SEP Projektmappe

**Projektmappe des Projektes**

Swat Engagement Pheretima

**Dokumentation des Projektes**

Gruppe L:

Karaca, Tarik

Flieter, Julian

Kekec, Ibrahim

Nemangou Tchatchou, Steve Gabin

Voigtländer, Jan

Inhaltsverzeichnis

[Projektbeschreibung 4](#_Toc512291565)

[Iteration I 7](#_Toc512291566)

[User-Stories 7](#_Toc512291567)

[Szenarien 10](#_Toc512291568)

[MSCs 12](#_Toc512291569)

[UML Klassendiagramm 17](#_Toc512291570)

[Funktionalitätsplanung 17](#_Toc512291571)

[Unittests 18](#_Toc512291572)

[Iteration II 19](#_Toc512291573)

[User-Stories 19](#_Toc512291574)

[MSCs 19](#_Toc512291575)

[UML Klassendiagramm 19](#_Toc512291576)

[Funktionalitätsplanung 19](#_Toc512291577)

[Unittests 19](#_Toc512291578)

[Systemtests 19](#_Toc512291579)

[Iteration III 21](#_Toc512291580)

[User-Stories 21](#_Toc512291581)

[MSCs 21](#_Toc512291582)

[UML Klassendiagramm 21](#_Toc512291583)

[Funktionalitätsplanung 21](#_Toc512291584)

[Unittests 21](#_Toc512291585)

[Systemtests 21](#_Toc512291586)

[Nutzerhandbuch 23](#_Toc512291587)

[Technische Anforderungen 23](#_Toc512291588)

[Installationsanleitung 23](#_Toc512291589)

[Bedienungsanleitung 23](#_Toc512291590)

## Projektbeschreibung

In diesem Abschnitt soll die Projektbeschreibung abgedruckt werden, die ihr als Aufgabenbeschreibung von eurem Betreuer erhalten habt. Sie dient als initiales Anforderungsdokument für eure Spezifikationsaktivitäten.

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** | CharakterBewegen |
| **User Story-Beschreibung** | Als Spieler möchte ich meinen Charakter steuern, sodass er sich auf der Landschaft frei bewegen kann. |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor(en)** |  |
| **Abhängigkeiten zu anderen User Stories** | LandschaftIT1 |
| **Zugehörige Szenarien** | * Hauptszenario: „Bewegung“ * Alternativszenarien: - * Ausnahmeszenarien: „Bewegung blockiert“ |

|  |  |
| --- | --- |
| **User Story-ID** | CharakterSpringen |
| **User Story-Beschreibung** | Als Spieler möchte ich mit meinem Charakter springen, um Hindernisse zu überwinden. |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor(en)** |  |
| **Abhängigkeiten zu anderen User Stories** | CharakterBewegen |
| **Zugehörige Szenarien** | * Hauptszenario: - (ja, es fehlt tatsächlich) * Alternativszenarien: - * Ausnahmeszenarien: - |

|  |  |
| --- | --- |
| **User Story-ID** | LandschaftIT1 |
| **User Story-Beschreibung** | Als Entwickler möchte ich, dass die Landschaft aus einer flachen Ebene besteht, um die Entwicklung zunächst einfach zu halten. |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor(en)** |  |
| **Abhängigkeiten zu anderen User Stories** | Spielansicht |
| **Zugehörige Szenarien** | <Zuordnung zu Szenarien>   * Hauptszenario: - * Alternativszenarien: - * Ausnahmeszenarien: - |

|  |  |
| --- | --- |
| **User Story-ID** | Spielansicht |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass das Spielgeschehen zweidimensional von der Seite dargestellt wird. |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor(en)** |  |
| **Abhängigkeiten zu anderen User Stories** |  |
| **Zugehörige Szenarien** | * Hauptszenario: - * Alternativszenarien: - * Ausnahmeszenarien: - |

|  |  |
| --- | --- |
| **User Story-ID** | AnzahlSpielerIT1 |
| **User Story-Beschreibung** | Als Entwickler möchte ich, dass zwei Spieler gegeneinander spielen, um die Entwicklung zunächst einfach zu halten. |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor(en)** |  |
| **Abhängigkeiten zu anderen User Stories** |  |
| **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. |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor(en)** |  |
| **Abhängigkeiten zu anderen User Stories** | SpielerZug |
| **Zugehörige Szenarien** | * Hauptszenario: „Rundenbasiert“ * Alternativszenarien: „RundenbasiertAlt“ * Ausnahmeszenarien: - |

