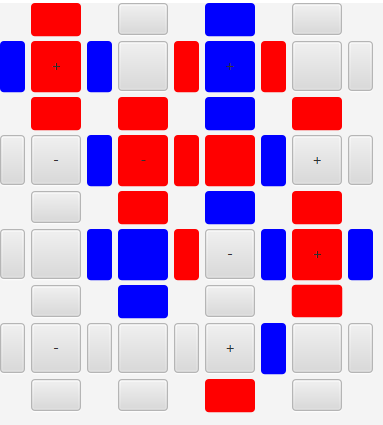
Software Entwicklung 2

Sommersemester 2019



Käsekästchen

Mascha Weis, Matrikelnummer: 37052

Susanne Weiß, Matrikelnummer: 37404

Alexander Kraus, Matrikelnummer: 37034

<https://gitlab.mi.hdm-stuttgart.de/sw191/kaesekaestchen>

1. **Kurzbeschreibung**

Bei unserem Projekt handelt es sich um ein Minispiel „Käsekästchen“ (Projektname: kaesekaestchen) für das zweite Semester des Studiengangs Medieninformatik an der Hochschule der Medien.

Es geht darum möglichst viele Kästchen zu erobern. Ein Kästchen wird erobert, wenn die vierte Strebe platziert wird. Diese dürfen willkürlich auf dem ganzen Spielfeld gesetzt werden. Um das Spiel spannender zu gestalten, gibt es die Möglichkeit Bonus- und/oder Minusfelder zu aktivieren, die dann zufällig erstellt werden. Zwei Spieler spielen abwechselnd. Während ein „normales“ Feld 1 Punkt gibt, geben Bonusfelder 3 und Minusfelder -2 Punkte. Sobald es keine leeren Felder mehr gibt, gewinnt der Spieler mit den meisten Punkten.

1. **Startklasse**

Die Main-Methode befindet sich in der Klasse **GUI**.

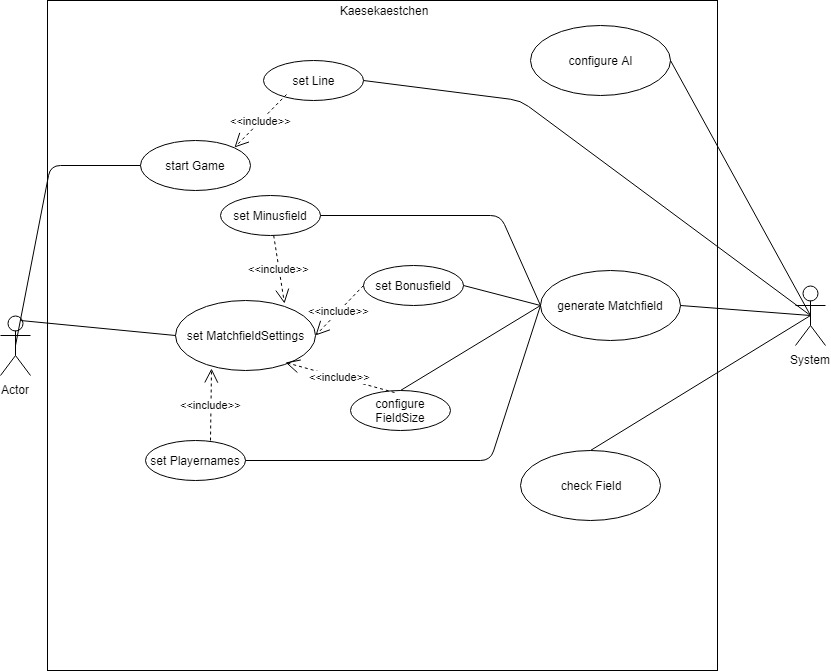
1. **Besonderheiten**

Unser Spiel wäre noch um einen AI Player erweiterbar (Ursprünglich geplant. Aus Zeitgründen leider nicht umsetzbar gewesen).

