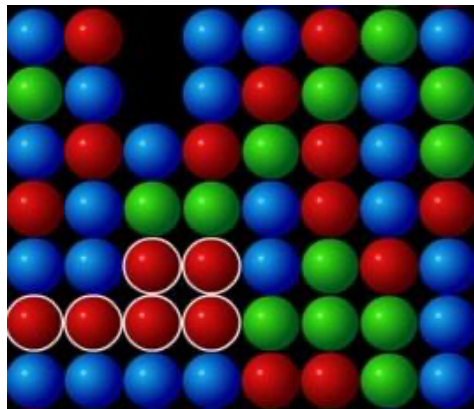


Projekt

Magic Marbles

Teil 1

Magic Marbles ist ein einfaches Computerspiel, bei dem ein Spieler horizontal und vertikal zusammenhängende gleichfarbige Murmeln auswählen kann und diese dann im nächsten Spielschritt verschwinden. Ziel des Spiels ist es, das gesamte Spielfeld zu leeren, wobei möglichst viele Murmeln auf einmal entfernt werden sollen. Damit erreicht man eine hohe Punktzahl zu erreichen. Übrig gebliebene einzelne Murmeln bedeuten am Spielende Punkteabzüge.



Ziel dieser Übung ist es, eine einfache Version von *Magic Marbles* zu entwickeln. In Teil 2 wollen wir dann dazu eine graphische Oberfläche bauen.

A) Implementierung des Spiels

Spielablauf:

Zu Beginn legt der Spieler die Spielfeldgröße fest, d.h., Ihre Version von *Magic Marbles* soll Spielfelder mit beliebiger Spielfeldgröße unterstützen. In weiterer Folge wird ein entsprechendes Spielfeld mit einer zufälligen Belegung von roten, grünen und blauen Murmeln erzeugt. Ab diesem Zeitpunkt wird (i) das Spielfeld auf der Konsole und (ii) die aktuellen Punkte ausgegeben sowie (iii) nach einem gültigen Zug gefragt. Dies wird wiederholt, bis keine weiteren gültigen Züge mehr getätigt werden können. Das Spiel ist beendet, wenn entweder das Spielfeld vollständig geleert wurde oder kein gültiger Zug mehr möglich ist (dies ist der Fall falls nur noch einzelne unzusammenhängende Murmeln übrig sind). Wurde das Ende erreicht, wird schließlich noch der erreichte Punktestand ausgegeben.

Punkteberechnung:

Die Anzahl der erreichten Punkte errechnet sich als Summe der bisher erreichten Punkte plus die Anzahl der auf einmal entfernten Murmeln zum Quadrat. Das heißt, werden in einem Zug 5 zusammenhängende Murmeln entfernt, so erhöht sich der Punktestand um 25 Punkte. Übrig gebliebene einzelne Murmeln am Spielende erniedrigen den Punktestand jeweils um 10, d.h., bleiben 2 Murmeln übrig, so werden 20 Punkte subtrahiert.

Exemplarischer Spielablauf:

```
Height: 5
Width: 4

  0  0  0  0
  1  2  3  4
01 b  g  g  r
02 b  g  r  b
03 g  g  g  g
04 r  r  r  r
05 g  r  r  r
*****
Current Score: 0
*****
```

Please select a non-empty field

Row: 5

Column: 4

```

    0 0 0 0
    1 2 3 4
01
02 b
03 b g g r
04 g g r b
05 g g g g
*****

```

Current Score: 49

Please select a non-empty field

Row: 5

Column: 3

```

    0 0 0 0
    1 2 3 4
01
02
03
04 b r
05 b r b
*****

```

Current Score: 113

Please select a non-empty field

Row: 5

Column: 2

```

    0 0 0 0
    1 2 3 4
01
02
03
04 r
05 r b

```

Final Score: 87

Game over!

Implementieren Sie das Spiel. Um Ihnen diese Aufgabe zu erleichtern, ist ein Klassengerüst vorgegeben (siehe `Vorgabe.zip`), welches folgende Pakete sowie Klassen enthält:

Paket `magicmarbles.model` (Umsetzung des Spiels):

- *Interface `MMField`*: Gibt die Methoden vor, die eine Implementierung von Magic Marbles anbieten muss.
- *Klasse `MMFieldImpl`*: In dieser Klasse soll das Spiel implementiert werden. In der Vorgabe sind hier die leeren Methodenrumpfe des Interfaces `Field.java` enthalten.
- *Klasse `MMException`*: Exception-Klasse für Exceptions bei der Auswahl von Positionen im Spiel.
- *Enumeration `MMFieldState.java`*: Gibt mögliche Werte vor, die ein Platz des Spielfeldes annehmen kann. Ein Platz kann entweder mit einer roten, grünen oder blauen Murmel besetzt sein oder kann leer sein.
- *Enumeration `MMState.java`*: Gibt mögliche Zustände vor, die das Spiel annehmen kann. Das Spiel kann entweder im Gange („running“) oder beendet („end“) sein.

Paket `magicmarbles.textui` (Konsolen-basierte Schnittstelle):

- *Klasse `Main.java`*: Implementiert eine Konsolen-basierte Anwendung, um mit dem Spiel zu interagieren.

