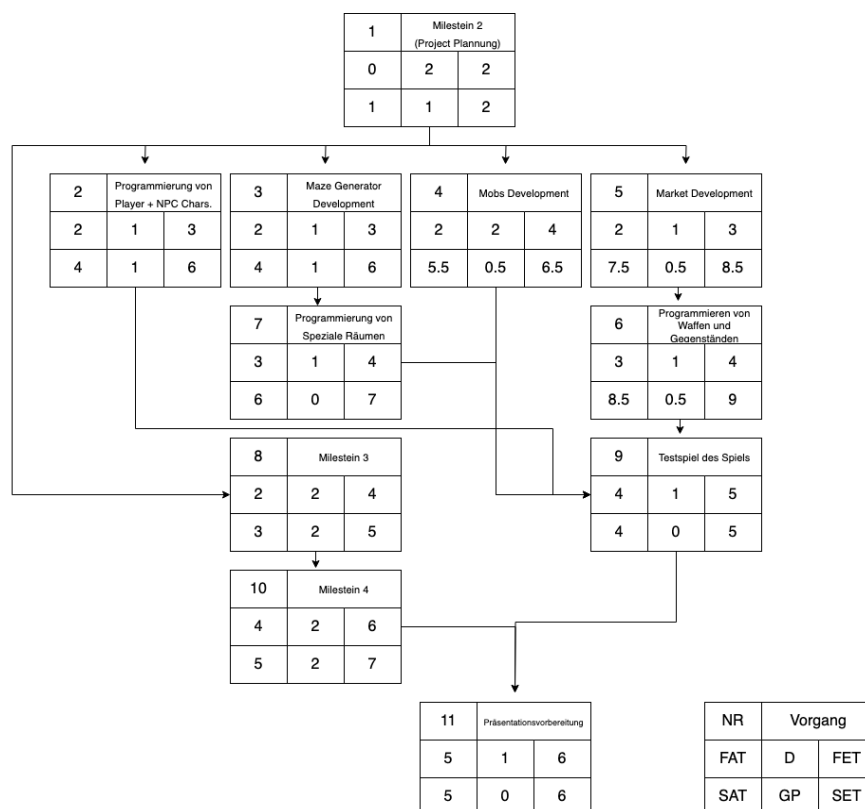


Abb.2: MPM-PLAN

## Product Backlog

Das Product Backlog stellt das Herzstück unseres Scrum-Projekts dar. Es enthält alle Anforderungen, Ideen und Verbesserungsvorschläge, die das Produkt voranbringen sollen. Es ist eine lebendige Liste, die sich ständig entwickelt und die sich an sich ändernde Bedürfnisse anpasst.

Items					Sprint	
Items (17)	Type	Priorität	Status	Beschreibung / Name		Beschreibung (lang)
is-5	Epic	hoch	wartend	Implementierung von ein procedural Maze Generator Skript		1 Erwerben des Algorithmus, dessen Labyrinth prozedural generiert werden und integrieren dieser in das Spiel. Das Labyrinth (sogar machen und zusätzliche Funktionen hinzufügen (Mobs, Events).
is-9	Epic	hoch	wartend	PlayerNPC Design und Implementation		1 Ermitteln der Statistiken des Hauptcharakter und Programmieren des Gameplay
is-10	Epic	hoch	wartend	Mobs Development		1 Designen, anfertigen, Erwerben von NPC's, die dem Spiel hinzuverfügt werden sollen, und Interaktion mit der Hauptfigur, falls vorhanden
is-11	Task	niedrig	wartend	Boss		2 Bestimmen der Statistiken von Mobs und Programmieren ihres allgemeinen Verhaltens 2 (aggressive Reichweite, Besondereheiten seiner Angriffe).
is-12	Task	niedrig	wartend	Räumen		2 Die Statistiken und Fähigkeiten des Bosses bestimmen, sein Erscheinungsbild entwerfen und alles codieren.
is-13	Task	niedrig	wartend	Market/Economy System Entwicklung und Development		3 Bestimmen Sie die Spawn-Raten der Räume. Beispiel: Bossraum (alle 5-10 Blockwerke), Marktbaum (oft), Item-Drop-Raum (seiten) usw.
is-14	Task	niedrig	wartend	Design und Development von Items		4 Den Markt im Spiel gestalten, Ermittlung von Verkaufs- und Einkaufspreisen von Waren.
is-15	langfristig	niedrig	wartend	Soundeffekte und Music		4 Bestimmen der Statistiken der Gegenstände im Spiel, Erwerben ihres Aussehens und 4 Bestimmung ihrer Sicherheit.
is-16	langfristig	niedrig	wartend	Story-Panel		5 Erstellung der Soundeffekte und Musik des Spiels
is-0	Task	hoch	abgeschlossen	Milesten 1 - Motivation		5 Erstellen von Zwischensequenzen, um die Geschichte des Spiels weiterzuentwickeln und sie dem Spieler zu präsentieren.
is-1	Task	hoch	abgeschlossen	HA: Gruppenbildung und Projektidee	-	Es wurde eine Dokumentation erstellt, die die Motivation des Projekts und seine Grenzen enthält. (infinite Maze - Projektdokumentation.pdf)
is-2	Task	mittel	abgeschlossen	Abschluss der freecodecamp Godot-Schulung	-	Markupfile: infinite_Maze-Hausaufgabe_1-Gruppenbildung_Proje.md
is-3	Task	mittel	abgeschlossen	Literatürübersicht zu prozeduralen Spielen, die mit Godot erstellt wurden.	-	Wir haben einen Kurs besucht, um zu lernen, wie man mit Godot arbeitet.
is-4	Task	mittel	abgeschlossen	Untersuchung von Algorithmen zur Labyrinthgenerierung	-	Wir haben unser Wissen über Godot erweitert, indem wir uns angesehen haben, wie verschiedene Arten von Spielen von verschiedenen Leuten auf YouTube entwickelt wurden.
is-5	Task	hoch	abgeschlossen	Milesten 2 - Projektplan	-	Wir haben verschiedene Algorithmen zur Labyrinthherstellung untersucht und uns für die Verwendung des rekursiven Backtrack-Algorithmus entschieden.
is-6	Task	mittel	abgeschlossen	Main-Menu	-	Planung des Entwicklungsteils des Projekts.
is-7	Task	mittel	abgeschlossen	Save-Funktion	-	Programmieren des Startmenüs des Spiels
						Programmierern der Save-Funktion, die es ermöglicht, das Spiel dort fortzusetzen, wo es aufhört hat.

*Abb.3: Product Backlog*

## Sprint Backlog

Das **Sprint Backlog** hingegen ist die Blaupause für den aktuellen Arbeitszyklus. Es enthält die Aufgaben, die während eines Sprints erledigt werden sollen, basierend auf den Prioritäten des Product Backlogs. Die Sprint Backlog-Elemente werden von Teammitgliedern übernommen und während des Sprints umgesetzt.