|  |  |
| --- | --- |
| **User Story-ID** | CharakterKampfunfähigkeit |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass Charaktere kampfunfähig werden, wenn sie zu viel Schaden einstecken oder aus der Spielkarte herausfallen. |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor(en)** |  |
| **Abhängigkeiten zu anderen User Stories** | CharakterAngriff, LandschaftIT1 |
| **Zugehörige Szenarien** | * Hauptszenario: „kampfunfähig“ * Alternativszenarien: „kampfunfähigAlt“ * Ausnahmeszenarien: - |

|  |  |
| --- | --- |
| **User Story-ID** | Siegbedingung |
| **User Story-Beschreibung** | Als Spieler möchte ich die Partie gewinnen, wenn alle gegnerischen Charaktere kampfunfähig sind. |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor(en)** |  |
| **Abhängigkeiten zu anderen User Stories** | CharakterKampfunfähigkeit |
| **Zugehörige Szenarien** | * Hauptszenario: - * Alternativszenarien: - * Ausnahmeszenarien: - |

|  |  |
| --- | --- |
| **User Story-ID** | SpielerZug |
| **User Story-Beschreibung** | Als Spieler möchte ich in meinem Zug meinen Charakter bewegen und mit einer Waffe einmal schießen. |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor(en)** |  |
| **Abhängigkeiten zu anderen User Stories** | CharakterAngriff, CharakterBewegen, CharakterSpringen |
| **Zugehörige Szenarien** | * Hauptszenario: „SpielerZug“ * Alternativszenarien: „SpielerZugAlt“ * Ausnahmeszenarien: - |

|  |  |
| --- | --- |
| **User Story-ID** | CharakterAngriff |
| **User Story-Beschreibung** | Als Spieler möchte ich mit einer Waffe zunächst den Abschusswinkel bestimmen und danach Projektile abschießen, um gegnerischen Charakteren Schaden zuzufügen. |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor(en)** |  |
| **Abhängigkeiten zu anderen User Stories** |  |
| **Zugehörige Szenarien** | * Hauptszenario: „Waffe benutzen“ * Alternativszenarien: - * Ausnahmeszenarien: - |

## Szenarien

Hauptszenario „Rundenbasiert“:

1. Spieler 1 ist an der Reihe.
2. Spieler 1 führt Bewegung aus und/oder greift an.
3. Spieler 1 beendet Zug
4. Spieler 2 bekommt signalisiert, dass er an der Reihe ist.
5. Spieler 2 führt Bewegung aus und/oder greift an.
6. Spieler 2 beendet Zug
7. Spieler 1 bekommt signalisiert, dass er an der Reihe ist.

Alternativszenario „RundenbasiertAlt“:

1. Spieler 1 ist an der Reihe.
2. Spieler 1 macht nichts und lässt die Zeit seines Zuges verstreichen.
3. Zug von Spieler 1 wird automatisch beendet.
4. Spieler 2 bekommt signalisiert, dass er an der Reihe ist.

Hauptszenario „kampfunfähig“:

1. Projektil trifft Spielcharakter
2. Spiel rechnet Schadenswert an
3. Spiel prüft, ob der Charakter noch Leben übrig hat.
4. Spiel erklärt Charakter für kampfunfähig
5. Spiel signalisiert dies beiden Spielern

Alternativszenario „kampfunfähigAlt“:

1. Charakter bewegt sich
2. Spiel prüft Position des Charakters auf Gültigkeit
3. Spiel definiert Charakter als 'aus der Welt gefallen' und somit kampfunfähig
4. Spiel signalisiert dies beiden Spielern

Hauptszenario „gewinnen“:

1. Spielcharakter von einem Spieler wird kampfunfähig
2. Spiel prüft, ob noch andere Charaktere am Leben sind
3. Spiel erklärt Spieler als Verlierer, da alle Charaktere kampfunfähig sind.
4. Spiel zeigt dies an, sowie dem Gewinner, dass er siegreich war
5. Spielrunde wird beendet.

