SEP Projektmappe

***PROJEKTMAPPE DES PROJEKTES***

SprEading Peace

**Dokumentation des Projektes**

Gruppe 29 – Spiel

Daumann, Lars Christian

Jüssen, Cedric

Knothe, Patrick

Rongen, Robin

Rühl, Florian

Schiller, Niklas

Schneider, Joel

Inhaltsverzeichnis

[Projektbeschreibung 4](#__RefHeading___Toc4979_2631342851)

[Einleitung 4](#__RefHeading___Toc4981_2631342851)

[Iteration l 4](#__RefHeading___Toc4983_2631342851)

[Iteration ll 4](#__RefHeading___Toc4985_2631342851)

[Iteration lll 5](#__RefHeading___Toc4987_2631342851)

[Iteration l 7](#__RefHeading___Toc5020_2631342851)

[User Stories Spiel 7](#__RefHeading___Toc4989_2631342851)

[bMSCs 11](#__RefHeading___Toc5058_2631342851)

[1.01 11](#__RefHeading___Toc5060_2631342851)

[1.02 12](#__RefHeading___Toc5062_2631342851)

[1.03 13](#__RefHeading___Toc5064_2631342851)

[1.04 14](#__RefHeading___Toc5066_2631342851)

[1.05 16](#__RefHeading___Toc5068_2631342851)

[1.06 16](#__RefHeading___Toc5070_2631342851)

[1.07 17](#__RefHeading___Toc5072_2631342851)

[1.08 17](#__RefHeading___Toc5074_2631342851)

[1.09, 1.10 18](#__RefHeading___Toc5076_2631342851)

[1.11 19](#__RefHeading___Toc5078_2631342851)

[1.12, 1.13 19](#__RefHeading___Toc5080_2631342851)

[1.14 20](#__RefHeading___Toc5082_2631342851)

[1.15 21](#__RefHeading___Toc5084_2631342851)

[hMsc 22](#__RefHeading___Toc5086_2631342851)

[UML-Klassendiagramm 23](#__RefHeading___Toc5088_2631342851)

[Funktionalitätsplanung 24](#__RefHeading___Toc9343_3881277333)

[Iteration ll 27](#__RefHeading___Toc5022_2631342851)

[User-Stories - Spiel – Iteration 2 27](#__RefHeading___Toc4993_2631342851)

[User Stories Modus „Deathmatch“: 28](#__RefHeading___Toc5090_2631342851)

[User Stories Modus „Zombie“: 29](#__RefHeading___Toc5092_2631342851)

[User Stories „Features“: 30](#__RefHeading___Toc4995_2631342851)

[bMSCs 31](#__RefHeading___Toc5094_2631342851)

[2.01 31](#__RefHeading___Toc5096_2631342851)

[2.02 32](#__RefHeading___Toc5098_2631342851)

[2.03 33](#__RefHeading___Toc5100_2631342851)

[2.04 34](#__RefHeading___Toc5102_2631342851)

[2.05 35](#__RefHeading___Toc9345_3881277333)

[2.06 36](#__RefHeading___Toc5106_2631342851)

[2.07 36](#__RefHeading___Toc5108_2631342851)

[2.08 37](#__RefHeading___Toc5110_2631342851)

[2.09 38](#__RefHeading___Toc5112_2631342851)

[2.10 39](#__RefHeading___Toc5114_2631342851)

[2.11 40](#__RefHeading___Toc5116_2631342851)

[2.12 40](#__RefHeading___Toc5118_2631342851)

[hMSC 41](#__RefHeading___Toc4997_2631342851)

[UML Klassendiagramm 42](#__RefHeading___Toc4999_2631342851)

[Funktionalitätsplanung 43](#__RefHeading___Toc5001_2631342851)

[Unittests 45](#__RefHeading___Toc5003_2631342851)

[Systemtests 47](#__RefHeading___Toc5024_2631342851)

[Iteration lll 49](#__RefHeading___Toc5026_2631342851)

[User Stories Spiel 49](#__RefHeading___Toc5007_2631342851)

[MSCs 54](#__RefHeading___Toc5009_2631342851)

[UML Klassendiagramm 67](#__RefHeading___Toc5011_2631342851)

[Funktionalitätsplanung 68](#__RefHeading___Toc5013_2631342851)

[Unittests 79](#__RefHeading___Toc9347_3881277333)

[Systemtests 79](#__RefHeading___Toc5017_2631342851)

[Technische Anforderungen 80](#__RefHeading___Toc9349_3881277333)

[Installationsanleitung 80](#__RefHeading___Toc9351_3881277333)

[Bedienungsanleitung 80](#__RefHeading___Toc9353_3881277333)

# Projektbeschreibung

## Einleitung

In dem zu entwickelnden Spiel, inspiriert von dem Atari 2600 Titel „Combat“, sollen Friedenspanzer in einem Wettstreit gegeneinander antreten, dessen Ziel es ist, die gegnerischen Friedenspanzer mit Blumen zu schmücken. Hat ein Friedenspanzer zu viel Blumenbeschuss erlitten, so wird er zur friedlichen Aufgabe überredet. Das Spielgeschehen soll dabei aus der Vogelperspektive gezeigt werden.

## Iteration l

In der ersten Iteration soll eine erste Version des Spiels mit reduziertem Umfang entwickelt werden. Zunächst soll nur ein Spieler einen Friedenspanzer mit der Tastatur, der Maus und Tastatur, oder dem Gamepad steuern können. Der Friedenspanzer soll fahren, sich drehen und Blumen verschießen können. Zusätzlich zu dem Friedenspanzer soll nur eine Zielscheibe in der leeren Spielwelt existieren. Wird die Zielscheibe von einer Blume des Friedenspanzers getroffen, so soll sie verschwinden und an einer anderen, zufällig gewählten Stelle neu erscheinen.

## Iteration ll

In der zweiten Iteration soll das Grundgerüst des Spiels vervollständigt werden, vor allem sollen zwei Spielmodi erstellt werden und die Zielscheibe aus der ersten Iteration soll es nun nicht mehr geben. Im ersten Spielmodus sollen bis zu fünf Friedenspanzer gleichzeitig gegeneinander antreten können. Der Spielmodus soll von einem Spieler allein und von zwei Spielern gemeinsam gespielt werden können, wobei jeder Spieler einen eigenen Friedenspanzer steuert. Die Friedenspanzer sollen auf fünf Teams aufgeteilt werden können. Die nicht spielergesteuerten Friedenspanzer sollen von einer KI gesteuert werden.

In dieser Iteration sollen zusätzlich Wände in der Spielwelt existieren, die nicht durchfahren oder durchschossen werden können. Die KI soll die gegnerischen Friedenspanzer auf sinnvolle Weise bekämpfen, was vor allem bedeutet, dass sie nicht an Wänden stecken bleibt und aktiv ihre Gegner jagen. Alle Friedenspanzer sollen über eine Friedensanzeige verfügen, die den Spielern ermöglicht zu sehen, wie viel Blumenbeschuss der Friedenspanzer noch ertragen kann, bevor er zur Aufgabe überredet wird. Im zweiten Spielmodus soll ein spielergesteuerter Friedenspanzer gegen eine unbegrenzte Anzahl nachrückender, KI-gesteuerter Friedenspanzern antreten und dabei möglichst viele dieser zur Aufgabe überreden. Je länger der Spieler es schafft, nicht selbst zur Aufgabe überredet zu werden, desto schwieriger soll dieser Spielmodus werden, indem immer mehr Friedenspanzer gleichzeitig den Spieler bekämpfen. Nach dem Spiel soll dann die Anzahl der überredeten KI-Gegner in einer Highscore-Tabelle, die sich die Spieler zwischen den Spielen anschauen können, gespeichert werden.

Darüber hinaus sollen die Friedenspanzer in dieser Iteration auch während des Fahrens in eine andere Richtung schießen können. Das heißt, dass sich die Kanone nun unabhängig von der Bewegung des Friedenspanzers steuern lassen soll. Hierbei soll es in dieser Iteration den Spielern auch möglich sein die Tastenbelegung ihrer Steuerung frei konfigurieren können. Zuletzt soll die bearbeitende Gruppe sich selbst 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 lll

In der letzten Iteration soll der Umfang des Spiels erweitert werden. Die Spieler sollen nicht nur auf einer, sondern auf drei verschiedenen Karten spielen können, die sich darin unterscheiden, wo die Wände stehen. Die Friedenspanzer sollen neben dem normalen Typ noch drei weitere Blumentypen verschießen können: einen Typ, der schneller fliegt, aber die Friedensanzeige des getroffenen Friedenspanzers weniger stark füllt; einen Typ, der bis zu zweimal von Wänden abprallt und ebenfalls die Friedensanzeige des getroffenen Friedenspanzers weniger stark füllt; und einen Typ, der sich im Flug in drei kleinere Blumen aufteilt, die langsam fliegen und nach einiger Zeit im Flug von alleine verschwinden. Der ausgewählte Blumentyp der Friedenspanzer soll dabei immer sichtbar sein.

Außerdem soll das Spielgeschehen durch Soundeffekte untermalt werden und die Spieler sollen den Friedenspanzern jeweils eins von sechs verschiedenem Aussehen zuweisen können. Abschließend soll die bearbeitende Gruppe sich auch in dieser Iteration selbst wieder drei weitere Features für das Spiel ausdenken. Diese Features sollen zum Thema der Aufgabenstellung passen und vor der Umsetzung mit dem Gruppenbetreuer abgesprochen werden.

*Anmerkungen*

* Das Spiel ist in der Programmiersprache Java zu entwickeln.
* Wir veröffentlichen besonders gelungene Software auf unserer SEP-Webseite. Hierzu ist es unbedingt erforderlich, dass das gesamte Spiel keine urheberrechtlich geschützten Inhalte (Bilder, Musik, etc.) enthält.

# Iteration l

## User Stories Spiel

|  |  |
| --- | --- |
| User-Story-ID | 1.01 |
| User-Story-  Beschreibung | Als Nutzer möchte ich eine Zielscheibe haben um ein Spielziel verfolgen zu können |
| Geschätzter  Realisierungsaufwand | 3 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories | / |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 1.02 |
| User-Story-  Beschreibung | Als Nutzer möchte ich eine Spielfigur haben um ein Spielziel verfolgen zu können |
| Geschätzter  Realisierungsaufwand | 4 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 1.03 |
| User-Story-  Beschreibung | Als Nutzer möchte ich den Panzer vorwärts, rückwärts, nach rechts/links und diagonal bewegen können um ihn auf dem Spielbrett positionieren zu können |
| Geschätzter  Realisierungsaufwand | 4 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories | 1.02 |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 1.04 |
| User-Story-  Beschreibung | Als Nutzer möchte ich die Spielfigur drehen können um sie auf dem Spielbrett positionieren zu können |
| Geschätzter  Realisierungsaufwand | 4 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories | 1.02 |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 1.05 |
| User-Story-  Beschreibung | Als Nutzer möchte ich die Spielfigur mit der Computertastatur steuern können um auf die Zielscheibe zielen zu können |
| Geschätzter  Realisierungsaufwand | 2 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories | 1.02 |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 1.06 |
| User-Story-  Beschreibung | Als Nutzer möchte ich die Spielfigur mit der Maus und der  Computertastatur steuern können um auf die Zielscheibe zielen zu können |
| Geschätzter  Realisierungsaufwand | 2 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories | 1.02 |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 1.07 |
| User-Story-  Beschreibung | Als Nutzer möchte ich die Spielfigur mit dem Gamepad steuern können um auf die Zielscheibe zielen zu können |
| Geschätzter  Realisierungsaufwand | 2 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories | 1.02 |
| Zugehörige Szenarien |  |
| User-Story-ID | 1.08 |
| User-Story-  Beschreibung | Als Nutzer möchte ich mit der Spielfigur Blumen verschießen können um die Zielscheibe treffen zu können |
| Geschätzter  Realisierungsaufwand | 5 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories | 1.01, 1.02 |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 1.09 |
| User-Story-  Beschreibung | Als Nutzer möchte ich die Zielscheibe mit geschossenen Blumen verschwinden lassen können um ein Spielziel zu haben |
| Geschätzter  Realisierungsaufwand | 3 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories | 1.01, 1.02, 1.08 |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 1.10 |
| User-Story-  Beschreibung | Als Nutzer möchte ich eine neue Zielscheibe an einem neuen zufälligen Ort haben um nach erreichen des ursprünglichen Ziels ein neues zu haben |
| Geschätzter  Realisierungsaufwand | 4 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories | 1.01, 1.02, 1.08 |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 1.11 |
| User-Story-  Beschreibung | Als Nutzer möchte ich einen Schuss pro Zeiteinheitsbegrenzung (z.B. 1 Schuss pro 0,3 Sek.) haben um nicht aus Zufall einen Treffer zu erreichen |
| Geschätzter  Realisierungsaufwand | 3 Stunden |
| Priorität | Niedrig |
| Autor(en) | Cedric Jüssen |
| Abhängigkeiten zu anderen User-Stories | 1.08 |
| Zugehörige Szenarien | Schuss |
| User-Story-ID | 1.12 |
| User-Story-  Beschreibung | Als Nutzer möchte eine Spielwelt haben, damit ich mich in ihr bewegen kann. |
| Geschätzter  Realisierungsaufwand | 4 Stunden |
| Priorität | Hoch |
| Autor(en) | Niklas Schiller |
| Abhängigkeiten zu anderen User-Stories | / |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 1.13 |
| User-Story-  Beschreibung | Als Nutzer möchte ich, dass die Spielwelt begrenzt ist, damit ich mich in ihr zurechtfinden kann. |
| Geschätzter  Realisierungsaufwand | 2 Stunden |
| Priorität | Hoch |
| Autor(en) | Niklas Schiller |
| Abhängigkeiten zu anderen User-Stories | 1.12 |
| Zugehörige Szenarien |  |

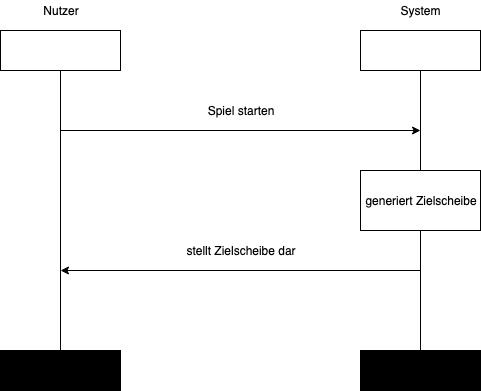
|  |  |
| --- | --- |
| User-Story-ID | 1.14 |
| User-Story-  Beschreibung | Als Nutzer möchte ich, dass die abgeschossenen Blumen verschwinden, wenn sie auf ein Hindernis treffen. |
| Geschätzter  Realisierungsaufwand | 1,5 Stunden |
| Priorität | Mittel |
| Autor(en) | Niklas Schiller |
| Abhängigkeiten zu anderen User-Stories | 1.06, 1.07, 1.08 |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 1.15 |
| User-Story-  Beschreibung | Als Nutzer möchte ich, dass der Panzer nicht durch Hindernisse fahren kann um den Spielbereich abgrenzen zu können |
| Geschätzter  Realisierungsaufwand | 4 Stunden |
| Priorität | Hoch |
| Autor(en) | Florian Rühl |
| Abhängigkeiten zu anderen User-Stories | 1.12, 1.13, 1.14 |
| Zugehörige Szenarien |  |

