



Die Hausaufgaben müssen von jedem Studierenden einzeln bearbeitet und abgegeben werden. Für die Hausaufgabe sind die aktuellen Informationen vom Blog <https://seblog.cs.uni-kassel.de/ws2122/programming-and-modelling/> zu berücksichtigen. Diese Hausaufgabe enthält die Aufgaben des Projekts.

Abgabefrist ist der 10.02.2022 - 23:59 Uhr

Abgabe

Für das Projekt benötigst du **kein neues** Repository. Es wird das gleiche Repository benutzt, das bereits in Hausaufgabe 4 angelegt wurde. Dieses kann über folgenden Link eingesehen oder auch erstellt werden, falls nicht bereits geschehen:

<https://classroom.github.com/a/S1VERCvA>

Aufgabe 1 - Vervollständigung

Vervollständige dein Mühle-Spiel. Es muss möglich sein, dass zwei Spieler gegeneinander Mühle spielen können, wobei dein Programm entsprechend auf die Eingaben reagiert und nur gültige Spielzüge zulässt. Die Regeln des Spiels¹ müssen korrekt implementiert sein.

Hierbei darf auch die bereitgestellte Musterlösung² verwendet werden.

Fehler aus vergangenen Hausaufgaben müssen korrigiert werden. Hierzu zählen neben dem reinen Programmcode auch die Tests. Arbeite Tests nach, die unsauber oder gar nicht implementiert sind, sodass sie dir dabei helfen können fehlerhafte Logik zu finden, die dann nachgebessert werden kann.

¹Spielregeln: https://en.wikipedia.org/wiki/Nine_men%27s_morris#Rules

²Musterlösung: <https://github.com/sekassel/pmws2122-files/tree/main/HA09>

Aufgabe 2 - Mensch gegen Computer

Implementiere einen Spielmodus, in welchem der Nutzer gegen den Computer spielen kann. Wenn der Computer an der Reihe ist, wird durch eine künstliche Intelligenz der nächste Spielzug getätigt.

Deine künstliche Intelligenz muss nicht besonders schlau sein. Allerdings muss sie immer, wenn ein valider Zug möglich ist, einen solchen tätigen und darf nicht aufgeben!

Neuen Spielmodus ermöglichen

Das Hauptmenü muss angepasst werden, sodass der Nutzer den neuen Spielmodus wählen kann.

Eine mögliche Umsetzung könnte so aussehen:

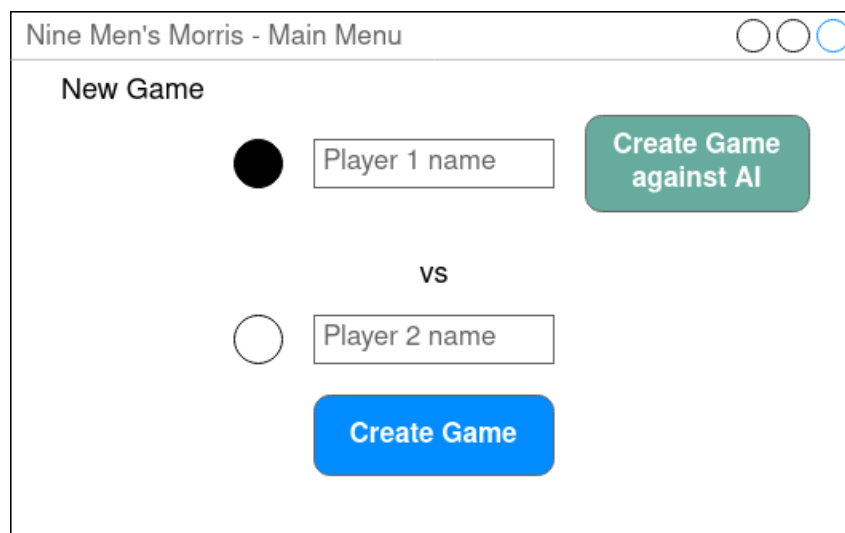


Abbildung 1: Angepasstes SetupScreen-Wireframe

Neben der FXML-Datei muss auch der SetupScreenController entsprechend erweitert werden. Wird der neue Button geklickt, muss nur noch bei Spieler 1 geprüft werden, ob er einen Namen eingegeben hat. Der zweite Spieler kann dann intern mit einem festgelegten Namen versehen werden. Die Auswahl des Spielmodus muss außerdem an den ModelService weitergereicht werden. Hierfür kann beispielsweise die `buildGame`-Methode um einen weiteren Parameter, der den Spielmodus angibt, erweitert werden. Innerhalb der `buildGame`-Methode kann dann je nach Spielmodus ein `brain` initialisiert werden.

Brain.java

Im package `de.uniks.pmws2122.ai` soll eine `Brain`-Klasse implementiert werden, die die Logik der künstlichen Intelligenz enthält.

Die Struktur der Klasse ist vorgegeben. Lade dir die Datei hier herunter:

<https://github.com/sekassel/pmws2122-files/blob/main/HA-Project-M/Brain.java>

Die darin angegebenen TODOs geben eine kleine Hilfestellung zur Implementierung.

Die KI einbinden

Im ModelService soll außerdem die nextTurn-Methode entsprechend des folgenden TODOs erweitert werden, um den Computer spielen zu lassen:

```
public void nextTurn() {
    // Set the game to the next player
    this.game.setCurrentPlayer(this.game.getCurrentPlayer().getNext());

    // Check if the game is in the next phase
    this.checkNextPhase();

    /*
     * TODO: if it is a human vs computer game,
     * and it's the computer's turn, let the computer make its move
     */
}
```

Um den Computer seinen Zug machen zu lassen, rufe die turn-Methode des brains auf.

Hilfestellung zu Random

Möglicherweise werden Zufallszahlen bei der Implementierung der KI benötigt. In Java gibt es verschiedene Möglichkeiten, um eine Zufallszahl in einem bestimmten Zahlenbereich zu erhalten:

<https://stackoverflow.com/questions/5887709>

Wenn du während der Entwicklungsphase reproduzierbare Zufallszahlen erhalten möchtest, um beim Ausprobieren deines Programms besser nachvollziehen zu können, was passiert, kannst du beispielsweise einen Seed setzen:

<https://stackoverflow.com/questions/12458383>

Information zur Fehlerbehebung

Im Laufe der Entwicklung können dir z. B. NullPointerExceptions begegnen. Lerne, den ausgegebenen Stacktrace auf der Konsole zu lesen. Häufig ist damit leicht zu erkennen, an welcher Stelle die Exception aufgetreten ist. Du kannst an geeigneten Stellen deines Codes Breakpoints setzen und dein Programm debuggen, um herauszufinden, was bei der Ausführung deines Programms passiert.

Informationen zur Abgabe und Präsentation

- Dein Programmcode muss pünktlich abgegeben werden (siehe Deadline).
- Im Laufe der zweiten Bearbeitungswoche wird ein Doodle gepostet (Ankündigung über Discord), über das sich die Studierenden einen Zeitslot auswählen müssen, in dem sie ihr Projekt präsentieren. Diese Präsentationen werden erst nach der Abgabe stattfinden.
- Die Präsentation wird über Discord stattfinden. Hierfür ist eine Webcam erforderlich, da wir deine Identität prüfen müssen. Halte hierfür deinen Personalausweis und deinen Studierendenausweis bereit.
- Es soll **keine** PowerPoint-Präsentation vorbereitet werden. Die Ergebnisse des Projektes werden am laufenden Programm und am Programmcode erklärt.
- Innerhalb von ca. 10 Minuten sollen die Ergebnisse von Aufgabe 2 als Demo präsentiert werden.
- Innerhalb von ca. 5 Minuten soll der Programmcode (Schwerpunkt künstliche Intelligenz) präsentiert werden. Die Prüfenden werden dabei Fragen stellen.
- Die Auswertung wird einige Zeit dauern, daher findet die Notenvergabe erst entsprechend später statt.