**Projektmappe des Projektes**

Swat Engagement Pheretima

**Dokumentation des Projektes**

Gruppe L:

Tarik Karaca

Julian Flieter

Jan Voigtländer

Ibrahim Kekec

Steve Nemangou

Inhaltsverzeichnis

[Projektbeschreibung 3](#_Toc513952264)

[Iteration I 5](#_Toc513952265)

[MSCs 10](#_Toc513952266)

[UML Klassendiagramm 12](#_Toc513952267)

[Funktionalitätsplanung 12](#_Toc513952268)

[Iteration II 14](#_Toc513952269)

[User-Stories 14](#_Toc513952270)

[MSCs 18](#_Toc513952271)

[UML Klassendiagramm 18](#_Toc513952272)

[Funktionalitätsplanung 21](#_Toc513952273)

[Unittests 24](#_Toc513952274)

[Systemtests 25](#_Toc513952275)

[Iteration III 27](#_Toc513952276)

[User-Stories 27](#_Toc513952277)

[MSCs 29](#_Toc513952278)

[UML Klassendiagramm 35](#_Toc513952279)

[Funktionalitätsplanung 36](#_Toc513952280)

[Unittests 38](#_Toc513952281)

[Systemtests 39](#_Toc513952282)

[Nutzerhandbuch 41](#_Toc513952283)

[Technische Anforderungen 41](#_Toc513952284)

[Installationsanleitung 41](#_Toc513952285)

[Bedienungsanleitung 41](#_Toc513952286)

## Projektbeschreibung

**Einleitung**

In dem zu entwickelnden Spiel sollen bis zu fünf Spieler gegeneinander spielen können, indem jeweils ein Spieler ein Team aus bewaffneten Würmern steuern. Das Ziel jedes Teams ist es dabei, alle anderen Teamgegner zu besiegen. Ein Team ist besiegt, wenn alle Würmer des Teams kampfunfähig sind, entweder weil sie zu viel Beschuss einstecken mussten, oder weil sie aus der Spielkarte herausgefallen sind. Das Spielgeschehen soll zweidimensional von der Seite gezeigt werden. Außerdem soll das Spiel an einem Computer gespielt werden, an dem die Spieler nacheinander jeweils eine Spielrunde spielen. In einer Spielrunde kann der Spieler, der am Zug ist, einen seiner Würmer bewegen und einmal eine Waffe mit diesem Wurm benutzen.

**Iteration I**

In der ersten Iteration soll eine erste Version des Spiels mit reduziertem Umfang entwickelt werden. Zunächst sollen nur zwei Spieler, die jeweils nur einen Wurm steuern, gegeneinander spielen. Die Landschaft der Spielkarte soll aus einer flachen Ebene bestehen, auf der die Würmer sich nach rechts und links bewegen und springen können. Jedem Spieler steht nur eine Waffe zur Verfügung, die Projektile abschießt. Vor dem Schuss soll der Spieler den Winkel bestimmen können, in dem die Waffe schießt.

**Iteration II**

In dieser Iteration wird das Grundgerüst des Spiels vervollständigt, vor allem sollen Wind und Gravitation für die Spielobjekte simuliert werden. Die Stärke und Richtung des Windes soll sich in jeder Runde zufällig verändern und so die Flugbahn der abgeschossenen Projektile beeinflussen. Außerdem sollen die Projektile und Würmer von einer simulierten Gravitation beeinflusst werden. Fallen Würmer unter eine bestimmte Grenze in der Spielwelt, werden sie kampfunfähig. Diese Grenze soll in jeder Runde leicht steigen. Das Spiel soll auf vier verschiedenen Karten, die sich im Aufbau ihrer Landschaft unterscheiden, gespielt werden können. Diese Landschaften sollen komplexer sein als eine Ebene. Sie sollen außerdem von den Waffen der Würmer zerstört werden können. Den Würmern soll eine Auswahl von drei verschiedenen Waffen zur Verfügung stehen, die jeweils andere Projektile verschießen. Die Projektile sollen sich in dem Schaden unterscheiden, den sie an getroffenen Würmern anrichten, sie sollen verschieden stark von Wind und Gravitation beeinflusst werden, sie sollen die Umgebung auf verschiedene Arten zerstören und sie sollen verschieden aussehen. Außerdem sollen die beiden Spieler jeweils ein Team aus bis zu fünf Würmern steuern. Die Spieler sind weiterhin abwechselnd dran und können weiterhin nur mit einem Wurm in ihrem Zug laufen und schießen. Wann immer ein Wurm stirbt, soll ein Replay der letzten Sekunden angezeigt werden. Zuletzte soll eine weitere Waffe eingeführt werden. Diese Waffe soll Würmern über den Verlauf von mehreren Runden Schaden zufügen. Sofern ein von dieser Waffe getroffener Wurm in die Nähe anderer Würmer gelangt, sollen diese Würmer ebenfalls von dem Effekt der Waffe betroffen werden.

**Iteration III**

In der letzten Iteration wird der Umfang des Spiels erweitert. Bis zu fünf Spieler sollen jeweils mit einem Team gegeneinander spielen könne. Jeder Spieler kann seinem Team und den Würmern aus seinem Team individuelle Namen geben, die dann im Spiel angezeigt werden. Dazu soll der Schaden, den die einzelnen Würmer schon genommen haben, visualisiert werden. Anstatt zu schießen soll jeder Spieler, wenn er am Zug ist, auch mit einem seiner Würmer eine Sonderaktion ausführen können. Die Anzahl

Sonderaktionen, die jeder Spieler pro Spiel ausführen darf, soll dabei begrenzt sein. Die möglichen Sonderaktionen sind einen Wurm Teleportieren, eine Miene legen, einen Luftschlag anfordern und ein Standgeschütz aufstellen. Das Standgeschütz soll in jeder Runde automatisch auf andere Teams feuern. Zuletzt soll die bearbeitende Gruppe sich selber drei weitere Features für das Spiel ausdenken. Diese Features sollen zum Thema der Aufgabenstellung passen, und vor der Umsetzung mit dem Gruppenbetreuer abgesprochen werden.

# Iteration I

User-Stories

|  |  |
| --- | --- |
| **User Story-ID** | Bewegung |
| **User Story-Beschreibung** | Als Spieler möchte ich meinen Charakter steuern können, sodass er sich auf der Landschaft frei bewegen kann. |
| **Geschätzter Realisierungsaufwand** | Mittel |
| **Priorität** | 3 Tage |
| **Autor(en)** | Jan Voigtländer |
| **Abhängigkeiten zu anderen User Stories** | keine |
| **Zugehörige Szenarien** | Hauptszenario:   * 1. Spieler drückt Bewegungstaste * 2. Spiel erhält Tastencode * 3. Spiel wertet Bewegung aus * 4. Spiel aktualisiert Charakterposition * 5. Spiel gibt visuelle Rückmeldung an Spieler   Alternativszenarien:  Ausnahmeszenarien:   * 1. Spieler drückt Bewegungstaste * 2. Spiel erhält Tastencode * 3. Spiel wertet Bewegung aus * 4. Spiel stellt Ungültigkeit der Bewegung fest * 5. Spiel aktualisiert Charakterposition nicht |

|  |  |
| --- | --- |
| **User Story-ID** | Springen |
| **User Story-Beschreibung** | Als Spieler möchte ich mit meinem Charakter springen können, um Hindernisse zu überwinden. |
| **Geschätzter Realisierungsaufwand** | 3 Std |
| **Priorität** | Mittel |
| **Autor(en)** | Tarik Karaca |
| **Abhängigkeiten zu anderen User Stories** | Bewegung |
| **Zugehörige Szenarien** | Hauptszenario:   * 1. Spieler drückt Bewegungstaste * 2. Spiel erhält Tastencode * 3. Spiel wertet Bewegung aus * 4. Spiel aktualisiert Charakterposition * 5. Spiel gibt visuelle Rückmeldung an Spieler   Alternativszenarien:  Ausnahmeszenarien:   * 1. Spieler drückt Bewegungstaste * 2. Spiel erhält Tastencode * 3. Spiel wertet Bewegung aus * 4. Spiel stellt Ungültigkeit der Bewegung fest * 5. Spiel aktualisiert Charakterposition nicht |