## bMSCs

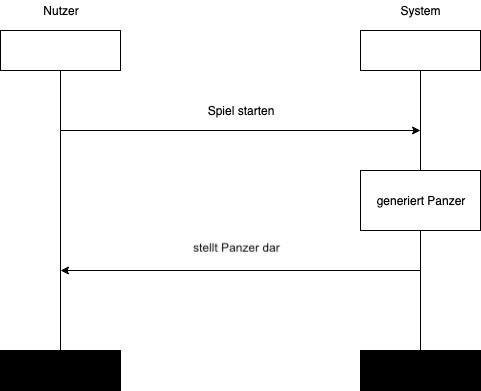
### 1.01

* Als Nutzer möchte ich eine Zielscheibe haben um ein Spielziel verfolgen zu können



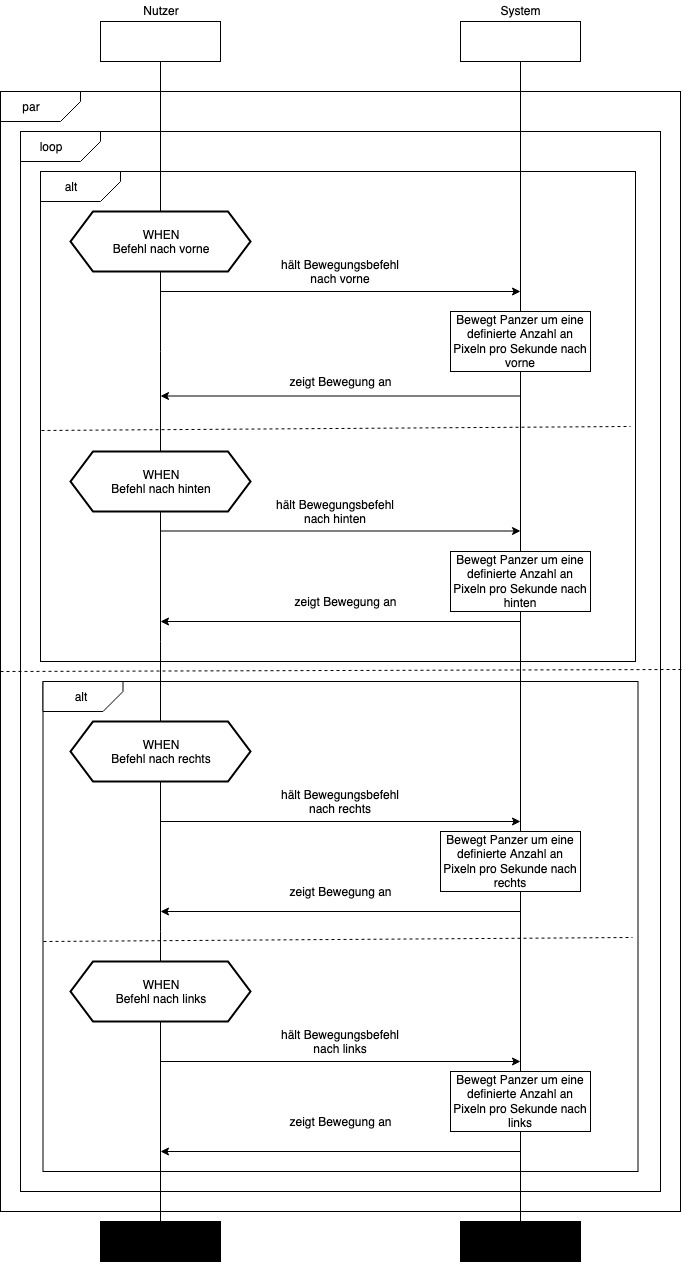
### 1.02

* Als Nutzer möchte ich eine Spielfigur haben um ein Spielziel verfolgen zu können



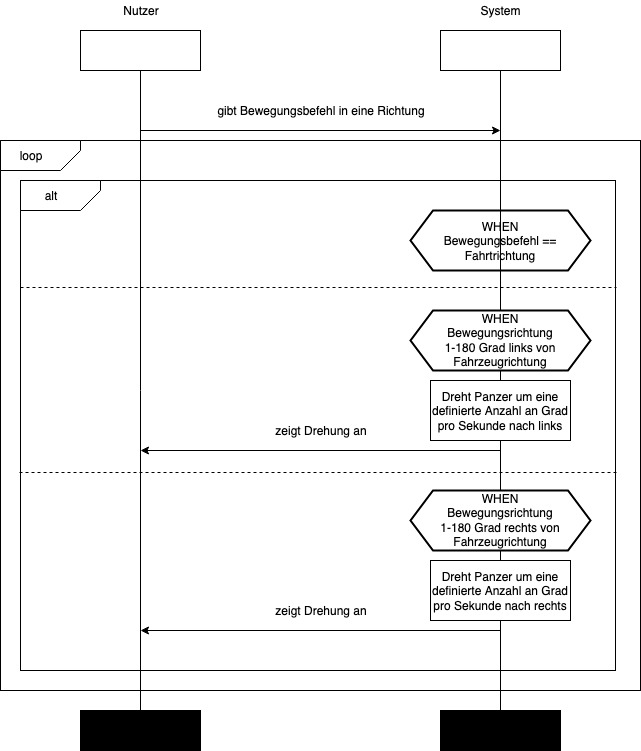
### 1.03

* Als Nutzer möchte ich den Panzer vorwärts, rückwärts, nach rechts/links und diagonal bewegen können um ihn auf dem Spielbrett positionieren zu können



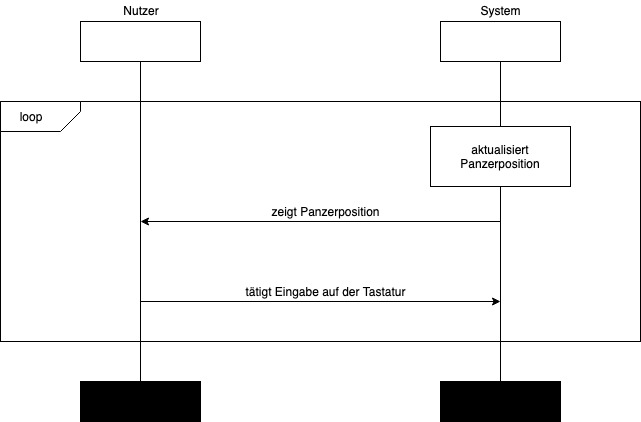
### 1.04

* Als Nutzer möchte ich die Spielfigur drehen können um sie auf dem Spielbrett positionieren zu können



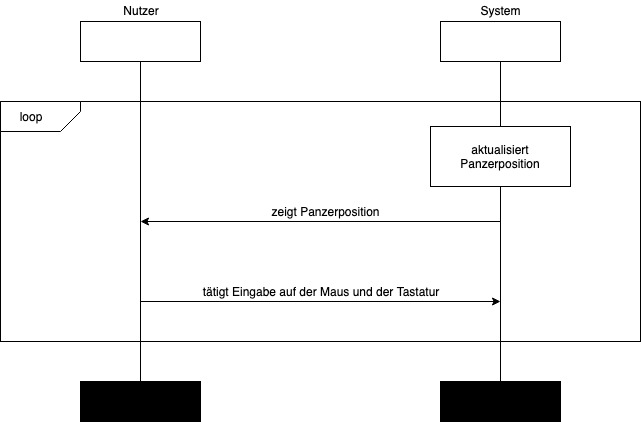
### 1.05

* Als Nutzer möchte ich die Spielfigur mit der Computertastatur steuern können um auf die Zielscheibe zielen zu können



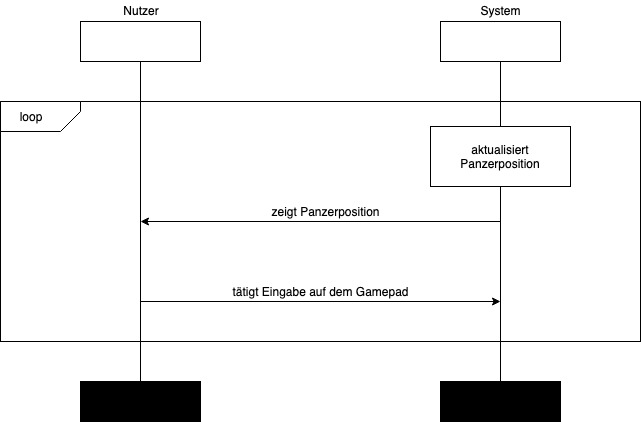
### 1.06

* Als Nutzer möchte ich die Spielfigur mit der Maus und der Computertastatur steuern können um auf die Zielscheibe zielen zu können



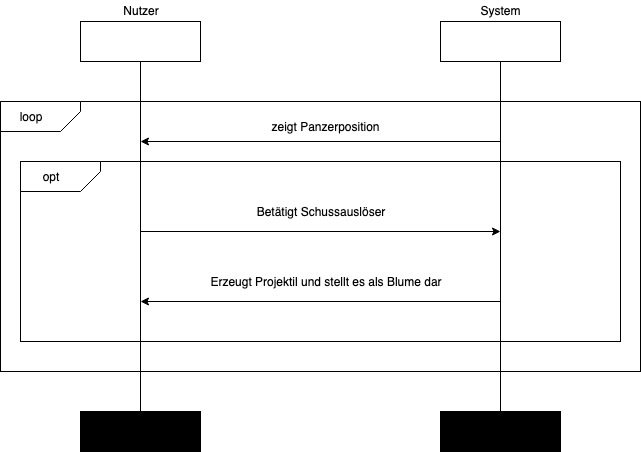
### 1.07

* Als Nutzer möchte ich die Spielfigur mit dem Gamepad steuern können um auf die Zielscheibe zielen zu können



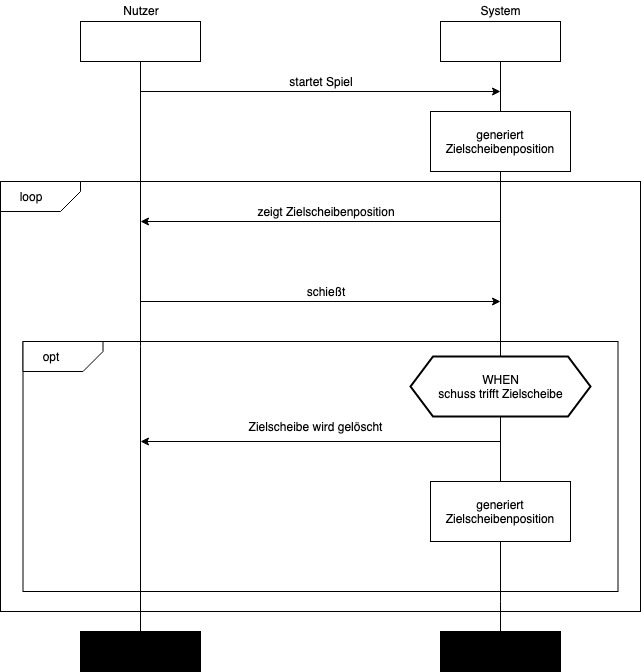
### 1.08

* Als Nutzer möchte ich mit der Spielfigur Blumen verschießen können um die Zielscheibe treffen zu können



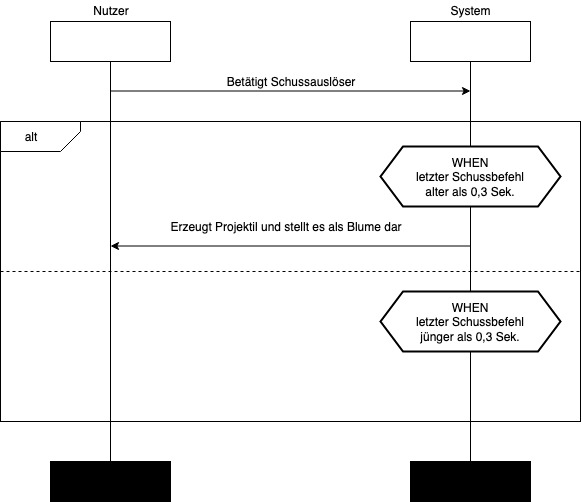
### 1.09, 1.10

* Als Nutzer möchte ich die Zielscheibe mit geschossenen Blumen verschwinden lassen können um ein Spielziel zu haben
* Als Nutzer möchte ich eine neue Zielscheibe an einem neuen zufälligen Ort haben um nach erreichen des ursprünglichen Ziels ein neues zu haben



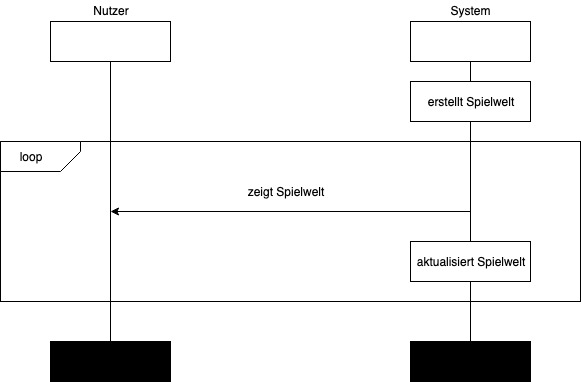
### 1.11

* Als Nutzer möchte ich einen Schuss pro Zeiteinheitsbegrenzung (z.B. 1 Schuss pro 0,3 Sek.) haben um nicht aus Zufall einen Treffer zu erreichen



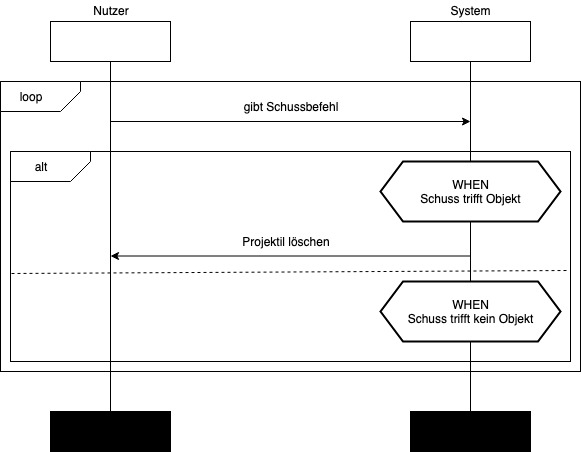
### 1.12, 1.13

* Als Nutzer möchte eine Spielwelt haben, damit ich mich in ihr bewegen kann.
* Als Nutzer möchte ich, dass die Spielwelt begrenzt ist, damit ich mich in ihr zurechtfinden kann.



