Hochschule Ravensburg-Weingarten
Technik | Wirtschaft | Sozialwesen

Studiengang Angewandte Informatik
Grafische Bedienoberflächen
Prof. Dr. S. Keller

09/03/2016

ÜBUNGSBLATT 5

Erweiterung auf Multithreading und Spieleinstellungen zentral verwalten

1. Erweitern Sie Ihr Spiel zur Multithreading-Variante

Bei dieser Variante wird für jeden der Gegner ein eigener Thread erstellt und die Gegner verfügen über eine hinterlegte Logik.

Zur Umstellung Ihres Spiels müssen Sie dazu eine Instanz der Klasse **MTConfiguration** erstellen und in dieser einige Einstellungen vornehmen (z.B. die Wartezeit der Gegner, die Anzahl an Schüssen, die der Spieler zu Beginn hat etc.). Die vorhandenen Einstellungsmöglichkeiten entnehmen Sie bitte der Javadoc-Dokumentation in Moodle.

Haben Sie alle nötigen Einstellungen vorgenommen, übergeben Sie das MTConfiguration-Objekt dem Controller mit **controller.setMultiThreaded(activeConfiguration).**

Eine beispielhafte Instanz der MTConfiguration hätte folgende Gestalt:

```
MTConfiguration conf = new MTConfiguration();
conf.setAlgorithmAStarActive(true);
conf.setAvoidCollisionWithObstacles(true);
conf.setAvoidCollisionWithOpponent(false);
conf.setMinimumTime(800);              // 0,8 Sekunden
conf.setShotGetsOwnThread(true);       // nicht unbegrenzte Anzahl Schüsse
conf.setOpponentStartWaitTime(5000);   // 5 Sekunden am Anfang Schlaf
conf.setOpponentWaitTime(2000);        // Gegner warten vor jedem Zug 2 Sekunden
conf.setShotWaitTime(500);            // ein Schuss benötigt eine halbe Sekunde
conf.setRandomOpponentWaitTime(false);  // keine zufällige Wartezeit
conf.setDynamicOpponentWaitTime(false); // immer gleichlang warten
controller.setMultiThreaded(conf);
```

2. Realisieren Sie die Munitionsanzeige in der Toolbar

In der Munitionsanzeige steht für eine vorhandene Munitionseinheit, die der Spieler besitzt, ein kleines Bild. Sobald der Spieler mehr als 3 Munitionseinheiten besitzt, (das ist der Wert zu Spielanfang), wird sein genauer Munitionsvorrat als Text angezeigt. Der Text nimmt den Platz des ersten Munitionsbildes ein. Der restliche Platz der Munitionsanzeige wird mit Munitionsbildern aufgefüllt. Sobald der Spieler wieder 3 Munitionseinheiten besitzt, verschwindet die Textanzeige und die Munitionsanzeige besteht wie gehabt nur aus Bildern. Hat der Spieler keine Munition mehr, so ist auch die Munitionsanzeige leer.

Das Munitionsbild soll nicht als JLabel realisiert werden, sondern als JPanel, in das mit den **Zeichenfunktionen von Java2D** ein Image gezeichnet wird. Dieses Panel kann dann wie ein JLabel im Container platziert werden.

Blinken der Munitionsanzeige.

Steht dem Spieler keine Munition zur Verfügung und versucht er trotzdem einen Schuss abzugeben, dann blinkt die Munitionsanzeige fünfmal auf. Wenn der Spieler während des Blinkens eine Munitionseinheit aufsammelt, so wird das Blinken unterbrochen.

- Für das Blinken brauchen Sie einen eigenen Thread, welcher in periodischen Abständen der Munitionsanzeige einen Rand gibt und diesen wieder löscht. Wenn Sie das Blinken ohne einen eigenen Thread zu realisieren versuchen, dann wird das restliche Spiel in der Zeit des Blinkens zum Stillstand kommen.
- Versucht der Spieler mehrmals kurz nach einander einen Schuss abzugeben, während er keine Munition zur Verfügung hat, so darf in diesem Fall das Blinken nicht mehrmals gestartet werden.

3. Spieleinstellungen über den Button Settings anzeigen und modifizieren

Die wichtigsten Spieleinstellungen werden in einer zentralen Klasse „**Settings**“ verwaltet. In der Klasse werden die Spieleinstellungen in einer **Map** abgelegt. Dazu erhält jede Einstellung einen Namen und einen Wert. Die Klasse stellt Methoden bereit, mithilfe derer man Spieleinstellungen speichern, ändern, lesen und löschen kann.

Der Spieler hat nun die Möglichkeit alle Spieleinstellungen, die in der Klasse „Settings“ gespeichert sind, über den Button „ Setting“ in der **Toolbar** anzuzeigen.

Ein Klick auf den Button „Settings“ öffnet ein Dialogfenster vom Typ JDialog, welches die Einstellungen des Spiels anzeigt und anpassbar macht.

Beispiel: