



## **Anforderungsdokument – 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	Management Summary .....	3
2	Systemumfang und Kontext .....	4
3	Funktionale Anforderungen .....	5
4	Nicht-funktionale Anforderungen .....	9
5	Glossar .....	10

## 1 Management Summary

Dieses Software Engineering Projekt konzentriert sich auf die Entwicklung eines fesselnden Endless-Runner-Spiels, bei dem der Spieler das Ziel verfolgt, eine möglichst große Distanz zu überwinden, ohne von Hindernissen oder Gegnern getroffen zu werden. Das Herzstück des Spiels liegt in seinem ansprechenden Spielerlebnis, welches die Nutzer dazu ermutigt, ihre eigenen Fähigkeiten zu verbessern und ihren Highscore immer wieder zu übertreffen.

Die Funktionalitäten des Spiels erstrecken sich über verschiedene Aspekte. Der Spieler hat die Möglichkeit, die Spielfigur präzise zu steuern, geschickt Hindernissen auszuweichen und Gegner durch den Einsatz einer Waffe zu besiegen. Das Sammeln unterschiedlichster Power-Ups während des Spiels eröffnet dem Spieler zusätzliche Strategien und Möglichkeiten, seinen aktuellen Versuch zu optimieren. Während eines laufenden Versuchs erhält der Spieler kontinuierlich Rückmeldung über seinen Fortschritt, angezeigt in Form der zurückgelegten Meter und gesammelten Punkte. Diese transparente Darstellung ermöglicht es dem Spieler, seine Leistung zu überwachen und den Anreiz zu schaffen, stetig besser zu werden.

Abseits des eigentlichen Spielgeschehens bietet das übersichtliche Hauptmenü dem Spieler eine Vielzahl von Interaktionsmöglichkeiten. Hier kann er nicht nur seine letzten Läufe und Statistiken einsehen, sondern auch die Lautstärke von Ton und Musik anpassen. Zusätzlich hat der Spieler die Option, gesammelte Punkte gegen temporäre Power-Ups einzutauschen, die über mehrere Runden hinweg wirksam sind. Ebenso startet der Spieler von hier unkompliziert ein neues Spiel respektive einen neuen Versuch.

In Ergänzung zu den funktionalen Anforderungen liegt ein besonderer Fokus auf nicht-funktionalen Aspekten wie Leistung, Zuverlässigkeit, Ladezeiten und Benutzeroberfläche. Das Ziel ist es, den Nutzern eine umfassende und zufriedenstellende Spielerfahrung zu bieten, die durch hohe Qualität in der Benutzerinteraktion, eine intuitive Benutzeroberfläche und reibungslose Performance geprägt ist. Die Entwicklung zielt darauf ab, ein durchweg ansprechendes und unterhaltsames Spielerlebnis zu garantieren, das den Erwartungen der Spieler gerecht wird.

## 2 Systemumfang und Kontext

Die Anwendung eines UML-Anwendungsfalldiagramms stellt eine effektive Methode dar, um die Abgrenzungen eines Systems zu definieren und den Umfang des Systems auf prägnante Weise zu veranschaulichen. Die klare visuelle Strukturierung von Akteuren und deren Interaktion mit den verschiedenen Anwendungsfällen schafft Transparenz über die Funktionen und Möglichkeiten, die das System bietet. In der nachfolgenden Abbildung ist daher ein UML-Anwendungsfalldiagramm dargestellt, das die Kernaspekte des zu entwerfenden Endless-Runner-Spiels umfasst und dementsprechend Auskunft über die möglichen Aktionen, die der Spieler ausführen kann, gibt.

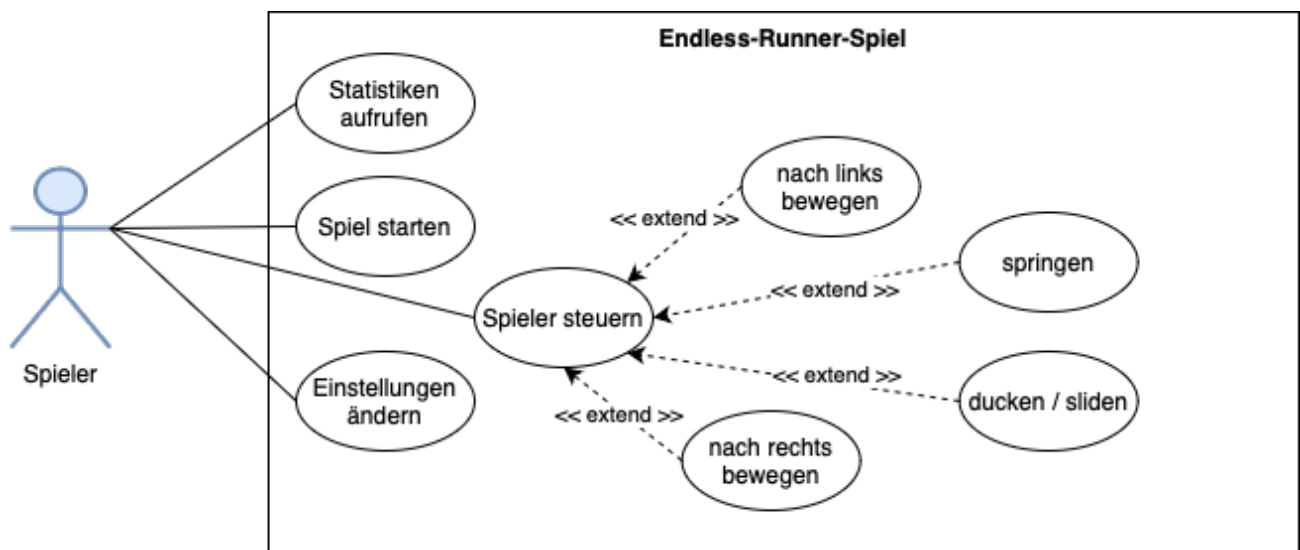


Abbildung 1: UML-Anwendungsfalldiagramm

In diesem UML-Anwendungsfalldiagramm repräsentiert das Endless-Runner-Spiel die klaren Grenzen des Systems. Der einzige Akteur ist der Spieler bzw. Nutzer, der durch die Interaktion mit dem System verschiedene Anwendungsfälle auslöst. Diese Anwendungsfälle umfassen "Statistiken aufrufen", "Einstellungen ändern" und "Spiel starten".

Der Anwendungsfall "Spieler steuern" nimmt eine zentrale Rolle ein, da er sämtliche Bewegungen des Spielcharakters, wie springen, ducken oder in eine Richtung laufen, koordiniert, und stellt daher die Schlüsselkomponente während der Spielausführung dar. Die Anwendungsfälle "Einstellungen ändern" und "Statistiken aufrufen" beschreiben zwei von mehreren Möglichkeiten bzw. Untermenüs, auf welche der User vom Hauptmenü aus zugreifen kann. Zusammengefasst ermöglicht dieses Anwendungsfalldiagramm eine prägnante Darstellung der Systemgrenzen. Es verdeutlicht auf einen Blick, wie der Spieler mit der Endless-Runner-Anwendung interagieren kann, welche Optionen im Menü für den Nutzer verfügbar sind und welche Aktionen im Kontext eines laufenden Spiels ausgeführt werden können.

### 3 Funktionale Anforderungen

Zur präzisen Beschreibung der funktionalen Anforderungen werden sogenannte User Stories verwendet, welche im Zuge des Projektmanagements häufig für diesen Zweck eingesetzt werden. User Stories konzentrieren sich auf spezifische Benutzerinteraktionen und Bedürfnisse, wodurch die Entwicklung unmittelbar auf die Anforderungen des Endnutzers ausgerichtet wird. Definierte Akzeptanzkriterien für jede User Story dienen als klare Maßstäbe zur Beurteilung der erfolgreichen Umsetzung, indem sie messbare Ergebnisse und klare Erwartungen festlegen. Die Einführung von Story Points ermöglicht zudem eine relative Schätzung des Aufwands jeder einzelnen User Story. Dieses Schätzverfahren unterstützt dabei, den Entwicklungsaufwand abzuschätzen und Prioritäten festzulegen. Die nachfolgenden Abbildungen repräsentieren die wichtigsten User Stories dieses Projekts in Form von Karten. Die Story Points reichen hierbei von eins bis fünf und spiegeln eine Bandbreite von geringstem bis höchstem Aufwand wider.