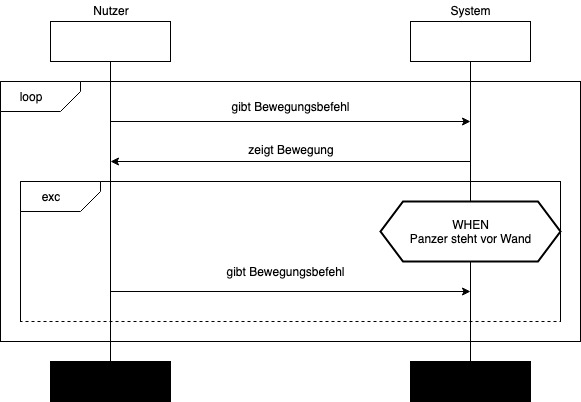
### 1.14

* Als Nutzer möchte ich, dass die abgeschossenen Blumen verschwinden, wenn sie auf ein Hindernis treffen.

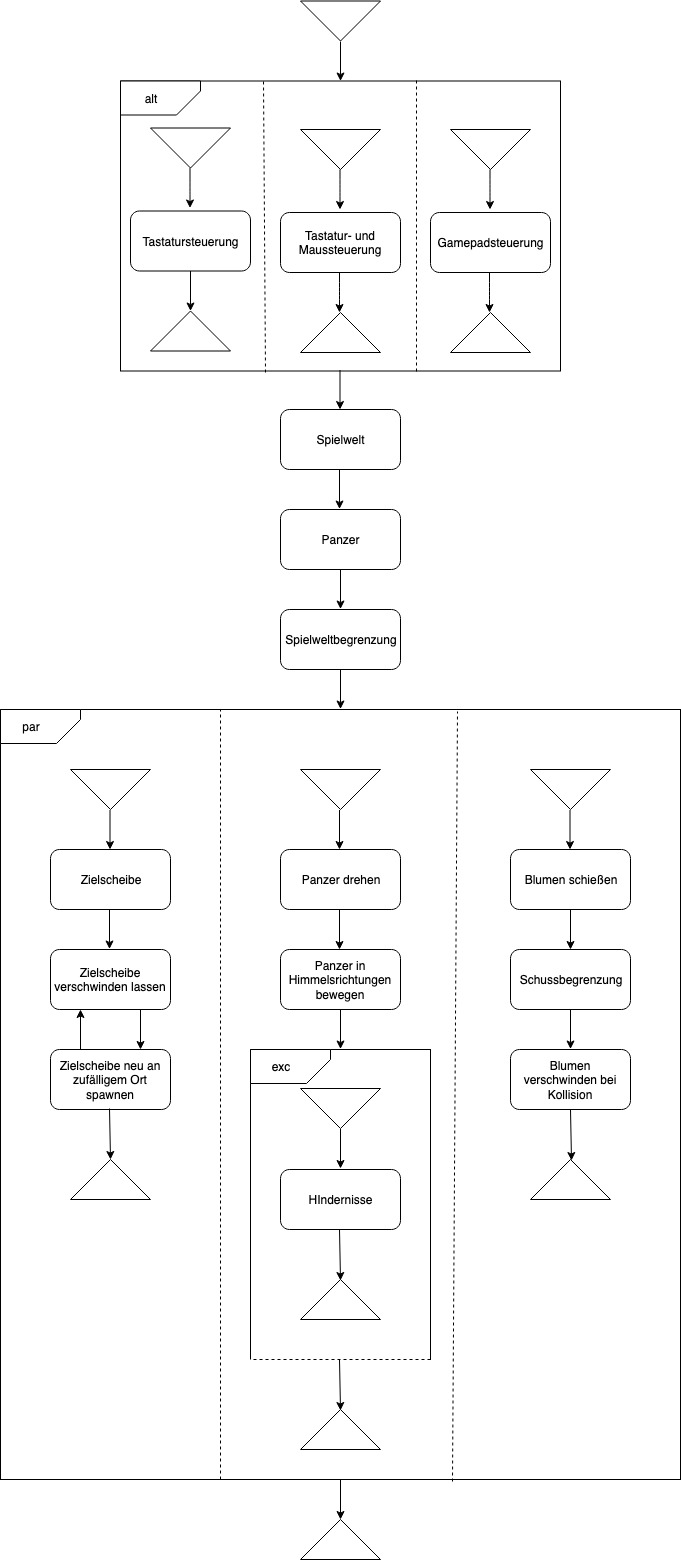


### 1.15

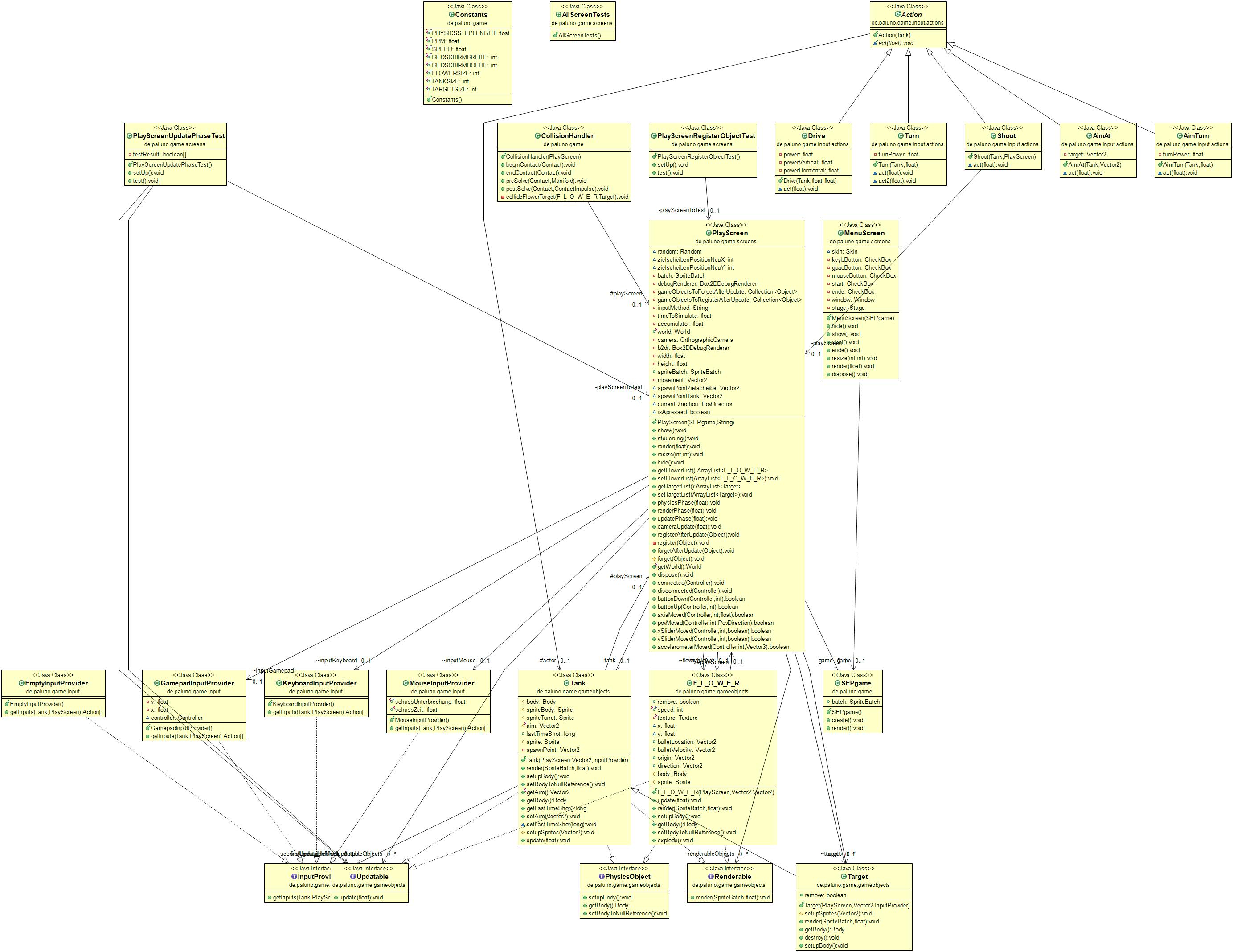
* Als Nutzer möchte ich, dass der Panzer nicht durch Hindernisse fahren kann um den Spielbereich abgrenzen zu können



## hMsc



## UML-Klassendiagramm



## Funktionalitätsplanung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Funktionalität | Geschätzter Aufwand | Verantwortlicher | Abhängige Funktionalitäten | Quellcode-referenz | Status |
| 1. | SEPGame |  | Florian Rühl |  | - | Fertig |
| 2. | Constants |  |  |  |  | Fertig |
| 3. | CollisionHandler |  |  |  |  | Fertig |
| 3.1 | beginContact() | 4h | Niklas Schiller |  | 29-68 | Fertig |
| 3.2 | collideFlowerTarget() | 0,5h | Robin Rongen |  | 84-90 | Fertig |
| 4. | F\_L\_O\_W\_E\_R |  |  |  |  | Fertig |
| 4.1 | F\_L\_O\_W\_E\_R() | 0,5h | Cedric Jüssen |  | 39-53 | Fertig |
| 4.2 | Update() | 0,5h | Robin Rongen |  | 55-61 | Fertig |
| 4.3 | Render() | 0,5h | Cedric Jüssen |  | 63-67 | Fertig |
| 4.4 | setupBody() | 2h | Robin Rongen |  | 69-95 | Fertig |
| 4.5 | Explode() | 1h | Joel Schneider |  | 109-130 | Fertig |
| 5. | Tank | Insgesamt: 4h |  |  |  | Fertig |
| 5.1 | Tank() |  | Florian Rühl | Playscreen |  | Fertig |
| 5.2 | Render() |  | Florian Rühl | Playscreen |  | Fertig |
| 5.3 | setupBody() |  | Florian Rühl | Playscreen |  | Fertig |
| 5.4 | Update() |  | Florian Rühl | Playscreen |  | Fertig |
| 6. | Target |  |  | Tank |  | Fertig |
| 6.1 | Target() | 10min | Patrick Knothe |  | 27-33 | Fertig |
| 6.2 | Render() | 2min | Patrick Knothe |  | 40-42 | Fertig |
| 6.3 | Destroy() | 1h | Joel Schneider |  | 50-71 | Fertig |
| 6.4 | setupBody() | 2h | Robin Rongen |  | 82-110 | Fertig |
| 7. | GamepadInputProvider |  |  |  |  | Fertig |
| 7.1 | Action() |  |  |  |  | Fertig |
| 8. | KeyboardInputProvider |  |  |  |  | Fertig |
| 8.1 | Action() | 1h | Lars Daumann |  |  | Fertig |
| 9. | MouseInputProvider |  |  |  |  | Fertig |
| 9.1 | Action() | 0.5h | Joel Schneider | KeyboardInputProvider |  | Fertig |
| 10. | AimAt |  |  |  |  | Fertig |
| 11. | AimTurn | 0.5h | Lars Daumann |  |  | Fertig |
| 12. | Drive | 1h | Robin Rongen |  |  | Fertig |
| 13. | Shoot | 0.5h | Lars Daumann |  |  | Fertig |
| 14. | Turn |  |  |  |  | Fertig |
| 14.1 | Turn() | 0.5h | Cedric Jüssen |  | 0-18 |  |
| 14.2 | Act() | 0.5h | Lars Daumann |  | 19-22 | Fertig |
| 14.3 | Act2() | 0.5h | Cedric Jüssen |  | 23-27 | Fertig |
| 15. | MenuScreen |  |  |  |  | Fertig |
| 15.1 | MenuScreen() | 1h | Patrick Knothe |  | 41-50 | Fertig |
| 15.2 | Show() | 2h | Patrick Knothe |  | 56-74 | Fertig |
| 15.3 | Start() | 0.5h | Lars Daumann |  | 76-88 | Fertig |
| 15.4 | Ende() | 5min | Lars Daumann |  | 90-93 | Fertig |
| 15.5 | Resize() | 2min | Patrick Knothe |  | 96-98 | Fertig |
| 15.6 | Render() | 20min | Patrick Knothe |  | 100-110 | Fertig |
| 15.7 | Dispose() | 5min | Patrick Knothe |  | 112-115 | Fertig |
| 16. | PlayScreen |  |  |  |  | Fertig |
| 16.1 | PlayScreen() | 0.5h | Joel Schneider |  | 84-92 | Fertig |
| 16.2 | Show() | 1h | Niklas Schiller |  | 94-110 | Fertig |
| 16.3 | Steuerung() | 0.5h | Lars Daumann |  | 112-123 | Fertig |
| 16.4 | Render() | 2h | Patrick Knothe |  | 125-174 | Fertig |
| 16.5 | physicsPhase() | 0.5 | Robin Rongen |  | 194-203 | Fertig |
| 16.6 | Dispose() | 5min | Patrick Knothe |  | 278-282 | Fertig |

# Iteration ll

## User-Stories - Spiel – Iteration 2

|  |  |
| --- | --- |
| User-Story-ID | 2.01 |
| User-Story-  Beschreibung | Als Nutzer möchte ich, dass Wände in der Spielwelt existieren um mich vor den Gegnern verstecken zu können |
| Geschätzter  Realisierungsaufwand | 3 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 2.02 |
| User-Story-  Beschreibung | Als Nutzer möchte ich eine Lebensanzeige über meiner Figur haben um jederzeit sehen zu können, wie viele Schüsse ich noch aushalte, ohne zu verlieren |
| Geschätzter  Realisierungsaufwand | 2 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 2.03 |
| User-Story-  Beschreibung | Als Nutzer möchte ich die Möglichkeit haben während der Fahrt in eine Richtung, in eine andere Richtung schießen zu können um meine Spielweise aggressiv oder defensiv gestalten zu können |
| Geschätzter  Realisierungsaufwand | 3 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 2.04 |
| User-Story-  Beschreibung | Als Nutzer möchte ich die Steuerung und Tastenbelegung für meinen Panzer selbst bestimmen können um nach meinen Wünschen besser spielen zu können |
| Geschätzter  Realisierungsaufwand | 3 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

## User Stories Modus „Deathmatch“:

|  |  |
| --- | --- |
| User-Story-ID | 2.05 |
| User-Story-  Beschreibung | Als Nutzer möchte ich mit zwei Spielern gleichzeitig spielen können um mich mit Freunden messen zu können |
| Geschätzter  Realisierungsaufwand | 4 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 2.06 |
| User-Story-  Beschreibung | Als Nutzer möchte ich die Spielfiguren in fünf Teams aufteilen können um verschiedene Feindgruppen zu haben |
| Geschätzter  Realisierungsaufwand | 2 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 2.07 |
| User-Story-  Beschreibung | Als Nutzer möchte ich, dass die nicht spielergesteuerten Gegner trotzdem klug agieren um das Spiel spannender zu gestalten |
| Geschätzter  Realisierungsaufwand | 4 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

## User Stories Modus „Zombie“:

|  |  |
| --- | --- |
| User-Story-ID | 2.08 |
| User-Story-  Beschreibung | Als Nutzer möchte ich, dass dieser Spielmodus mit jedem überredeten Friedenspanzer schwieriger wird um eine Herausforderung zu haben |
| Geschätzter  Realisierungsaufwand | 3 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 2.09 |
| User-Story-  Beschreibung | Als Nutzer möchte ich zwischen den Spielen die Möglichkeit haben meine Highscores in einer Tabelle einsehen zu können, die persistent gespeichert werden um mich immer weiter verbessern zu können |
| Geschätzter  Realisierungsaufwand | 5 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