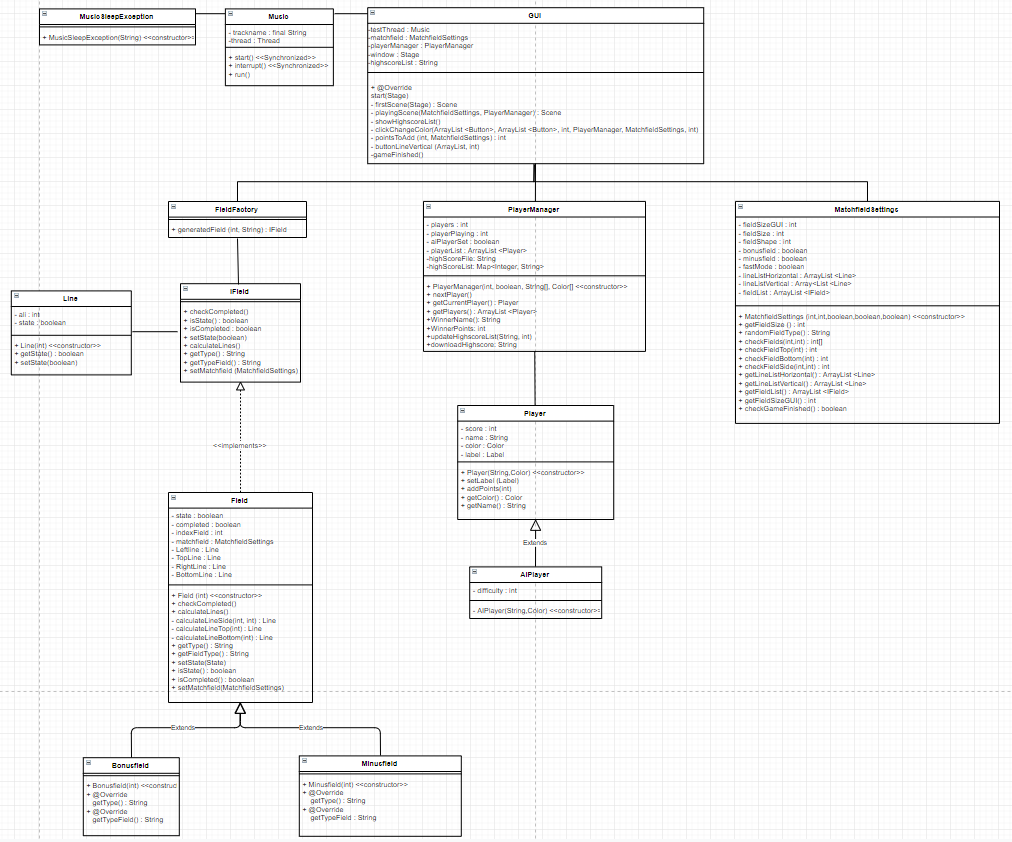
Für die Matchfield Einstellungen gibt es noch weitere Möglichkeiten, wie „fieldShape“ und „fastMode“ (auch dies aus Zeitgründen nicht Umsetzbar gewesen).

Java 11.0.2 SDK wird verwendet

1. **UML-Klassendiagramme**
   1. Use-Case Diagramm



* 1. Klassendiagramm



1. **Stellungnahme**

Architektur

Interface: IField, Field implements iField

Vererbung: Bonusfield extends Field und Minusfield extends Field, GUI extends Application, AIPlayer extends Player

Packages: CustomExceptions, Matchfield, PlayerManager, Threads

Factory: FieldFactory generiert IField

Clean Code

Keine Public Members vorhanden, sind alle private.

Wenige Static Methoden wurden benutzt.

Getter Methoden geben Kopien der Referenz zurück.

Tests

Mehrere Klassen (FieldTest, LineTest, MatchfieldSettingsTest, PlayerTest) werden getestet, inklusive Negativtest.

GUI

Komplexeres GUI, zwei Scenes mit Starteinstellungen und Spielfeld.

Starteinstellungen: Borderpane,Gridpane, Labels, Textfield und Buttons

Spielfeld (variable Größe): Gridpane, Buttons, Label

Logging

Verschiedene Loglevels (Debug, Info und Error) vorhanden, Sinnvolle Anwendung in verschiedene Klassen, Logging von Exceptions und Threads

UML

Use-Case und Klassendiagramm vorhanden (Im Repository und in der Dokumentation).

Threads

Ein Thread vorhanden in Klasse Music um Hintergrundmusik abzuspielen.

Streams

Lambda Funktion vorhanden (Klick von Buttons)

Methode checkGameFinished

Streams zur Sortierung einer Map vorhanden

Methode: updateHighscoreList

Seite 6: Ausgefüllter Bewertungsbogen (Excel) für das Projekt Bewertungsbogen ist auch im Repository!

