

Achtung: Lesen Sie zuerst die gesamte Aufgabenstellung. Falls irgendetwas unklar sein sollte, fragen Sie!

Aufgabenstellung

Entwickelt wird eine digitale Version des Spiels „Mensch ärgere dich nicht“. Das Spiel wird in zwei Stufen entwickelt:

- Entwicklung einer lokalen Version (Meilenstein II, 23.5.2014)
- Entwicklung einer Netzversion (Meilenstein III, vorauss. bis KW 24)

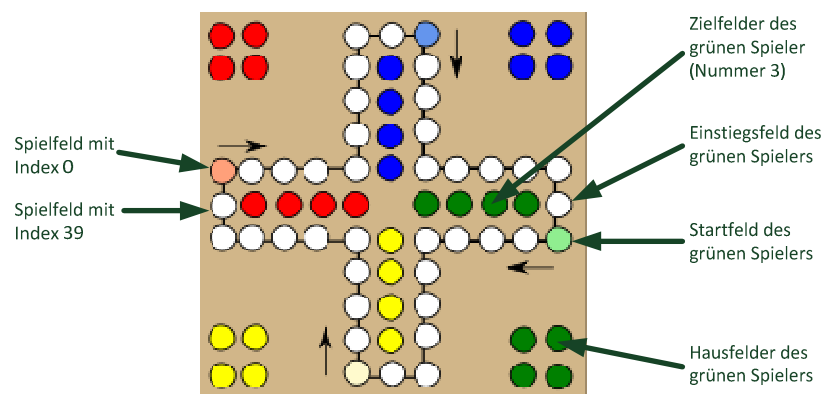
Die Abgaberichtlinien zu den Meilensteinen finden Sie jeweils in Moodle im Abgabeformular. Starten Sie mit der Minimalversion in Moodle. Im jar-Archiv befindet sich ein vollständiges Eclipse-Projekt, welches Sie in Ihren Workspace importieren können.

Hinweise zur Abnahme

- Arbeiten Sie in Zweiergruppen. Die Abnahme erfolgt einmalig am Ende des Semesters in Einzelkolloquien (KW 25 – 27). Diese entscheiden über den Schein. Zu den Meilensteinen werde ich mir von den Gruppen den jeweiligen Stand vorführen lassen.
- Die im Minimalprojekt vorgegebene Struktur sollte möglichst nicht verändert werden. Dies gilt insbesondere für die Schnittstellen. Sie dürfen natürlich Erweiterungen einführen. Eigene Spielentwürfe, die signifikant von den Vorgaben abweichen, werden nicht akzeptiert!
- Die Spiellogik ist im Regelwerk unten vorgegeben und darf nicht verändert werden (auch wenn Sie evtl. die eine oder andere Regel anders kennen sollten). Die Einhaltung der Regeln kann über eine JUnit-Testklasse, die Sie im Minimalprojekt finden, automatisiert geprüft werden.

Regelwerk „Mensch ärgere dich nicht“

Es spielen maximal vier Spieler. Jeder Spieler erhält vier Spielsteine einer Farbe. Die Spielsteine stehen zu Beginn auf den Hausfeldern des Spielers. Ziel des Spiels ist es, alle Spielfiguren der Spieler auf ihre jeweiligen Zielfelder zu bekommen.



Spielbrett Mensch ärgere dich nicht (Quelle: Wikipedia)

Im Spielverlauf würfeln und ziehen die Spieler in der definierten Reihenfolge (rot, blau, grün, gelb) auf der grafischen Benutzeroberfläche. Das Spiel sollte vollständig den Ablauf regeln und kein fehlerhaftes Verhalten zulassen.

Der folgende Zustandsautomat definiert die Spielregeln:

Ausgangszustand	Ereignis (was tut der Spieler)	Spielzustand bevor das Ereignis stattfindet.	Automatische Reaktion des Spiels	Neuer Zustand
Der aktuelle Spieler darf würfeln	Der aktuelle Spieler hat eine Würfelzahl < 6 gewürfelt.	Keiner der Spielsteine des aktuellen Spielers kann ziehen && es sind noch mehr als 1 Würfelversuch übrig.	Anzahl der Würfelversuche wird um 1 reduziert.	Der aktuelle Spieler darf würfeln
Der aktuelle Spieler darf würfeln	Der aktuelle Spieler hat eine Würfelzahl < 6 gewürfelt.	Keiner der Spielsteine des aktuellen Spielers kann ziehen && Anzahl der Würfelversuche ist 1.	Das System wählt den nächsten Spieler als aktuellen Spieler.	Der aktuelle Spieler darf würfeln
Der aktuelle Spieler darf würfeln	Der aktuelle Spieler hat eine Würfelzahl == 6 gewürfelt.	Mindestens ein Spielstein des aktuellen Spielers ist zu Hause && sein Startfeld ist frei && Anzahl der Würfelversuche ist größer gleich 1	Das System setzt einen beliebigen Spielstein des aktuellen Spielers aus dem Haus auf dessen Startfeld && Anzahl der Würfelversuche wird um 1 erhöht	Startfeld des aktuellen Spielers ist besetzt. Der aktuelle Spieler darf würfeln
Startfeld des aktuellen Spielers ist besetzt. Der aktuelle Spieler darf würfeln	Der aktuelle Spieler hat eine Würfelzahl <= 6 gewürfelt.	Der Spielstein des aktuellen Spielers auf dem Startfeld kann ziehen.	Das System zieht den Spielstein auf dem Startfeld automatisch && wählt den nächsten Spieler als aktuellen Spieler.	Der aktuelle Spieler darf würfeln
Startfeld des aktuellen Spielers ist besetzt. Der aktuelle Spieler darf würfeln	Der aktuelle Spieler hat eine Würfelzahl <= 6 gewürfelt.	Der Spielstein des aktuellen Spielers auf dem Startfeld ist blockiert.	Das System wählt den nächsten Spieler als aktuellen Spieler.	Der aktuelle Spieler darf würfeln
Der aktuelle Spieler darf würfeln	Der aktuelle Spieler hat eine Würfelzahl < 6 gewürfelt.	Mindestens ein Spielstein des aktuellen Spielers kann ziehen && dessen Startfeld ist nicht belegt.		Der aktuelle Spieler darf ziehen
Der aktuelle Spieler darf würfeln	Der aktuelle Spieler hat eine Würfelzahl == 6 gewürfelt.	Der aktuelle Spieler hat keinen Spielstein mehr zu Hause && mindestens einer seiner Spielstein kann ziehen.		Der aktuelle Spieler darf ziehen
Der aktuelle Spieler darf ziehen	Der aktuelle Spieler wählt einen seiner Spielsteine zum Ziehen aus	Zug bringt nicht alle Spielsteine ins Ziel	Das System setzt den gewählten Spielstein auf das neue Feld && wählt den nächsten Spieler.	Der aktuelle Spieler darf würfeln
Der aktuelle Spieler darf ziehen	Der aktuelle Spieler wählt einen seiner Spielsteine zum Ziehen aus	Zug bringt alle Spielsteine ins Ziel	Das System setzt den gewählten Spielstein auf das neue Feld && zeigt den Sieger	Ende

Es gelten außerdem folgende Zusatzregeln:

- Spielsteine können sich gegenseitig schlagen. Es besteht allerdings kein Schlagzwang. Der geschlagene Stein muss wieder nach Hause.
- Es kann immer nur ein Spielstein auf einem Feld stehen.
- Ein Spieler darf niemals sich selbst schlagen.
- Die Felder im Ziel werden ab dem Einstiegsfeld fortlaufend gezählt. Damit ein Stein in das Ziel einfahren darf, muss er direkt auf ein freies Feld im Ziel fahren können.
- Innerhalb der Zielfelder darf nicht gefahren werden. Belegte Felder dürfen jedoch übersprungen werden.

Struktur der Basisanwendung

Klasse	Aufgaben
Game	Zuständig für die Initialisierung des Spiels und die Steuerung des Gesamtspiels. Das Game repräsentiert die zentrale Schnittstelle nach außen. Auf dem Spiel ist es möglich zu würfeln und zu ziehen.
GameBoard	Die Klasse GameBoard repräsentiert den Spielzustand (aktuelle Position der Spielsteine) und den Aufbau des Spiels (Struktur und Zuordnung der Spielfelder). Sie ist notwendig für die Abstimmung von Spiellogik und Spielanzeige. Zustandsänderungen werden durch die Klassen Game oder Player vorgenommen. Die Klasse GameBoard selbst sollte möglichst nicht verändert werden.
Field	Ein Feld kann allgemeines Spielfeld oder direkt einem Spieler zugeordnet sein. Die allgemeinen Felder werden von der Klasse Game verwaltet. Die spieterspezifischen Felder werden von den Spielern selbst verwaltet. Auf einem Feld kann ein Spielstein stehen.
Player	Repräsentiert einen Spieler. Ein Spieler verwaltet seine persönlichen Felder und zieht mit seinen Spielfiguren.
Pawn	Repräsentiert eine Spielfigur. Jeder Spieler hat vier Spielfiguren, die ihm exklusiv zugeordnet sind.

Die Klassen DiceView, GameView, Main und FieldView realisieren die graphische Benutzeroberfläche. GameView nutzt GameBoard zur Anzeige des Spielverlaufs.

Die Testklasse

Die Klasse TestClient dient zum Test des korrekten Spielablaufs. Die Klasse arbeitet String-basiert, d.h. sie vergleicht anhand einer String-Repräsentation, ob der aktuelle Spielzustand korrekt ist. Im Rahmen des Tests wird ein vollständiges Beispiel (instructions.txt) durchgespielt und der Ist-Zustand Ihres Spiels jeweils mit den Sollvorgaben (example-game.txt) verglichen.

Die String-Repräsentation des Spielzustands ist wie folgt aufgebaut:

- Der Spieler, der den nächsten Zug durchführen darf: 1 (rot), 2 (blau), 3 (grün), 4 (gelb).
- Nächster Zug. Zur Auswahl stehen m (move für ziehen) oder r (roll für würfeln).
- Wert des letzten Würfelwurfs.
- Die vier Hausfelder eines jeden Spielers: Spielerzahl falls belegt, sonst 0.
- Die vier Zielfelder eines jeden Spielers: Spielerzahl falls belegt, sonst 0.
- Die vierzig Felder im Spiel: Spielerzahl falls belegt, sonst 0.