Sprint 1

Name:	Level 1: Labyrinth + Hauptfigur			Beginn in
Projekt Ziel:	Die Entstehung der primitivsten Version des Spiels in spielbarer Form.	Dauer:	5 Arbeitstage	16
Prozess Verbesserung:	-	Items:	16	
Start Datum:	27.11.2023	Story Points:	0	Tagen
End Datum:	03.12.2023			

Sprint Metrik

Aufgaben geplant:	16	Key Achievements:	<ul style="list-style-type: none"><li>- Maze Generator</li><li>- Dorf</li><li>- Hauptcharakter</li></ul>
Aufgaben abgeschlossen:	0		
Aufgaben übrig:	16		
Story Points geplant:	44	Zusammenfassung Performance:	
Story Points erledigt:	0		
Story Points übrig:	44		

Sprint Backlog

Items (16)	Typ	Priorität	Status	Beschreibung	Story Points	verantwortlich	Notizen
1	Task	hoch	wartend	Entwerfen einer Version des Labyrinth-Algorithmus, die an die Bedürfnisse des Spiels angepasst ist.	2	Mahmutcan	
2	Task	hoch	wartend	Vorbereitung von Wand- und Bodentexturen.	2	Mahmutcan	
3	Task	hoch	wartend	Codierung des Labyrinth Algorithmus.	5	Mahmutcan	
4	Task	niedrig	wartend	Vorbereitung der Textur der Hauptfigur.	1	Barkan	
5	Task	hoch	wartend	Programmierung der Bewegungen der Hauptfiguren.	3	Oğuzhan	
6	Task	hoch	wartend	Der Charakter wird mit dem Labyrinth verbunden und bewegt sich darin.	4	Oğuzhan	
7	Task	mittel	wartend	Erstellung einer vollständigen NPC mit Dialogen Liste.	2	Barkan	
8	Task	mittel	wartend	NPCs mit Dialogen programmieren.	4	Barkan	
9	Task	niedrig	wartend	NPCs Texturen vorbereiten.	1	Barkan	
11	Task	niedrig	wartend	Erstellung der Dorfskizze.	2	Ramadan	
11	Task	niedrig	wartend	Vorbereitung von Dorftexturen	1	Ramadan	
13	Task	hoch	wartend	Programmierung des Dorfes	3	Ramadan	
14	Task	hoch	wartend	Entwicklung eines Algorithmus, der bestimmt, wo Mobs im Labyrinth erscheinen.	4	Idil	
15	Task	hoch	wartend	Programmierung des Mob-Erstellungsalgorithmus.	4	Idil	
16	Task	niedrig	wartend	Erstellung einer vollständigen NPCs ohne Dialogen Liste.	2	Dilara	
17	Task	niedrig	wartend	NPCs ohne Dialogen programmieren.	4	Dilara	Sie werfen einzelne Dialogzeilen weg, wenn die Hauptfigur vorbeigeht.

Stand Story Points

Abb.4: Sprint Backlog



## Produktumfang

In diesem Projekt wollten wir ein wiederspielbares 2D-Puzzlespiel entwickeln. Um dieses Produkt zu erstellen, haben wir einen Labyrinthgenerator, einen Spielercharakter, einen Dorfplatz und Zwischenereignisse entwickelt.

Mit unserem Produkt wollten wir dem Benutzer eine angenehme und unterhaltsame Zeit bieten. Während dieser Zeit sollten sie versuchen, die Labyrinthe zu lösen oder Minispiele in den Aktivitätsräumen zu spielen. Der Kampfgeist muss durch die Prüfung mit einem Schwierigkeitsgrad befriedigt werden, der nicht störend ist. Unser Spiel bietet dieses Erlebnis.

Wenn der Kunde das Spiel kauft, erhält er ein 2D-Puzzlespiel. Er kann dieses Spiel schätzungsweise 1 bis 2 Stunden in einer einzigen Sitzung spielen und in regelmäßigen Abständen zum Spiel zurückkehren, um seine Freizeit zu füllen. Er kann jedes Mal eine neue Erfahrung machen.

Aufgrund der begrenzten Zeit und Ressourcen unseres Spiels konnten wir nicht alle Verbesserungen, von denen wir geträumt und vorgeschlagen hatten, auf der Ideenebene verwirklichen. Es gibt viele Verbesserungen und Innovationen, die wir in Zukunft umsetzen können, wenn wir mehr Ressourcen und Zeit haben. Unser oberster Tagesordnungspunkt ist Diversität. Unser Spiel ist sowohl optisch als auch mechanisch offen für Abwechslung.

## Technische Infrastruktur

Unser Projekt nutzt die Leistung der Godot-Spiel-Engine, um ein immersives Spielerlebnis zu schaffen. Godot ist eine Open-Source, plattformübergreifender Spiel-Engine, die eine Vielzahl von Funktionen und Möglichkeiten bietet, um unser Spiel zum Leben zu erwecken. Wir werden Godot als die primäre Spielentwicklungsumgebung nutzen. Godots benutzerfreundliche Oberfläche, zusammen mit seiner umfangreichen Dokumentation und aktiven Gemeinschaft, macht es zu einer idealen Wahl für unser Projekt. Das Spiel wird mit Godot 4.1.2, um den Zugriff auf die neuesten Funktionen und Verbesserungen sicherzustellen. Eine der wichtigsten technischen Aspekte unseres Projekts ist die dynamische Erzeugung von Labyrinthen. Godot ermöglicht es uns, komplexe Spiellogik und Mechaniken mühelos umzusetzen. Unser Projekt wird intensiv von Godots szenenbasiertem System Gebrauch machen, bei dem Spielelemente als Knoten strukturiert und hierarchisch organisiert sind. Dieser Ansatz vereinfacht den Entwicklungsprozess und fördert die Wiederverwendung von Code. Godots Flexibilität ermöglicht es uns, einen Algorithmus zu implementieren, der einzigartige Layouts für Labyrinthe in jeder Spielsitzung generiert. Diese dynamische Labyrintherstellung gewährleistet, dass keine zwei Spiele gleich sind und die Spieler immer herausgefordert und begeistert sind. Charakteranimationen und Verhalten werden mit Godots integrierten Tools implementiert. Um das Benutzererlebnis zu verbessern, werden wir Spielstände speichern und den Fortschritt verfolgen. Godots

Datei-E/A-Fähigkeiten werden genutzt, um den Spielfortschritt in regelmäßigen Abständen zu speichern, damit die Spieler ihre Reise jederzeit fortsetzen können. Derzeit entwickeln wir unser Spiel nur für Windows. Unsere technische Infrastruktur ist darauf ausgelegt, die Fähigkeiten des Godot-Spiel-Engines zu nutzen und ein nahtloses und fesselndes Spielerlebnis zu bieten.

## Komponentendiagramm

Das Komponentendiagramm unseres Projekts finden Sie unten. Die Kommunikation und Organisation zwischen den Komponenten wird von Godot verwaltet und die Kommunikation zwischen ihnen wird während der Entwicklungsphase nicht berücksichtigt. Aus diesem Grund ist Godot im Diagramm als Hauptmodul dargestellt.

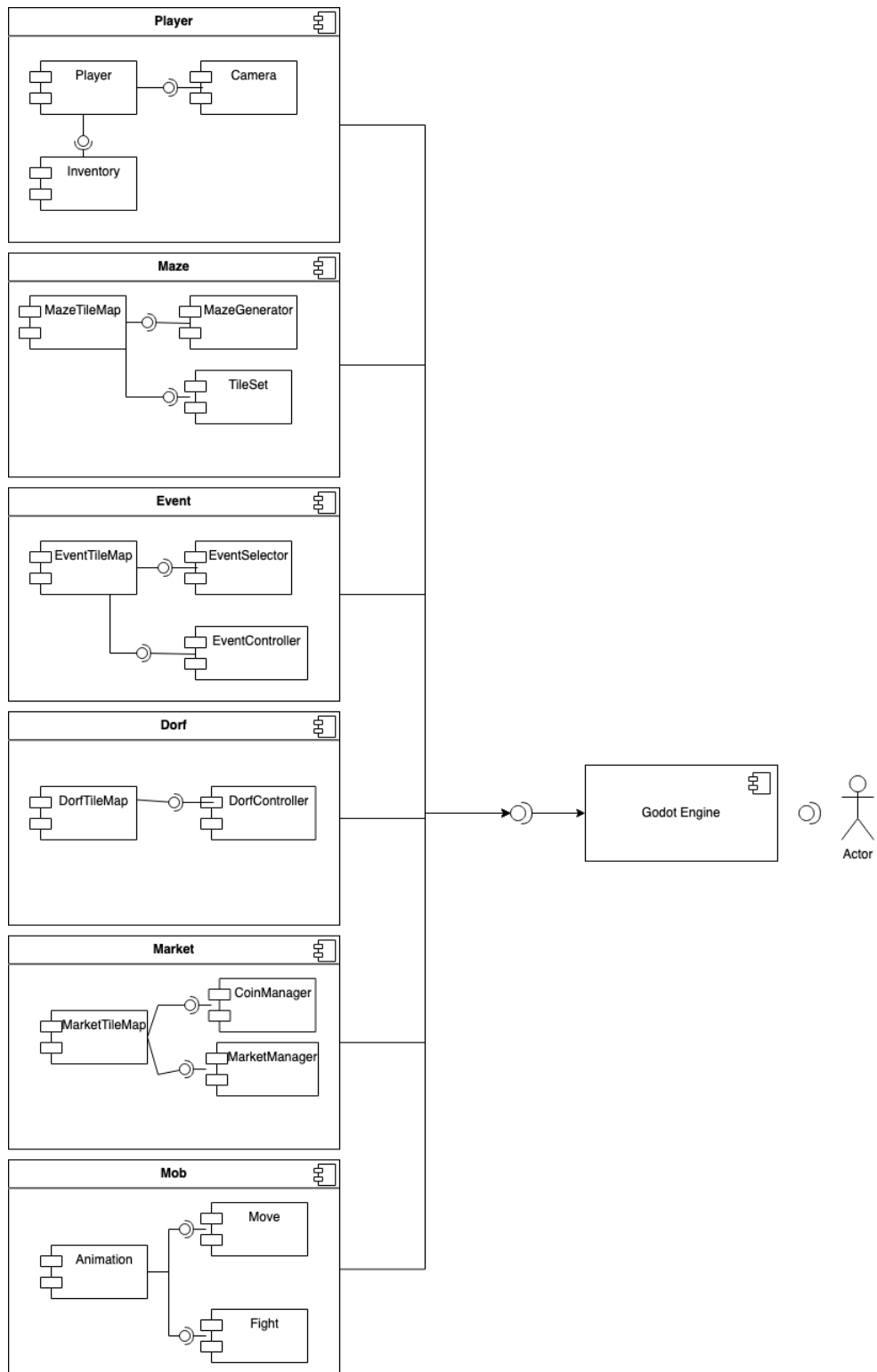


Abb. 5: Komponentendiagramm

## Anforderungsanalyse

### Funktionale Anforderungen

**FA1**-Der Spieler muss in der Lage sein, die Hauptfigur auf der x- und y-Achse zu bewegen und Angriffe auszuführen.

**FA2**-Das Dungeon-Labyrinth, das der Spieler betritt, muss bei jedem Spieldurchgang neu und einzigartig erstellt werden.

**FA3**-Es sollte ein Hauptmenü geben, das den Benutzer zu Beginn des Spiels begrüßt, und der Benutzer sollte über dieses Menü Anpassungen vornehmen und wählen können, ob er dort weitermachen möchte, wo er aufgehört hat, oder eine neue Runde beginnen möchte.

**FA4**-Die Gesundheit des Charakters, der durch die Interaktionen des Spielers mit der Umgebung beeinflusst wird, muss während des Spiels überwacht werden und der Charakter muss sterben können.

**FA5**-Es muss sichergestellt werden, dass der Spieler zu keinem Zeitpunkt auf einen Feind trifft, der ein Weiterspielen unmöglich macht.

### Nicht-funktionale Anforderungen

**NFA1**- Das Spiel muss auf der Windows-Plattform lauffähig sein. Abhängig vom zukünftigen Stand des Projekts können unterschiedliche Plattformen hinzugefügt werden.

**NFA2**- Das Spiel sollte eine isometrische Draufsicht im Pixel-Art-Stil haben und die Charaktere sollten durch Minifiguren im Hauptspielbereich des Spiels dargestellt werden. Detaillierte Ansichten sollten zusammen mit den Statusinformationen der Charaktere in einem anderen Teil des Bildschirms angezeigt werden.

**NFA3**- Wenn der Spieler eine Pause braucht, sollte er in der Lage sein, das Spiel zu unterbrechen, sich vom Computer zu entfernen und es dann fortzusetzen, wenn er zurückkommt.

## Use Cases

**Titel:** Save Funktion

**Kurzbeschreibung:** Speichern Sie den aktuellen Stand des Spiels, damit Sie später dort weitermachen können, wo Sie aufgehört haben.

**Akteure:** Spieler

**Vorbedingungen:** Das Spiel muss gestartet und gespielt werden.

**Auslösendes Ereignis:**

- 1- Das Spiel beenden.
- 2- Es ist Zeit zum automatischen Speichern.

**Beschreibung des Ablaufs:**

- 1- Bereite die zu speichernden Daten vor.
- 2- Schreibe in die Protokolldatei.

**Nachbedingung Erfolg:** Die Speicherdatei des Spiels wurde erfolgreich gespeichert.

**Nachbedingung Fehlschlag:** Eine Fehlermeldung informiert Sie darüber, dass das Spiel nicht gespeichert werden kann und daher nicht an der Stelle fortgesetzt werden kann, an der Sie aufgehört haben.

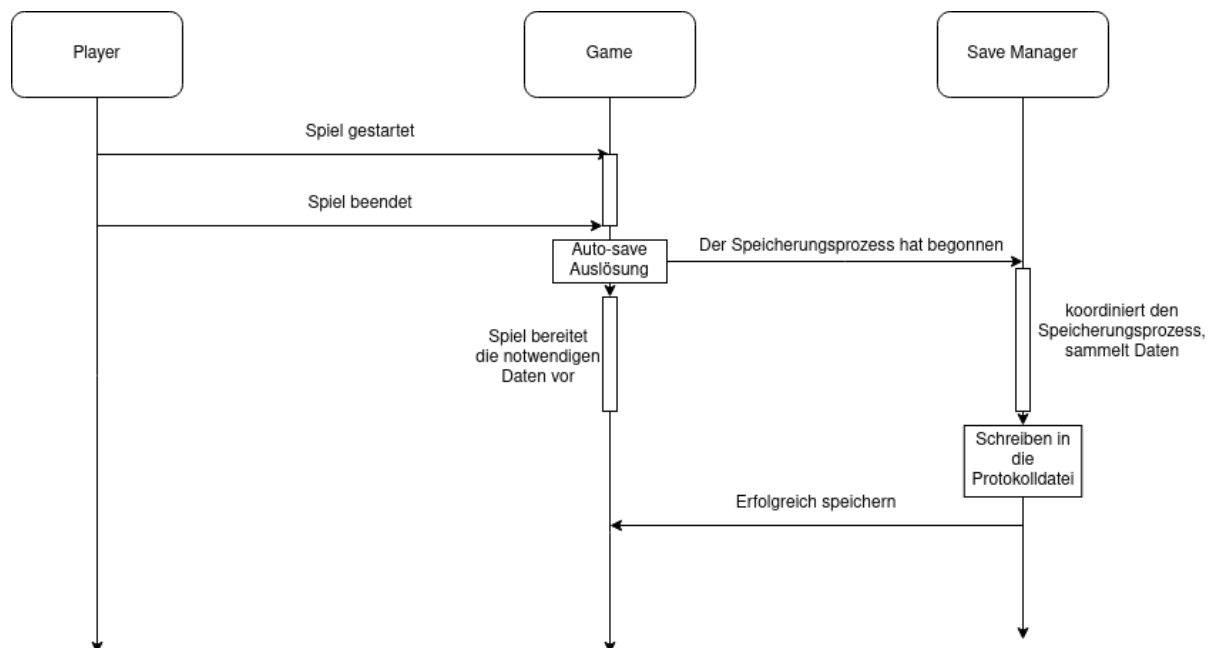


Abb. 6: Sequenzdiagramm des Use-Case "Save Funktion"

**Titel:** Erschaffung des Labyrinth-Dungeons

**Kurzbeschreibung:** Wenn eine neue Runde im Spiel beginnt, wird mithilfe von Labyrinth-Erstellungsalgorithmen ein Labyrinth-Dungeon erstellt, in dem der Spieler in dieser Runde voranschreitet.

**Akteure:** Spiel

**Vorbedingungen:** Dies ist das erste Mal, dass die Tour gestartet wird und das Labyrinth noch nie zuvor gebaut wurde.

**Auslösendes Ereignis:** Auswahl der Schaltfläche „Neues Spiel“ im Hauptmenü.

**Beschreibung des Ablaufs:**

- 1- Erstelle ein neues Dungeon-Labyrinth-System, indem Sie den Algorithmus anwenden.
- 2- Bewahre dieses neue Produkt auf.

**Nachbedingung Erfolg:** Das Labyrinth ist für andere Module zugänglich.

**Nachbedingung Fehlschlag:** Starte das Spiel nicht und geben Sie eine Fehlermeldung aus.

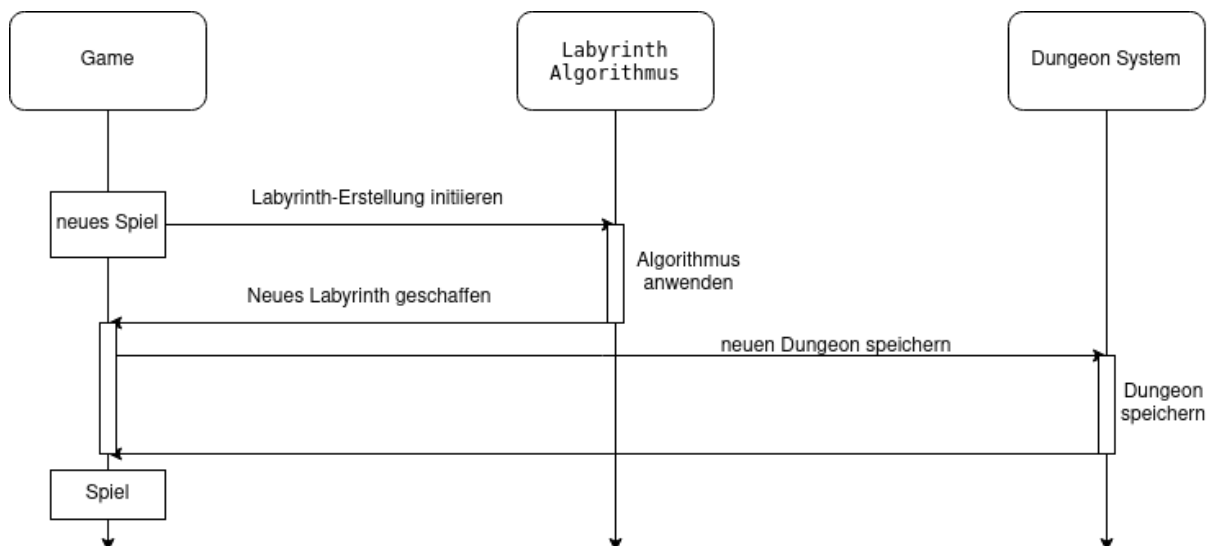


Abb.7: Sequenzdiagramm des Use-Case "Erschaffung des Labyrinth-Dungeons"

**Titel:** Spielerbewegung und Umweltauswirkungen

**Kurzbeschreibung:** Bewegung der Charaktere(Spieler, Mob, Npc) mit der Umgebung

**Akteure:** Labyrinth, Spieler, Mob, Npc

**Vorbedingungen:** Der Standort des Charakters oder die Bewegung, die er ausführen möchte, liegt innerhalb vorgegebener Grenzen.

**Auslösendes Ereignis:** Der Charakter leitet eine Aktion ein.

**Beschreibung des Ablaufs:**

- 1- Der Charakter beschließt, eine Bewegung-Aktion einzuleiten.

2- Das System prüft, ob diese Bewegung-Aktion durchgeführt wird oder nicht.

3- Als Ergebnis der Aktion befindet sich das Zeichen innerhalb der angegebenen Grenzen.

**Nachbedingung Erfolg:** Charaktere können innerhalb der festgelegten Grenzen bleiben und sich nicht an unerwünschten Orte bewegen.

**Nachbedingung Fehlschlag:** Das Vorhandensein von Charakteren außerhalb der Karte oder an Orten, an denen sie nicht auf der Karte sein sollten. Ineinander verschachtelte Charaktere.

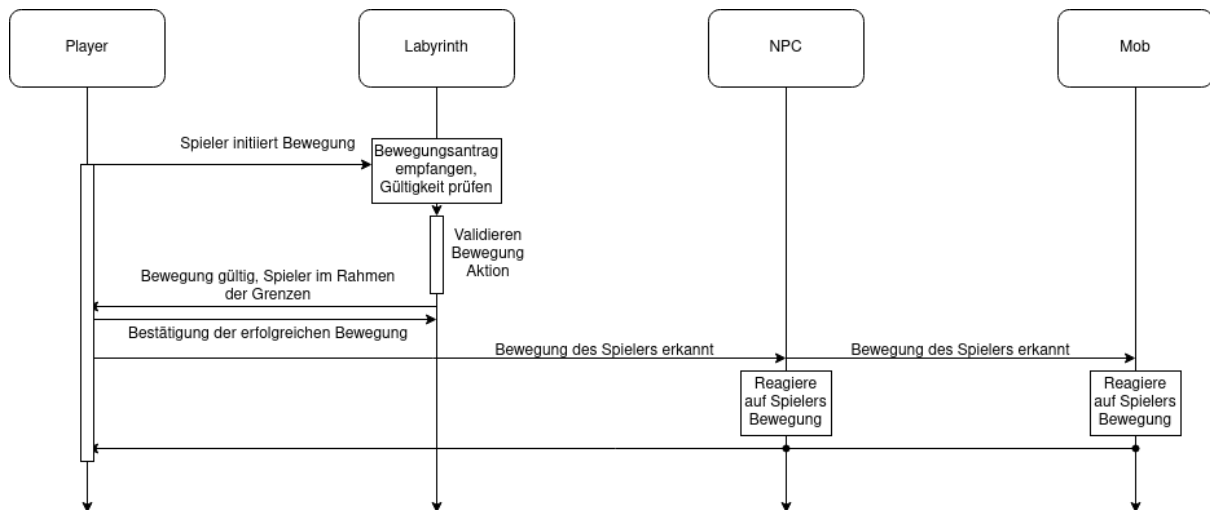


Abb.8: Sequenzdiagramm des Use-Case "Spielerbewegung und Umweltauswirkungen"

**Titel:** Management von Kämpfen

**Kurzbeschreibung:** Algorithmus, der die Auswirkungen von Spieler und Mobs in Kriegen berechnet, die im Spiel auftreten können.

