

1 Teilnehmer/innen des Teams:

Klasse: BI19a	Team: Sirdan Sirius Schürmann, Adan Mohamed M226B LB2
------------------	--

2 Anforderungsdefinition (Meilenstein A)

„Coincatcher“

Auftrag: (Allgemeine Beschreibung)

Nutzen: Das Spiel soll dem Endnutzer möglichst viel Spass bereiten.

Szenario:

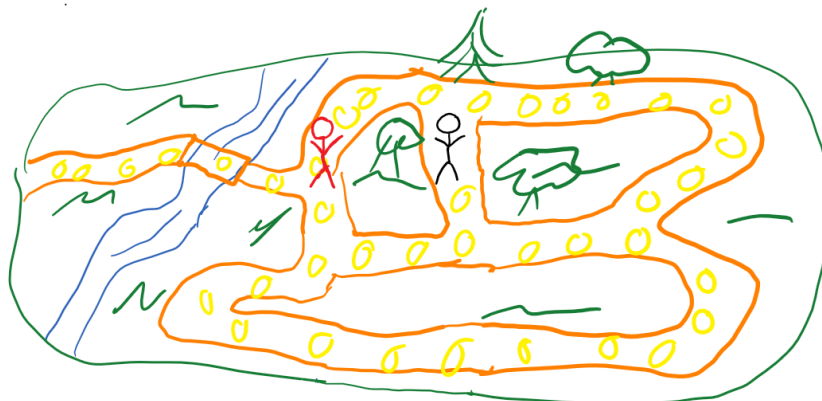
Das Spiel namens Coincatcher umfasst eine Welt, welche über verschiedene begehbare Wege verfügt. Das Ziel des Spielers besteht darin alle Münzen, welche auf der Karte verteilt sind, einzusammeln, um das Spiel zu gewinnen. Jedoch gibt es auch feindliche Kreaturen, die versuchen, den Spieler zu erreichen. Falls dies geschieht, wird das Spiel beendet und als verloren gewertet.

Details:

- Es soll eine Map existieren, welche von begehbaren und nicht begehbaren Wegen/Bereichen unterscheidet.
- Es sollen einsammelbare Münzen auf der Map verteilt sein
- Es gibt feindliche Kreaturen, welche den Spieler versuchen zu jagen

Machbarkeitsabklärung:

- Folgende Skizze veranschaulicht das Projekt.



MUSS Kriterien: (Konkrete Features, die umzusetzen sind)	Folgende Features sollen implementiert werden (Funktionalität): <ul style="list-style-type: none">• Eine Map mit begrenzt begehbaren Bereichen• Einsammelbare Münzen• Main Character, welcher über WASD gesteuert werden kann• Feindliche Kreatur• Counter für die Anzahl an Coins• Verschiedene Levels/Maps• Game Over Screen, falls man getötet wird
KANN Kriterien: (Konkrete Features, die optional sind)	Folgende Features können zusätzlich implementiert werden: (Kreativität) <ul style="list-style-type: none">• Einsammelbare Special Abilitys/Power Ups• Verlangsamung durch Brücken• Möglichkeit, Feinde zu besiegen• Mehr Feinde• (Als Spieler die Möglichkeit die Welt selbständig zu dekorieren oder evtl. sogar zu kreieren)