Hauptszenario „SpielerZug“:

1. Spieler bekommt signalisiert, dass er an der Reihe ist
2. Spieler bewegt seinen Charakter.
3. Spieler erreicht maximale Bewegdistanz.
4. Spieler wählt Waffe aus.
5. Spieler benutzt die Waffe.
6. Projektil wird mit der Kamera verfolgt bis es trifft.
7. Runde wird beendet.
8. Spieler 2 wird benachrichtigt, dass er an der Reihe ist.

Alternativszenario „SpielerZugAlt“:

1. Spieler bekommt signalisiert, dass er an der Reihe ist
2. Spieler wählt direkt Waffe aus.
3. Spieler zielt mit der Waffe.

Hauptszenario „Bewegung“:

1. Spieler drückt Bewegungstaste
2. Spiel prüft Bewegung auf Richtigkeit (kein Hindernis im Weg, Bewegungsmaximaldistanz nicht erreicht)
3. Spiel führt Bewegung aus

Ausnahmeszenario „Bewegung blockiert“:

1. Spieler drückt Bewegungstaste
2. Spiel prüft Bewegung auf Richtigkeit
3. Spiel stellt fehlerhafte Bewegung fest.
4. Bewegung wird nicht ausgeführt.

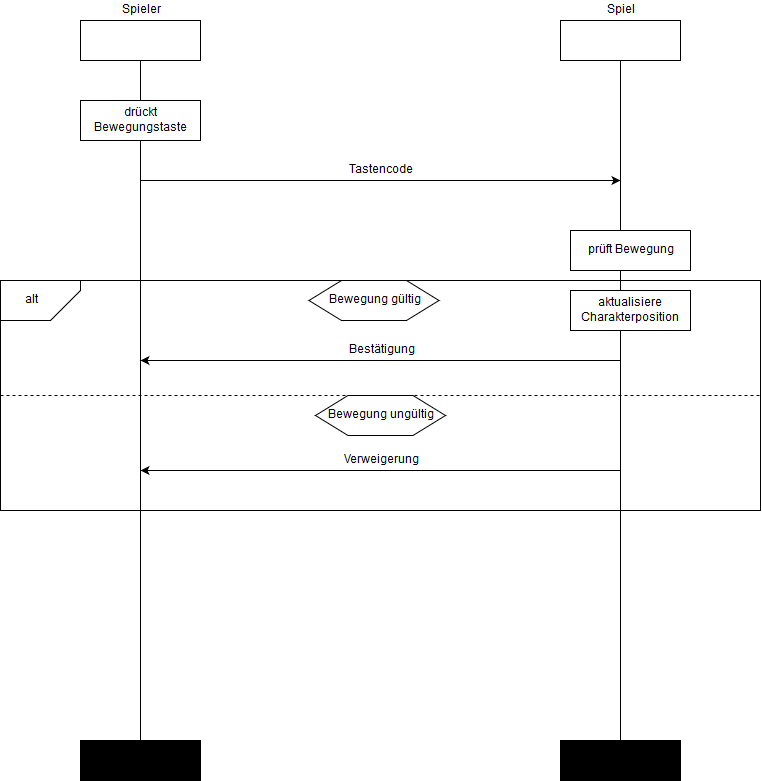
Hauptszenario „Waffe benutzen“:

1. Spieler wählt Waffe aus
2. Ziel GUI wird eingeblendet
3. Spieler bewegt die Maus zum zielen
4. Spieler drückt 'Feuer' Taste
5. Projektil wird abgeschossen