## User Stories „Features“:

|  |  |
| --- | --- |
| User-Story-ID | 2.10 |
| User-Story-  Beschreibung | Als Nutzer möchte ich ein Geschwindigkeitsboost haben um aus bestimmten Situationen besser fliehen zu können |
| Geschätzter  Realisierungsaufwand | 2 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

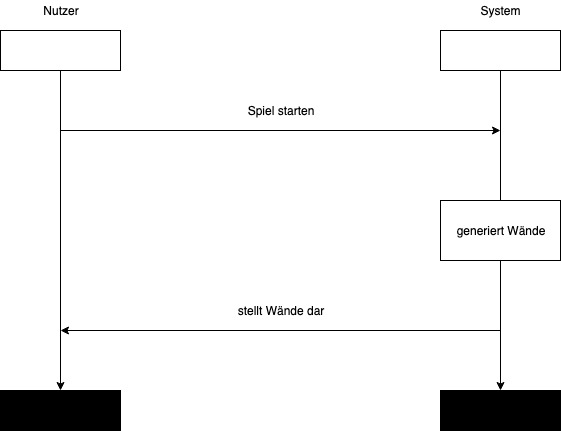
|  |  |
| --- | --- |
| User-Story-ID | 2.11 |
| User-Story-  Beschreibung | Als Nutzer möchte ich ein Heal-pack haben um mich in bestimmten Situationen heilen zu können |
| Geschätzter  Realisierungsaufwand | 2 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

|  |  |
| --- | --- |
| User-Story-ID | 2.12 |
| User-Story-  Beschreibung | Als Nutzer möchte ich ein Poison-Pack haben um einen erhöhten Schwierigkeitsgrad für mich und die gegnerischen Panzer zu haben |
| Geschätzter  Realisierungsaufwand | 2 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

## bMSCs

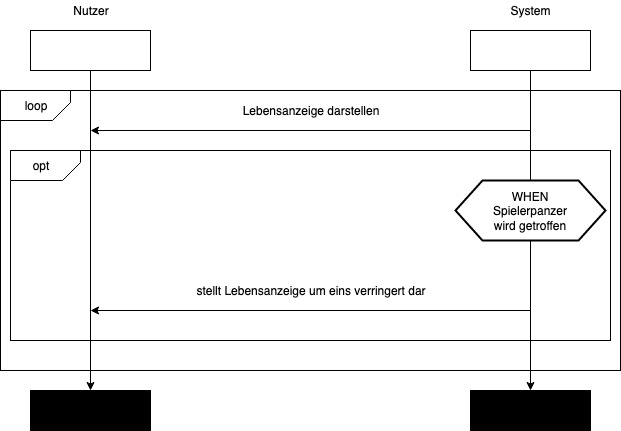
### 2.01

* Als Nutzer möchte ich, dass Wände in der Spielwelt existieren um mich vor den Gegnern verstecken zu können



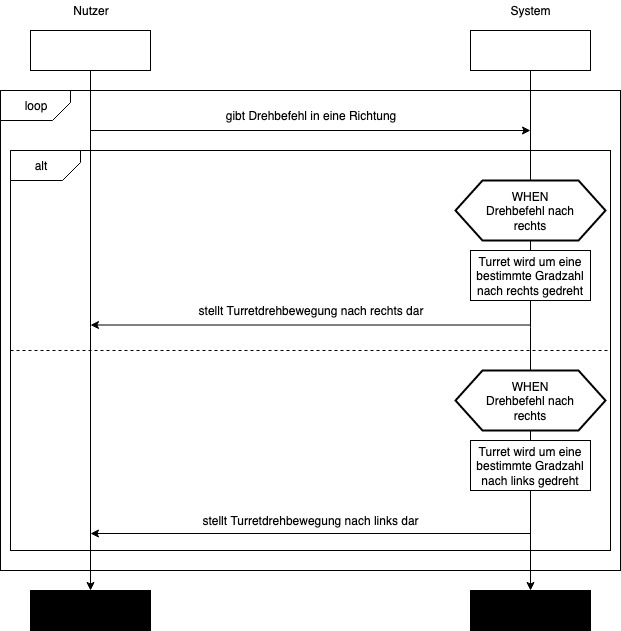
### 2.02

* Als Nutzer möchte ich eine Lebensanzeige über meiner Figur haben um jederzeit sehen zu können, wie viele Schüsse ich noch aushalte, ohne zu verlieren



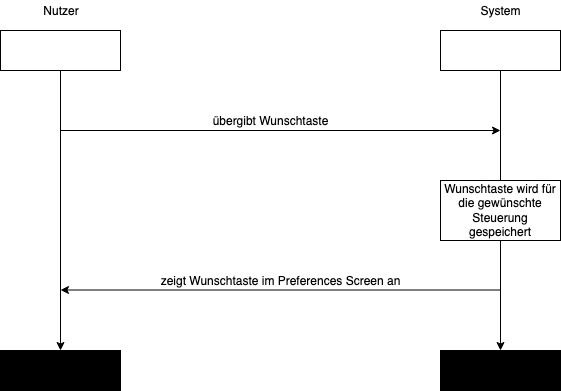
### 2.03

* Als Nutzer möchte ich die Möglichkeit haben während der Fahrt in eine Richtung, in eine andere Richtung schießen zu können um meine Spielweise aggressiv oder defensiv gestalten zu können



### 2.04

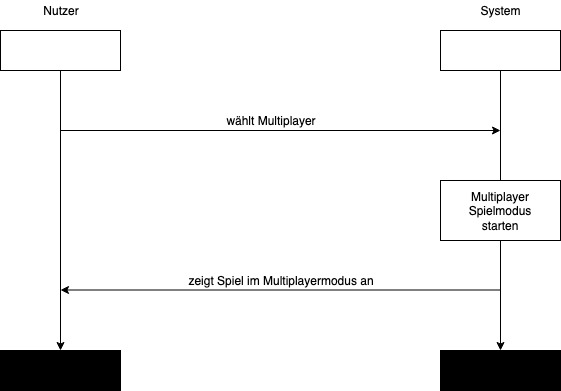
* Als Nutzer möchte ich die Steuerung und Tastenbelegung für meinen Panzer selbst bestimmen können um nach meinen Wünschen besser spielen zu können



### 

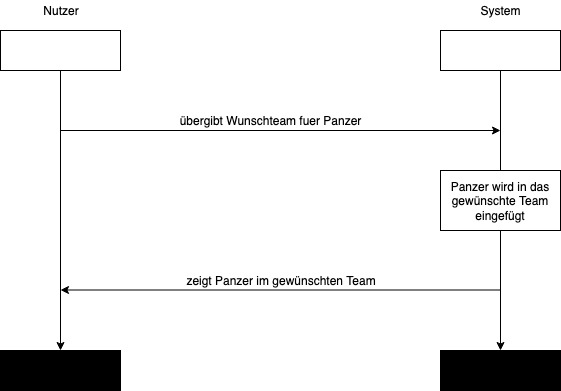
### 2.05

* Als Nutzer möchte ich mit zwei Spielern gleichzeitig spielen können um mich mit Freunden messen zu können



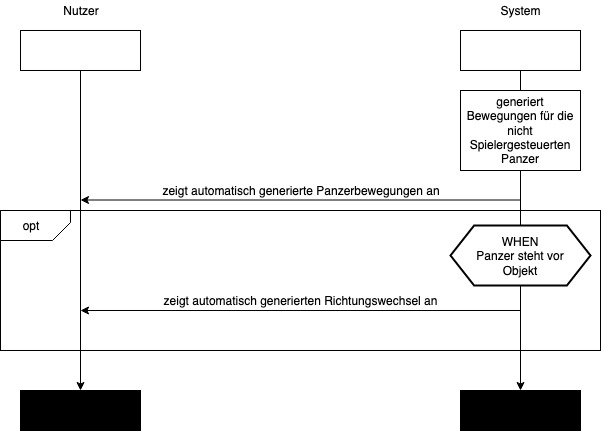
### 2.06

* Als Nutzer möchte ich die Spielfiguren in fünf Teams aufteilen können um verschiedene Feindgruppen zu haben



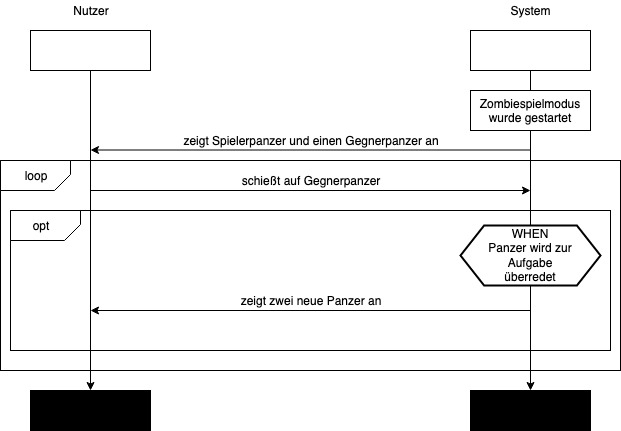
### 2.07

* Als Nutzer möchte ich, dass die nicht spielergesteuerten Gegner trotzdem klug agieren um das Spiel spannender zu gestalten



### 2.08

* Als Nutzer möchte ich, dass dieser Spielmodus mit jedem überredeten Friedenspanzer schwieriger wird um eine Herausforderung zu haben



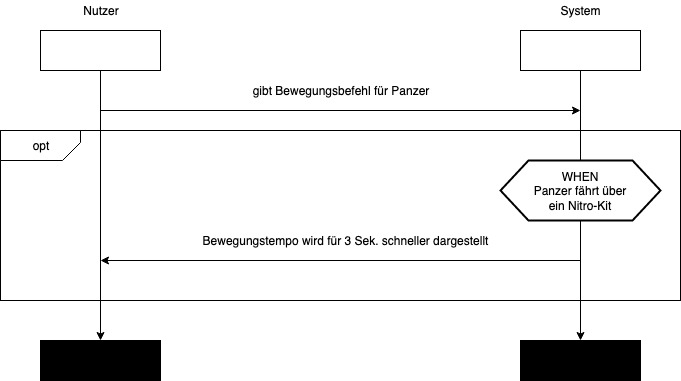
### 2.09

* Als Nutzer möchte ich zwischen den Spielen die Möglichkeit haben meine Highscores in einer Tabelle einsehen zu können, die persistent gespeichert werden um mich immer weiter verbessern zu können



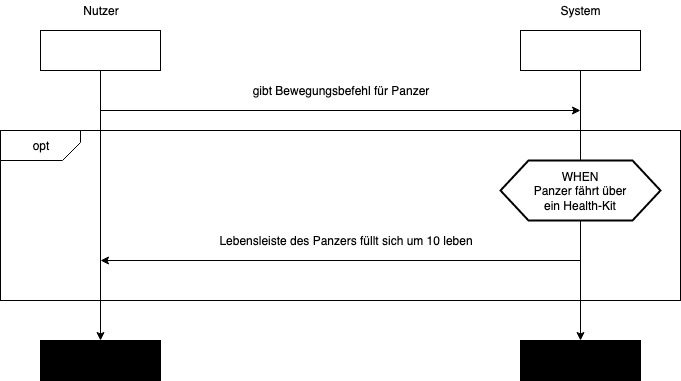
### 2.10

* Als Nutzer möchte ich ein Geschwindigkeitsboost haben um aus bestimmten Situationen besser fliehen zu können



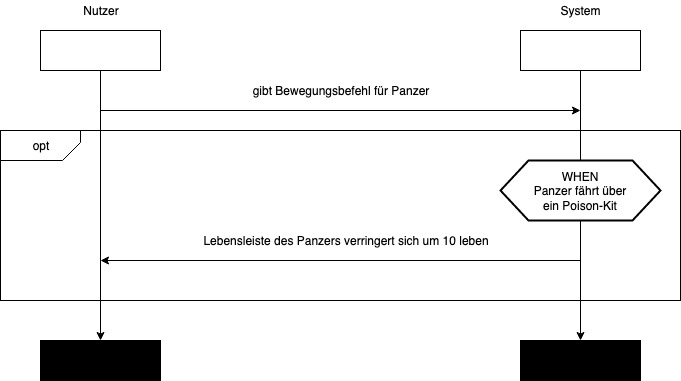
### 2.11

* Als Nutzer möchte ich ein Heal-pack haben um mich in bestimmten Situationen heilen zu können