**Akteure:** Spieler, Mobs

**Vorbedingungen:** Berechnung des erlittenen und verursachten Schadens basierend auf Statistiken und Gegenständen über die Charakter(Spieler und Mobs) verfügt.

**Auslösendes Ereignis:** Ein Charakter, der einem anderen Charakter Schaden zufügt.

**Beschreibung des Ablaufs:**

1- Ein Charakter beschließt, einen anderen Charakter anzugreifen.

2- Sobald der Charakter in Angriffsbereichweite ist, greift er an und der verursachte Schaden wird berechnet.

3- Wenn es welche gibt, kommen Animationen zum Angreifen und Erleiden von Schaden ins Spiel.

**Nachbedingung Erfolg:** Genaue Berechnung von Interaktionen und Aktivierung von Animationen, falls vorhanden.

**Nachbedingung Fehlschlag:** Animationen passieren nicht. Charaktere verursachen keinen Schaden.

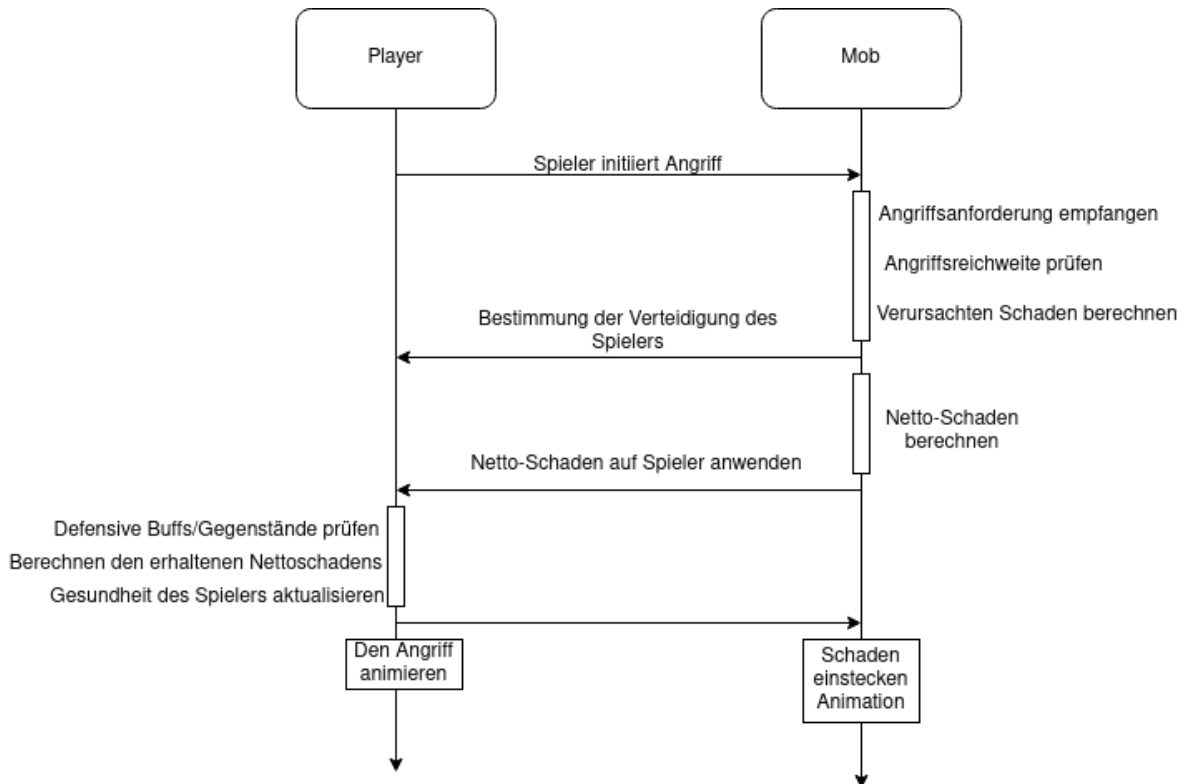


Abb.9: Sequenzdiagramm des Use-Case "Management von Kämpfen"

## Risiken

**1.Risiko:** Der prozedurale Generierung Algorithmus funktioniert möglicherweise nicht wie vorgesehen und führt möglicherweise zu bedeutungslosen Ergebnissen.

**Vorsicht:** Gleich zu Beginn des Projekts werden wir unsere gesamte Belegschaft dem Entwurf und Test dieses Algorithmus widmen und das verbleibende Projekt so aufbauen, dass der uns zur Verfügung stehende Algorithmus möglich ist.

**Aufgetreten:** Probleme mit dem Algorithmus führten zu unerwarteten Mustern.

**Auswirkung:** Verzögerungen bei der Entwicklung; erhebliche Fehlersuche und Verfeinerung erforderlich.

**Unvorhergesehen:** Die Art der spezifischen Algorithmusprobleme war nicht vollständig vorhersehbar.

**2.Risiko:** Charakter- und Umgebung Zeichnungen können aus Zeit- und Ressourcenmangel unvollständig sein.

**Vorsicht:** Die Wahl des Pixelart-Stils, der vergleichsweise einfacher zu produzieren ist, und die Unterstützung durch künstliche Intelligenz bei der visuellen Produktion.

**Aufgetreten:** Trotz dieser Vorsichtsmaßnahmen war es schwierig, alle erforderlichen Zeichnungen der Figuren und der Umgebung innerhalb des vorgegebenen Zeitrahmens fertigzustellen. Die Komplexität bestimmter Objekte und unerwartete Iterationen im Entwurfsprozess trugen zu einigen Verzögerungen bei.

**3.Risiko:** Das Spiel stürzt aufgrund eines Fehlers ab, der irgendwann im Spiel auftritt.

**Vorsicht:** Wir bereiten ein sauberes Endprodukt vor, indem wir das Spiel mehrmals spielen und testen und alle bei den Tests festgestellten Fehler beheben.

**Aufgetreten:** Angesichts der zeitlichen Beschränkungen während der Entwicklung wurde es schwierig, die ursprünglich geplanten umfangreichen Tests gründlich



durchzuführen. Diese Einschränkung beeinträchtigte unsere Fähigkeit, alle potenziellen Probleme vor der Veröffentlichung zu erkennen. Diese Erfahrung machte deutlich, wie wichtig es ist, ein Gleichgewicht zwischen der Entwicklungsgeschwindigkeit und umfassenden Tests zu finden, und unterstrich die Notwendigkeit effizienter Testverfahren innerhalb eines begrenzten Zeitrahmens.

**4.Risiko:** Der Spaßfaktor des Spiels wird in der Praxis nicht so interessant sein, wie wir denken, und wir haben möglicherweise ein langweiliges Spiel gemacht.

**Vorsicht:** Wir werden das Spiel von ein paar Leuten außerhalb des Teams testen, deren Feedback einholen und wenn uns der Unterhaltungsfaktor nicht ausreicht, werden wir es mit verschiedenen Mechaniken anreichern.

2. Es ist ein Risiko eingetreten. Aufgrund der Menge an Ressourcen, die wir für die Entwicklung des Spiels bereitstellen mussten, konnten wir keine Ressourcen für die visuelle Darstellung des Spiels bereitstellen. Aus diesem Grund haben wir für die Erstellung dieser Version des Spiels vorgefertigte Texturen verwendet, die für unsere Nutzung lizenziert wurden oder für die wir eine schriftliche Genehmigung des Herstellers erhalten haben.