|  |  |
| --- | --- |
| **User Story-ID** | Landschaft |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass die Landschaft aus einer flachen Ebene besteht, damit die Charaktersteuerung einfach ist. |
| **Geschätzter Realisierungsaufwand** | ½ Tag |
| **Priorität** | Hoch |
| **Autor(en)** | Tarik Karaca |
| **Abhängigkeiten zu anderen User Stories** | keine |
| **Zugehörige Szenarien** | Hauptszenario:  Alternativszenarien:  Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Spielansicht |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass das Spielgeschehen zweidimensional von der Seite dargestellt wird, um die Spielkomplexität gering zu halten. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Hoch |
| **Autor(en)** | Julian Flieter |
| **Abhängigkeiten zu anderen User Stories** | keine |
| **Zugehörige Szenarien** | Hauptszenario:  Alternativszenarien:  Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Spieleranzahl |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass zwei Spieler gegeneinander spielen, um das Spiel übersichtlich und den Spielfluss hoch zu halten. |
| **Geschätzter Realisierungsaufwand** | Gering |
| **Priorität** | Hoch |
| **Autor(en)** | Steve Nemangou |
| **Abhängigkeiten zu anderen User Stories** | keine |
| **Zugehörige Szenarien** | Hauptszenario:  Alternativszenarien:  Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Rundenbasiert |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass Spieler jeweils nacheinander am Zug sind, sodass ein rundenbasiertes Spiel entsteht, damit faire Abwechslung herrscht. |
| **Geschätzter Realisierungsaufwand** | Mittel |
| **Priorität** | 6 Stunden |
| **Autor(en)** | Ibrahim Kekec |
| **Abhängigkeiten zu anderen User Stories** | Spieleranzahl |
| **Zugehörige Szenarien** | Hauptszenario:   * 1. Spieler ist am Zug * 2. Spieler bewegt Charakter * 3. Spieler greift an * 4. Anderer Spieler ist am Zug   Alternativszenarien:   * 1. Spieler ist am Zug * 2. Spieler greift an * 3. Anderer Spieler ist am Zug   Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Kampfunfähigkeit |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass Charaktere kampfunfähig werden, wenn sie zu viel Schaden einstecken oder aus der Spielkarte herausfallen. |
| **Geschätzter Realisierungsaufwand** | Gering |
| **Priorität** | 2 Tage |
| **Autor(en)** | Jan Voigtländer |
| **Abhängigkeiten zu anderen User Stories** | Bewegung, Landschaft-IT1, Angreifen |
| **Zugehörige Szenarien** | Hauptszenario:   * 1. Projektil trifft Charakter * 2. Spiel wertet Schaden aus * 3. Spiel überprüft verbleibendes Leben * 4. Spiel erklärt Charakter als kampfunfähig   Alternativszenarien:   * 1. Charakter bewegt sich * 2. Spiel überprüft Position * 3. Spiel stellt fest, dass Charakter aus der Karte gefallen ist * 4. Spiel erklärt Charakter als kampfunfähig   Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Gewinnen |
| **User Story-Beschreibung** | Als Spieler möchte ich die Partie gewinnen, wenn alle gegnerischen Charaktere kampfunfähig sind. |
| **Geschätzter Realisierungsaufwand** | Gering |
| **Priorität** | 1 Tag |
| **Autor(en)** | Ibrahim Kekec |
| **Abhängigkeiten zu anderen User Stories** | Kampfunfähigkeit |
| **Zugehörige Szenarien** | Hauptszenario:   * 1. Spiel stellt Kampfunfähigkeit eines Charakters fest * 2. Spiel bestimmt Gewinner * 3. Spiel beendet Runde und zeigt Gewinner an   Alternativszenarien:  Ausnahmeszenarien: |

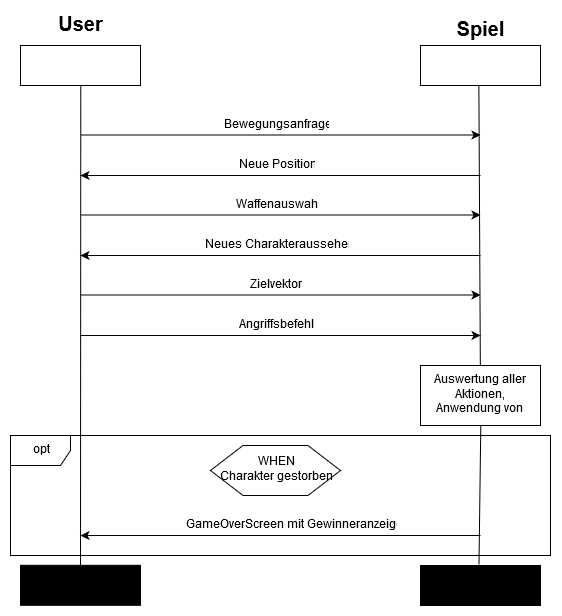
|  |  |
| --- | --- |
| **User Story-ID** | Rundenablauf |
| **User Story-Beschreibung** | Als Spieler möchte ich in meinem Zug meinen Charakter bewegen und mit einer Waffe einmal schießen können, um mehr Möglichkeiten des Spielens zu haben. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Mittel |
| **Autor(en)** | Julian Flieter |
| **Abhängigkeiten zu anderen User Stories** | Bewegung, Angreifen, Rundenbasiert |
| **Zugehörige Szenarien** | Hauptszenario:  Alternativszenarien:  Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Angreifen |
| **User Story-Beschreibung** | Als Spieler möchte ich mit einer Waffe zunächst den Abschusswinkel bestimmen und danach Projektile abschießen, um gegnerischen Charakteren gezielt Schaden zufügen zu können. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Mittel |
| **Autor(en)** | Tarik Karaca |
| **Abhängigkeiten zu anderen User Stories** | keine |
| **Zugehörige Szenarien** | Hauptszenario:   * 1. Spieler kann angreifen * 2. Spiel zeigt Zielrichtung an * 3. Spieler wählt Zielrichtung * 4. Spieler schießt * 5. Spiel erzeugt Projektil   Alternativszenarien:   * 1. Spieler kann angreifen * 2. Spiel zeigt Zielrichtung an * 3. Spieler schießt * 4. Spiel erzeugt Projektil   Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Neuer Spieler am Zug |
| **User Story-Beschreibung** | Als Spieler möchte ich deutlich darüber informiert werden welcher Spieler am Zug ist, um leichter den Überblick behalten zu können. |
| **Geschätzter Realisierungsaufwand** | 2 Std |
| **Priorität** | Mittel |
| **Autor(en)** | Steve Nemangou |
| **Abhängigkeiten zu anderen User Stories** | Rundenbasiert |
| **Zugehörige Szenarien** | Hauptszenario:   * 1. Neuer Spieler ist am Zug * 2. Spiel gibt visuelle Rückmeldung in Form von Text * 3. Spiel ändert Aussehen des am Zug befindlichen Charakters   Alternativszenarien:  Ausnahmeszenarien: |

