

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.
- 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.

Welt:

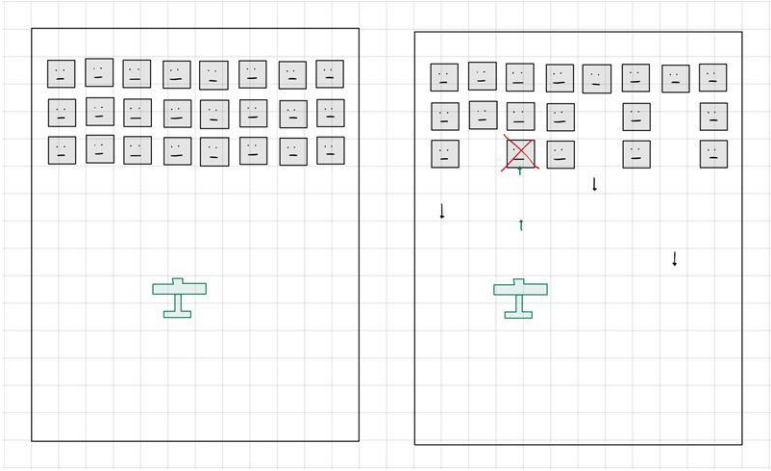
- Rechteckige Fläche (Eventuell mit Hindernissen)
- Weltall-Setting

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.

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

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

2.1 Planung LB2

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

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:

...

(Funktionsmodell: Skizze, Bild, Mindmap, Blockdiagramm zur obigen Anforderungsdefinition mit Legende)

Legende:

-

3.2 Anwendungsfälle (UseCases)

Folgende Anwendungsfälle sind hier detailliert dokumentiert:

...

(Detaillierte Beschreibung der UseCases)

Legende:

...

3.3 Ablauf

Aus Benutzersicht ist folgender Ablauf des Programms zu erwarten:

...

(Storyboard)

| | |
|--|--|
| | |
| | |
| | |

4 Testvorschrift (LB2 Meilenstein B2: Teamaufgabe 2)

Testbeschrieb 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](#)