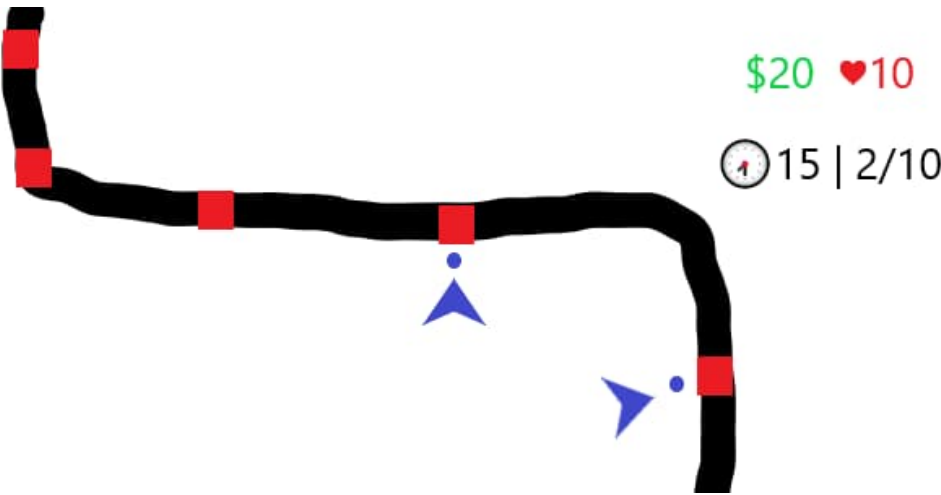


## 1 Teilnehmer/innen des Teams:

Klasse: BI19a	Team: Stählin Matteo Habtom Abrham
------------------	--

## 2 Anforderungsdefinition (Meilenstein A)

„Tower Defense“	
<b>Auftrag:</b> (Allgemeine Beschreibung)	<p><b>Nutzen: Unterhaltung</b></p> <p><b>Szenario:</b></p> <p>Tower Defense Game. Es gibt einen einfarbigen Pfad. Denn die Gegner folgen. Mann soll auf Kästchen drücken können, um Türme zu platzieren die Automatisch auf Gegner Schiessen. Man gewinnt, wen man keine Gegner durchlässt. Es gibt einen Wave timer der die Zeit zwischen den Gegner spawns anzeigt und eine Anzeige auf welcher Wave man ist. Für besiegte</p> <p><b>Details:</b></p> <ul style="list-style-type: none"><li>• Pfad den Gegner folgen.</li><li>• Geld mit dem man Towers kaufen kann.</li><li>• Tower die auf Kugeln Schiessen.</li><li>• Leben die man Verliert, wen Gegner durchkommen.</li></ul> <p><b>Machbarkeitsabklärung:</b></p> <ul style="list-style-type: none"><li>• Gegner Folgen der Farbe des Pfades.</li><li>•</li></ul> 

<b>MUSS</b> <b>Kriterien:</b> (Konkrete Features, die umzusetzen sind)	<b>Folgende Features sollen implementiert werden (Funktionalität):</b> <ul style="list-style-type: none"><li>• Geld mit dem man Towers kaufen kann.</li><li>• Gegner die einem Pfad folgen.</li><li>• Mehrere Levels.</li><li>• Leben, die Man verliert falls ein Gegner den Pfad beendet.</li><li>• Towers die auf Gegner schiessen.</li></ul>
<b>KANN</b> <b>Kriterien:</b> (Konkrete Features, die optional sind)	<b>Folgende Features können zusätzlich implementiert werden: (Kreativität)</b> <ul style="list-style-type: none"><li>• Mehrere Gegnerarten (Schnellere oder solche mit mehr Leben).</li><li>• Mehrere Tower arten.</li><li>• Magie mit Abklingzeit die man irgendwo einsetzen kann.</li></ul>

## 2.1 Planung LB2

<i>MS</i>	<i>Tätigkeit / Abgabe</i>	<i>Soll-Datum</i>	<i>Ist-Datum</i>
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		
B	<b>Teamaufgabe 1:</b> <ul style="list-style-type: none"><li>➤ <b>Abgabe: Lösungsdesign</b> (Analyse, Design: Funktionsmodell, UseCase, GUI, Storyboard)</li></ul>		
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 Projektabnahme</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:

...

(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\*\*](#)