Hinweise:

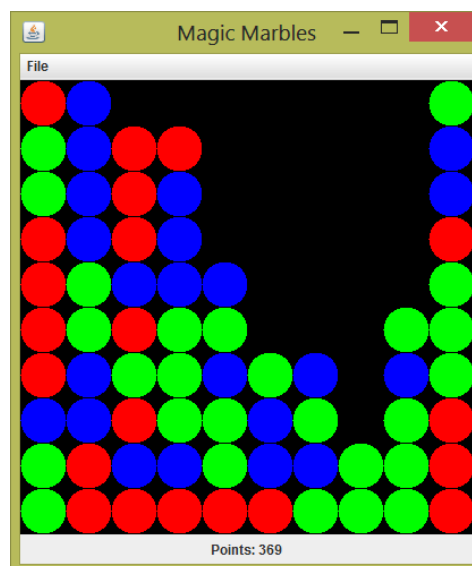
- Beachten Sie das automatische „Nachrutschen“ der Murmeln bei auftretenden Löchern zwischen den Murmeln. Hierbei rutschen die Murmeln einerseits in vertikaler Richtung nach (d.h., nach unten). Treten überdies leere Spalten zwischen den Murmeln auf, so werden die Murmeln in horizontaler Richtung

verschoben (d.h., nach rechts). Das heißt, die Murmeln bewegen sich bei auftretenden Löchern nach rechts unten (siehe Spielablauf).

- Beachten Sie überdies, dass nach jedem ausgeführten Zug überprüft werden muss, ob noch weitere gültige Züge möglich sind, d.h., ob das Ende des Spiels erreicht wurde.
- Beachten Sie schließlich, dass die kommenden Übungen 10 auf dieser Übung aufbauen werden – d.h., mit einer sauberen Ausarbeitung dieser Übung schaffen Sie sich eine gute Ausgangsbasis für die beiden folgenden Übungen!

Teil 2

In dieser Übung soll das Spiel *Magic Marbles* entsprechend der MVC-Architektur in Swing umgesetzt werden. Dabei soll eine eigene graphische Komponente zur Darstellung des Feldes und zur Benutzerinteraktion implementiert werden. Die folgende Abbildung zeigt, wie die Applikation aussehen könnte. Das erzeugte Spielfeld soll eine **fixe, von Ihnen gewählte** Größe haben.

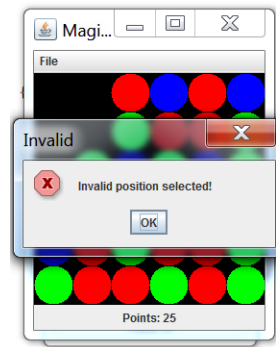


Die Aufgabe umfasst somit:

- Modell-Implementierung:
Realisieren Sie ein Modell für *Magic Marbles* nach dem MVC-Prinzip. Definieren Sie ein `MMListener`-Interface und stellen Sie eine `MMEventObject`-Klasse bereit.
- View-Implementierung:
Implementieren Sie eine View zur Anzeige des Spielfeldes und zur Selektion von Zellen (siehe Abbildung oben). Die View reagiert auf Mouse-Clicks und selektiert damit entsprechende Felder. Des Weiteren horcht die View auf Änderungen im Modell und zeichnet sich daraufhin neu.

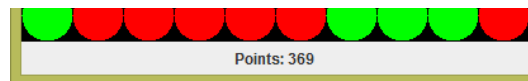
- Dialog bei falschen Eingaben:

Wurde vom Benutzer ein Feld selektiert, das keinen gültigen Spielzug darstellt (Feld frei oder Feld mit einem isolierten Stein), soll dies durch einen Dialog angezeigt werden.



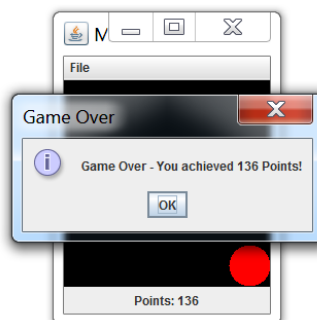
- Anzeige des Spielstandes:

Unter dem eigentlichen Spielfeld, sollen die derzeit erreichten Punkte angezeigt werden. Bei jeder Änderung wird unmittelbar der neue Spielstand angezeigt.



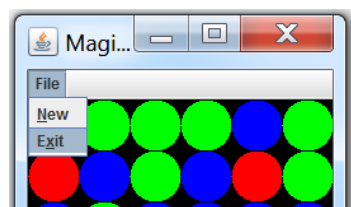
- Dialog mit Spielergebnis:

Am Ende des Spiels soll das Spielergebnis in einem eigenen Dialog angezeigt werden.



- Menü:

Die Applikation soll ein Menü mit Menüeinträgen „New“ und „Exit“ haben. Damit kann man jederzeit das Programm beenden bzw. ein neues Spiel beginnen. Die Initialisierung des Spiels erfolgt wiederum mit dem oben gezeigten Dialog.



Hinweise zu Swing:

Popup-Dialoge: Einen Dialog können Sie einfach mit JOptionPane erzeugen. Mit

```
JOptionPane.showMessageDialog(parentComponent, message, title, messageType)
```

kann man einen Dialog mit einer Meldung `message` erzeugen, wobei für `parentComponent` der Frame angegeben und als `messageType` eine Konstante von JOptionPane wie `JOptionPane.ERROR_MESSAGE`, `JOptionPane.WARNING_MESSAGE` oder `JOptionPane.INFORMATION_MESSAGE` verwendet werden soll.

Zeichnen einer Steins in bestimmter Farbe: Zeichnen von Figuren in einer bestimmten Farbe erfolgt durch 2 Schritte: (1) Setzen der Farbe in Graphics-Kontext und (2) dann Zeichnen der Figur, z.B.

```
g.setColor(Color.RED);  
g.fillOval(100, 100, 10, 10);
```

setzt die Farbe rot im Graphics-Kontext `g` und zeichnet dann einen gefüllten Kreis an Position 100/100 mit Durchmesser 10.