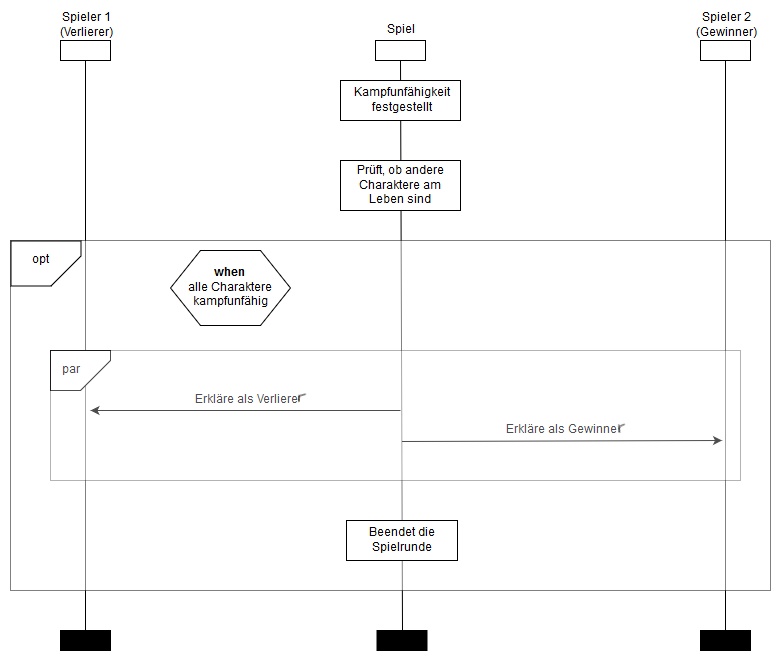
## MSCs

MSC-Diagramme inkl. Bezeichner und, falls nötig, erläuterndem Text (beispielsweise getroffene Annahmen)