## MSCs

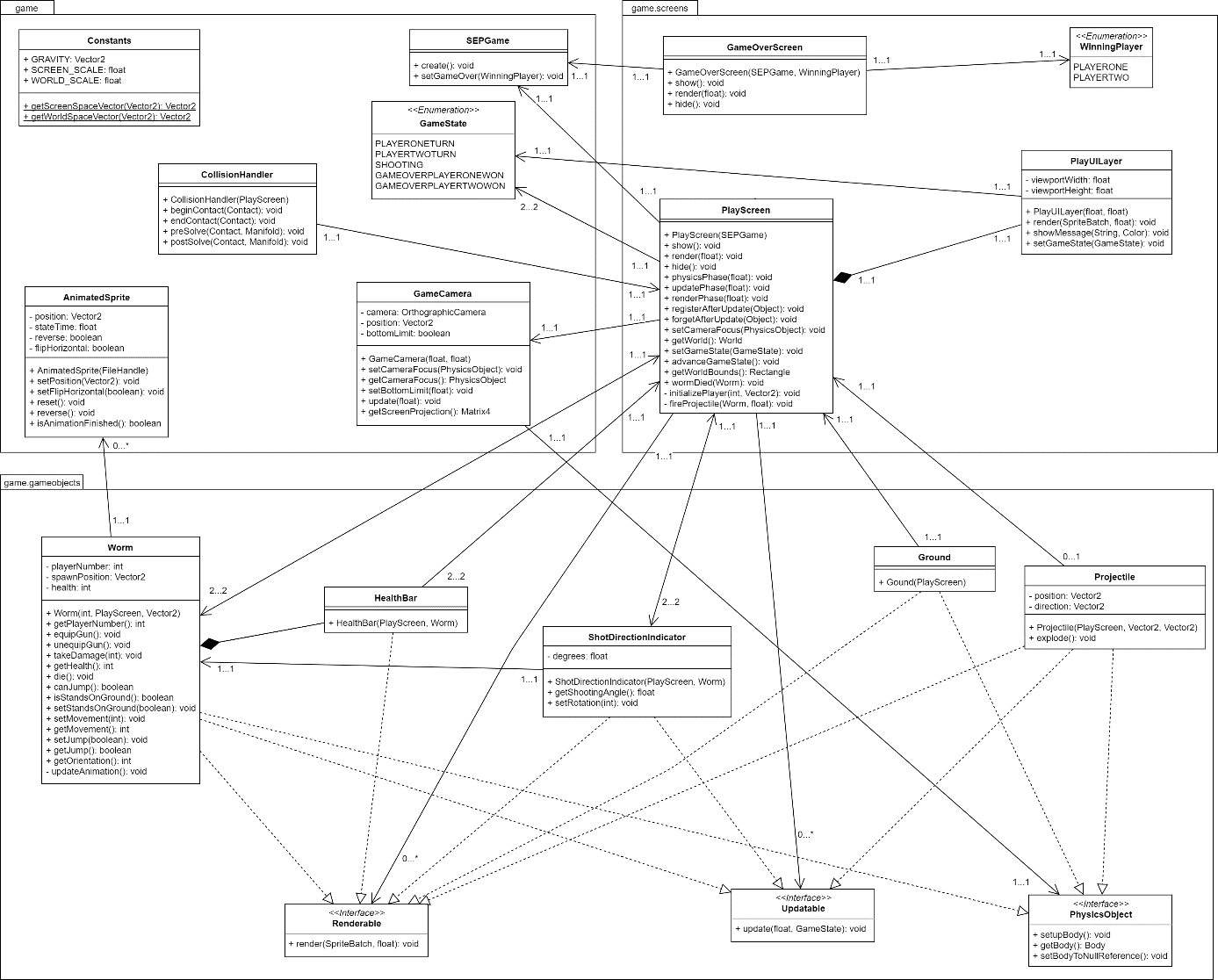
bMSC Spielerzug (Julian Flieter):



hMSC (Julian Flieter):



## UML Klassendiagramm



## 

## Funktionalitätsplanung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Quellcode-**  **referenz** | **Status** |
| **1.** | **PlayScreen** |  |  |  |  |  |
| 1.1 | Verwaltung des Spielablaufs | 1 Tag | Tarik Karaca |  | PlayScreen | Fertig |
| 1.2 | Darstellung der Spielwelt | 2 Tage | Tarik Karaca |  | PlayScreen | Fertig |
| 1.3 | Eingabeverwaltung | 1 Tag | Tarik Karaca |  | PlayScreen | Fertig |
| 1.4 | Initialisierungen der Spielwelt | 2 Tage | Tarik Karaca |  | PlayScreen | Fertig |
| 1.5 | Verwaltung der Spielwelt | Halber Tag | Tarik Karaca |  | PlayScreen | Fertig |
| **2.** | **GameOverScreen** |  |  |  |  |  |
| 2.1 | Visualisierung des Spielendes | 3 Tage | Ibrahim Kekec |  | GameOverScreen | Fertig |
| **3.** | **PlayUILayer** |  |  |  |  |  |
| 3.1 | Graphische Oberfläche | 2 Tage | Steve Nemangou |  | PlayUILayer | Fertig |
| **4.** | **CollisionHandler** |  |  |  |  |  |
| 4.1 | Verwaltungen von Kollisionen | 5 Tage | Julian Flieter |  | CollisionHandler | Fertig |
| **5.** | **SEPGame** |  |  |  |  |  |
| 5.1 | Initialisierungen des Spiels | 1 Tag | Julian Flieter |  | SEPGame | Fertig |
| **6.** | **AnimatedSprite** |  |  |  |  |  |
| 6.1 | Darstellung der Animationen | 2 Tage | Tarik Karaca |  | AnimatedSprite | Fertig |
| 6.2 | Initialisierung von Animationen | 1 Tag | Tarik Karaca |  | AnimatedSprite | Fertig |
| **7.** | **GameCamera** |  |  |  |  |  |
| 7.1 | Verwaltung der Spielansicht | 1 Tag | Tarik Karaca |  | GameCamera | Fertig |
| 7.2 | Verfolgung von Spielobjekten | 2 Std. | Tarik Karaca |  | GameCamera | Fertig |
| **8.** | **Worm** |  |  |  |  |  |
| 8.1 | Behandlung von Charakterbewegungen | Halber Tag | Jan Voigtländer |  | Worm | Fertig |
| 8.2 | Verwaltung von Spielercharakteren | 1 Tag | Jan Voigtländer |  | Worm | Fertig |
| 8.3 | Visualisierung von Spielercharakteren | 3 Tage | Jan Voigtländer |  | Worm | Fertig |
| **9.** | **Ground** |  |  |  |  |  |
| 9.1 | Initialisierung und Visualisierung der Landschaft | 6 Tage | Steve Nemangou |  | Ground | Fertig |
| **10.** | **Projectile** |  |  |  |  |  |
| 10.1 | Initialisierung und Visualisierung der Projektile | 6 Tage | Steve Nemangou |  | Projectile | Fertig |
| **11.** | **ShotDirectionIndicator** |  |  |  |  |  |
| 11.1 | Initialisierung und Visualisierung des Abschusswinkels | 4 Tage | Ibrahim Kekec |  | ShotDirectionIndicator | Fertig |
| 11.2 | Berechnen des Abschusswinkels | 2 Tage | Ibrahim Kekec |  | ShotDirectionIndicator | Fertig |
| **12.** | **HealthBar** |  |  |  |  |  |
| 12.1 | Visualisierung der Lebensanzeige | 1 Tag | Jan Voigtländer |  | HealthBar | Fertig |

