



## **Projektplan – Projekt Software Engineering**

<b>Erstellt von:</b>	Jan Wunderlich
<b>Matrikelnummer:</b>	92205002
<b>Studiengang:</b>	Master of Science Informatik
<b>Tutor:</b>	Prof. Dr. Markus Kleffmann
<b>Datum:</b>	25.12.2023

## **Inhaltsverzeichnis**

1	Ziele, Umfang und angestrebtes Ergebnis des Projekts .....	3
2	Anvisierte Zielgruppe.....	5
3	Potenzielle Projektrisiken und Gegenmaßnahmen .....	6
4	Zeitplan und Meilensteine .....	7

# 1 Ziele, Umfang und angestrebtes Ergebnis des Projekts

Das Ziel des Projekts besteht in der Entwicklung eines sogenannten Endless-Runner- oder Infinite-Runner-Spiels. In diesem Genre übernimmt der Spieler die Kontrolle über eine Figur, die ununterbrochen läuft, und versucht, so lange wie möglich zu überleben. Dies erreicht er, indem er geschickt sämtlichen Hindernissen ausweicht und auftauchende Gegner bezwingt. Der Spieler verliert, sobald die Spielfigur von einem Hindernis oder Gegner getroffen wird. In diesem Moment endet das Spiel bzw. der aktuelle Versuch. Folgende Komponenten sind integraler Bestandteil des Projekts und sollen in der finalen Anwendung nahtlos integriert sein:

## 1. Grundgerüst:

- ☐ Initialisierung des Spiels durch Pygame, Erschaffung eines Fensters und Laden aller erforderlicher Ressourcen, einschließlich Grafiken für Hintergrund, Spieler, Gegner, Power-Ups und mehr.
- ☐ Das Design soll über alle Spielelemente hinweg, wie den Hintergrund, die Spieler, die Gegner und das Menü im Cyberpunk-Style gehalten werden.
- ☐ Definition von Kernparametern wie die Scrollgeschwindigkeit des Hintergrunds, die Sprunghöhe des Spielers, die Schussgeschwindigkeit und die Häufigkeit von Hindernissen oder Gegnern.

## 2. Spieler:

- ☐ Spielerbewegungen einschließlich links, rechts, springen und ducken oder sliden ermöglichen.
- ☐ Umsetzung der Schussmechanik, wobei die Möglichkeit geschaffen wird, mehrere Kugeln gleichzeitig abzufeuern, welche ein Spieler mithilfe eines Power-Ups freischalten kann.

## 3. Gegner:

- ☐ Zufällige Erzeugung von Gegnern auf der rechten Seite des Bildschirms.
- ☐ Definition und Implementierung von unterschiedlichen Fähigkeiten für Gegner wie Laufen, Springen oder das Abfeuern von Waffen.

## 4. Power-Ups und Items:

- ☐ Zufällige Generierung von Power-Ups, die der Spieler einsammeln kann.
- ☐ Power-Ups bieten verschiedene Effekte, wie ein zusätzliches Leben oder verbesserte Waffen.

## 5. Spiellogik und Fortschrittsanzeige:

- ☐ Anzeige von zurückgelegten Metern sowohl während des Spiels als auch im Endbildschirm mithilfe eines Meterzählers.
- ☐ Umsetzung einer Highscore-Verwaltung, um Spielerergebnisse zu speichern und zu aktualisieren.

- ☐ Spielende-Logik, um ein Spiel zu beenden, wenn der Spieler von einem Gegner getroffen wird oder gegen ein Hindernis läuft (Kollisionserkennung).
6. Startbildschirm und Optionen:
- ☐ Hauptmenü-Bildschirm, über den der Spieler das Spiel starten kann.
  - ☐ Integration von Optionen zur Steuerung der Musik und Soundeffekten.
7. Benutzerinteraktion:
- ☐ Tastatursteuerung, um Interaktion mit der Spielfigur zu realisieren.
  - ☐ Maussteuerung für die Navigation im Spielmenü.
8. Spielneustart, -pause und -beendigung:
- ☐ Option zum Neustart des Spiels nach dem Tod des Spielers.
  - ☐ Spieler können Spiel pausieren und zum Hauptbildschirm zurückkehren.

Diese Komponenten bilden den groben Umfang des zu entwerfenden Endless-Runner-Spiels und können im Laufe des Projekts, sollte ein entsprechender Bedarf erkannt werden, ggf. um weitere Elemente ergänzt werden. Das angestrebte Ergebnis dieses Software Engineering Projekts ist die Entwicklung einer voll funktionsfähigen, ansprechenden und unterhaltsamen Endless-Runner-Anwendung, die alle definierten Komponenten umfasst sowie dem Nutzer ein fesselndes Spielerlebnis beschert.