ID: 1	<b>Benutzeroberfläche anzeigen</b>
Als Spieler (Rolle) möchte ich eine Benutzeroberfläche für das Spiel sehen (Funktion), damit ich die Anwendung bedienen kann (Nutzen).	
Akzeptanzkriterien:	
<input type="checkbox"/> Fenster öffnet sich beim Starten der Anwendung	Story Points: 1
<input type="checkbox"/> Fenster besitzt festgelegte Größe und passenden Hintergrund	

Abbildung 2: User Story - Benutzeroberfläche anzeigen

ID: 2	<b>Spieler steuern</b>
Als Spieler (Rolle) möchte ich meinen Charakter bewegen können (Funktion), damit ich Hindernissen und Gegnern ausweichen kann (Nutzen).	
Akzeptanzkriterien:	
<input type="checkbox"/> Spielercharakter reagiert auf Tastatureingaben und bewegt sich	Story Points: 3
<input type="checkbox"/> Bewegung des Spielers wird durch Bildschirmgrenzen limitiert	

Abbildung 3: User Story - Spieler steuern

ID: 3

Gegner mit verschiedenen Attacken

Als Spieler (Rolle)  
möchte ich die Fähigkeiten meiner Gegner kennenlernen (Funktion),  
damit ich meine Strategie anpassen kann (Nutzen).

Akzeptanzkriterien:

- ☐ Gegner sind sichtbar und visuell einfach zu unterscheiden
- ☐ Gegner besitzen verschiedene Fähigkeiten (Springen, Schießen)

Story  
Points:  
4

Abbildung 4: User Story - Gegner mit verschiedenen Attacken

ID: 4

Funktionale Spiellogik

Als Spieler (Rolle)  
möchte ich das ein Spiel korrekt beendet und neugestartet wird (Funktion),  
damit ich mich verbessern kann (Nutzen).

Akzeptanzkriterien:

- ☐ Spiel wird bei Kollision mit Gegner oder Hindernis beendet
- ☐ Neustart startet Spiel / Versuch neu

Story  
Points:  
5

Abbildung 5: User Story - Funktionale Spiellogik

ID: 5

Fortschritt anzeigen

Als Spieler (Rolle)  
möchte ich einen Meterzähler sehen (Funktion),  
damit ich meinen Fortschritt verfolgen kann (Nutzen).

Akzeptanzkriterien:

- ☐ Meterzähler erhöht sich basierend auf zurückgelegter Strecke
- ☐ Korrekte Zurücksetzung bei Spielneustart

Story  
Points:  
1

Abbildung 6: User Story - Fortschritt anzeigen

ID: 6

## Hauptmenü anzeigen

Als Spieler (Rolle)  
möchte ich ein Hauptmenü sehen (Funktion),  
damit ich das Spiel starten oder Optionen auswählen kann (Nutzen).

### Akzeptanzkriterien:

- ☐ Hauptmenü wird bei Anwendungsstart angezeigt
- ☐ Spieler kann Spiel starten oder Einstellungen anpassen

Story  
Points:  
2

Abbildung 7: User Story - Hauptmenü anzeigen

ID: 7

## Power-Ups einsammeln

Als Spieler (Rolle)  
möchte ich Power-Ups einsammeln können (Funktion),  
damit die Verbesserung des Highscores leichter fällt (Nutzen).

### Akzeptanzkriterien:

- ☐ Power-Ups erscheinen zufällig auf dem Bildschirm
- ☐ Einsammeln eines Power-Ups schaltet jeweilige Funktion frei

Story  
Points:  
3

Abbildung 8: User Story - Power-Ups einsammeln

ID: 8

## Belohnungen für Leistungen

Als Spieler (Rolle)  
möchte ich Belohnungen für meine Leistungen erhalten (Funktion),  
damit ich Anreize für kontinuierliches Spielen habe (Nutzen).