# Iteration II

## User-Stories

|  |  |
| --- | --- |
| **User Story-ID** | Kräfte |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass Projektile von Kräften wie Wind und Gravitation beeinflusst werden, um ein realistischeres und fordernderes Spielerlebnis zu bekommen. |
| **Geschätzter Realisierungsaufwand** | Gering |
| **Priorität** | Mittel |
| **Autor(en)** | Julian Flieter |
| **Abhängigkeiten zu anderen User Stories** | Keine |
| **Zugehörige Szenarien** | * Hauptszenario:   1. Nächster Spieler am Zug  2. Spiel berechnet neue Windstärke  3. Spiel aktualisiert Windstärke  4. Spieler führt Zug aus   * Alternativszenarien: * Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Zerstörbare Spielwelt |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass die Spielwelt zerstörbar ist, um mehr Spielmöglichkeiten zu besitzen. |
| **Geschätzter Realisierungsaufwand** | Hoch |
| **Priorität** | Mittel |
| **Autor(en)** | Tarik Karaca |
| **Abhängigkeiten zu anderen User Stories** | Spielstart |
| **Zugehörige Szenarien** | * Hauptszenario:   1. Projektil trifft Spielwelt  2. Spiel berechnet Schaden an Spielwelt  3. Spiel wendet Schaden an  4. Spiel zeichnet Spielwelt neu   * Ausnahmeszenarien: * Alternativszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Waffen |
| **User Story-Beschreibung** | Als Spieler möchte ich mindestens drei verschiedene Waffen zur Auswahl haben, um für jede Spielsituation gewappnet zu sein. |
| **Geschätzter Realisierungsaufwand** | Mittel |
| **Priorität** | Mittel |
| **Autor(en)** | Jan Voigtländer |
| **Abhängigkeiten zu anderen User Stories** | Keine |
| **Zugehörige Szenarien** | * Hauptszenario:  1. Spieler wählt Angriff 2. Spiel prüft verfügbare Waffen 3. Spiel zeigt verfügbare Waffenauswahl an 4. Spieler wählt Waffe 5. Spiel aktualisiert Charakteraussehen für neue Waffe 6. Spiel zeigt Ziel-GUI für ausgewählte Waffe 7. Spieler führt Angriff aus  * Ausnahmeszenarien: * Alternativszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Spielgrenze |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass die Charaktere sterben, wenn sie unter eine bestimmte Grenze der Spielwelt fallen, um mehr Spielmöglichkeiten zu besitzen. |
| **Geschätzter Realisierungsaufwand** | Gering |
| **Priorität** | Mittel |
| **Autor(en)** | Ibrahim Kekec |
| **Abhängigkeiten zu anderen User Stories** | Spielstart, zerstörbare Spielwelt |
| **Zugehörige Szenarien** | * Hauptszenario:   1. Runde endet  2. Spiel erhöht untere Spielgrenze  3. Spiel visualisiert neue untere Spielgrenze  4. Spiel zeichnet Spielwelt neu  5. Nächste Runde beginnt   * Ausnahmeszenarien: * Alternativszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Spielstart |
| **User Story-Beschreibung** | Als Spieler möchte ich mindestens vier verschiedene Spiellandschaften haben, um Abwechslung im Spiel zu haben. |
| **Geschätzter Realisierungsaufwand** | Hoch |
| **Priorität** | Hoch |
| **Autor(en)** | Steve Nemangou |
| **Abhängigkeiten zu anderen User Stories** | Keine |
| **Zugehörige Szenarien** | * Hauptszenario:   1. Spiel startet  2. Spiel zeigt Startmenü  3. Spieler wählt ein von vier Spielwelten und Charakterzahl aus  4. Spiel generiert Spielwelt mit Charakteren  5. Spiel zeichnet Spielwelt  6. Runde startet   * Ausnahmeszenarien: * Alternativszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Spezialwaffe |
| **User Story-Beschreibung** | Als Spieler möchte ich eine Spezialwaffe zur Verfügung haben, welche Charakteren Schaden über Zeit verursacht, um mehr Spielmöglichkeiten zu besitzen. |
| **Geschätzter Realisierungsaufwand** | Mittel |
| **Priorität** | Mittel |
| **Autor(en)** | Jan Voigtländer |
| **Abhängigkeiten zu anderen User Stories** | Waffen |
| **Zugehörige Szenarien** | * Hauptszenario:   1. Charakter wird von Spezialwaffe getroffen  2. Spiel markiert Charakter als infiziert  3. Spiel wendet jede Runde Schaden auf Charakter an   * Ausnahmeszenarien: * Alternativszenarien:   2. Spiel markiert Charakter als infiziert  3. Spiel prüft auf andere Charaktere in Reichweite  4. Spiel markiert andere Charaktere auch als infiziert  5. Spiel erhöht Schaden an Charakteren jede Runde |

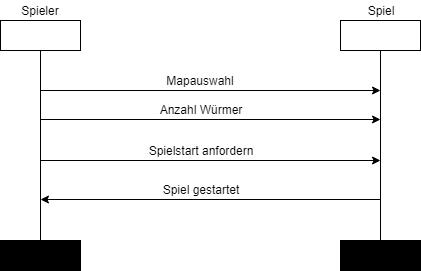
|  |  |
| --- | --- |
| **User Story-ID** | Team |
| **User Story-Beschreibung** | Als Spieler möchte ich ein Team von bis zu fünf Charakteren besitzen, damit sich mehr Spielmöglichkeiten ergeben und die Spielrunden länger dauern. |
| **Geschätzter Realisierungsaufwand** | Gering |
| **Priorität** | Hoch |
| **Autor(en)** | Tarik Karaca |
| **Abhängigkeiten zu anderen User Stories** | Spielbare Welten |
| **Zugehörige Szenarien** | Hauptszenario:  Alternativszenarien:  Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Replay |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass ein Replay vom Tod eines Charakters gezeigt wird, wenn dieser stirbt. |
| **Geschätzter Realisierungsaufwand** | Hoch |
| **Priorität** | Gering |
| **Autor(en)** | Steve Nemangou |
| **Abhängigkeiten zu anderen User Stories** | Keine |
| **Zugehörige Szenarien** | Hauptszenario:   1. Projektil wird abgeschossen 2. Replay wird erstellt 3. Wurm stirbt 4. Replay der letzten 10s wird wiedergegeben 5. Replay wird gelöscht   Alternativszenarien:  Ausnahmeszenarien:   1. Projektil wird abgeschossen 2. Replay wird erstellt 3. Kein Wurm stirbt 4. Replay wird gelöscht |