## Technischer Entwurf als OOM

In diesem Kapitel erklären wir die technischen Details unserer geplanten Lösung. Wir greifen die Architektur unseres Systems auf und erläutern den technischen Entwurf. Dieser ist von zentraler Bedeutung, da er die Vorlage für die Entwicklung unseres Projekts darstellt. Im Zentrum steht dabei die Erfüllung der Anforderungen.

Das objektorientierte Modell (OOM) stellt ein Softwaresystem als eine Sammlung von miteinander interagierenden Objekten dar. Diese Objekte sind Instanzen von Klassen und repräsentieren reale oder abstrakte Entitäten im System. Das objektorientierte Modell reduziert die Komplexität von Software, indem es ermöglicht, den Fokus auf die Strukturierung von Daten und Funktionalitäten zu legen.

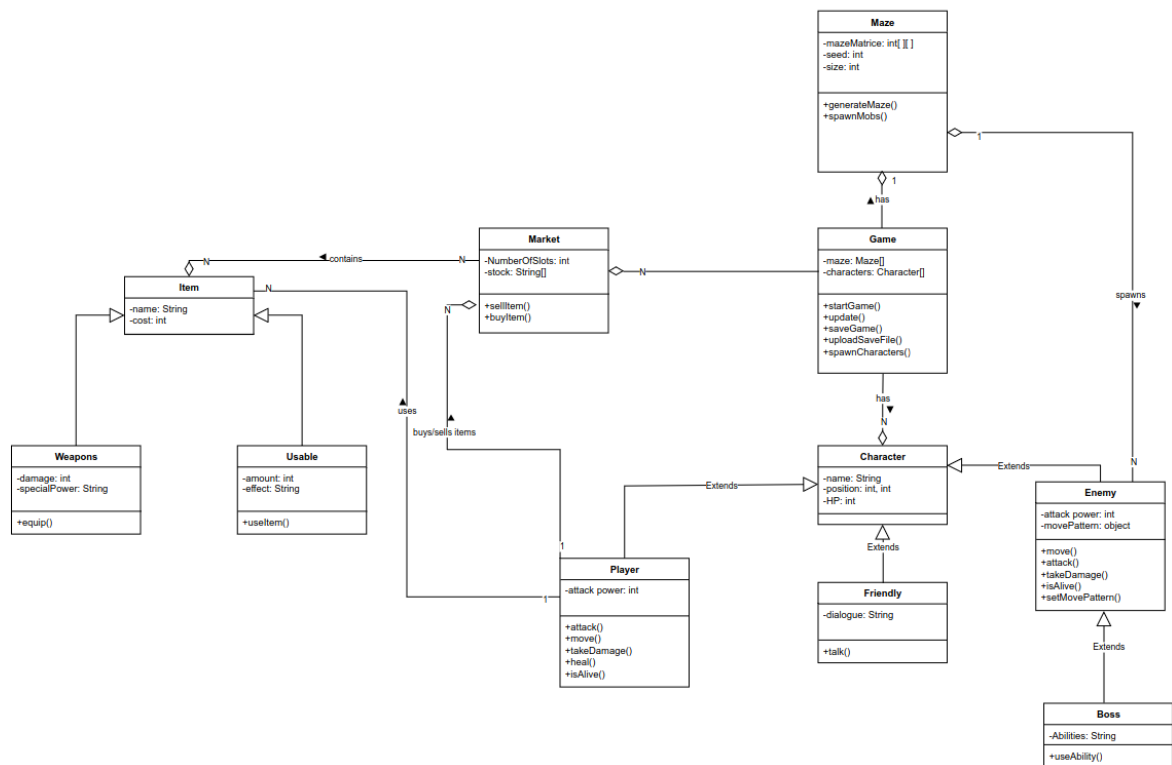


Abb. 10: UML Diagramm

## Qualitätssicherung – Testing

### A. Testplanung

Der Software-Test wird auf Komponententest durchgeführt. Der Systemtest ist ein Black Box-Test. Wir haben kritische funktionen zum Beständigkeit der Spiel gewählt und getestet. Die testgetriebene Entwicklung der Unit-Testfälle und der Komponenten erfolgt in der Godot Engine unter Verwendung des GUT(Godot Unit Test-Framework).

MazeGen\_RTM Table

Integration _Test_ID	Description	Vorbedingungen	Testschritte	Erwartete Ergebnis	Erreichte tes Ergebnis	Pass/Fail	Verf.Use Case
#1	Haupt menu zu Village	-	1-Öffne den Spiel.2-Clicke zum new game oder continue options.	Das Spiel wechselt zur Dorfszene.	Spieler ist im Dorf.	Pass	-
#2	Village to Dungeon	Spieler steht auf der Village	1- Betrete den Eingang des Dungeons vom Dorf aus.	Das Spiel wechselt zur Dungeonszene.	Spieler ist im Dungeo n.	Pass	-
#3	Dungeon to Village	Spieler sind in dem Dungeon	1- Verlasse den Dungeon über den Dorfausgang.	Das Spiel wechselt zur Dorfszene.	Spieler ist im Dorf.	Pass	-

Abb.11: MazeGen\_RTM Table

- Die Test Case Tabelle sollte hier enthalten sein, aber da die Größe der Tabelle eine Platzierung auf der Seite verhindert, wurde sie als Anhang am Ende dieses Dokuments hinzugefügt.

## Ergebnisse



Abb. 12: ein Bildschirmfoto aus dem Spiel



Abb. 13: Angriffsanimation

Unser Ziel war es zunächst, ein vollständiges und großes Spiel zu schaffen, das umfangreiche und verschiedene Minispiele beinhaltet, deren visuelle Komponenten wir selbst entwickelten. Allerdings war es für alle in unserem Team der erste Spielentwicklungsprozess und es fehlte uns an Erfahrung. Wir wussten nicht, was wir tun konnten und was nicht. Das Endprodukt lag unter unseren ursprünglichen Vorstellungen.

Die Entwicklung des Labyrinth-Generierungsalgorithmus war einfacher als erwartet, aber wir hatten viel größere Schwierigkeiten als gedacht bei der Entwicklung der Nebenkompontenten, die wir geplant hatten, um die Vielfalt des Spiels zu erhöhen.

Außerdem haben wir bei der Auswahl der Version den Fehler gemacht zu glauben, dass die neueste die beste sei. Wir sind ein unerfahrenes Team und müssen lernen, bevor wir Dinge tun, aber alle Beispiele in der Literatur, auf die wir Zugriff haben, wurden für Godot 3.x erstellt und die Syntax von GDScript hat sich in Godot 4.x geändert. Aus diesem Grund hatten wir Schwierigkeiten, Ressourcen zu finden und die Beispiele zu verstehen, und wir hatten Probleme, mit ihnen umzugehen und die Syntax anzupassen. Unser Team hatte in diesem Teil die größten Schwierigkeiten, aber wir wollten ihn nicht ändern, da wir ihn am Anfang geschrieben und spezifiziert hatten und das Problem gelöst hatten.

Selbst wenn wir alle anderen Komponenten unseres Spiels entfernen, verspricht das Lösen der Labyrinth und das ständige Sammeln neuer Erfahrungen ein unterhaltsames Erlebnis für den Spieler. Wir glauben, dass wir den Wiederspielwert erreicht haben, den wir als Hauptmerkmal unseres Spiels festgelegt haben, und wir sind stolz darauf.