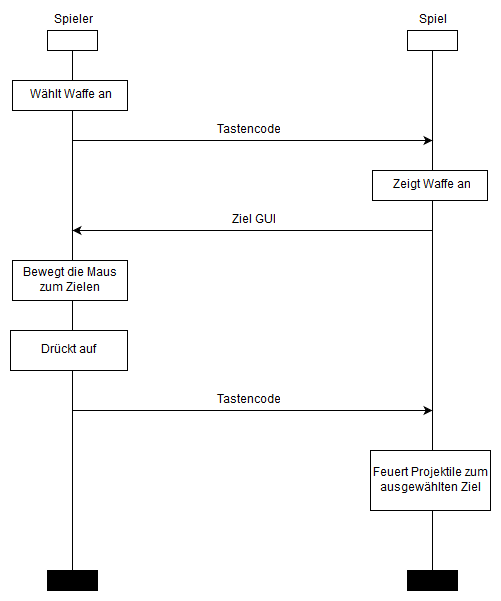
bMSC Bewegung



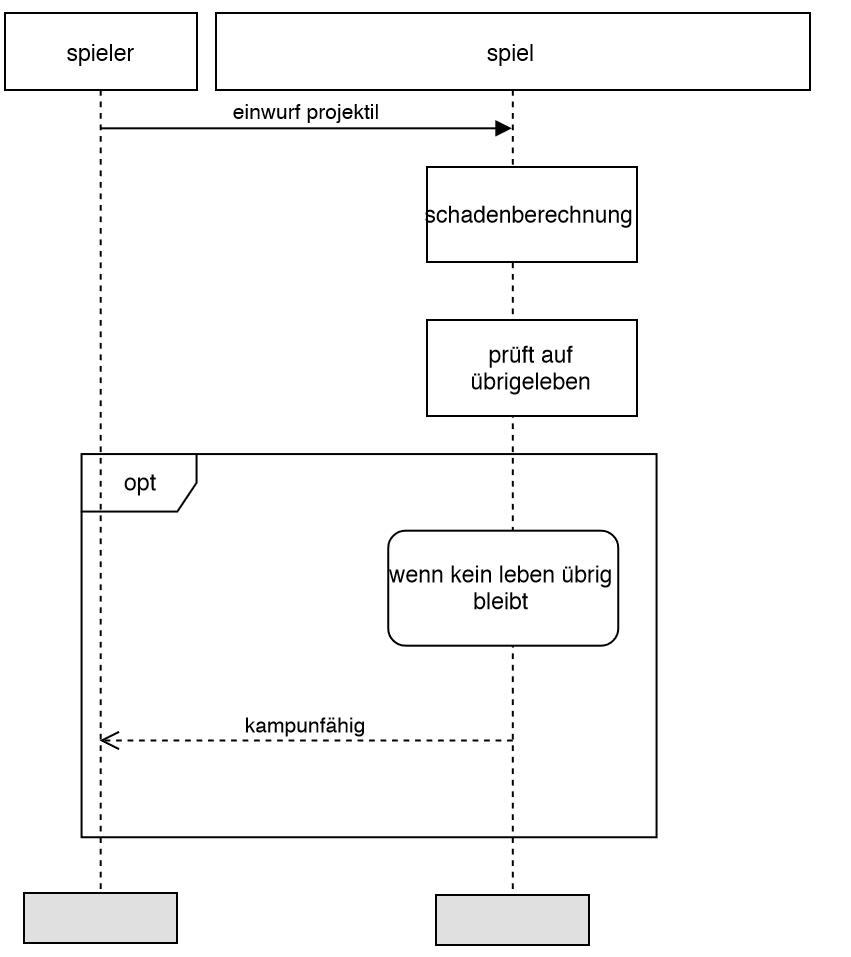
bMSC Gewinnen



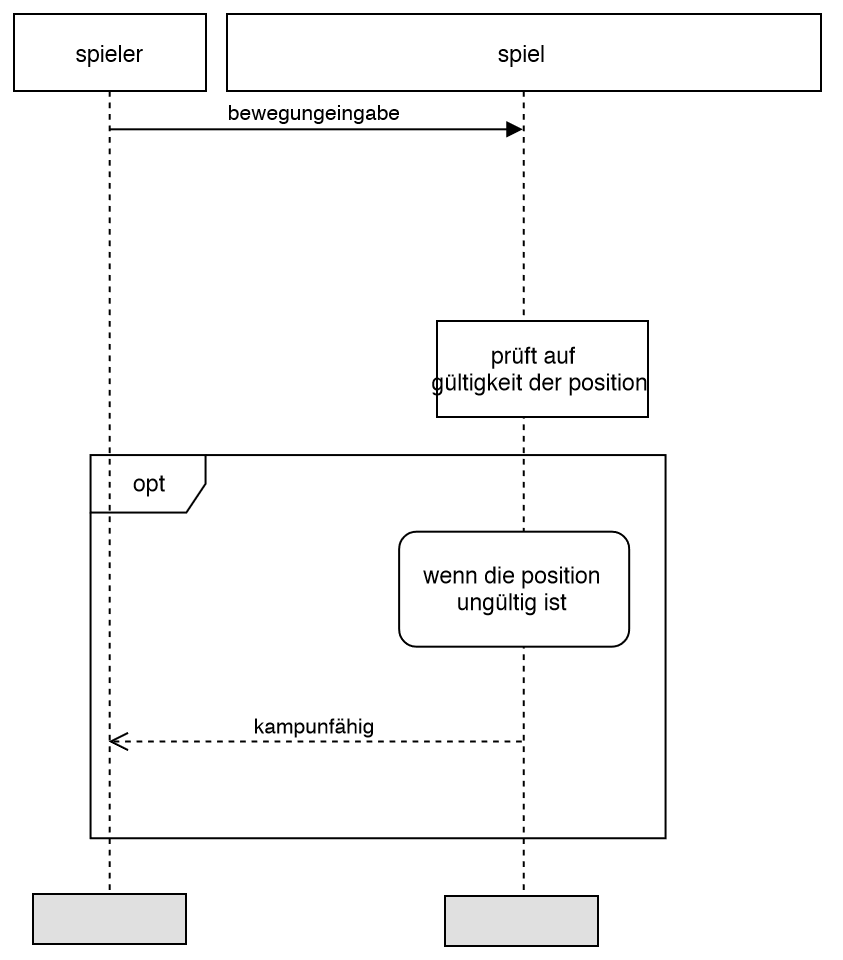
bMSC Waffe benutzen



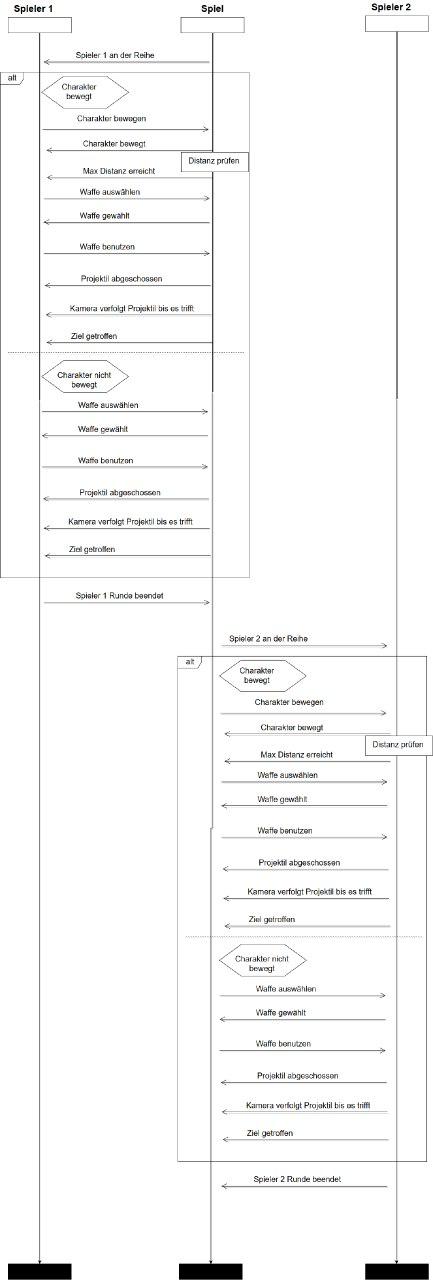
bMSC kampfunfähigSchaden



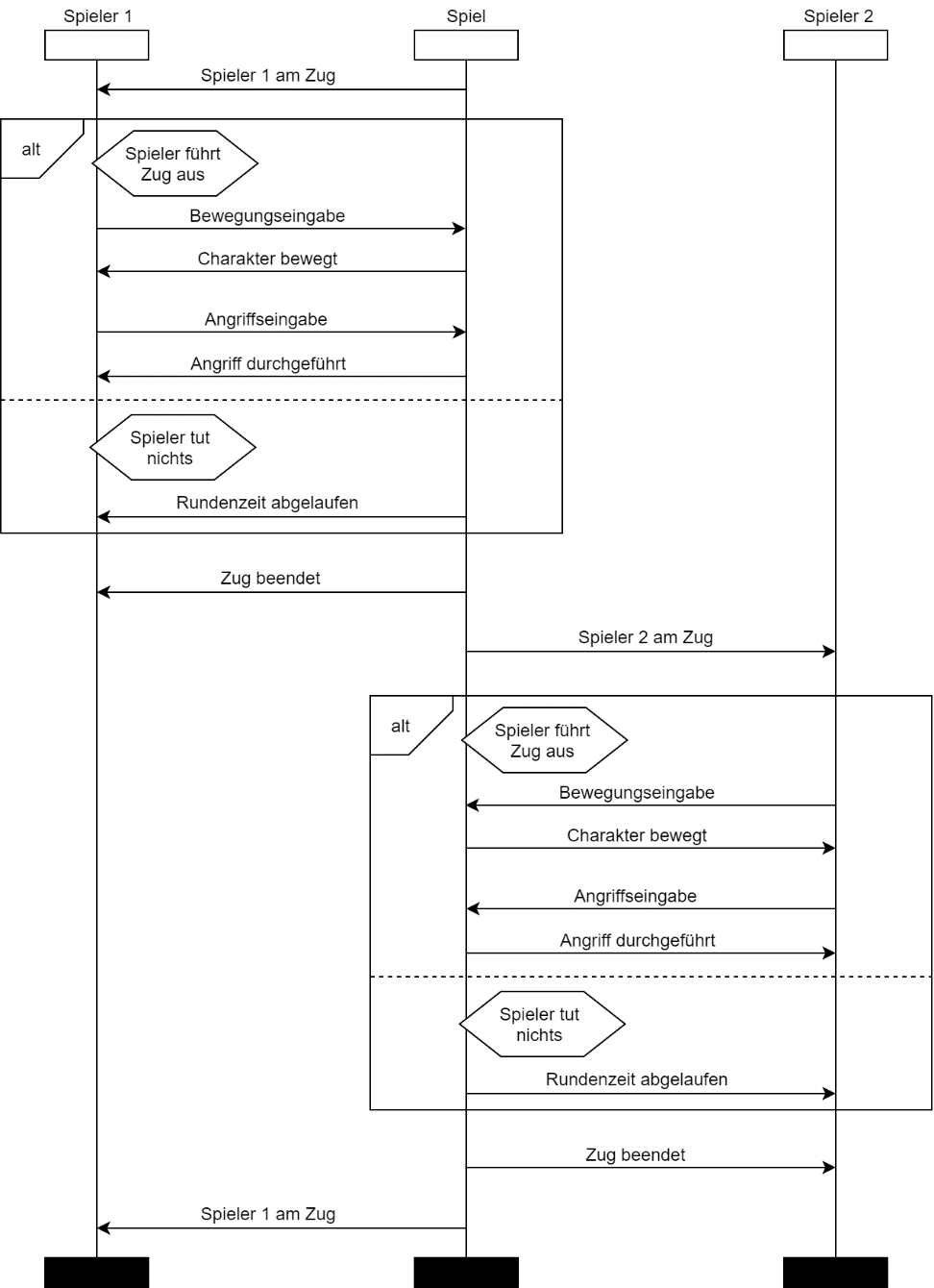
bMSC kampfunfähigUmgebung