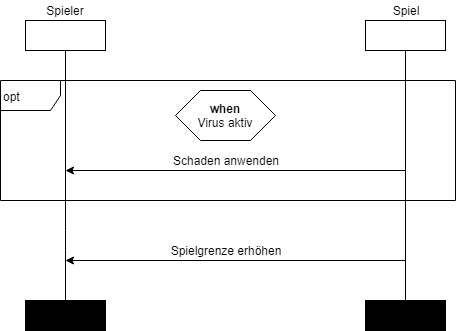
## 

## MSCs

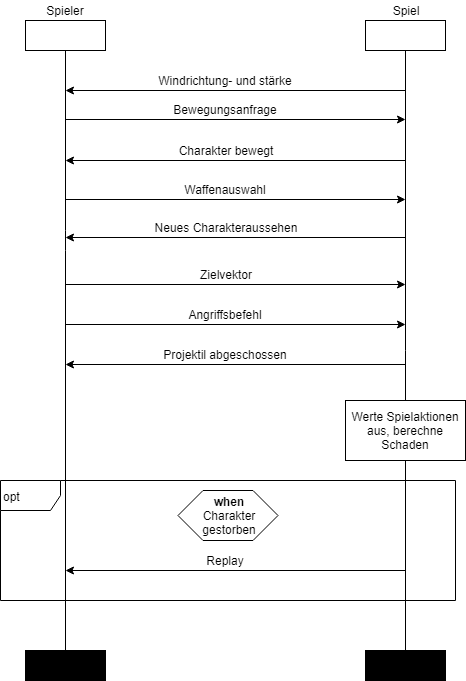
Spielstart



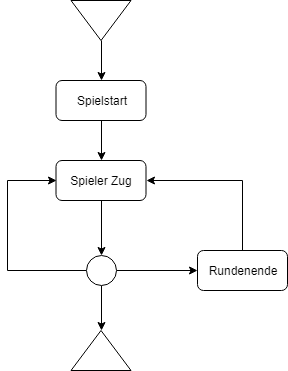
Rundenende



Spieler Zug



hMSC



## UML Klassendiagramm

## Funktionalitätsplanung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Quellcodereferenz** | **Status** |
| **1.** | **PlayScreen** |  | **Tarik Karaca** |  |  |  |
| 1.1 | Darstellung des Replays | 10 Minuten |  | Snapshot | PlayScreen.java | Fertig |
| 1.2 | Darstellung der Spielwelt | 1 Stunde |  | World | PlayScreen.java | Fertig |
| **2.** | **World** |  | **Tarik Karaca** |  |  |  |
| 2.1 | Verwaltung der Spielwelt | 3 Tage |  | PlayScreen, Ground | World.java | Fertig |
| 2.2 | Verwaltung der Spielweltobjekte | 2 Tage |  | Player, Worm, Weapon | World.java | Fertig |
| 2.3 | Verarbeitung dynamischer Spielweltzerstörung | 2 Tage |  | Weapon, Ground | World.java, Ground.java | Fertig |
| 2.4 | Verarbeitung und Aktualisierung der unteren Spielweltgrenze | 1 Tag |  | - | World.java | Fertig |
| **3.** | **Player** |  | **Jan Voigtländer** | **World** |  |  |
| 3.1 | Verwaltung der Charaktere | 1 Tag |  | Worm | Player.java | Fertig |
| 3.2 | Verwaltung der Indikatoren | 1 Tag |  | WindIndicator, ShotDirectionIndicator | Player.java | Fertig |
| 3.3 | Erzeugung von benötigten Objekten für Replays | 2 Tage |  | Snapshot | Player.java | Fertig |
| 3.4 | Verwaltung von Charakterwaffen | 2 Tage |  | Weapon | Player.java | Fertig |
| **4.** | **Worm** |  | **Jan Voigtländer** | **Player** |  |  |
| 4.1 | Verarbeitung und Berechnung von Charaktereingaben | 5 Tage |  | InputHandler, World | Worm.java | Fertig |
| **5.** | **Weapon** |  | **Jan Voigtländer** | **Worm** |  |  |
| 5.1 | Verwaltung von Munition | 15 Minuten |  | - | WeaponTpe.java | Fertig |
| 5.2 | Erzeugung von Projektilen | 1 Stunde |  | Projectile | Weapon.java |  |
| **6.** | **Projectile** |  | **Steve Nemangou** | **Weapon, ShotDirectionIndicator** |  |  |
| 6.1 | Erzeugung und Verwaltung verschiedener Projektiltypen | 5 Tage |  | Weapon | Projectile.java | Fertig |
| **7.** | **Explosion** |  | **Steve Nemangou** | **Projectile** |  |  |
| 7.1 | Visualisierung und Berechnung von Projektilexplosionen und deren Wirkung | 3 Tage |  | Projectile | Explosion.java | Fertig |
| **8.** | **Snapshot** |  | **Steve Nemangou** | **World** |  |  |
| 8.1 | Verwaltung für Replays benötigter Daten | 7 Tage |  | World, PlayScreen, Player | Snapshot.java | Fertig |
| **9.** | **AssetManager** |  | **Tarik Karaca** |  |  |  |
| 9.1 | Verwaltung und Bereitstellung aller Assets | 1 Tag |  | - | AssetManager.java | Fertig |
| **10.** | **InputHandler** |  | **Jan Voigtländer** |  |  |  |
| 10.1 | Verwaltung und Verteilung aller Peripherie-Eingaben | 2 Tage |  | - | InputHandler.java | Fertig |
| **11.** | **WindHandler** |  | **Julian Flieter** |  |  |  |
| 11.1 | Verwaltung der rundenbasierten Windbedingungen | 2 Tage |  | - | WindHandler.java | Fertig |
| **12.** | **WindIndicator** |  | **Julian Flieter** | **WindHandler** |  |  |
| 12.1 | Visualisierung der Aktuellen Windbedingungen | 2 Tage |  | WindHandler, PlayScreen | WindIndicator.java | Fertig |
| **13.** | **CollisionHandler** |  | **Julian Flieter** |  |  |  |
| 13.1 | Verwaltung von Objektkollisionen in der Spielwelt | 3 Tage |  | Worm, Pojectile, World | CollisionHandler.java | Fertig |
| **14** | **MenuScreen** |  | **Julian Flieter** | **PlayScreen** |  |  |
| 14.1 | Visualisierung des Startmenüs und Verwaltung der Spielstartparamteter. | 1 Tag |  | PlayScreen | MenuScreen.java | fertig |
| **15** | **WeaponUI** |  | **Julian Flieter** | **PlayScreen, Weapon** |  |  |
| 15.1 | Visualisierung und Verwaltung der Waffenauswahl. | 2 Tage | Julian Flieter | PlayScreen, Weapon | WeaponUI.java | Fertig |
| **16** | **Erstellung der Maps** |  | **Julian Flieter** |  |  | **Fertig** |
| **16** | **GameOverScreen** |  | **Ibrahim Kekec** | **PlayScreen** |  |  |
| 16.1 | Visualisierung des „Game Over“ Bildschirms | 1 Tag |  | PlayScreen | GameOverScreen.java | Fertig |
| **17** | **Virus** |  | **Ibrahim Kekec** | **Weapon** |  |  |
| 17.1 | Erzeugung und Verwaltung des Wirkungsbereiches des Virus | 2 Tage |  | Weapon | Worm.java, createVirusFixture(), setIsInfected() | Fertig |
| **18** | **Modultests** |  | **Ibrahim Kekec** |  |  |  |
| 18.1 | Testen der Methode „shoot“ der Klasse Weapon | 1 Tag |  | Weapon | WeaponTest.java | Fertig |
| 18.2 | Testen der Methode „takeDamage“ der Klasse Worm | 1 Tag |  | Worm | WormTest.java | Fertig |

## Unittests

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Getestete Funktionalität** | **Quellcode**  **Referenz** | **Status** |
| **1.** | Getestet, ob die Methode „shoot“ die Munition pro Schuss verringert. | WeaponTest.java | Test bestanden |
| **2.** | Getestet, ob bei der Methode „takeDamage“, der richtige Wert als Schaden abgezogen wird. | WormTest.java | Test bestanden |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **11.06.2018** | | |
| **Tester** | Julian Flieter | | |
| **SW-Version** | V 1.0 | | |
| **Vorbedingung(en)** | Es existieren noch lebende Charaktere. Spieler 1 tötet einen Charakter von Spieler 2. | | |
| **Schritt** | Aktion (User) | Erwartete Reaktion (System) | √ / X |
| **1** | Spieler 1 wählt eine Waffe aus. | Spieler 1 sieht visuell die ausgewählte Waffe. | √ |
| **2** | Spieler 1 zielt und schießt auf den Charakter des 2. Spielers. | Der Charakter des Spieler 2 stirbt. | √ |
| **3** | Es wird ein Replay abgespielt. | Beide Spieler bekommen ein Replay des Todes abgespielt. | √ |
| **Nachbedingung(en)** | Spieler 2 verliert einen Charakter und ist jetzt am Zug. | | √ |
| **Testurteil** | Der Test wurde erfolgreich abgeschlossen. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **C** | **11.06.2018** | | |
| **Tester** | Julian Flieter | | |
| **SW-Version** | V 1.0 | | |
| **Vorbedingung(en)** | Es existieren noch lebende Charaktere. Spieler 1 ist am Zug. | | |
| **Schritt** | Aktion (User) | Erwartete Reaktion (System) | √ / X |
| **1** | Spieler 1 wählt die Spezialwaffe aus. | Spieler 1 sieht visuell die ausgewählte Waffe. | √ |
| **2** | Spieler 1 zielt und schießt auf den Charakter des Gegenspielers | Der Charakter des Spielers 2 wird getroffen und als markiert angezeigt. | √ |
| **3** | Spieler 1 beendet seine Runde | Der markierte Charakter des Spielers 2 erleidet Schaden zu Beginn seiner Runde. | √ |
| **Nachbedingung(en)** | Charaktere in der Nähe des Markierten werden ebenfalls markiert. | | √ |
| **Testurteil** | Der Test wurde erfolgreich abgeschlossen. | | |

