

## 1 Teilnehmer/innen des Teams:

Klasse: BI19a	Team: Oliver Ammann, Senthil Nagendran, Athavan Ranganathan
------------------	----------------------------------------------------------------

## 2 Anforderungsdefinition (Meilenstein A)

### „EINWANDERER!“

#### Auftrag:

(Allgemeine Beschreibung)

#### Nutzen: Mit dem Spiel sollen

#### Szenario:

- Ein Kampffjet (Spieler) und mehrere UFOs (Gegner->Bots) auf einer rechteckigen Fläche.
- Der Spieler befindet sich im unteren und die Gegner im oberen Bereich der Welt.
- Der Spieler ist in der Lage sich in alle Richtungen zu bewegen.
- Die Gegner bewegen sich als ganze Gruppe horizontal und nach unten.
- Der Spieler kann gerade nach oben schiessen.
- Die Gegner schiessen gerade nach unten.
- Der Spieler und die Bots versuchen sich gegenseitig zu vernichten.
- Sobald alle Gegner-UFOs eliminiert wurden beginnt die nächste Welle.
- Das Spiel endet wenn der Spieler getroffen wird oder mit einem UFO zusammenstösst oder wenn alle Level geklärt werden.

#### Welt:

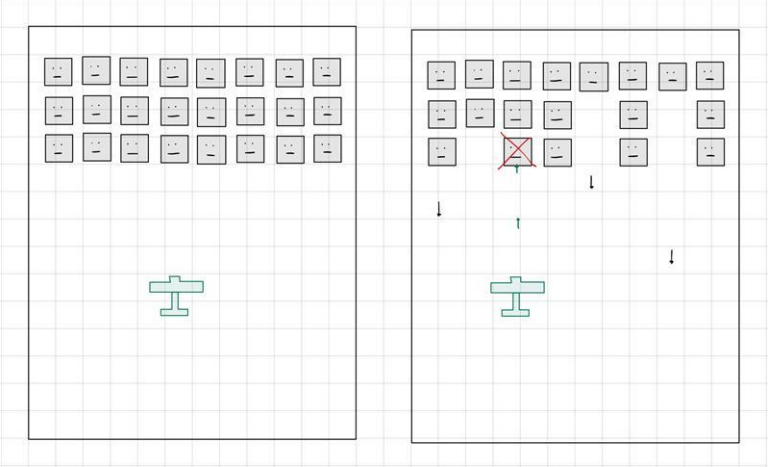
- Rechteckige Fläche (Eventuell mit Hindernissen)
- Weltall-Setting

#### Aktoren:

- Kampffjet(Player): Zweidimensionales Flugzeug mit einer Kanone
- UFOs(Bots): Zweidimensionale UFOs mit einer Kanone
- Schuss (Bullet): Schussobjekt das aus den Flugobjekten geschossen wird
- Statistic board: Kills / Score etc. Simpler Akteur

#### Details:

- Rechts befinden sich Statistik-Anzeigen (Punkte/Kills)
- Das Spiel ist zweidimensional.
- Das Spiel beinhaltet mehrere Level. Darin hat es mehr/stärkere/schnellere Gegner. Evtl. hat es einen Boss-Gegner.

	<p><b>Machbarkeitsabklärung:</b></p> <ul style="list-style-type: none"> <li>Folgende Features sind vorab untersucht worden und ..... (Skizze / Mockup)</li> </ul> 
<p><b>MUSS Kriterien:</b> (Konkrete Features, die umzusetzen sind)</p>	<p><b>Folgende Features sollen implementiert werden (Funktionalität):</b></p> <ul style="list-style-type: none"> <li>Welt: Hintergrund, keine speziellen Funktionen</li> <li>Der Spieler (Kampfjet) kann sich in jede Richtung bewegen / ausweichen</li> <li>Der Kampfjet kann auf Gegner schiessen</li> <li>Die Gegner schiessen den Kampfjet ab (jeweils der unterste von einer Spalte)</li> <li>Die Gegner bewegen sich synchron nach links, rechts und unten</li> <li>Verschiedene Levels</li> </ul>

<b>KANN</b> <b>Kriterien:</b> (Konkrete Features, die optional sind)	<p><b>Folgende Features können zusätzlich implementiert werden: (Kreativität)</b></p> <ul style="list-style-type: none"> <li>• "Wände" können als Schutz für den Kampfjet dienen. Hinter diesen Wänden kann er nicht getroffen werden.</li> <li>• Gegner bewegen sich unregelmässig / asynchron / schneller</li> <li>• Gegner verfügen über mehr Trefferpunkte</li> <li>• Gegner schießen schneller oder mit einem anvisierendem Schuss(verfolgt Spieler)</li> <li>• Boss-Gegner implementieren</li> <li>• Power-Ups für den Spieler</li> <li>• Auswählen des Kampfjets (verschiedene Modelle)</li> </ul>
----------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 2.1 Planung LB2