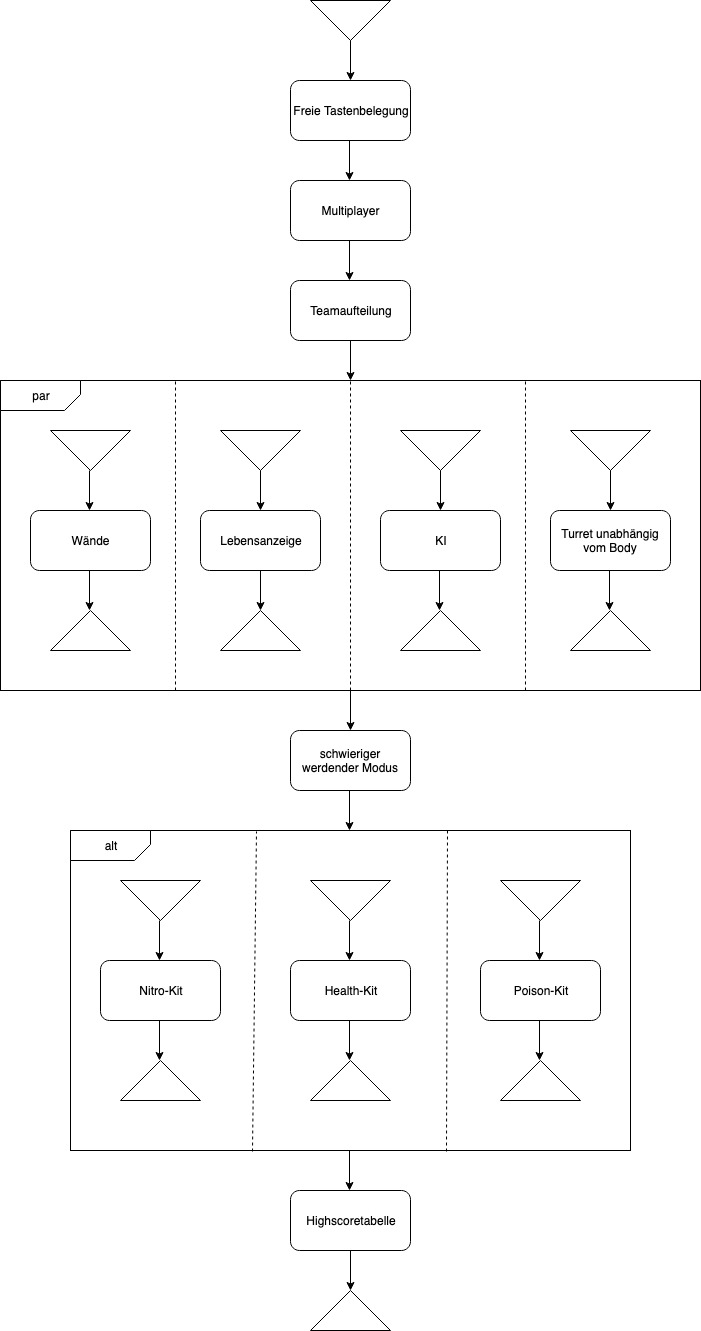


### 2.12

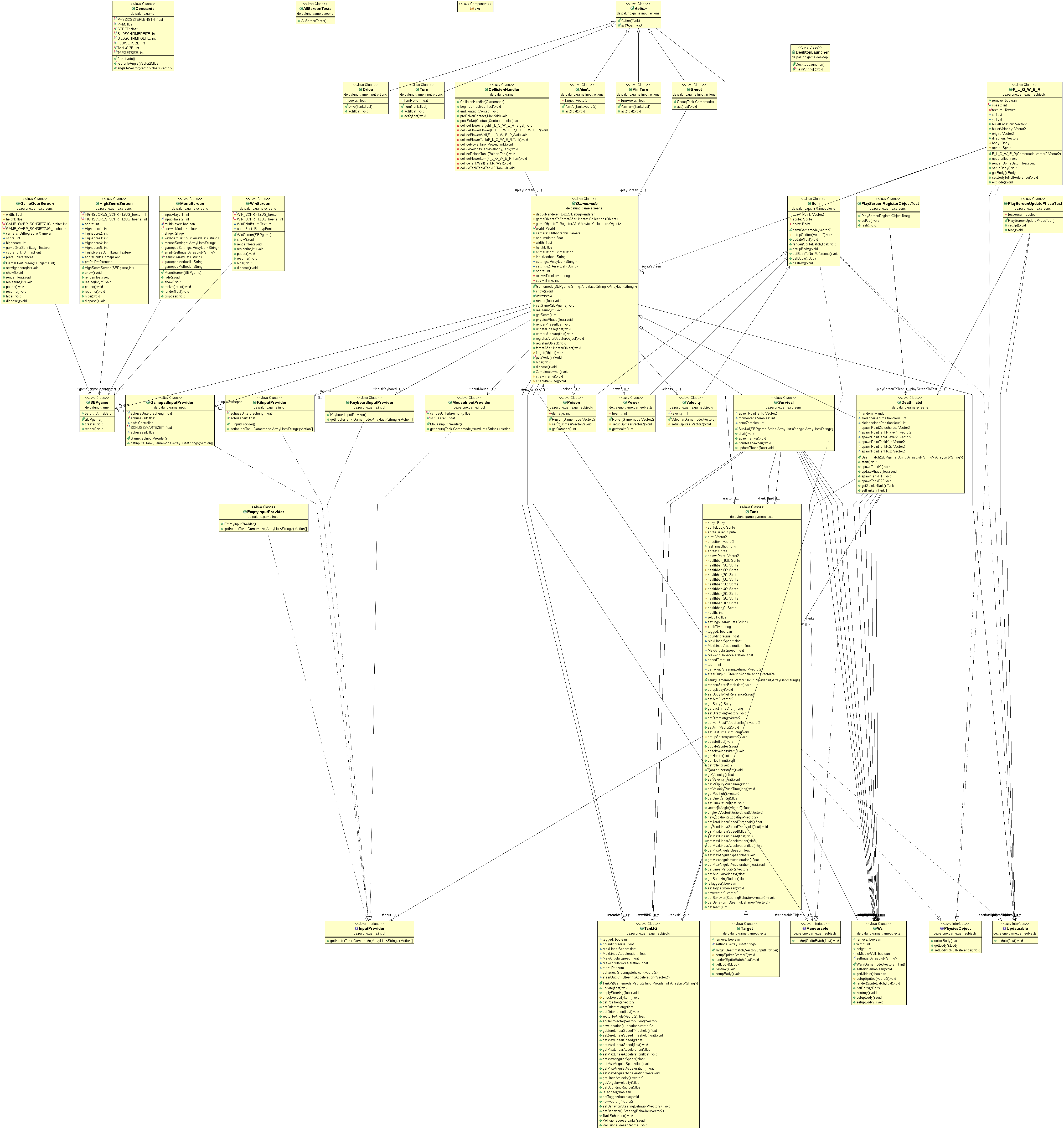
* Als Nutzer möchte ich ein Poison-Pack haben um einen erhöhten Schwierigkeitsgrad für mich und die gegnerischen Panzer zu haben



## hMSC



## UML Klassendiagramm



## Funktionalitätsplanung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Funktionalität | Geschätzter Aufwand | Verantwortlicher | Abhängige Funktionalitäten | Quellcode-referenz | Status |
| 1 | **CollisionHandler Klasse zum Erkennen von Kollisionen implementiert** | 6h | Niklas Schiller |  | de.paluno.game.CollisionHandler | Fertig |
| 1.1 | CollidePowerTank methode zum behandeln von Kollisionen zwischen Panzern und dem Heilungs Item | 10 min | Lars Daumann |  | de.paluno.game.CollisionHandler.CollidePowerTank() | Fertig |
| 1.2 | CollideVelocityTank methode zum behandeln von Kollisionen zwischen Panzern und dem Nitro Item | 10 min | Lars Daumann |  | de.paluno.game.CollisionHandler.CollideVelocityTank() | Fertig |
| 1.3 | CollidePoisonTank methode zum behandeln von Kollisionen zwischen Panzern und dem Gift Item | 10 min | Lars Daumann |  | de.paluno.game.CollisionHandler.CollidePosonTank() | Fertig |
| 1.4 | CollideFlowerItem methode zum behandeln von Kollisionen zwischen Blumen und Items | 10 min | Lars Daumann |  | de.paluno.game.CollisionHandler.CollideFlowerItem() | Fertig |
| 2 | **Item Klasse zum erstellen der einzelnen Itemtypen** | 30 min | Lars Daumann |  | de.paluno.game.gameobjects.Item | Fertig |
| 3 | **TankKi Klasse zum computersteuern der Ki Panzer** |  |  | Tank | de.paluno.game.gameobjects.TankKi | Fertig |
| 3.1 | checkVelocityItem methode zum prüfen, wie lange der Nitroeffekt aktiv ist | 30 min | Lars Daumann |  | de.paluno.game.gameobjects.TankKi.checkVelocityItem() | Fertig |
| 3.2 | applySteering methode zum hinzufügen des KI behaviours | 30 min | Cedric Jüssen |  | de.paluno.game.gameobjects.TankKi.applySteering() | Fertig |
| 3.3 | ((restliches)) | 3h | Patrick Knothe |  | de.paluno.game.gameobjects.TankKi | Fertig |
| 4 | **Velocity item Klasse für den Nitro effekt** | 15 min | Lars Daumann |  | de.paluno.game.gameobjects.Velocity | Fertig |
| 5 | **Power item Klasse zum heilen** | 15 min | Lars Daumann |  | de.paluno.game.gameobjects.Power | Fertig |
| 6 | **Poison item klasse für das gift item** | 15 min | Lars Daumann |  | de.paluno.game.gameobjects.Poison | Fertig |
| 7 | **Wall Klasse zum begrenzen der Spielwelt** | 3h | Patrick Knothe | Tank | de.paluno.game.gameobjects.Wall | Fertig |
| 8 | **KeyboardInputProvider zum steuern des Spiels mittels der Tastatur** | 30 min | Lars Daumann | Inputprovider | de.paluno.game.input.keyboardInputProvider | Fertig |
| 9 | **GamepadInputProvider zum steuern des Spiels mittles eines gamepads** | 2h | Patrick Knothe | Inputprovider | de.paluno.game.input.GamepadInputProvider | Fertig |
| 10 | **KiInputProvider zum computersteuern der Ki panzer** | 0,5h | Cedric Jüssen | Inputprovider | de.paluno.game.input.KiInputProvider | Fertig |
| 11 | **MenuScreen für einen Menü-bildschirm** | 4h | Robin Rongen | Screen | de.paluno.game.screens.MenuScreen | Fertig |
| 12 | **Deathmatch Klasse für den Deathmatch spielmodus** |  |  | Gamemode | de.paluno.game.screens.Deathmatch | Fertig |
| 12.1 | updatePhase methode zum aktualisieren der spieldetails | 3h | Cedric Jüssen | Gamemode | de.paluno.game.screens.Deathmatch.updatePhase() | Fertig |
| 12.2 | spawnTankKi methode zum spawnen der Ki Panzer | 15 mins | Cedric Jüssen |  | de.paluno.game.screens.Deathmatch.spawnTankKi() | Fertig |
| 12.3 | spawnTankP1 methode zum spawnen des spieler 1 | 15 min | Lars Daumann |  | de.paluno.game.screens.Deathmatch.spawnTankP1() | Fertig |
| 12.4 | spawnTankP2 methode zum spawnen des zweiten spielers | 15 min | Robin Rongen |  | de.paluno.game.screens.Deathmatch.spawnTankP2() | Fertig |
| 12.5 | ((Restliches)) |  | Florian Rühl | Gamemode | de.paluno.game.screens.Deathmatch | Fertig |
| 13 | **GameOverScreen für einen Bildschirm bei spielende** | 2 h | Joel Schneider | Screen | de.paluno.game.screens.GameoverScreen | Fertig |
| 14 | **HighScoreScreen zum anzeigen des Highscores** | 1.5 h | Joel Schneider | Screen | de.paluno.game.screens.HighScoreScreen | Fertig |
| 15 | **Survival klasse für den survival spielmodus** |  |  | Gamemode | de.paluno.game.screens.Survival | Fertig |
| 15.1 | Zombiespawner methode zum spawnen der KI-Panzer | 2h | Niklas Schiller | TankKi | de.paluno.game.screens.Survival.Zombieswaner() | Fertig |
| 15.2 | UpdatePhase zum aktualisieren der Spieldetails | 10 min | Florian Rühl | Gamemode | de.paluno.game.screens.Survival.updatePhase() | Fertig |
| 15.3 | ((Restliches)) |  | Florian Rühl | Gamemode | de.paluno.game.screens.Survival | Fertig |
| 16 | **Gamemode klasse zum erstellen der Gamemodes** | 2 h | Florian Rühl | SEPGame | de.paluno.game.screens.Gamemode | Fertig |
| 16-1 | spawnItems zum spawnen der Items | 0,5h | Lars Daumann |  | de.paluno.game.screens.gamemode.spawnItems() | Fertig |
| 16.2 | checkItemLife zum respawnen der items | 0,5h | Lars Daumann |  | de.paluno.game.screens.gamemode.checkItemLife() | Fertig |
| 17 | **Tank klasse zum erstellen der Panzer** |  |  |  | de.paluno.game.gameobjects.Tank | Fertig |
| 17.1 | checkVelocityItem zum prüfen, ob der Nitro effekt aktiv ist | 0,5h | Lars Daumann |  | de.paluno.game.gameobjects.Tank.checkVelocityItem() | Fertig |
| 17.2 | updateSprites zum aktualisieren der panzer sprites | 0,5h | Lars Daumann |  | de.paluno.game.gameobjects.Tank.cupdateSprites() | Fertig |
| 17.3 | Render zum rendern des Panzers | 0,5h | Niklas Schiller | Gamemode | de.paluno.game.gameobjects.Tank.render() | Fertig |

## Unittests

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Getestete Funktionalität | Quellcode Referenz | Status |
| 1 | Erzeugen von SEPgame Objekten | Testgameobjects > Zeile 30 | Fertig |
| 2 | Testen der vectorToAngle() Methode | testConstants > zeile 29 | Fertig |
| 3 | Testen der angleToVector() Methode | testConstants > Zeile 30 | Fertig |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 28.05.2019 | | |
| **Tester** | Sedat Köyden | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Es wurde sich erfolgreich durch das Hauptmenü navigiert um Zombie zu starten | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer startet Zombie | Das System öffnet den Survival Screen | **√** |
| 2 | Der Benutzer nutzt die Panzer Steuerung um den Gegner zur Aufgabe zu überreden | Das System löscht den Gegner Panzer und erschafft zwei neue Panzer | **√** |
| 3 | Der Benutzer wird überwältigt | Das System zeigt den Gameover Screen | **√** |
| **Nachbe-dingung(en)** |  | | **√** |
| **Testurteil** | Test bestanden. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 28.05.2019 | | |
| **Tester** | Kadir Kilincarslan | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Hauptmenü ist geöffnet | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer klickt Preferences an | Das System öffnet den Preferences Screen | **√** |
| 2 | Der Benutzer verändert die Steuerungsoptionen | Das System zeigt die veränderte Steuerung an | **√** |
| 3 | Der Benutzer startet einen Spielmodus | Das System öffnet den neuen Screen mit der veränderten Steuerung | **√** |
| **Nachbe-dingung(en)** |  | | **√** |
| **Testurteil** | Test bestanden. | | |

# Iteration lll

## User Stories Spiel

Map-Auswahl

|  |  |
| --- | --- |
| User-Story-ID | 3.01 |
| User-Story-  Beschreibung | Als Nutzer möchte ich auf drei verschiedenen Maps spielen können, um eine größere Abwechslung zu haben. |
| Geschätzter  Realisierungsaufwand | 6 Stunden |
| Priorität | hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

Flower-Auswahl

|  |  |
| --- | --- |
| User-Story-ID | 3.02 |
| User-Story-  Beschreibung | Als Nutzer möchte ich ein Blumen-auswahl-Menü auf dem Spielscreen haben, um sehen zu können, welche Blume ich aktuell verschießen kann |
| Geschätzter  Realisierungsaufwand | 3 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories | 3.03, 3.04, 3.05 |
| Zugehörige Szenarien |  |

FastFlower

|  |  |
| --- | --- |
| User-Story-ID | 3.03 |
| User-Story-  Beschreibung | Als Nutzer möchte ich eine Blume verschießen können, die schneller fliegt, als die normale Blume, dafür aber weniger Schaden verursacht um offensiver Spielen zu können |
| Geschätzter  Realisierungsaufwand | 1 Stunde |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

BounceFlower

|  |  |
| --- | --- |
| User-Story-ID | 3.04 |
| User-Story-  Beschreibung | Als Nutzer möchte ich eine Blume verschießen können, die bis zu zweimal von Wänden abprallt und bei der dritten Kollision mit einer Wand verschwindet und weniger Schaden verursacht, um defensiver Spielen zu können |
| Geschätzter  Realisierungsaufwand | 2 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories | 3.02, 3.03, 3.05 |
| Zugehörige Szenarien |  |

TrippleFlower

|  |  |
| --- | --- |
| User-Story-ID | 3.05 |
| User-Story-  Beschreibung | Als Nutzer möchte ich eine Blume verschießen können, die sich im Flug in drei einzelne Blumen aufteilt, die aber nach einiger Zeit von selbst verschwinden um im Nahkampf besser agieren zu können |
| Geschätzter  Realisierungsaufwand | 2 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories | 3.02, 3.03, 3.04 |
| Zugehörige Szenarien |  |