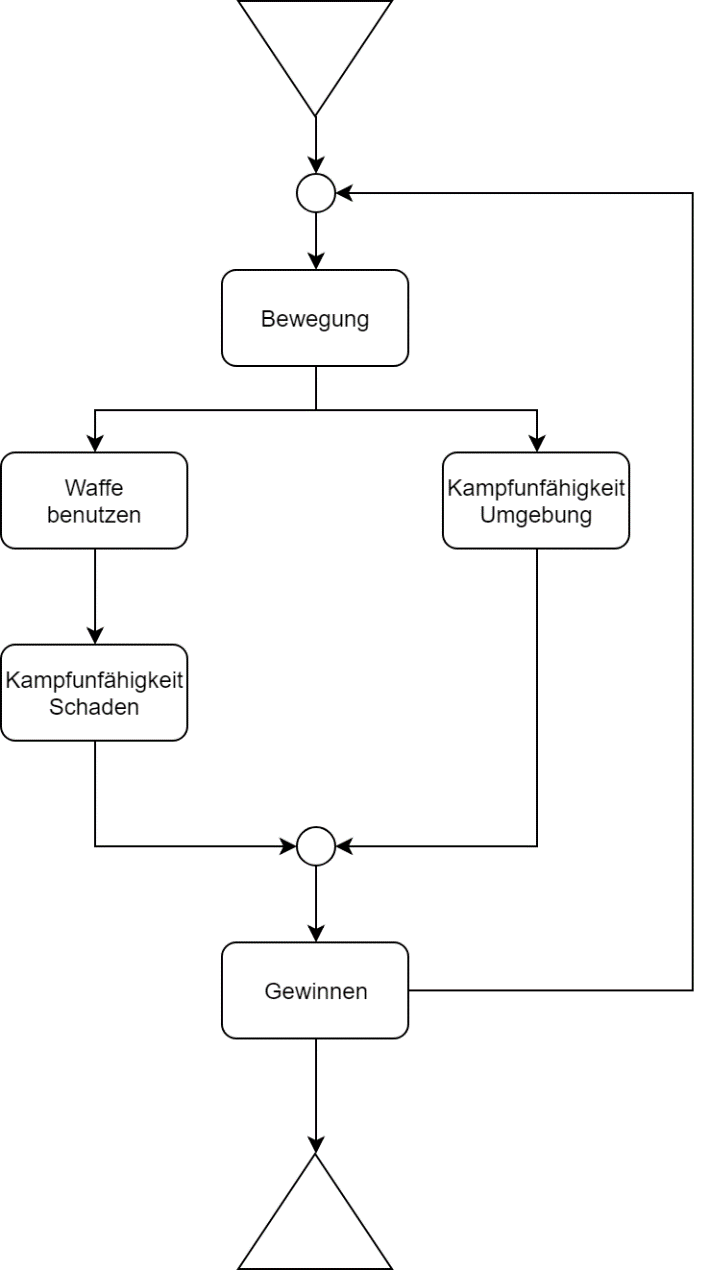
bMSC SpielerZug



bMSC Rundenbasiert



hMSC



## UML Klassendiagramm

Erläuterung des Klassendiagramms:

## Funktionalitätsplanung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Quellcode-**  **referenz** | **Status** |
| **1.** | **Route** |  |  |  |  |  |
| 1.1 | Suche nach POI/ Straßen | 1 Tag | Max Muster | 1.2 |  | fertig |
| 1.2 | Erstellen von Routen | 2 Tage | Max Muster | 1.3, 1.4 |  | In Bearbeitung |
| 1.2.1 | …. |  |  |  |  |  |
| **2.** | **Anzeige** |  |  |  |  |  |
| 2.1 | Lokales Speichern der Routen | 4 Tage | Anja Muster | 2.4, 2,7 |  | fertig |
| … |  |  |  |  |  |  |

Template:

