

GDI Übungsblatt

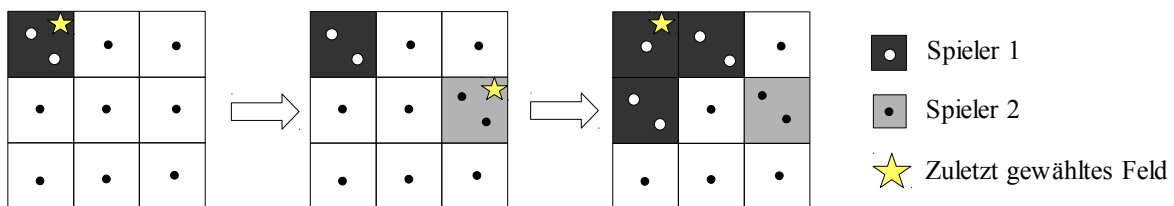
Aufgabe 1

Das Spiel „Jumping Cube“ ist **eine** Spiel für zwei Teilnehmer, das auf einem $n \times m$ Spielfeld basiert. Auf jedem Feld liegt ein Würfel, dessen Wert zu Beginn des Spiels den Wert „Eins“ zeigt. Die beiden Spieler wählen abwechselnd einen Würfel. Der Wert des Würfels wird um eins erhöht und dem Spieler zugeordnet, der den Würfel gewählt hat. Es dürfen nur Würfel gewählt werden, die **sich** selbst, oder noch keinem Spieler zugeordnet sind.

Hat ein Würfel einen höheren Wert als die Anzahl der angrenzenden Felder (ohne diagonale Nachbarn), wird der Wert des Würfels zurück auf 1 gesetzt und die umliegenden Felder werden um 1 erhöht und dem Spieler zugeordnet. Falls ein betroffener Würfel das Kriterium wiederum erfüllt, wird das Ereignis erneut ausgelöst.

Das Spiel ist beendet, wenn ein Spieler im Besitz aller Würfel ist.

Das folgende Bild beschreibt beispielhaft einen Auszug eines Spielablaufes. Im ersten Bild gehört der obere linke Würfel einem Spieler. Der zweite Spieler wählt dann einen anderen Würfel. Der erste Spieler wählt anschließend den selben Würfel wie im letzten Zug. Dieser Würfel hat nun den Wert „drei“. Da er nur zwei Nachbarn besitzt, **springen** die Werte dieses Würfels an die Nachbarn **über**. Die Werte der Würfel werden angepasst und die betroffenen Würfel werden dem Spieler zugeordnet.



Implementieren Sie dieses Spiel für die Konsole. Überlegen Sie sich eine geeignete Ausgabe, die Wert und Besitzer der Würfel verdeutlicht. Vor Beginn des Spiels soll die Größe des Spielfeldes gewählt werden können.

Eine mögliche Klassenaufteilung des Spiels wäre z.B.: Spiel, Spielfeld, Würfel

Tipp: Für eine einfache Umsetzung von Aufgabe 3 empfiehlt es sich, ein Interface „Spieler“ zu verwenden, sodass lediglich die Implementierungen ausgetauscht werden müssen.

Aufgabe 2

Nach jedem Spielzug soll es möglich sein das Spiel speichern zu können. Beim Start des Programms soll der Benutzer die Möglichkeit haben, ein gespeichertes Spiel fortsetzen zu können.

Beachten Sie, dass das Spiel mit dem korrekten Spieler fortgesetzt wird.

Aufgabe 3

Implementieren Sie einen einfachen Computergegner. Dieser soll aus den gültigen Zügen zufällig einen auswählen.

Vor Beginn des Spiels soll für beide Spieler gewählt werden können, ob dieser ein Mensch oder Computergegner ist.

Beim Speichern (Aufgabe 2) müssen die Spielertypen nicht mit gespeichert werden. Sie können beim Laden erneut abgefragt werden.

Folgender Code generiert eine Zufallszahl im Bereich von 0-9

```
import java.util.Random;  
...  
Random randomGenerator = new Random();  
int someRandomNumber = randomGenerator.nextInt(10);  
...
```

Die Dokumentation zur Random-Klasse finden Sie hier:

<http://java.sun.com/javase/6/docs/api/java/util/Random.html>