# Iteration III

## User-Stories

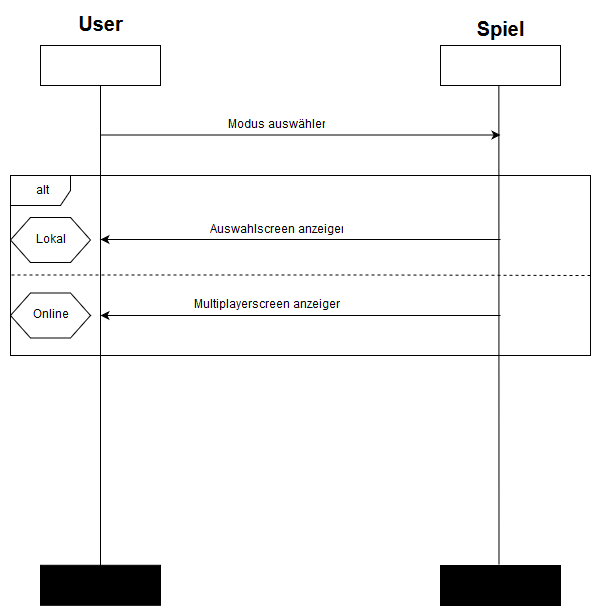
|  |  |
| --- | --- |
| **User Story-ID** | Spezialaktionen |
| **User Story-Beschreibung** | Als Spieler möchte ich, zusätzlich zu den normalen Waffen, alternativ noch Spezialaktionen zur Verfügung haben, um mehr Spielmöglichkeiten zu besitzen. |
| **Geschätzter Realisierungsaufwand** | 1 Woche |
| **Priorität** | Hoch |
| **Autor(en)** | Jan Voigtländer |
| **Abhängigkeiten zu anderen User Stories** |  |
| **Zugehörige Szenarien** | * Hauptszenario: * Neuer Charakter ist am Zug * Spieler wählt Spezialwaffe/-aktion * Spieler führt Spezialaktion aus * Nächster Spieler ist am Zug * Alternativszenarien: - * Ausnahmeszenarien: * Neuer Charakter ist am Zug * Spieler wählt Spezialwaffe/-aktion * Spieler hat keine Munition für Spezialwaffe/-aktion mehr * Spiel weist Aktion zurück, Spieler weiterhin am Zug |

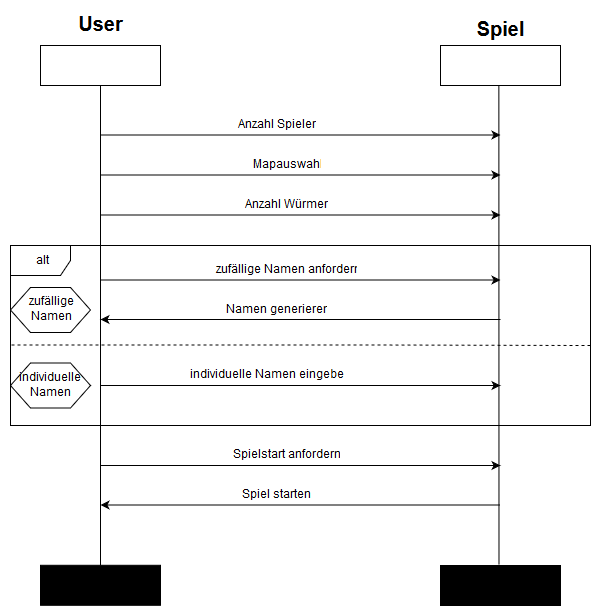
|  |  |
| --- | --- |
| **User Story-ID** | Mehr Spieler |
| **User Story-Beschreibung** | Als Spieler möchte ich mit bis zu 4 Freunden gleichzeitig spielen können, um mehr Spielspaß und längere Spielsitzungen zu bekommen. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Mittel |
| **Autor(en)** | Julian Flieter |
| **Abhängigkeiten zu anderen User Stories** |  |
| **Zugehörige Szenarien** | * Hauptszenario: * Spielstart * Spieler wählt Spielmodus (Lokal <-> Multiplayer) * Spieler/Host wählt Anzahl Teilnehmer aus * Spieler/Host wählt Anzahl Charaktere pro Spieler aus * Jeder Spieler wählt Team- und Charakternamen * Alle Spieler stellen sich auf Bereit * Spiel beginnt * Alternativszenarien: - * Ausnahmeszenarien: - |

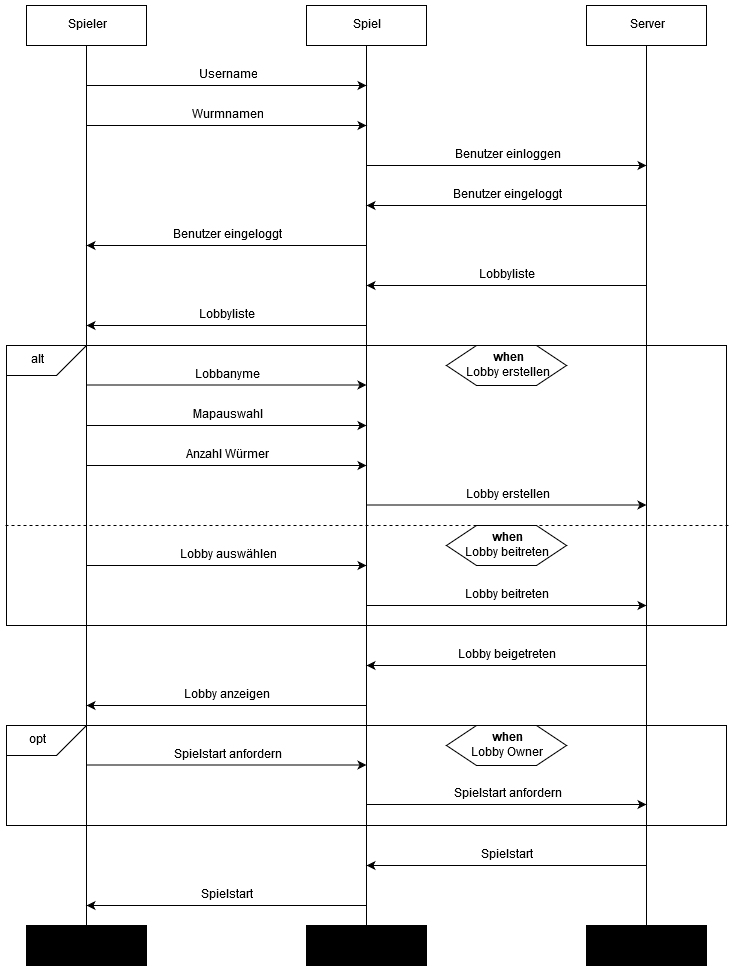
|  |  |
| --- | --- |
| **User Story-ID** | Teamnamen |
| **User Story-Beschreibung** | Als Spieler möchte ich mein Team und meine Charaktere vor der Runde benennen können, um mehr Abwechslung und Kreativität geboten zu bekommen. |
| **Geschätzter Realisierungsaufwand** | <1 Tag |
| **Priorität** | Mittel |
| **Autor(en)** | Steve Nemangou |
| **Abhängigkeiten zu anderen User Stories** |  |
| **Zugehörige Szenarien** | * Hauptszenario: Siehe Mehr Spieler * Alternativszenarien: - * Ausnahmeszenarien: - |

|  |  |
| --- | --- |
| **User Story-ID** | Multiplayer |
| **User Story-Beschreibung** | Als Spieler möchte ich Multiplayer Unterstützung haben, um auch dann mit meinen Freunden spielen zu können, wenn sie nicht zu Besuch sind. |
| **Geschätzter Realisierungsaufwand** | 3 Wochen |
| **Priorität** | Mittel |
| **Autor(en)** | Tarik Karaca |
| **Abhängigkeiten zu anderen User Stories** |  |
| **Zugehörige Szenarien** | * Hauptszenario: Siehe Mehr Spieler * Alternativszenarien: - * Ausnahmeszenarien: - |