Floweranzeige über den Tanks

|  |  |
| --- | --- |
| User-Story-ID | 3.06 |
| User-Story-  Beschreibung | Als Nutzer möchte ich zu jedem Zeitpunkt sehen können, welcher Blumentyp gerade bei welchem Panzer aktiv ist um mich entsprechend zu verteidigen |
| Geschätzter  Realisierungsaufwand | 2 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

Soundeffekte

|  |  |
| --- | --- |
| User-Story-ID | 3.07 |
| User-Story-  Beschreibung | Als Nutzer möchte ich Soundeffekte während des Spiels haben um auch auditives Feedback zu erhalten und besser unterhalten zu werden |
| Geschätzter  Realisierungsaufwand | 2 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

Panzerfarben

|  |  |
| --- | --- |
| User-Story-ID | 3.08 |
| User-Story-  Beschreibung | Als Nutzer möchte ich den Panzern unterschiedliche Aussehen geben können um sie unterscheiden zu können und den Teams zuordnen zu können |
| Geschätzter  Realisierungsaufwand | 3 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

WinScreens

|  |  |
| --- | --- |
| User-Story-ID | 3.09 |
| User-Story-  Beschreibung | Als Nutzer möchte ich zwischen den Spielen Winscreens haben um sehen zu können welches Team das Spiel gewonnen hat |
| Geschätzter  Realisierungsaufwand | 3 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

Features:

Capture The Flag

|  |  |
| --- | --- |
| User-Story-ID | 3.10 |
| User-Story-  Beschreibung | Als Nutzer möchte ich einen „Capture The Flag“-Modus haben, um eine größere Vielfalt zu haben |
| Geschätzter  Realisierungsaufwand | 8 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

Survival mit Geld

|  |  |
| --- | --- |
| User-Story-ID | 3.11 |
| User-Story-  Beschreibung | Als Nutzer möchte ich einen Modus haben, in dem ich, durch das überreden von Panzern, ingame-Währung erwirtschaften kann, um diese gegen Powerups eintauschen zu können |
| Geschätzter  Realisierungsaufwand | 5 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

Schatzkammer

|  |  |
| --- | --- |
| User-Story-ID | 3.12 |
| User-Story-  Beschreibung | Als Nutzer möchte ich einen Raum im Spiel haben, in dem ich die erwirtschaftete ingame-Währung gegen Powerups eintauschen kann, um bessere Voraussetzungen für die nächste Spielrunde zu schaffen |
| Geschätzter  Realisierungsaufwand | 3 Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories | 3.11 |
| Zugehörige Szenarien |  |

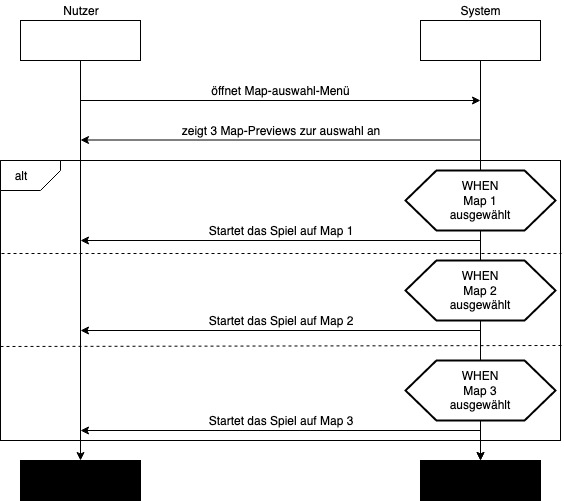
Bossfight

|  |  |
| --- | --- |
| User-Story-ID | 3.13 |
| User-Story-  Beschreibung | Als Nutzer möchte ich einen Boss bekämpfen können, der sich durch mehr Leben kennzeichnet, um eine schwierigere Herausforderung zu haben |
| Geschätzter  Realisierungsaufwand | Stunden |
| Priorität | Hoch |
| Autor(en) | Joel Schneider |
| Abhängigkeiten zu anderen User-Stories |  |
| Zugehörige Szenarien |  |

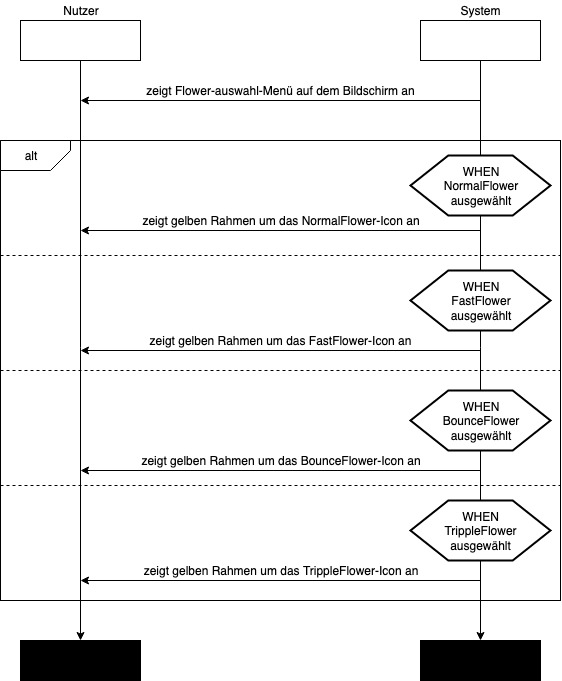
## MSCs

bMSCs

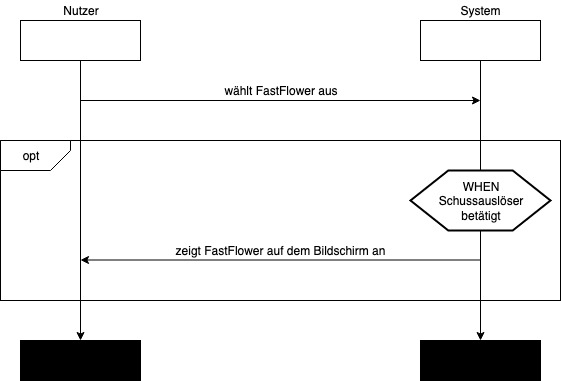
Map-Auswahl – 3.01



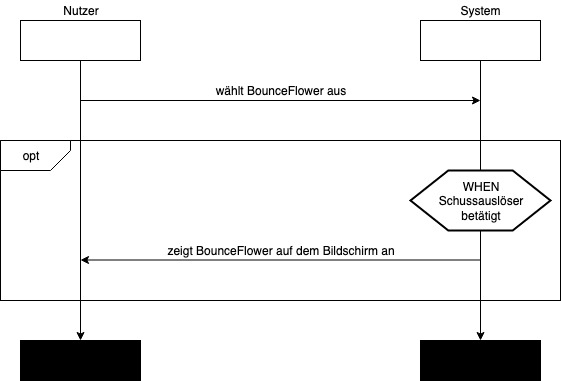
Flower-Auswahl – 3.02



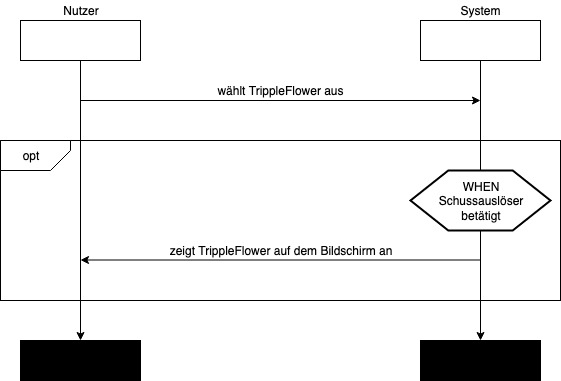
FastFlower – 3.03



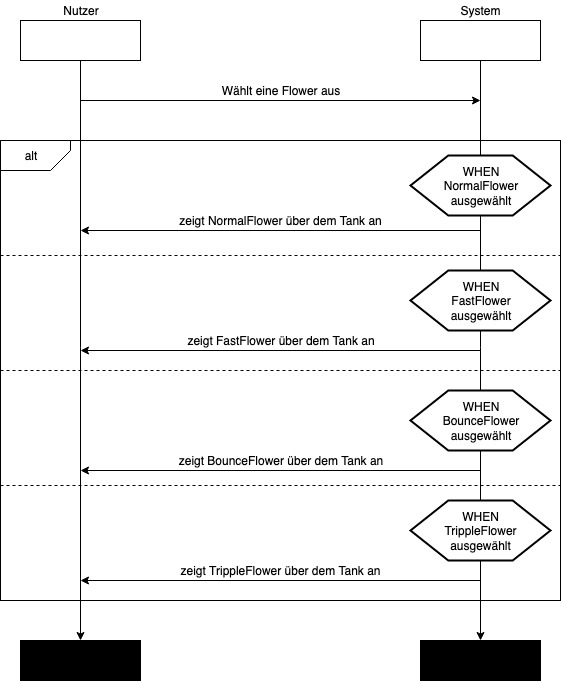
BounceFlower – 3.04



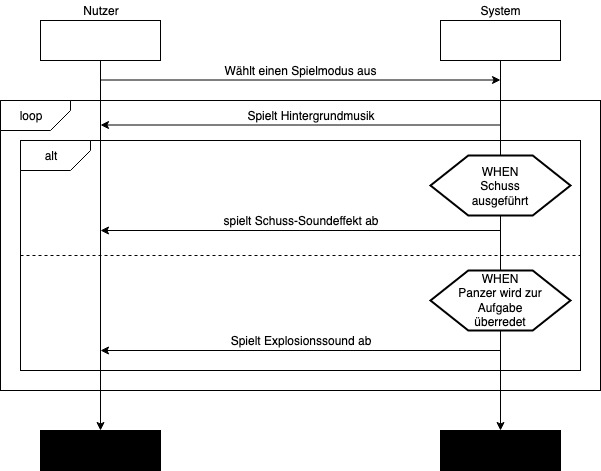
TrippleFlower – 3.05



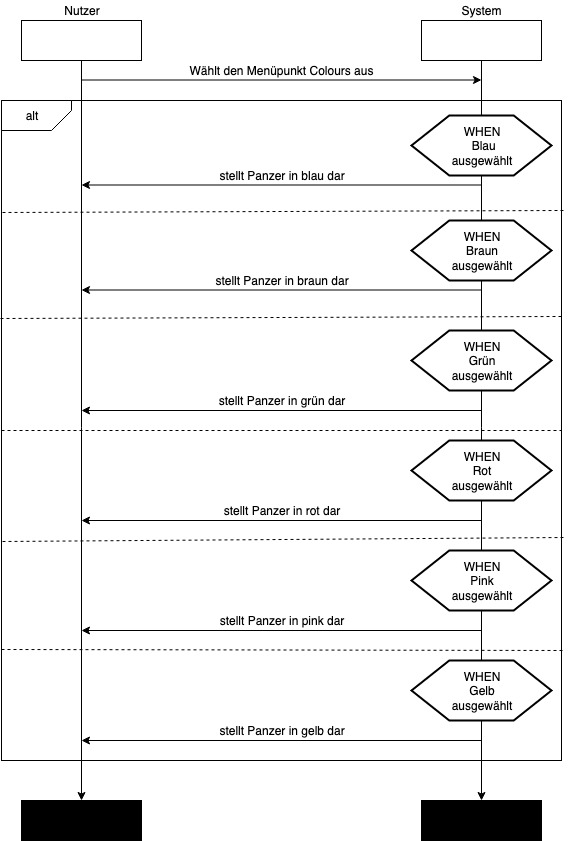
Floweranzeige über den Tanks – 3.06



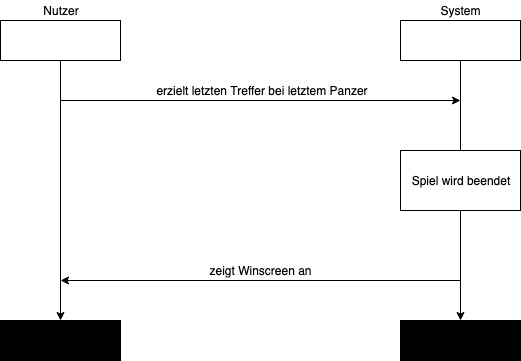
Soundeffekte – 3.07



Panzerfarben – 3.08

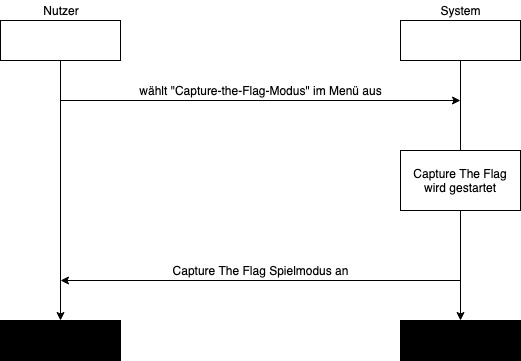


Winscreens – 3.09

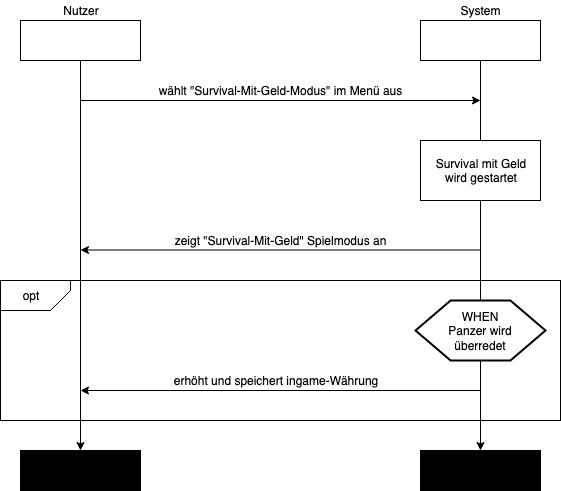


Features:

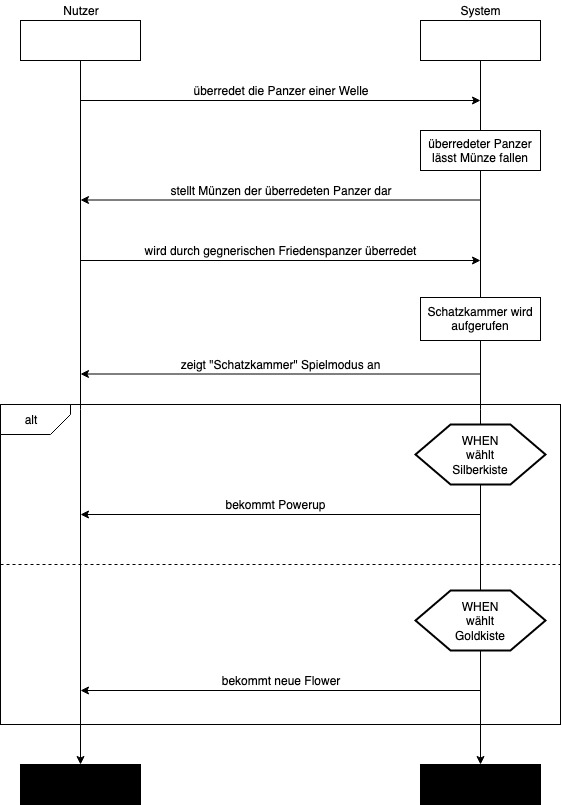
Capture The Flag – 3.10



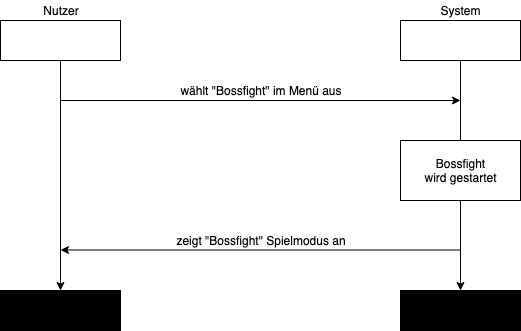
Survival mit Geld – 3.11



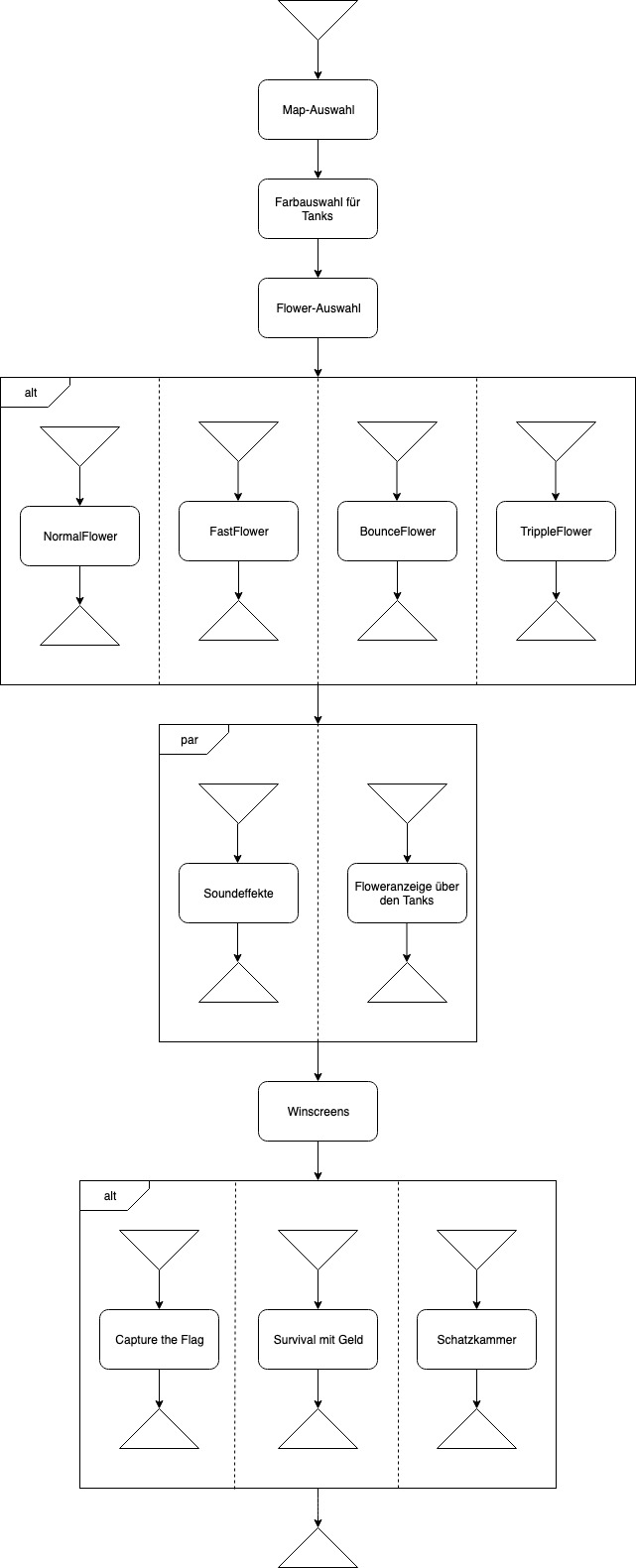
Schatzkammer – 3.12



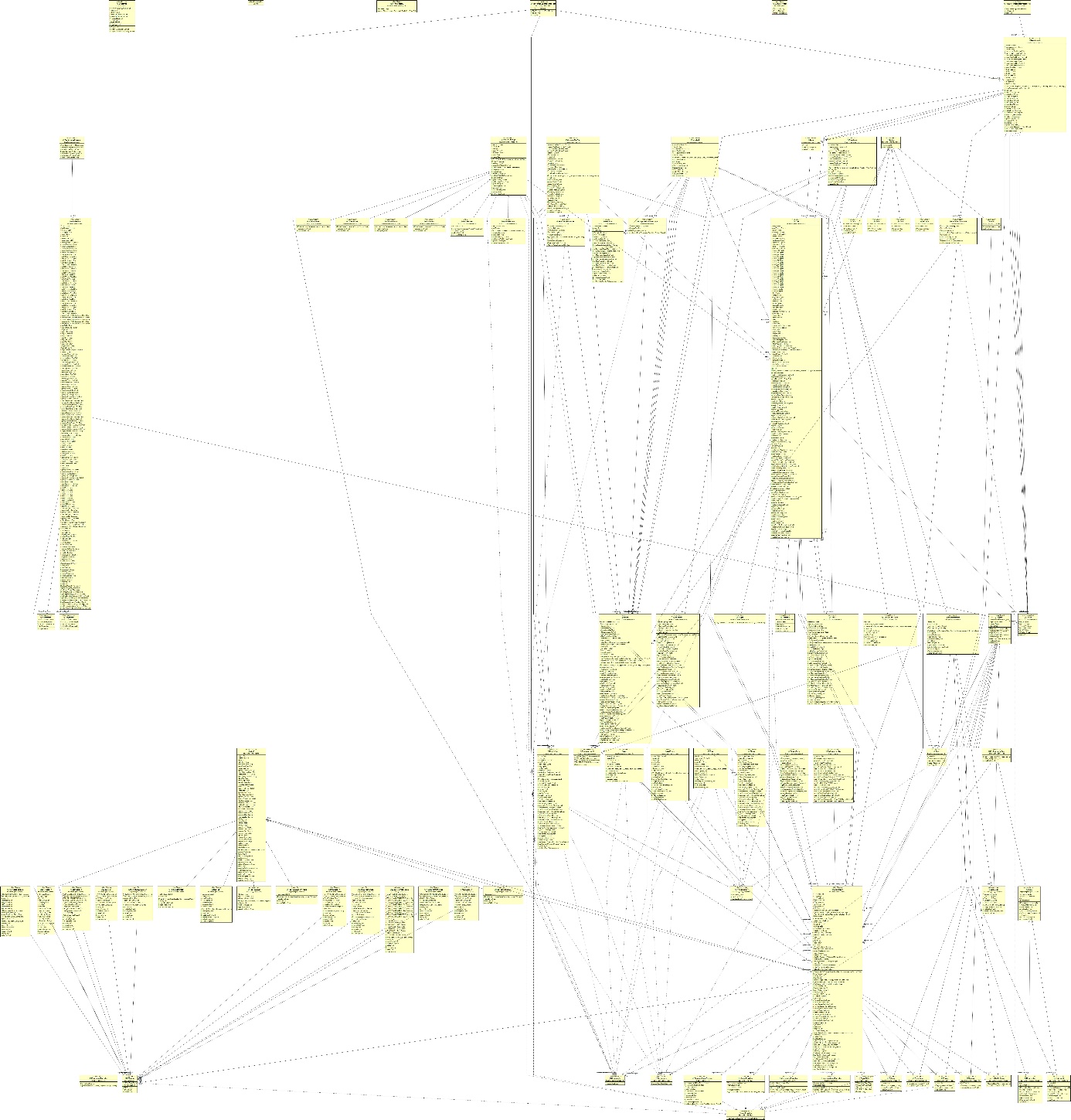
Bossfight – 3.13



hMSC



## UML Klassendiagramm



## Funktionalitätsplanung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwort-licher** | **Abhängige**  **Funkt.** | **Quellcode-**  **Referenz**  **De.paluno…** | **Status** |
| **1.** | **Verschiedene Anzeigeoberflächen** |  |  |  |  |  |
| 1.1 | Anzeige für Deathmatch | 3h | Lars Daumann | 1.5 | screens.Hud\_Deathmatch,  game. CollisionHandler. killFeed() | fertig |
| 1.2 | Anzeige für Survival | 2h | Lars Daumann | 1.5 | screens.Hud\_Survival | fertig |
| 1.3 | Anzeige für Capture the Flag | 2h | Lars Daumann | 1.5 | screens.Hud\_CaptureTheFlag | fertig |
| 1.4 | Anzeige für Schatzkammer | 2h | Lars Daumann | 1.5 | screens.Hud\_Schatzkammer,  screens.Hud\_Survival\_mit\_Geld | fertig |
| 1.5 | Abstrakte Anzeige | 3h | Lars Daumann |  | screens.Hud | fertig |
| **2.** | **Verschiedene Flowertypen** |  |  |  |  |  |
| 2.1 | Schnelle Flower | 1h | Lars Daumann | 2.5 | game.gameobjects.FastFlower | fertig |
| 2.2 | Abprallende Flower | 1h | Lars Daumann | 2.5 | game.gameobjects.BounceFlower | fertig |
| 2.3 | Zerteilende Flower | 4h | Lars Daumann | 2.5 | game.gameobjects.TripleFlower,  game.gameobjects.SplittedFlower | fertig |
| 2.4 | Normale Flower | 1h | Lars Daumann | 2.5 | game.gameobjects.FastFlower | fertig |
| 2.5 | Abstrakte Flower | 3h | Lars Daumann |  | game.gameobjects.F\_L\_O\_W\_E\_R | fertig |
| 2.6 | Kollision der Flowers | 1h | Lars Daumann | 2.1, 2.2, 2.3, 2.4, 2.5 | game.CollisionHandler.collideFlowerTank(),  game.CollisionHandler.collideFlowerWall() | fertig |
| **3.** | **Verschiedene Karten** |  |  |  |  |  |
| 3.1 | Auswahl der Karten | 1h | Lars Daumann |  | screens.MenuScreen.chooseMap() | fertig |
| **4.** | **Verschiedene Tankfarben** |  |  |  |  |  |
| 4.1 | Auswahl der Farben | 1h | Lars Daumann |  | screens.MenuScreen.chooseColour(),  screens.OwnChangeListener.updateColourSettings(),  gameobjects.Colour,  gameobjects.Tank.setupColours() | fertig |
| **5.** | **Spieler- & Teamanzahl** |  |  |  |  |  |
| 5.1 | Auswahl der Teams | 1h | Lars Daumann |  | screens.MenuScreen.chooseTeam() | Fertig |
| 5.2 | Auswahl der Spieleranzahl | 1h | Lars Daumann |  | screens.MenuScreen.chooseTeam(),  screens.MenuScreen.firstTable(),  screens.OwnChangeListener.updateTeamSettings(), | Fertig |
| **6.** | **Anpassung der Steuerung** | 1h | Lars Daumann |  | screens.MenuScreen.chooseSettings(),  screens.OwnChangeListener.updateKeyboardSettings(),  screens.OwnChangeListener.updateMouseSettings(),  screens.OwnChangeListener.updateGamepadSettings() | Fertig |
| **7.** | **Capture the Flag Modus** | 14 h | Robin Rongen |  | Screens.CaptureTheFlag.start()  Spawntankp2()  Spawntankp1()  Spawnkitanks()  Spawnbase()  updatePhase()  checkround2()  checkwin()  resetflagholder1()  resetflagholder2()  gameobjects.Tank.respawn()  respawn2()  reset() | Fertig |
| 7.1 | Animationen für Capture the Flag | 5h | Robin Rongen |  | Game.Gameobjects.tank.render()  Updateflagsprites()  Game.Gameobjects.tankki.render() | Fertig |
| 7.2 | Objekte für Capture The Flag | 2h | Robin Rongen |  | Game.Gameobjects.flagholder | Fertig |
| 7.3 | Collision für Capture the Flag | 1h | Robin Rongen |  | game.CollisionHandler.collideFlowerFlagholder()  collideTankFlagHolder()  ctfRespawn() | Fertig |
| 7.4 | TankKi checkt Sicht auf anderen Panzer | - | Cedric Jüssen | 7.2 | KiInputProviderCTF.  TankInVisible();  Gamemode. | Fertig |
| 7.5 | TankKi checkt nächstes Target | - | Cedric Jüssen | 7.4 | KiInputProviderCTF.naesterTank(); .naesterTankKi(); | Fertig |
| 7.6 | TankKi bekommt Target und beschisst es | 0,5h | Cedric Jüssen | 7.5 | KiInputProviderCTF.getInputs(); .Ziel() | Fertig |
| 7.7 | Setze Arrive für Ki | 4h | Cedric Jüssen | 7.2 | CaptureTheFlag().UpdatePhase2(); . setKiVerhalten | Fertig |
| **8** | **Funktionen der Schatzkammer** | 12h | Niklas Schiller |  | Screens.Schatzkammer  Gameobjects.Kiste  Gameobjects.Kiste\_silber  Gameobjects.Kiste\_gold  Gameobjects.Exit\_Door  Gameobjects.Rechteckiges\_Item | Fertig |
| 8.1 | Kollision zwischen Tank und Kiste | 20min | Niklas Schiller | 8 | Game.CollisionHandler.collideTankKiste() | Fertig |
| 8.2 | Verlassen der Schatzkammer | 20min | Niklas Schiller | 8 | Game.CollisionHandler.collideTankExit()  Gameobjects.Exit\_Door | Fertig |
| **9** | **Funktionen des Survivalscreens mit Geld** | 8h | Niklas Schiller |  | Screens.Survival\_mit\_Geld  Gameobjects.Coin | Fertig |
| 9.1 | Einsammeln der Münzen | 20min | Niklas Schiller | 9 | Game.CollisionHandler.collideTankCoin() | Fertig |
| **10** | **TankKi für DM** |  |  |  |  | Fertig |
| 10.1 | TankKi checkt Sicht auf anderen Panzer | 4h | Cedric Jüssen |  | KiInputProviderDM.TankInVisible(); Gamemode.getObjektFix() | Fertig |
| 10.2 | TankKi checkt nächstes Target | 5h | Cedric Jüssen | 10.1 | KiInputProviderDM.naesterTank(); .naesterTankKi(); | Fertig |
| 10.3 | TankKi bekommt Target und speichert dessen Koordinaten | 5h | Cedric Jüssen | 10.2 | KiInputProviderDM.Ziel() | Fertig |
| 10.4 | TankKi beschießt Target | 1h | Cedric Jüssen | 10.3 | KiInputProviderDM.getInputs() | Fertig |
| 10.5 | Setze Arrive für Ki | 3h | Cedric Jüssen | 10.3 | Deathmatch.  UpdatePhase();  .setArriveforKi(); | Fertig |
| **11** | **Tanks(Ki) Teams und Ende in DM** |  |  |  |  | Fertig |
| 11.1 | TankKi können null sein und haben Teams | 2h | Cedric Jüssen |  | Deathmatch.spawnTankKi(); .spawnTankP1(); .spawnTankP2(); | Fertig |
| 11.2 | Schaue ob es keinen Player mehr gibt und Ki nichts mehr sieht | 0,5h | Cedric Jüssen |  | Deathmatch.AllPlayerDead(); .AllKiSeeNothing() | Fertig |
| 11.3 | Setzte unentschieden für Ki | 0,5h | Cedric Jüssen | 11.2 | Deathmatch.  UpdatePhase() | Fertig |
| **12** | **TiledMaps laden und darstellen** | **1 Woche** |  |  |  | Fertig |
| 12.1 | Vordergrundlayer vor dem Panzer rendern und Hintergrundlayer hinter dem Panzer | 1 Tag | Florian Rühl | 12 | screens.Gamemode.show(), screens.Gamemode.render(), screens.Deathmatch.render() | Fertig |
| 12.2 | Objekte der Tiledmap als Box2d Objekte erstellen | 5 Tage | Florian Rühl | 12 | screens.Gamemode.buildTilemapBodies(), screens.Gamemode.getFixturedef(), screens.Gamemode.getShapeFromRectangle(), screens.Deathmatch.start(), screens.Survival.start(), screens.Survival\_mit\_Geld.start(), screens.Schatzkammer.start(), screens.Bossfight.start(), screens.CaptureTheFlag.start(), gameobjects.Wall | Fertig |
| 12.2 | Zerstörbare Bäume auf der Map spawnen | 2 Tage | Florian Rühl | 12, 13 | gameobjects.Tree,  game.CollisionHandler: 115 – 125, game.CollisionHandler.collideFlowerTree() | Fertig |
| 12.3 | Panzern ermöglichen über Wasser zu schließen, aber nicht darüber zu fahren | 3 Tage | Florian Rühl | 12, 13 | gameobjects.Water, gameobjects.F\_L\_O\_W\_E\_R: 71 | Fertig |
| 12.4 | Mehrere Maps erstellen | 1 Woche | Florian Rühl | 12 |  | Fertig |
| **13** | **Das KI Verhalten der Panzer verbessern** | **2 Wochen** | **Florian Rühl** | **12** |  | Fertig |
| 13.1 | Die KI daran hindern durch wände zu schauen | 1 Woche | Florian Rühl | 12, 13 | gameobjects.TankKi.rayCast(), gameobjects.TankKi.haslineofsight(), screens.Survival.updatePhase(),  Screens.Survival\_mit\_Geld.updatePhase() | Fertig |
| 13.2 | Der KI ein Gedächtnis mit der letzten ihr bekannten Spielerposition geben | 3 Tage | Florian Rühl | 12, 13 | gameobjects.TankKi.setGhosts() | Fertig |
| 13.3 | Die KI automatisch um Wände und Hindernisse herum fahren lassen | 1,5 Wochen | Florian Rühl | 12, 13 | gameobjects.TankKi.update(), gameobjects.TankKi: 68 – 73, gameobjects.TankKi.setPrioritySteering(), gameobjects.whiskersAvoid, Screens.Survival.updatePhase(),  screens.Survival\_mit\_Geld.updatePhase() | Fertig |
| **14** | **Panzer können verschiedene Blumen verschießen** | **1 Woche** | **Joel Schneider** |  | **Gameobjects.FlowerFactory**  **Gameobjects.FastFlower**  **Gameobjects.NormalFlower**  **Gameobjects.Tank.useFlower**  **Input.actions.shoot.act**  **Input.KeyboardInputProvider.chooseFlower**  **Input.MouseInputProvider.chooseFlower**  **Gameobjects.Blumenauswahl**  **Gameobjects.TankKi.KiBlumenauswahl**  **Gameobjects.TankKi.renderFlower**  **Gameobjects.Tank.renderFlower** | **Fertig** |
| 14.1 | Verschiedene Blumentypen werden Panzerbezogen angezeigt | 3 Tage | Joel Schneider | bla | Screens.Gamemode.checkRenderSelection  Screens.Gamemode.RenderselectionP2  Screens.Gamemode.RednderSelectionP1  Input.KeyboardInputProvider.chooseFlower  Input.MouseInputProvider.chooseFlower  Gameobjects.Blumenauswahl  Gameobjects.Tank.updateFlowerAuswahl | Fertig |
| 15 | Tanks werden in Teams eingeteilt | 4 Tage | Joel Schneider |  | Gameobjects.Team  Gameobjects.Tank.isAlive  Screens.Deathmatch.setupTeams  Screens.Deathmatch.addTanks  Screens.Deathmatch.SpawnTankKi  Screens.Deathmatch.numberOfTeamsAlive  Screens.Deathmatch.getWinningTeam  Screens.Deathmatch.EndScreenAufruf | Fertig |