## **2 Anvisierte Zielgruppe**

Bei der Definition der Zielgruppe für das Endless-Runner-Spiel sind mehrere entscheidende Überlegungen anzustellen, die maßgeblich in die Planung und Entwicklung des Spiels einfließen. Neben der Altersgruppe, die mit dem Spiel angesprochen wird, spielen auch die Interessen und Vorlieben der Nutzer eine besonders wichtige Rolle. Ebenso darf die gewünschte Spielerfahrung nicht außer Acht gelassen werden.

Obwohl das Endless-Runner-Spiel grundsätzlich von einem vielfältigen Publikum gespielt werden kann, richtet es sich primär an Gamer und Personen, die zumindest gelegentlich Computer- oder Videospiele nutzen. Die Hauptzielgruppe sind folglich Jugendliche und junge Erwachsene. Die Spielmechanik und -logik werden bewusst so gestaltet, dass das Spiel leicht zugänglich ist, aber dennoch eine gewisse Herausforderung bietet. Eine progressiv zunehmende Schwierigkeit ermöglicht es sowohl Einsteigern als auch erfahrenen Spielern, sich auf ihrem eigenen Niveau zu verbessern und Fortschritte zu erzielen.

Das im Cyberpunk-Style gehaltene Design des Spiels spricht darüber hinaus insbesondere Science-Fiction-Fans an. Eine futuristische Ästhetik, gepaart mit innovativen Spielelementen, bietet eine einzigartige Spielerfahrung. Die spielerischen, visuellen und inhaltlichen Merkmale der Anwendung werden dabei gezielt ausgerichtet, um die Interessen der beschriebenen Zielgruppe bestmöglich zu bedienen.

### 3 Potenzielle Projektrisiken und Gegenmaßnahmen

Für die Realisierung eines adäquaten Risikomanagements sind drei aufeinanderfolgende Schritte erforderlich. Zunächst erfolgt im Rahmen der Risikoidentifikation eine umfassende Analyse, um potenzielle Risiken während der Projektdurchführung zu identifizieren. Anschließend erfolgt eine Risikobewertung, die unter anderem die angenommene Eintrittswahrscheinlichkeit sowie die möglichen Auswirkungen des Risikos auf Kosten oder Ergebnisqualität ermittelt. Der letzte Schritt besteht in der Maßnahmenplanung, bei der geeignete Gegenmaßnahmen zur Prävention oder Korrektur der Risiken entwickelt werden. Die Ergebnisse dieser Analyse werden in einem Risikoinventar zusammengefasst, das in der nachfolgenden Tabelle abgebildet ist. Die Risikokennzahl setzt sich dabei aus dem Produkt der Eintrittswahrscheinlichkeit und der Schadenshöhe des jeweiligen Risikos zusammen. Die Gegenmaßnahmen sind in 2 Arten, nämlich präventive (p) und korrektive (k) Maßnahmen, untergliedert.

Name des Risikos	Tragweite	Schadenshöhe (1-5)	Eintrittswahrscheinlichkeit (1-5)	Risikokennzahl	Maßnahmen (k oder p)
<b>Technische Komplexität</b>	Schwierigkeiten bei der Implementierung	4	3	12	Entwicklungsprozess in kleine Komponenten unterteilen (p), Funktionalitäten priorisieren und ggf. Abstriche machen (k)
<b>Ressourcenknappheit</b>	Verzögerungen bei Implementierung oder Dokumentation	2	2	4	Effiziente Planung und Priorisierung der Aufgaben (p), Projektdauer verlängern (k)
<b>Unvorhergesehene Änderungen im Projektumfang</b>	Zusätzlicher Arbeitsaufwand und Zeitbedarf	3	4	12	Detaillierte Anforderungsanalyse (p), flexible Anpassung der Anforderungen / Aufgaben (k)
<b>Qualitätsprobleme</b>	Fehler in der Anwendung und ggf. Leistungseinbußen	2	2	4	Einzelnen Funktionen testen (p), intensive Fehleranalyse und -korrektur (k)
<b>Ungenügende Planung</b>	Unklare Ziele und Projektverzögerungen	3	2	6	Durchführung einer gründlichen Projektplanung (p), flexibles Projektmanagement (k)
<b>Ungenügende Testabdeckung</b>	Unentdeckte Fehler und Qualitätsprobleme	4	1	4	Festlegung klarer Testziele und -kriterien (p), Erweiterung der Testabdeckung (k)

Tabelle 1: Risikoinventar

## **4 Zeitplan und Meilensteine**

Die Erstellung eines klaren und verständlichen Zeitplans mitsamt Meilensteinen ist entscheidend für den Erfolg eines jeden Projekts. Um diese Komplexität zu bewältigen und den zeitlichen Ablauf optimal zu planen, wird in diesem Software Engineering Projekt auf ein Gantt-Diagramm zurückgegriffen. Dieses visuelle Instrument ermöglicht nicht nur eine übersichtliche Präsentation der Projektzeitleiste, sondern auch eine gezielte Identifikation von kritischen Meilensteinen sowie Abhängigkeiten und ist aus Übersichtlichkeitsgründen auf der folgenden Seite im Querformat abgebildet.