## Unittests

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Getestete Funktionalität** | **Quellcode**  **Referenz** | **Status** |
| **…** |  |  |  |

# Iteration II

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** | <Eindeutiger Identifizierer> |
| **User Story-Beschreibung** | <Text der User Story mittels Satzschablone (s. Foliensatz „Anforderungen“)> |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** | <Wichtigkeit der User Story hinsichtlich der Aufgabenstellung> |
| **Autor(en)** | Max Mustermann |
| **Abhängigkeiten zu anderen User Stories** | <Auflistung verwandter User Stories> |
| **Zugehörige Szenarien** | <Zuordnung zu Szenarien>   * Hauptszenario: * Alternativszenarien: * Ausnahmeszenarien: |

## MSCs

MSC-Diagramme inkl. Bezeichner und, falls nötig, erläuterndem Text (beispielsweise getroffene Annahmen)

## UML Klassendiagramm

Erläuterung des Klassendiagramms:

## Funktionalitätsplanung

Template:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Quellcodereferenz** | **Status** |
| **…** |  |  |  |  |  |  |

## Unittests

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Getestete Funktionalität** | **Quellcode**  **Referenz** | **Status** |
| **…** |  |  |  |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 05.05.2017 | | |
| **Tester** | Martina Musterfrau | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt den Benutzername „Max Mustermann“ auf der Tastatur ein. | Das System zeigt „Max Mustermann“ auf dem Display an. | **√** |
| 2 | Der Benutzer gibt das Passwort „geheim“ auf der Tastatur ein. | Das System zeigt das Passwort durch „\*“-Symbole zensiert an. | **√** |
| 3 | Der Benutzer klickt auf „Anmelden“. | Das System zeigt die Meldung „Anmeldung erfolgreich“ auf dem Display an. | **X** |
| **Nachbe-dingung(en)** | Nutzer ist am System angemeldet, Anmeldezeitpunkt ist im System gespeichert. | | **X** |
| **Testurteil** | Test nicht bestanden. | | |

# Iteration III

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** | <Eindeutiger Identifizierer> |
| **User Story-Beschreibung** | <Text der User Story mittels Satzschablone (s. Foliensatz „Anforderungen“)> |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** | <Wichtigkeit der User Story hinsichtlich der Aufgabenstellung> |
| **Autor(en)** | Max Mustermann |
| **Abhängigkeiten zu anderen User Stories** | <Auflistung verwandter User Stories> |
| **Zugehörige Szenarien** | <Zuordnung zu Szenarien>   * Hauptszenario: * Alternativszenarien: * Ausnahmeszenarien: |

## MSCs

MSC-Diagramme inkl. Bezeichner und, falls nötig, erläuterndem Text (beispielsweise getroffene Annahmen)

## UML Klassendiagramm

Erläuterung des Klassendiagramms:

## Funktionalitätsplanung

Vorlage:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Quellcodereferenz** | **Status** |
| **…** |  |  |  |  |  |  |

## Unittests

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Getestete Funktionalität** | **Quellcode**  **Referenz** | **Status** |
| **…** |  |  |  |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 05.05.2017 | | |
| **Tester** | Martina Musterfrau | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt den Benutzername „Max Mustermann“ auf der Tastatur ein. | Das System zeigt „Max Mustermann“ auf dem Display an. | **√** |
| 2 | Der Benutzer gibt das Passwort „geheim“ auf der Tastatur ein. | Das System zeigt das Passwort durch „\*“-Symbole zensiert an. | **√** |
| 3 | Der Benutzer klickt auf „Anmelden“. | Das System zeigt die Meldung „Anmeldung erfolgreich“ auf dem Display an. | **X** |
| **Nachbe-dingung(en)** | Nutzer ist am System angemeldet, Anmeldezeitpunkt ist im System gespeichert. | | **X** |
| **Testurteil** | Test nicht bestanden. | | |

# Nutzerhandbuch

## Technische Anforderungen

Technische Mindestanforderungen, welche das Programm benötigt, um wie gewünscht bedienbar zu sein.

## Installationsanleitung

Genaue Erläuterung, wie das entwickelte Programm vollkommen funktionsfähig auf einem Rechner in Betrieb genommen werden kann.

## Bedienungsanleitung

Genaue Erläuterung, wie das entwickelte Programm zu bedienen ist.