In Zukunft kann das Produkt durch die Entwicklung neuer Minispiele und neuer Feinde diversifiziert werden. Artikel können dem Markt hinzugefügt werden. Durch das Hinzufügen einer Charakteroption können verschiedene Spieltypen gefördert werden. Damit ist das von uns nun erstellte Produkt der erste Prototyp eines Spiels, das in seine endgültige Form weiterentwickelt werden kann.

## Zusammenfassung

Von Anfang des Projekts an war es unser Ziel, ein eigenständiges Spiel zu entwickeln, das die Dynamik von Plattformen und komplexen Labyrinthen nahtlos in die Godot Engine integriert. Wir haben mit der Implementierung zentraler Gameplay Elemente wie der dynamischen Erstellung von Labyrinthen und dem interaktiven Verhalten der Charaktere begonnen und dabei erhebliche Fortschritte gemacht, obwohl wir mit natürlichen Herausforderungen konfrontiert waren.

Die Zeitdruck schränkte die Implementierung fortgeschrittener Funktionen und die Verbesserung der Plattformkompatibilität ein und erforderte eine sorgfältige Prioritätensetzung, um ein Gleichgewicht zwischen strengen Tests und der Entwicklungsgeschwindigkeit zu wahren.

Trotz dieser Herausforderungen sind wir stolz darauf, eine Kernfunktionalität des Spiels erreicht zu haben, die den Spielern ein fesselndes und differenziertes Spielerlebnis verspricht. Die Fähigkeit unseres Teams, sich den Herausforderungen anzupassen und flexibel zu sein, hat sich positiv auf die Entwicklung des Projekts ausgewirkt.

Mit Blick auf die Zukunft erkennen wir die Notwendigkeit von Verbesserungen bei der Erweiterung der Plattformkompatibilität, bei der Verfeinerung der Charakterentwicklungsfunktionen und bei der Berücksichtigung herausragender Designaspekte. Die nächsten Schritte beinhalten eine kontinuierliche Verpflichtung zu regelmäßigen Updates, die Berücksichtigung von Nutzerfeedback und die iterative Verfeinerung des Spiels. Ein wichtiger Aspekt unserer Verbesserungsstrategie ist das gründliche Testen, um die Robustheit und Qualität künftiger Versionen zu gewährleisten. Dieser iterative Prozess stellt sicher, dass das Projekt seinen Weg in Richtung eines noch fesselnderen und ausgefeilteren Indie-Spielerlebnisses fortsetzt.

### Persönliche Zusammenfassungen:

Mahmutcan: Spiele zu entwickeln war für mich eine neue Erfahrung, aber es war ein Kindheitstraum. Ich liebe es, Spiele zu spielen, und ich wollte schon immer auf der Entwicklungsseite sein. Dank dieses Projekts habe ich diese Chance bekommen. Es war eine schwierigere und umfassendere Aufgabe, als ich dachte, aber ich denke, dass sich die Mühe angesichts der gesammelten Erfahrung gelohnt hat.

Oğuzhan: Wir wollten mithilfe künstlicher Intelligenz visuelle Elemente erstellen, konnten dies jedoch aufgrund der Schwierigkeit der Animationen nicht erreichen, ansonsten hätten wir keine größeren Probleme. Auch wenn es stressig war, weil wir unsere Zeit nicht richtig nutzen konnten, denke ich, dass wir es erfolgreich geschafft haben.

Ramadan: Ursprünglich hatten wir geplant, ein Labyrinthspiel zu erstellen, das sich durch verschiedene Level bewegt und mit der Zeit wächst. Wir konnten erfolgreich ein Labyrinthspiel entwickeln, das in jedem Level einzigartig ist. Das Hinzufügen von Feinden, Gegenständen, Märkten und einem Statistiksistem, das wir später in Betracht gezogen hatten, hat sich jedoch als nicht besonders nützlich erwiesen. Das Zusammenfügen der von uns erstellten Spielszenen war für uns ziemlich schwierig. Wir haben versucht, die Szenen manuell einzeln hinzuzufügen, um das Problem zu lösen. In Zukunft könnte das Kampfsystem verbessert und ein Marktsystem hinzugefügt werden.

Barkan: Ich war nicht nur begeistert davon, Computerspiele zu spielen, sondern auch sehr interessiert an der Spieleentwicklung, daher war ich sehr enthusiastisch für das Projekt. Ich hatte zuvor einmal eine Spiel-Demo erstellt, aber sie war mit der Unity-Spiel-Engine gemacht, und da ich Godot nicht kannte, hatte ich einige Schwierigkeiten. Obwohl wir am Anfang viele Ideen hatten, haben wir beim Umsetzen gekämpft. Wir haben zusammen gelernt, während wir vorankamen. Als wir dem Abschluss des Projekts näher kamen, habe ich erkannt, wie wichtig die Planung ist. Trotz aller Schwierigkeiten sind wir nun am Ende angelangt. Dies war ein Start für unsere zukünftigen Projekte.

İdil: Das erste Mal mit Godot in die Spielentwicklung einzusteigen, war eine wertvolle Erfahrung. Als Neuling habe ich Einblicke in die technischen Aspekte gewonnen, indem ich zum Algorithmus für die Labyrinthgenerierung beigetragen habe. Diese erste Erfahrung hat nicht nur mein Verständnis erweitert, sondern mir auch wichtige Lektionen über effektive Projectplanung vermittelt.

Dilara: Ich hatte noch nie zuvor im Gaming-Bereich gearbeitet, aber es war ein Bereich, auf den ich neugierig war. Dank dieses Projekts hatte ich die Möglichkeit, diese Gegend zu erleben. Zu Beginn des Projekts hatte ich es etwas schwer, weil ich keine Kenntnisse hatte. Aber mit Hilfe der Videos, die ich mir angeschaut habe, und des Teamleiters konnten wir meiner Meinung nach einen guten Job machen.