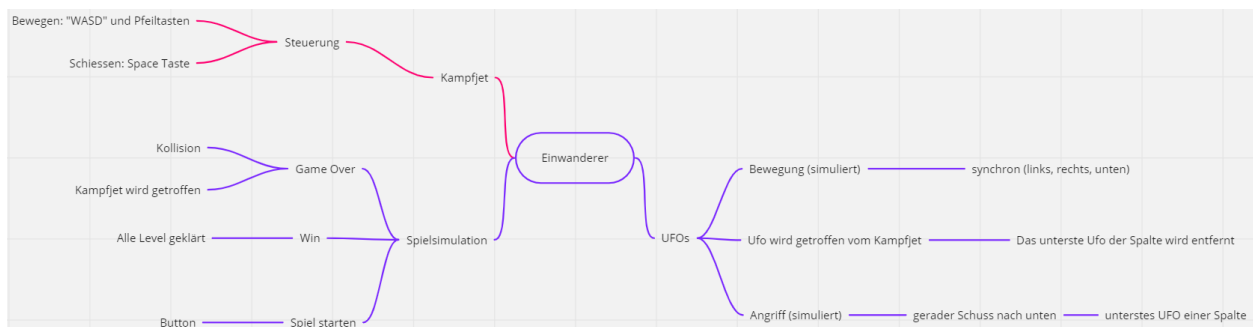
MS	Tätigkeit / Abgabe	Soll-Datum	Ist-Datum
A	<b>Projektstart</b> <ul style="list-style-type: none"> <li>➤ Team Bildung</li> <li>➤ <b>Wahl / Ausarbeitung der Anforderungsdefinition</b></li> </ul> Abnahme Anforderungsdefinition durch Lehrperson	06.12.22	06.12.22
B	<b>Teamaufgabe 1:</b> <ul style="list-style-type: none"> <li>➤ <b>Abgabe: Lösungsdesign</b> (Analyse, Design: Funktionsmodell, UseCase, GUI, Storyboard)</li> </ul>	13.12.22	13.12.22
B2	<b>Teamaufgabe 2:</b> <ul style="list-style-type: none"> <li>➤ Abgabe: Testvorschrift und Testfälle</li> </ul>		
C	<b>Einzelaufgabe 3:</b> <ul style="list-style-type: none"> <li>➤ Abgabe Szenario (.zip) mit Inline-Dokumentation, Systemdokumentation (UML Klassen-, Sequenzdiagramm)</li> <li>➤ <b>Fachgespräch Projektabschluss</b></li> </ul>		
C2	<b>Einzelaufgabe 4:</b> <ul style="list-style-type: none"> <li>➤ Abgabe: Ausgefüllter Systemtest</li> </ul>		

### 3 Lösungsdesign (Meilenstein B: Teamaufgabe 1)

Anhand der Analyse wurde folgendes Lösungsdesign entworfen:

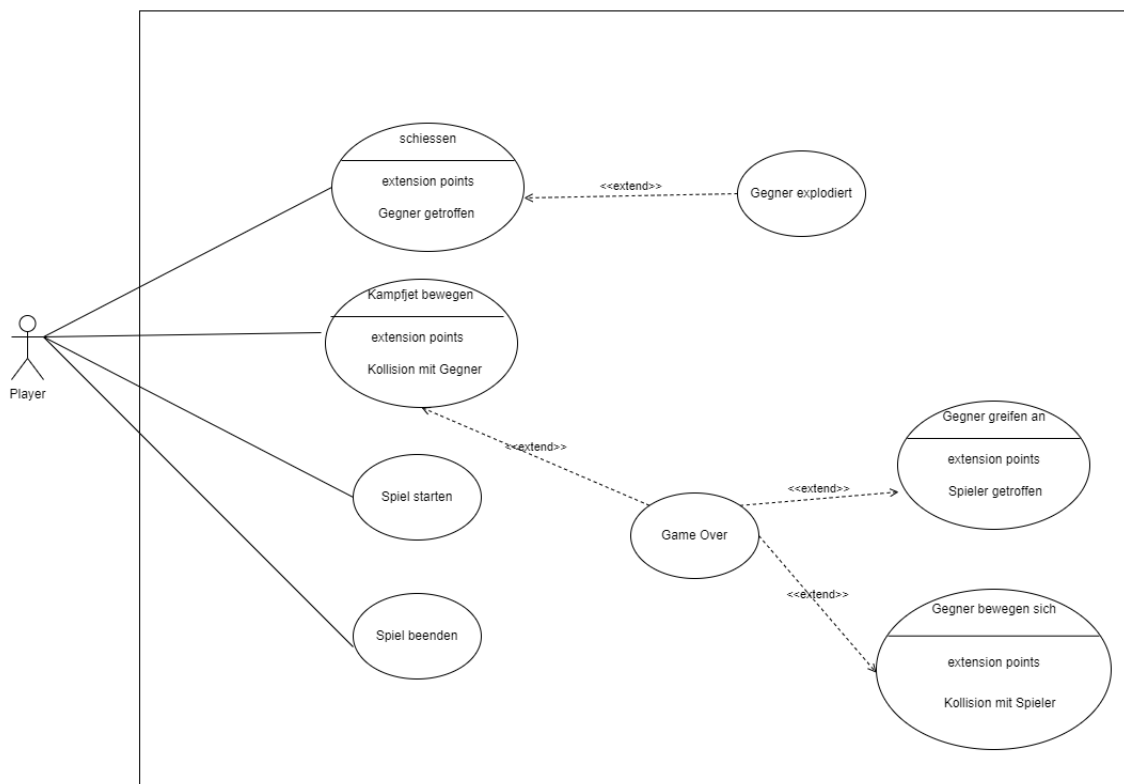
#### 3.1 Funktionsmodell

Im Folgenden sind die erwarteten Eingaben und Ausgaben beschrieben / dargestellt:



#### 3.2 Anwendungsfälle (UseCases)

Folgende Anwendungsfälle sind hier detailliert dokumentiert:



### 3.3 Ablauf

Aus Benutzersicht ist folgender Ablauf des Programms zu erwarten:

...

(Storyboard)

Bild wird kommen	<b>Ausgangslage:</b> Spiel wird gestartet Der Kampfjet befindet sich unten in der Mitte im Weltall Über dem Kampfjet (oben) befinden sich alle gegnerische UFOs
Bild wird kommen	<b>Useraktivität:</b> «WASD»/ Pfeiltasten: Kampfjet steuern Space Taste: schiessen  <b>Kampfjetaktivitäten:</b> Kampfjet schiesst UFOs ab Kampfjet weicht den Schüssen aus
Bild wird kommen	<b>Ufo Aktivitäten (simuliert):</b> UFOs bewegen sich synchron (links, rechts, unten) Das unterste UFO einer Spalte schiesst einen geraden Schuss nach unten
Bild wird kommen	Wen der Kampfjet von einem UFO getroffen wird oder es zu einer Kollision kommt, ist das Spiel zu ende.

## 4 Testvorschrift (LB2 Meilenstein B2: Teamaufgabe 2)

Testbeschreibung und vorbereitetes Testprotokoll siehe Dokument

[M226B\\_LB2\\_Testvorschrift\\_MS-B2.docx](#)

## 5 Systemdokumentation (Meilenstein C: individuelle Aufgabe 3)

Das erstellte Java-Projekt (Greenfoot-Szenario) ist hier detailliert abgelegt:

[M226B\\_Aufgabe\\_3\\_Szenario\\_IhrName.zip](#)

### 5.1 Statisches Design: Klassendiagramm

Folgend die statische Struktur des Szenarios

...

(UML Klassendiagramm mit Assoziationen und Kardinalitäten)

### 5.2 Umfang / Abgrenzung / Änderungen gegenüber Design

Aufgrund unten beschriebener Umstände sind Anpassungen des ursprünglichen Lösungsdesigns gemacht worden:

...

(Umstände / Anpassungen / Veränderungen)

### 5.3 Funktionalität der Implementation.

Zusätzlich zu der Inline-Dokumentation sind hier folgende Funktionen detailliert beschrieben:

...

(Ausführliche Beschreibung der internen Funktionen  
oder Verweis zum Inline-Kommentar mit Javadoc! (`/** @param @return */`))

### 5.4 Dynamische Struktur: Sequenzdiagramm

Ein zentraler Ablauf eines UseCases ist im Folgenden dargestellt:

...

(Darstellung eines zentralen Ablaufs mittels Sequenzdiagramm)

**Trace: ...**

...

## 6 Bedienungsanleitung (Meilenstein C: individuelle Aufgabe 3)

...

## 7 Testprotokoll (LB2 Meilenstein C2: individuelle Aufgabe 4)

Ausgefülltes Testprotokoll siehe Dokument

[M226B\\_LB2\\_Testvorschrift\\_MS-C2\\_Name.docx](#)