2.1 Planung LB2

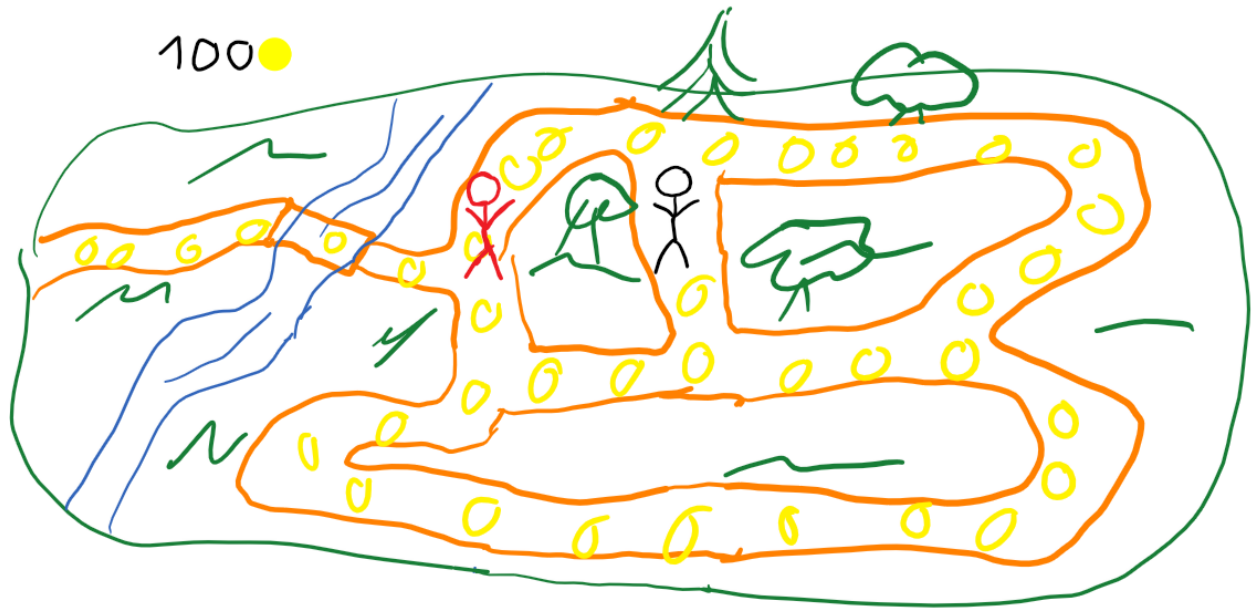
<i>MS</i>	<i>Tätigkeit / Abgabe</i>	<i>Soll-Datum</i>	<i>Ist-Datum</i>
A	Projektstart <ul style="list-style-type: none"> ➤ Team Bildung ➤ Wahl / Ausarbeitung der Anforderungsdefinition Abnahme Anforderungsdefinition durch Lehrperson	06.12.2022	13.12.2022
B	Teamaufgabe 1: <ul style="list-style-type: none"> ➤ Abgabe: Lösungsdesign (Analyse, Design: Funktionsmodell, UseCase, GUI, Storyboard) 	13.12.2022	13.12.2022
B2	Teamaufgabe 2: <ul style="list-style-type: none"> ➤ Abgabe: Testvorschrift und Testfälle 	20.12.2022	
C	Einzelaufgabe 3: <ul style="list-style-type: none"> ➤ Abgabe Szenario (.zip) mit Inline-Dokumentation, Systemdokumentation (UML Klassen-, Sequenzdiagramm) ➤ Fachgespräch Projektabnahme 	xx.xx.2023	
C2	Einzelaufgabe 4: <ul style="list-style-type: none"> ➤ Abgabe: Ausgefüllter Systemtest 	xx.xx.2023	

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:

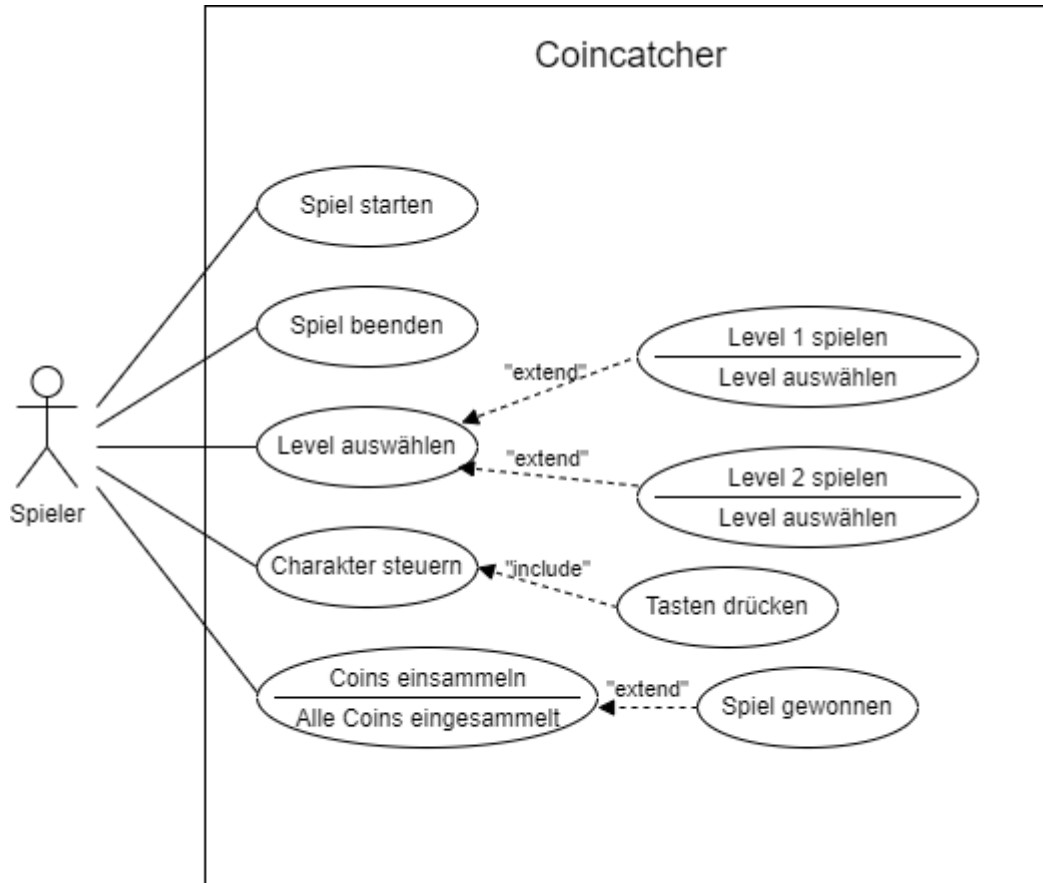


Legende:

- Der Maincharacter (schwarz) kann mit den Tasten WASD bewegt werden und kann Münzen einsammeln, welche sich auf den begehbaren Wegen befinden.
- Die Anzahl an gesammelten Münzen ist ersichtlich und wird jeweils hochgezählt.
- Der automatisierte Gegner (rot) versucht den Spieler zu erreichen. Falls dies geschieht, wird das Spiel beendet.
- Wenn der Maincharacter alle Münzen eingesammelt hat, wird das Spiel als gewonnen gewertet und beendet.
- Zu Beginn des Spiels kann das gewünschte Level ausgewählt werden.

3.2 Anwendungsfälle (UseCases)

Folgende Anwendungsfälle sind hier detailliert dokumentiert:





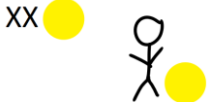



Legende:

Der Spieler kann das Spiel starten und Stoppen. Zu Beginn des Spiels kann der Spieler ein Level auswählen, indem auf jenes geklickt wird. Der Charakter kann via WASD gesteuert werden und kann Coins einsammeln. Falls alle Coins eingesammelt werden, hat man das Spiel gewonnen.

3.3 Ablauf

Aus Benutzersicht ist folgender Ablauf des Programms zu erwarten:

Storyboard:

	<p>Wenn das Spiel gestartet wird, kann der Spieler zu-nächst das gewünschte Level auswählen.</p>
	<p>Sobald das Level ausgewählt wurde, wird die Map generiert und das Spiel beginnt</p>
	<p>Nun versucht der Spieler alle Coins (Münzen), welche auf der gesamten Map verteilt sind, einzusammeln.</p>
	<p>Der automatisierte Gegner versucht den Spieler zu erreichen und zu töten.</p>
	<p>Falls alle Coins eingesammelt wurden, wird das Spiel als gewonnen gewertet und beendet</p>
	<p>Falls der Spieler vom Gegner erwischt, bevor alle Coins eingesammelt wurden, wird das Spiel als verloren gewertet</p>

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*](#)