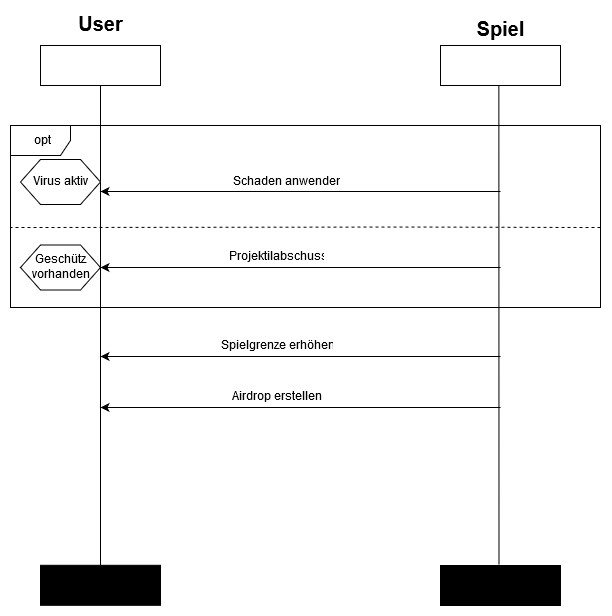
## MSCs

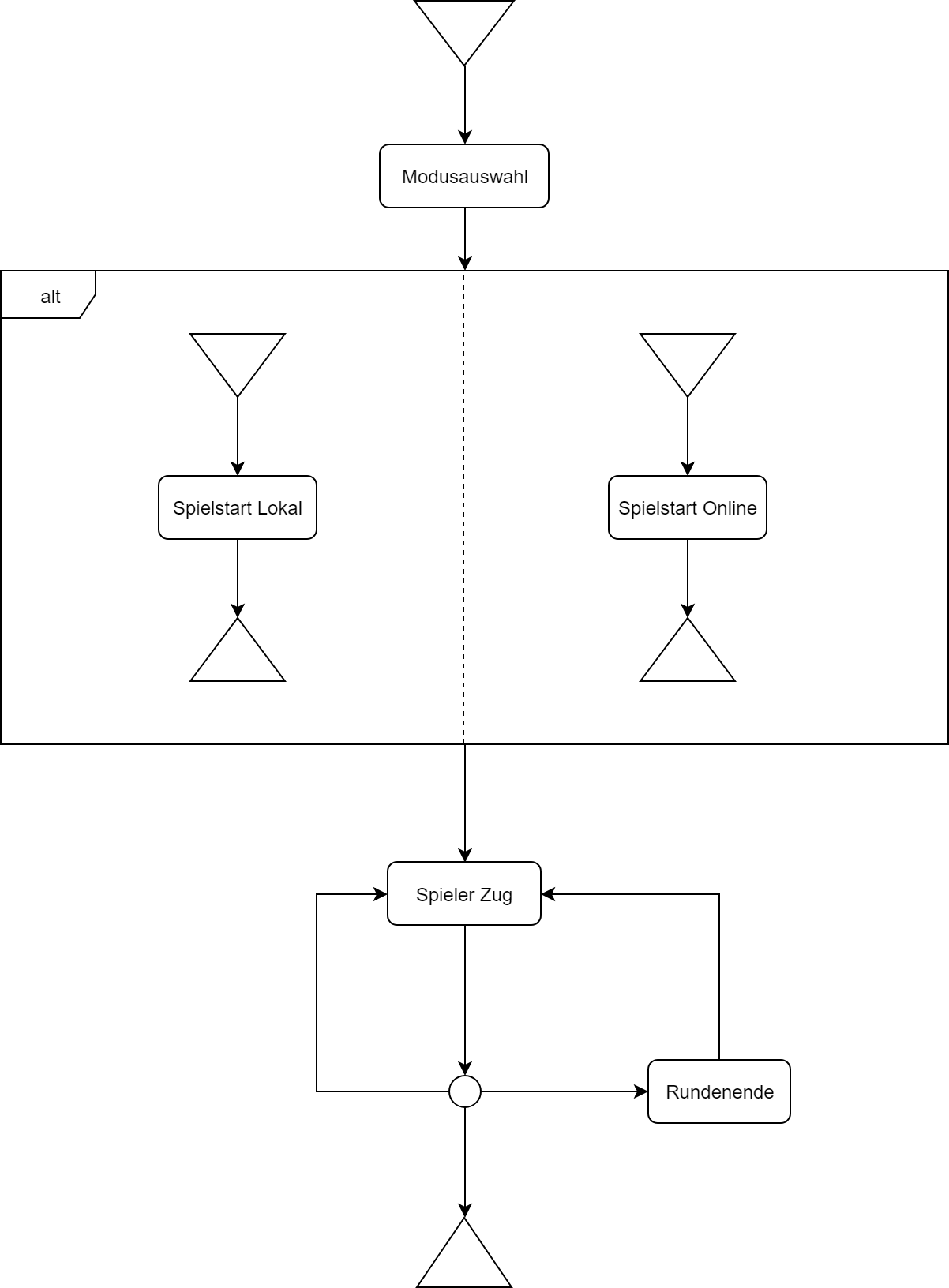
Modusauswahl  


Spielstart Lokal  


Spielstart Online  


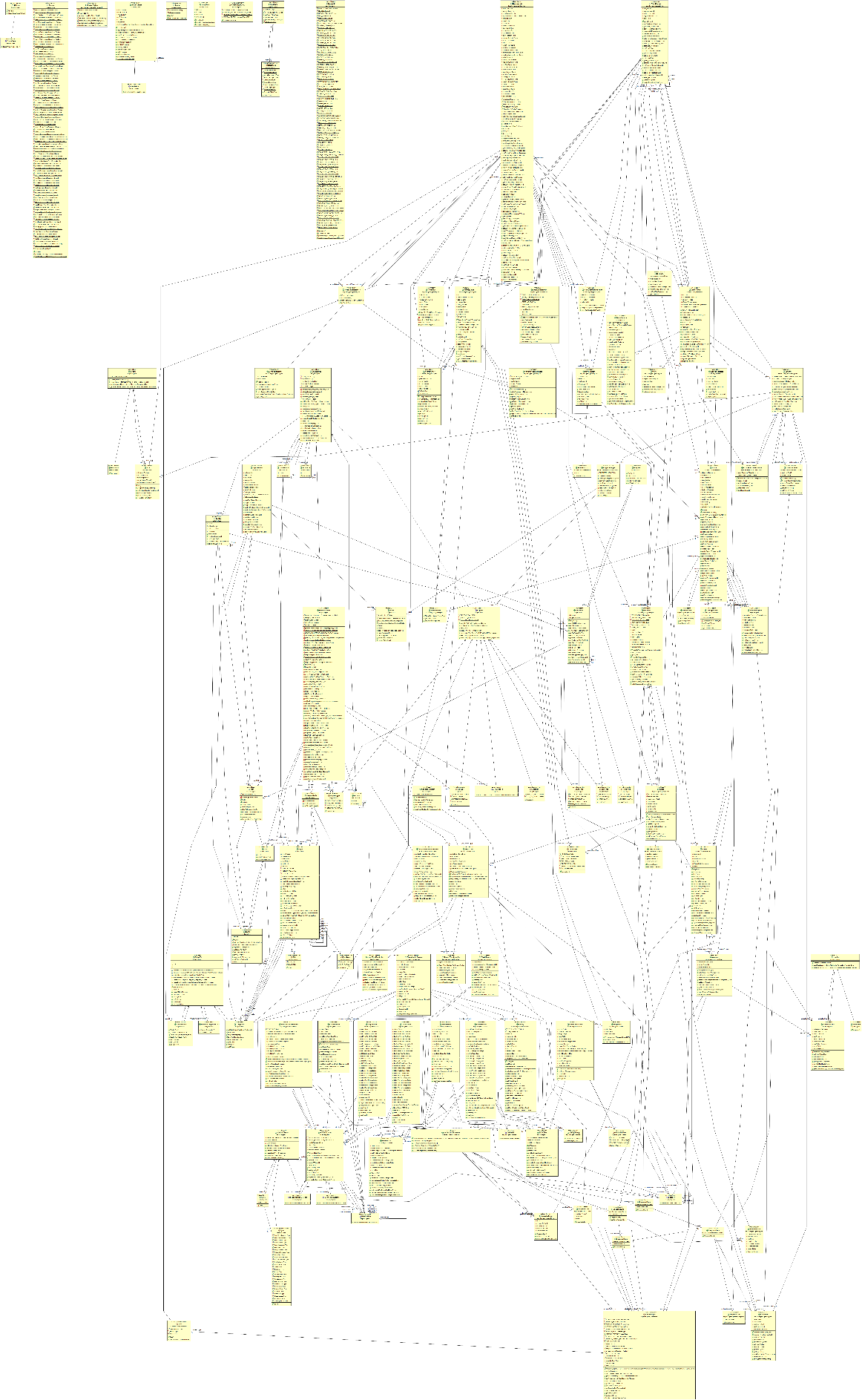
Spieler Zug  


Rundenende  


hMSC  


## UML Klassendiagramm

Erläuterung des Klassendiagramms:



## Funktionalitätsplanung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Quellcodereferenz** | **Status** |
| **1** | **Multiplayer** |  |  |  |  |  |
| 1.1 | Client-Server-Logik | 3 Wochen | Tarik Karaca |  | Server-Package, Interfaces-Package, NetworkWorldHandler.java, InterpolationWorldHandler.java, WorldHandler.java, NetworkClient.java, DataHandler.java | fertig |
| 1.2 | Erweiterung des Hauptmenüs | 1 Woche | Julian Flieter |  | LoginScreen.java, LocalScreen.java, ModiSceen.java, LobbyScreen.java, PlayerLobbyScreen.java, CreditsScreen.java | fertig |
| 1.3 | Merging | 5 Tage | Jeder selbst | 1.1 | Alles | fertig |
| **2** | **Spezialwaffen** |  |  | **3** |  |  |
| 2.1 | Luftschlag | 1 Woche | Ibrahim Kekec |  | WeaponType.java, Weapon.java | fertig |
| 2.2 | Geschütz | 2 Wochen | Steve Nemangou |  | Turret.java, WorldHandler.java | fertig |
| 2.3 | Landmine | 7 Tage | Steve Nemangou |  | Projectile.java, WorldHandler.java | fertig |
| 2.4 | Teleporter | 2 Tag | Julian Flieter |  | Weapon.java (shoot-Methode) | fertig |
| **3** | **Airdrop** |  | **Jan Voigtländer** | **2** |  |  |
| 3.1 | Abwurf einer Lootkiste mit zufälliger Waffe jede Runde | 5 Tage |  |  | WorldHandler.java, AirdropCrate.java, AirdropChute.java | fertig |
| **4** | **Bis zu 5 Spieler** |  |  |  |  |  |
| 4.2 | Erweiterung der Spiellogik | 2 Tage | Jan Voigtländer | 9.1 | WorldHandler.java, Player.java, PlayUILayer.java | fertig |
| **5** | **Team- & Charakternamen** |  | **Julian Flieter** |  |  |  |
| 5.1 | Erweiterung des Hauptmenüs | 2 Tage |  |  | LocalScreen.java,  LoginScreen.java | fertig |
| 5.2 | Spiellogik | 1 Tag |  |  | Player.java, Worm.java | fertig |
| 5.3 | Visualisierung | 2 Tage |  |  | HealthBar.java, | fertig |
| **6** | **Visuelle Benachrichtigung über Ereignisse** |  | **Jan Voigtländer** |  |  |  |
| 6.1 | HoverText für Drop-Pickup & Schaden | 1 Tag |  |  | HoverText.java, WorldHandler.java | fertig |
| **7** | **Sound** |  | **Ibrahim Kekec** |  |  |  |
| 7.1 | Hauptmenümusik | 1 Tag |  |  | MenuScreen.java | fertig |
| 7.2 | SFX Ingame | 3 Tage |  |  | WorldHandler.java | fertig |
| 7.3 | SFX Menü | 3 Tage |  |  | SEPGame.java | fertig |
| **8** | **Anderes** |  |  |  |  |  |
| 8.1 | Headshot System | 2 Tag | Ibrahim Kekec |  | Worm.java | fertig |
| **9** | **Spielwelt** |  | **Julian Flieter** |  |  |  |
| 9.1 | Erweiterung der Spielkarten | <1 Tag |  |  | TiledMap | fertig |
| 9.2 | Erweiterung Waffen Interface | ½ Tag |  |  | WeaponUI.java | fertig |
| **10** | **Modultests** |  | **Ibrahim Kekec** |  |  |  |

## Unittests

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Getestete Funktionalität** | **Quellcode**  **Referenz** | **Status** |
| **1** | Getestet, ob die Methode „getWeapon“ mit dem gegebenen Parameter, die richtige Waffe zurückgibt. | PlayerTest.java | Test bestanden |
| **2** | Getestet, ob die Methode „shoot“ die Munition pro Schuss verringert. | WeaponTest.java | Test bestanden |
| **3** | Getestet, ob bei der Methode „takeDamage“, der richtige Wert als Schaden abgezogen wird. | WormTest.java | Test bestanden |
| **4** | Getestet, ob die Methode „equipWeapon“, die richtige Waffe anlegt. | WormTest2.java | Test bestanden |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | 11.06.2018 | | |
| **Tester** | Jan Voigtländer | | |
| **SW-Version** | V 1.0 | | |
| **Vorbedingung(en)** | Spiel ist gestartet. Server ist gestartet. Lobby zum Beitreten existiert. | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Spieler wählt den Spielmodus „Online“ aus. | Das Spielt wechselt zum Login-Screen. | **√** |
| 2 | Spieler gibt einen Namen für sein Team und seine Charaktere ein. | Spiel zeigt die eingegebenen Informationen an. | **√** |
| 3 | Spieler bestätigt die eingegebenen Informationen. | Spiel verbindet sich mit dem Server und zeigt die Liste verfügbarer Lobbies an. | **√** |
| 4 | Spieler wählt eine Lobby zum Beitreten. | Das Spiel zeigt den Wartebildschirm an. | **√** |
| 5 | Host startet das Spiel. | Die Welt wird generiert und die Charaktere der Teilnehmer hineingesetzt. | **√** |
| **Nachbedingung(en)** | Das Spiel startet und ist spielbar. | | **√** |
| **Testurteil** | Bestanden | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | 11.06.2018 | | |
| **Tester** | Jan Voigtländer | | |
| **SW-Version** | V 1.0 | | |
| **Vorbedingung(en)** | Spielrunde ist beendet. Zu Beginn der neuen Runde wird eine Kiste mit zufälligem Inhalt (Spezialwaffe) am oberen Ende der Karte erstellt und auf dem Spielfeld deponiert. | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Spieler bewegt seine Figur auf die Kiste. | Kiste verschwindet. Charakter erhält 1 Munition für die Spezialwaffe. Das Spiel gibt textuelle Informationen über die Art der aufgenommenen Waffe aus. | **√** |
| 2 | Spieler wählt die Spezialwaffe mit Munition aus. | Waffe wird ausgewählt und ist benutzbar. | **√** |
| **Nachbedingung(en)** |  | | **√** |
| **Testurteil** | Bestanden | | |

# Nutzerhandbuch

## Technische Anforderungen

Java 8 oder neuer.  
Eine CPU, die 20xx tauglich ist.  
Eine Grafikkarte, die bunte Farben berechnen kann.  
Eine Tastatur, mit unten genannten Tasten.  
Ein Bildschirm, der bunte Farben anzeigen kann.  
Ein Betriebssystem, das Programme ausführen kann.  
Eine Floppy Disk mit mindestens 250MB Speicherplatz.  
Eine Maus, die Bewegungen erkennt.  
Strom.  
Einen Menschen oder ähnlichen humanoiden Affen, der das alles zusammenbaut und bedient.

Oder kurz gesagt: Sollte auf allen Rechnern laufen.

## Installationsanleitung

Sicherstellen, dass Java 8 oder neuer installiert ist. Die neuste Version gibt es [hier](http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html). Dann die mitgelieferte „Swat Engagement Pheretima.jar“ ausführen und spielen.

## Bedienungsanleitung

W,A,S,D zur Steuerung des Charakters  
Leertaste zum Springen  
Enter für Aktionen (schießen, Item benutzen)  
F1-9 Waffenauswahl (siehe Auswahlmenü ingame)  
↑, ↓ Zum zielen