# §Teilnehmer/innen des Teams:

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

# Anforderungsdefinition (Meilenstein A)

|  |  |
| --- | --- |
| „EINWANDERER!“ | |
| **Auftrag:**  (Allgemeine Beschreibung) | Nutzen: Mit dem Spiel sollen**Szenario:**  * Ein Kampfjet (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:**   * Kampfjet(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 Aktor   **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.   **Machbarkeitsabklärung:**   * Folgende Features sind vorab untersucht worden und ..... (Skizze / Mockup) |
| **MUSS**  **Kriterien:**  (Konkrete Features, die umzusetzen sind) | **Folgende Features sollen implementiert werden (Funktionalität):**   * Welt: Hintergrund, keine speziellen Funktionen * Der Spieler (Kampfjet) kann sich in jede Richtung bewegen / ausweichen * Der Kampfjet kann auf Gegner schiessen * Die Gegner schiessen den Kampfjet ab (jeweils der unterste von einer Spalte) * Die Gegner bewegen sich synchron nach links, rechts und unten * Verschiedene Levels |

|  |  |
| --- | --- |
| **KANN**  **Kriterien:**  (Konkrete Features, die optional sind) | **Folgende Features können zusätzlich implementiert werden: (Kreativität)**   * “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) |

## Planung LB2

|  |  |  |  |
| --- | --- | --- | --- |
| *MS* | *Tätigkeit / Abgabe* | *Soll-Datum* | *Ist-Datum* |
| A | ProjektstartTeam BildungWahl / Ausarbeitung der Anforderungsdefinition Abnahme Anforderungsdefinition durch Lehrperson | 06.12.22 | 06.12.22 |
| B | Teamaufgabe 1:Abgabe: Lösungsdesign  (Analyse, Design: Funktionsmodell, UseCase, GUI, Storyboard) | 13.12.22 |  |
| B2 | Teamaufgabe 2:Abgabe: Testvorschrift und Testfälle |  |  |
| C | Einzelaufgabe 3:Abgabe Szenario (.zip) mit Inline-Dokumentation, Systemdokumentation (UML Klassen-, Sequenzdiagramm)Fachgespräch Projektabnahme |  |  |
| C2 | Einzelaufgabe 4:Abgabe: Ausgefüllter Systemtest |  |  |

# Lösungsdesign (Meilenstein B: Teamaufgabe 1)

Anhand der Analyse wurde folgendes Lösungsdesign entworfen:

## Funktionsmodell

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

...

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

Legende:

* Eingabe: Steuerung des Kampfjets in alle Richtungen (mit WASD & Pfeiltasten). Schiessen mit Space-Taste.
* Ausgabe: Ufos

## Anwendungsfälle (UseCases)

Folgende Anwendungsfälle sind hier detailliert dokumentiert:

...

(Detaillierte Beschreibung der UseCases)

Legende:

...

## Ablauf

Aus Benutzersicht ist folgender Ablauf des Programms zu erwarten:

...

(Storyboard)

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

# Testvorschrift (LB2 Meilenstein B2: Teamaufgabe 2)

Testbeschrieb und vorbereitetes Testprotokoll siehe Dokument   
***M226B\_LB2\_Testvorschrift\_MS-B2.docx***

# Systemdokumentation (Meilenstein C: individuelle Aufgabe 3)

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

***M226B\_ Aufgabe\_3\_Szenario\_IhrName.zip***

## Statisches Design: Klassendiagramm

Folgend die statische Struktur des Szenarios

...

(UML Klassendiagramm mit Assoziationen und Kardinalitäten)

## Umfang / Abgrenzung / Änderungen gegenüber Design

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

...

(Umstände / Anpassungen / Veränderungen)

## 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 \*\*/)

## Dynamische Struktur: Sequenzdiagramm

Ein zentraler Ablauf eines UseCases ist im Folgenden dargestellt:

...

(Darstellung eines zentralen Ablaufs mittels Sequenzdiagramm)

**Trace: ...**

*...*

# Bedienungsanleitung (Meilenstein C: individuelle Aufgabe 3)

...

# Testprotokoll (LB2 Meilenstein C2: individuelle Aufgabe 4)

Ausgefülltes Testprotokoll siehe Dokument   
***M226B\_LB2\_Testvorschrift\_MS-C2\_Name.docx***