### Akzeptanzkriterien:

- ☐ Spieler erhält Punkte oder Münzen für erreichte Distanz
- ☐ Spieler kann Münzen für Upgrades oder temporäre Power-Ups nutzen

Story  
Points:  
2

Abbildung 9: User Story - Belohnungen für Leistungen

ID: 9

## Waffe abfeuern

Als Spieler (Rolle)

möchte ich eine Waffe abfeuern können (Funktion),  
damit ich Gegner abschießen kann (Nutzen).

### Akzeptanzkriterien:

- ☐ Spieler kann Waffe per Tastendruck abfeuern
- ☐ Gegner verliert leben / verschwindet, wenn er getroffen wird

Story  
Points:

3

Abbildung 10: User Story - Waffe abfeuern



## 4 Nicht-funktionale Anforderungen

Während funktionale Anforderungen direkt mit bestimmten Features oder Funktionen in Verbindung stehen, beziehen sich nicht-funktionale Anforderungen auf andere Aspekte eines Systems wie qualitative Eigenschaften, Randbedingungen oder Leistungsmerkmale. Hier werden also keine Handlungen oder Ergebnisse definiert, sondern stattdessen wird beschrieben, wie das System bestimmte Qualitäten erfüllen soll. Nachfolgend sind die wichtigsten Punkte aufgelistet, die das Endless-Runner-Spiel, welches in diesem Software Engineering Projekt entwickelt wird, erfüllen soll.

- Leistung
  - Das Spiel sollte flüssig und reaktionsschnell laufen, ohne Verzögerungen oder Ruckeln, um ein optimales Spielerlebnis zu gewährleisten.
- Benutzeroberfläche
  - Die Benutzeroberfläche sollte benutzerfreundlich und intuitiv gestaltet sein, um eine einfache Interaktion und Navigation für Spieler zu ermöglichen.
- Zuverlässigkeit
  - Das Spiel sollte stabil und zuverlässig sein, ohne häufige Abstürze, Fehler oder Bugs, um eine positive Spielerfahrung sicherzustellen.
- Barrierefreiheit
  - Das Spiel sollte barrierefrei und für Spieler mit unterschiedlichen Fähigkeiten oder Erfahrungen geeignet sein.
- Plattformkompatibilität
  - Die Anwendung sollte sowohl unter Windows 10 als auch unter Windows 11 vollumfänglich und ohne Probleme funktionieren.
- Wartbarkeit
  - Der Quellcode des Spiels sollte gut dokumentiert und strukturiert sein, um zukünftige Wartungen und Aktualisierungen bzw. Anpassungen zu erleichtern.
- Ladezeiten
  - Das Spiel sollte schnelle Ladezeiten aufweisen, um die Wartezeit für die Spieler zu minimieren und ein reibungsloses Spielerlebnis zu realisieren.

## 5 Glossar

Begriff	Beschreibung / Erklärung
<b>Akzeptanzkriterien</b>	Klar definierte Kriterien, die erfüllt sein müssen, damit eine User Story als abgeschlossen bzw. erfüllt gilt.
<b>Cyberpunk</b>	Ein Genre in der Science-Fiction, das oft eine dystopische, technologisch fortschrittliche Welt mit starken visuellen Elementen betont.
<b>Endless-Runner</b>	Ein Spielgenre, bei dem der Spieler eine Figur steuert, die unendlich lange durch eine sich ständig verändernde Umgebung läuft.
<b>Gantt-Diagramm</b>	Eine visuelle Darstellung eines Projektzeitplans, die die zeitliche Abfolge von Aufgaben und Meilensteinen zeigt.
<b>Highscore</b>	Die höchste erreichte Punktzahl eines Spielers.
<b>Kollisionserkennung</b>	Die Fähigkeit des Spiels, Kollisionen zwischen verschiedenen Objekten im Spiel zu erkennen, wie z.B. der Spielerfigur und Hindernissen.
<b>Power-Ups</b>	Objekte oder Elemente im Spiel, die dem Spieler temporäre Vorteile oder Fähigkeiten verleihen.
<b>Story Points</b>	Ein Maß für den Aufwand, der benötigt wird, um eine bestimmte User Story abzuschließen.
<b>User Story</b>	Eine kurze, benutzerzentrierte Beschreibung einer Funktion oder eines Features aus der Sicht des Benutzers.