## Individuelle Performance

Name	Punkte
Mahmutcan	90
Oğuzhan	80
İdil	80
Dilara	65
Ramadan	80
Barkan	85



Items (16)																			
Items (16)	Typ	Priorität	Status	Beschreibung / Name		Sprint		Beschreibung (lang)											
id-8	Epic	hoch	abgeschlossen	Entwerfung von ein procedural Maze Generator Skript		Sprint		1. Entwerfen des Algorithmus, dessen Labyrinth procedural generiert werden und integrieren dieser in das Spiel. Das Labyrinth realistischer machen und zusätzliche Funktionen hinzufügen (Mobs, Events).											
id-9	Epic	hoch	abgeschlossen	Player/Npc Design und Implementation				1. (bewegen, angreifen). Entwerfen von NPC's, die dem Spiel hinzugefügt werden sollen, und Interaktion mit der Hauptfigur, falls vorhanden.											
id-10	Epic	mittel	abgeschlossen	Mobs Development				2. die Erstellung von Mobs und die Wahl sinnvoller Spawnpunkte, Gesundheits- und Schadenwerte zum Zeitpunkt des Angriffs.											
id-11	Task	mittel	wartend	Bösa				2. Die Statistiken und Fähigkeiten des Bösen bestimmen, sein Erscheinungsbild entwerfen und alles codieren.											
id-12	Task	niedrig	wartend	Räumen				Die Gestaltung der Räume, auf die der Spieler zwischen den Labyrinthblöcken trifft.											
id-13	Task	niedrig	wartend	Market/Economy System Entwicklung und Development				3. Bestimmen Sie die Spawn-Raten der Räume. Beispiel: Storeraum (alle 5-10 Stockwerke), Markt (alle 10-15 Stockwerke), Item-Drop-Raum (alle 10-15 Stockwerke) usw.											
id-14	Task	niedrig	wartend	Design und Development von Items				4. Den Markt im Spiel gestalten. Ermittlung von Verkaufs- und Einkaufspreisen von Waren.											
id-15	langfristig	wartend	wartend	Soundeffekte und Music				4. Bestimmung ihrer Seltbarkeit.											
id-16	langfristig	niedrig	wartend	Story-Panel				5. Erstellung der Soundeffekte und Musik des Spiels											
id-0	Task	hoch	abgeschlossen	Mission 1 - Motivation		-		5. Erstellen von Zeichensprossen, um die Geschichte des Spiels weitzuzumitteilen und sie dem Spieler zu präsentieren.											
id-1	Task	hoch	abgeschlossen	H1: Gruppenbildung und Projektkläre		-		Es wurde eine Dokumentation erstellt, die die Motivation des Projekts und seine Grenzen enthält. (infinite Maze - Projektdokumentation.pdf)											
id-2	Task	mittel	abgeschlossen	Abschluss der freecodecamp Godot-Schulung		-		Marktpf: Infinite_Maze-Hausaufgabe_1-Gruppenbildung_Projekt.md											
id-3	Task	mittel	abgeschlossen	Literaturbericht zu prozeduralen Spielen, die mit Godot erstellt wurden.		-		Wir haben einen Kurs besucht, um zu lernen, wie man mit Godot arbeitet.											
id-4	Task	mittel	abgeschlossen	Untersuchung von Algorithmen zur Labyrinthgenerierung		-		Wir haben unser Wissen über Godot erweitert, indem wir uns angesehen haben, wie verschiedene Arten von Spielen von verschiedenen Leuten auf YouTube entwickelt wurden.											
id-5	Task	hoch	abgeschlossen	Mission 2 - Projektplan		-		Wir haben verschiedene Algorithmen zur Labyrinthherstellung untersucht und uns für die Verwendung des rekursiven Backtrack-Algorithmus entschieden.											
id-6	Task	mittel	abgeschlossen	Main-Menu		-		Planung des Entwicklungsteils des Projekts.											
id-7	Task	mittel	abgeschlossen	Save-Funktion		-		Programmieren des Startmenüs des Spiels											
id-8	Epic	niedrig	abgeschlossen	Ton und Music		-		Programmieren der Save-Funktion, die es ermöglicht, das Spiel dort fortsetzen, wo es aufgehört hat.											
								Ton und Music											
								Die Produzenten von Undertale wurden kontaktiert und die Erlaubnis zur Nutzung ihrer Vermögenswerte eingeholt.											

## Sprint 1

Name:	Level 1: Labyrinth + Hauptfigur			
Projekt Ziel:	Die Entstehung der primitivsten Version des Spiels in spielbarer Form.	Dauer:	5 Arbeitstage	<div><div></div><div></div><div></div></div>
Prozess Verbesserung:	-	Items:	16	
Start Datum:	27.11.2023	Story Points:	0	
End Datum:	03.12.2023			
Sprint beendet				

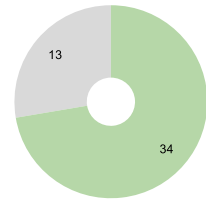
### Sprint Metrik

Aufgaben geplant:	16	Key Achievements:	- Maze Generator
Aufgaben abgeschlossen:	12		- Dorf
Aufgaben übrig:	4		- Hauptcharakter
Story Points geplant:	47	Zusammenfassung Performance:	
Story Points erledigt:	34		
Story Points übrig:	13		

## Sprint Backlog

[illegible]

## Stand Story Points



## Sprint 2

## Stand Story Points

Name:	Level 2: Boss + Mobs			
Projekt Ziel:	Die Entstehung der primitivsten Version des Spiels in spielbarer Form	Dauer:	10 Arbeitstage	
Prozess Verbesserung:	-	Items:	12	<div><div></div><div></div><div></div></div>
Start Datum:	04.12.2023	Story Points:	0	
End Datum:	15.12.2023			Sprint beendet



## Sprint Metrik

Aufgaben geplant:	12	Key Achievements:	-Boss
Aufgaben abgeschlossen:	12		-Mobs
Aufgaben übrig:	0		
Story Points geplant:	38	Zusammenfassung Performance:	
Story Points erledigt:	38		
Story Points übrig:	0		

## Sprint Backlog

[illegible]

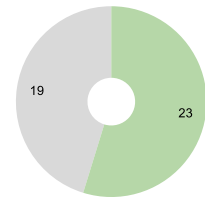
### Sprint 3

<b>Name:</b>	Level 3: Räume			
<b>Projekt Ziel:</b>	Das Spiel aus seiner spielbaren primitiven Form herauszu	<b>Dauer:</b>	8 Arbeitstage	
<b>Prozess Verbesserung:</b>	-	<b>Items:</b>	15	■ ■ ■
<b>Start Datum:</b>	15.12.2023	<b>Story Points:</b>	0	
<b>End Datum:</b>	26.12.2023			<b>Sprint beendet</b>

## Sprint Metrik

Aufgaben geplant:	15	Key Achievements:	-Räume
Aufgaben abgeschlossen:	8		
Aufgaben übrig:	7		
Story Points geplant:	42	Zusammenfassung Performance:	
Story Points erledigt:	23		
Story Points übrig:	19		

## Stand Story Points



## Sprint Backlog

[illegible]


Sprint 1				
Name:	Gib Deinem Sprint einen coolen Namen!			Noch
Projekt Ziel:	Das große Ziel für diesen Sprint.	Dauer:	10 Arbeitstage	3
Prozess Verbesserung:	Die Verbesserung des Arbeitsprozesses.	Items:	0	
Start Datum:	26.12.2023	Story Points:	0	
End Datum:	08.01.2024			Tage übrig

<b>Name:</b>	Gib Deinem Sprint einen coolen Namen!
<b>Projekt Ziel:</b>	Das große Ziel für diesen Sprint.
<b>Prozess Verbesserung:</b>	Die Verbesserung des Arbeitsprozesses.
<b>Start Datum:</b>	26.12.2023
<b>End Datum:</b>	08.01.2024

Dauer:	10 Arbeitstage
Items:	0
Story Points:	0

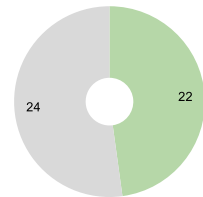
Noch  
**3**  
 Tage übrig

### Stand Story Points



A donut chart titled "Stand Story Points" showing two segments. The green segment on the right is labeled "22" and represents approximately 44% of the total. The grey segment on the left is labeled "24" and represents approximately 48% of the total. The chart has a white center.

Category	Value
Green Segment	22
Grey Segment	24



## Sprint Metrik

Aufgaben geplant:	20	Key Achievements:	Hier stehen die wichtigsten Dinge, die ihr in diesem Sprint erreicht habt. Das wird gemeinsam in der Sprint Retrospektive festgelegt.
Aufgaben abgeschlossen:	10		
Aufgaben übrig:	10		
Story Points geplant:	46	Zusammenfassung Performance:	Hier wird kurz aufgeschrieben, was in diesem Sprint gut gelaufen ist und was nicht. Das wird ebenfalls gemeinsam in der Sprint Retrospektive festgelegt.
Story Points erledigt:	22		
Story Points übrig:	24		

## Sprint Backlog

[illegible]