Bei genauerer Betrachtung des Gantt-Diagramms wird deutlich, dass für die Implementierung des Quellcodes in der Erarbeitungs- und Reflexionsphase zwar eine grobe Zeitspanne vorgesehen ist, jedoch keine spezifischen Startdaten, Dauerangaben oder Priorisierungen für einzelne Komponenten festgelegt sind. Dies resultiert aus der Entscheidung für ein agiles Vorgehensmodell in der Entwicklung, bei dem die exakte Reihenfolge und Priorisierung der Funktionen innerhalb des geplanten Zeitrahmens erst im Verlauf des Projekts festgelegt wird. Weitere Details zu diesem Vorgehensmodell werden in der Projektdokumentation ausführlich erläutert.

## Projekt: Software Engineering

### Endless-Runner-Game

Projektanfangsdatum: 13.11.23

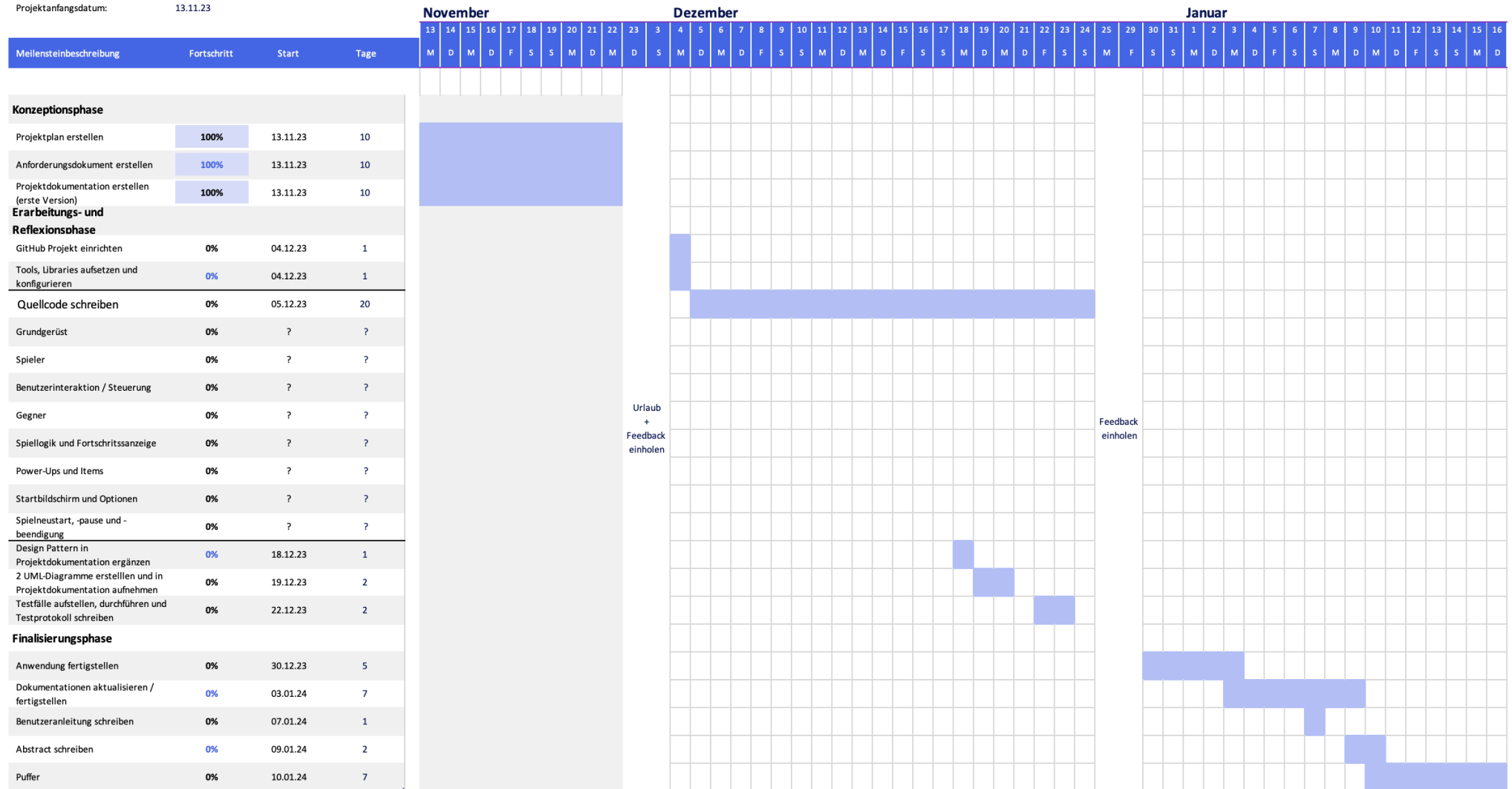


Abbildung 1: Zeitplanung als Gantt-Diagramm