| 16 | Spiel wird beendet durch Aufrufen von EndScreens | 2 Tage | Joel Schneider | 15 | Screens.Deathmatch.EndScreenAufruf  Screens.WinscreenCollection | Fertig |
| 17 | Endscreens verschönert | 4 Stunden | Joel Schneider |  | Screens.GameoverScreen  Screens.PlayerOneWinscreen  Screens.PlayerTwoWinscreen  Screens.WinScreen  Screens.KiWinScreen  Screens.HighScoreScreen | Fertig |
| 18 | Spiel hat Titelbildschirm | 1 Stunde | Joel Schneider |  | Screens.StartScreen | Fertig |
| 19 | Abstrakte Methode für das tactical Nuke Item eingefügt | 5min | Patrick Knothe | - | Screens.Gamemode.tacticalNuke2() | Fertig |
| 20 | Abstrakte Methode zum beenden der Sounds eingefügt | 5min | Patrick Knothe | - | Screens.Gamemode. stopSounds() | Fertig |
| 21 | gameState() | 10min | Patrick Knothe | GameState |  | Fertig |
| 22 | Methode zum stoppen der Sounds implementiert | 5min | Patrick Knothe |  | Screens.Survival.stopSounds() | Fertig |
| 23 | Nicht Spieler-Panzern soll die hälfte leben durch die tactical Nuke abgezogen werden | 5min | Patrick Knothe |  | Screens.Survival. tacticalNuke2() | Fertig |
| 24 | **ManualCreator Klasse mit statischen Methoden zur erzeugung einer txt datei mit der Bedienungsanleitung** | 3 stunden | Patrick Knothe | (Zugriff auf den benutzerordner (?)) | Game.ManualCreator | Fertig |
| **25** | **Bossfight modus implementiert** | 10 Stunden | Patrick Knothe | Gamemode | Screens.Bossfight | Fertig |
| 26 | **GameState enum zum erfassen des aktuellen Zustand des Spiels** | 10min | Patrick Knothe | - | Game.GameState | Fertig |
| 27 | **ErrorScreen** | 10 Stunden | Patrick Knothe | - |  | Fertig |
| 28 | **Kollision von Panzer mit tactical Nuke Item resultiert in aufruf der entsprechenden Methoden in den Screens** | 15min | Patrick Knothe | tacticalNuke() | Game.CollisionHandler. collideTankTacticalNuke() | Fertig |
| 29 | Kollisionsmethode zum kollidieren des panzers mit dem Trumpitem | 25min | Patrick Knothe | TrumpItem, WallBlock, | Game.CollisionHandler.collideTankTrumpItem() | Fertig |
| **30** | **TowerFlower zur Kollisionserkennung von den Tower-Panzern** | 15min | Patrick Knothe | F\_L\_O\_W\_E\_R\_ | Gameobjects.TowerFlower | Fertig |
| 31 | **TacticalNuke als zusätzliches Item implementiert** | 30 min | Patrick Knothe | Item | Gameobjects.TacticalNuke | Fertig |
| 32 | setSchussBol() methode zum ändern des schussbol booleans zur Kontrolle des automatischen Schießens der Ki Panzer | 5min | Patrick Knothe |  | Gameobjects.Tank.setSchussBol() | Fertig |
| 33 | getSchussBol() zum getten des schussbol booleans | Niedrig | Patrick Knothe |  | Gameobjects.Tank.getSchussBol( | Fertig |
| 34 | Mock Klasse von Tank implementiert | Niedrig | Patrick Knothe | Tank | Gameobjects.TestTank | Fertig |
| 35 | **Tower implementiert** | 2 Stunden | Patrick Knothe | Tank | Gameobjects.Tower | Fertig |
| 36 | **TrumpItem zum bauen von Mauern im Spiel** | Niedrig | Patrick Knothe | Item | Gameobjects.TrumpItem | Fertig |
| 37 | **Waypoint zum implementieren des Wegpunktsystems in Bossfight** | 20 min | Patrick Knothe |  | Gameobjects.Waypoint | Fertig |
| 38 | **GamepadInputProvider stark erweitert** | 10 Stunden | Patrick Knothe | InputProvider | Input.GamepadInputProvider | Fertig |
| 39 | stopSounds() | 5min | Patrick Knothe |  | Screens.CaptureTheFlag.stopSounds() | Fertig |
| 40 | tacticalNuke2 implementiert (dieses item existiert in diesem Modus nicht, daher nur methodenkörper) | 2min | Patrick Knothe |  | Screens.CaptureTheFlag.tacticalNuke2 | Fertig |
| 41 | Sämtliche Sounds werden gestoppt | 5min | Patrick Knothe |  | Screens.Deathmatch.stopSounds() | Fertig |
| 42 | Bei Kollision mit tactical nuke Item wird den nicht spielerpanzern die hälfte leben abgezogen. | 15min | Patrick Knothe |  | Screens.Deathmatch.tacticalNuke2() | Fertig |
| 43 | **BossTank implementiert** | Mittel | Patrick Knothe | Tank | Gameobjects.BossTanmk | Fertig |

## 

Kommentar: Sämtliche Verweise auf Sound / das abspielen des Soundtracks in allen Klassen ist Patrick Knothe funktional zuzuweisen.

## Unittests

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Getestete Funktionalität | Quellcode Referenz | Status |
| 1 | Testen der getAim() methode der Tank klasse | Desktop>Default>TestAim(Zeile 28) | Fertig, bestanden |
| 2 | Testen der hauptfunktionalität der Waypoint Klasse, das speichern von Koordinaten | Core>Game> testwaypoint (Zeile 22) | Fertig, bestanden |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| 1. **Datum** | 02.07.2019 | | |
| **Tester** | Fatih Urgun | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Das Gamepad ist bereits an dem PC angeschlossen | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer wählt unter Team das Feld für Gamepad | Das System zeigt die Auswahl an | **√** |
| 2 | Der Benutzer startet einen beliebigen Spielmodus | Das System öffnet den Screen des entsprechenden Modus | **√** |
| 3 | Der Benutzer spielt das Spiel mit der Ausgewählten Steuerung | Das System nutzt die angegebene Steuerung | **√** |
| **Nachbe-dingung(en)** |  | | **√** |
| **Testurteil** | Test bestanden. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 02.07.2019 | | |
| **Tester** | Fatih Urgun | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Es wurde sich erfolgreich durch das Hauptmenü navigiert und das Team Menü geöffnet | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer wählt Anzahl und Teams der KiTanks | Das System zeigt die entsprechende Auswahl | **√** |
| 2 | Der Benutzer startet einen beliebigen Modus | Das System öffnet den Screen des entsprechenden Modus | **√** |
| 3 | Der Benutzer spielt mit den gewählten Einstellungen | Das System passt das Verhalten der KiTanks an die Auswahl an | **√** |
| **Nachbe-dingung(en)** |  | | **√** |
| **Testurteil** | Test bestanden. | | |

## Technische Anforderungen

* PC mit installiertem Java JRE
* Bei Auswahl der Kontrollersteuerung: verbundener Xbox 360 Controller
* Mindestens 26 mb freien Speicherplatz
* Angeschlossene Lautsprecher/Kopfhörer zum Hören des Soundtracks

## Installationsanleitung

1. Verschieben Sie die erhaltene .jar Datei an einen für Sie geeigneten Ort auf ihrem PC
2. Doppelklicken Sie die .jar Datei, um sie auszuführen

## Bedienungsanleitung

Genaue Erläuterung, wie das entwickelte Programm zu bedienen ist.

Zur Gamepadsteuerung:

Gehen Sie im Hauptmenü in die Preferences. In die Felder tragen Sie ein:

Fahren – hier tippen Sie ein: dpad, für Fahrsteuerung über das Steuerkreuz. LS, für den linken Stick oder RS für den rechten Stick.

Links drehen: Wählen Sie die Taste aus, die die Kanone des Panzers nach links drehen soll. Standardmäßig ist das die linke Schultertaste.

Rechts drehen: Wählen Sie die Taste aus, die die Kanone des Panzers nach rechts drehen soll. Standardmäßig ist das die rechte Schultertaste.

Flower hoch-/runterschalten - Wählen Sie die Taste aus, die den aktuell ausgewählten Blumentyp ändert. Standardmäßig sind das die rechte-/bzw. linke taste neben dem Xbox Logo.

Schiessen - Wählen Sie die Taste aus, die den aktuell ausgewählten Blumentyp abfeuert. Standardmäßig sind das die A Taste.

Pause - Wählen Sie die Taste aus, die das Spiel pausieren soll. Standardmäßig ist das die Y Taste.

Mögliche Einträge zur eigenen Belegung der Funktionen sind: RT für die rechte Schultertaste, LT für die linke Schultertaste, die Tasten A, B, X, Y, RX für die Taste rechts vom Xbox Knopf, LX für die Taste links vom Xbox Knopf, rsdown für das Runterdrücken des rechten Sticks, lsdown für das Runterdrücken des linken Sticks.

Zur Tastatursteuerung:

Gehen Sie im Hauptmenü in die Preferences. In die Felder tragen Sie ein:

Oben, Unten, Links, Rechts – hier tippen Sie die Tasten für die Fahrsteuerung ihres Panzers ein. Standardmäßig ist das das WASD layout.

Links drehen: Wählen Sie die Taste aus, die die Kanone des Panzers nach links drehen soll. Standardmäßig ist das der Buchstabe O.

Rechts drehen: Wählen Sie die Taste aus, die die Kanone des Panzers nach rechts drehen soll. Standardmäßig ist das der Buchstabe P.

Schiessen - Wählen Sie die Taste aus, die den aktuell ausgewählten Blumentyp abfeuert. Standardmäßig sind das die Leertaste.

x-Flower: Wählen Sie die jeweiligen Tasten für den jeweiligen Blumentypen aus.

Pause - Wählen Sie die Taste aus, die das Spiel pausieren soll. Standardmäßig ist das die Q Taste.

Die Tastatur & Maussteuerung funktioniert nach dem gleichen Prinzip, nur wird hier die Drehung der Kanone per Mauszeiger gesteuert und das Standardlayout ist anders.

Bitte achten Sie darauf, keine Funktion unbelegt oder einzelne Tasten doppelt zu belegen.

Nun müssen Sie unbedingt die gewünschte Steuerung im Hauptmenü unter ‚Teams‘ noch einstellen. Machen Sie für den jeweiligen Spieler einen Haken bei dessen gewünschter Steuerung. Die KI-Checkbox lässt den Panzer durch eine KI gesteuert werden (nicht in allen Spielmodi).

Sie können hier auch die Teamaufstellung der Panzer ändern (nicht in allen Spielmodi).

Viel Spaß mit SPREADING PEACE.