



Übung zur „Einführung in die Programmierung – Java-Projekt“, WS 21/22

Übungsleiter: Ingrid Schumacher <schumache@itm.uni-luebeck.de>
Klaus-Dieter Schumacher <schumacher@itm.uni-luebeck.de>

Aufgabenblatt 3

Übungsblatt vom: Sonntag, 12.12.2021
Abgabe der Übung: Sonntag, 16.01.2022 um 23:30 Uhr

Bearbeitungshinweise

- Bevor Sie beginnen zu programmieren, lesen Sie zunächst die gesamte Aufgabenstellung mit allen Teilaufgaben, damit Sie einen vollständigen Überblick über das von Ihnen zu erstellende Teilprojekt erhalten.
- Lauffähige Zwischenstände sollten als Sicherheitskopien angelegt werden, damit bei späteren Fehlern damit weitergearbeitet werden kann. Da es in Eclipse nicht möglich ist, mehrere Projekte gleichen Namens zu verwalten, werden die Kopien mit einer laufenden Nummer ergänzt: <Gruppennummer>-SchiffeA3<.x>. So entstehen Projektnamen wie <Gruppennummer>-SchiffeA3.1, je nachdem wie viele Sicherheitskopien angelegt wurden. Bevor Sie mit der Implementierung beginnen, kopieren Sie Ihr Ergebnisprojekt des vorherigen Aufgabenblatts in Eclipse gemäß der Vorgabe nach

<Gruppennummer>-SchiffeA3.1

- Beachten Sie bei der Bearbeitung der Aufgabe in Eclipse die Informationen in den folgenden Dateien:
 - Rahmenbedingungen.pdf
 - Java-Style-Guide.pdf
 - Projekt-Importieren-Exportieren-Umbenennen.pdf
- Es ist sinnvoll, das Programm nach jeder Änderung auf seine Funktionalität zu testen!
- Nur korrekt benannte, als zip-Archiv-Datei exportierte und mit openjdk 11 erstellte Projekte werden bewertet.

- Debug-Textausgaben, z. B. mit `System.out.println()` sind vor der Abgabe zu entfernen oder auszukommentieren.
- In Moodle finden Sie pro Aufgabenblatt eine Datei `Fragen.txt`. Kopieren Sie diese in den src-Ordner des aktuellen Projektstands. Sie enthält Fragen zu den Aufgabenteilen. Diese sind von Ihnen zusätzlich zu beantworten.

Lehrziele

- Die Klasse `HitMissRenderer` wird erstellt.
- Die gegnerischen Schiffe werden damit versteckt.
- Die Koordinaten des mit der Maus angeklickten Feldes werden berechnet.
- Die für Schiffstreffer erzielten Punkte von Spieler und Computergegner werden vergeben.
- Das Spielende wird implementiert.
- Das Java-Style-Guide ist beachtet worden.
- Die Dokumentation des aktuellen Projektes mit Javadoc ist erstellt.
- Die Lösung wurde als gezipptes Projekt mit dem Namen

`<Gruppennummer>-SchiffeA3<.x>.zip`

exportiert und in Moodle korrekt eingestellt. Zur laufenden Nummer einer Sicherheitskopie `<.x>` siehe **Bearbeitungshinweise** auf Seite 1 der Aufgabenstellung.

Aufgabe 3.1: Schiffe im Spielfeld des Gegners verstecken

Aktuell zeigt das Spiel auch die Schiffe des Gegners an. Auch dies soll geändert werden.

- a) Erstellen Sie im Paket `de.uniluebeck.itm.schiffeversenken.game` eine Klasse `HitMissRenderer`, welche von `GameFieldRenderer` erbt und die Funktion `protected Tile getTileAt(...)` überschreibt. Dem Konstruktor wird ein `GameField`-Objekt übergeben. Die Klasse `GameField` muss also importiert werden.
- b) Damit ein Schiff nicht mehr zu sehen ist, soll die Methode `protected Tile getTileAt(...)` nun, falls ein Schiff gezeichnet werden müsste, nur das `water` Tile bzw. das `water.hiddenshiphit` Tile rendern. Für den Fall, dass es sich um eine Stelle handelt, die bereits bombardiertes Wasser enthält, soll weiterhin das `waterHit` Tile zurückgegeben werden.
Hinweis: Greifen Sie auf die Matrix über die `get()`-Methode der Oberklasse zu, da die Variable dort `private` deklariert ist.
- c) Tauschen Sie im Anschluss im Konstruktor der Klasse `GameView` den `Renderer` für das gegnerische Feld durch ein Objekt der neuen Klasse `HitMissRenderer` aus. Danach dürfen die Schiffe im gegnerischen Spielfeld nicht mehr zu sehen sein. Testen Sie dies, bevor Sie mit Aufgabe 3.2 weiterarbeiten.

Aufgabe 3.2: Punkte für Schiffstreffer vergeben

Machen Sie sich nun mit der Klasse `GameController` vertraut.

Controller dienen dazu, auf Aktionen in der grafischen Oberfläche zu reagieren. Die Klasse `GameController` behandelt die Reaktion auf einen Mausklick in das gegnerische Spielfeld. Sie erbt von der abstrakten Klasse `Controller` (siehe Programming tutorial) und überschreibt deren Methode `clickedAt()`, der die aktuelle Mausposition übergeben wird.

Aus der Mausposition wird die Mausposition bezogen auf das gegnerische Spielfeld (`positionOnOpponentsField`) berechnet und die Dimension (Breite/Höhe) dieses Feldes ermittelt. Mit diesen Werten wird überprüft, ob überhaupt in das richtige Feld geklickt wurde. Die Variable `res` gibt die Größe der Tiles (Kästchen) auf dem Spielfeld an.

Mit den Werten für `tileX` und `tileY` wird auf das angeklickte Tile (Kästchen) zugegriffen. Liefert die Methode `bombard()` aus der Klasse `FieldTile` `true`, so ist ein Schiffsteil getroffen worden und der Spieler erhält Punkte. Sinkt das Schiff nach dem Treffer, wird überprüft, ob das Spiel bereits zu Ende ist, der Spieler gewonnen hat. Gibt es keinen Treffer, so kommt der Computerspieler an die Reihe, und zwar so lange, bis er nicht mehr trifft. Auch er erhält Punkte und es wird überprüft, ob er bereits gewonnen hat, also alle Schiffe des Spielers versenkt hat.

- a) Ihre Aufgabe ist es, die Methode `clickedAt()` zu vervollständigen. Aus der bereits berechneten Mausposition `positionOnOpponentsField` und der Variablen `res` sind zunächst die Matrixkoordinaten (Spalte, Zeile) `tileX` und `tileY` zu ermitteln.

Tipp: Lassen Sie sich dazu zusätzlich die Werte von `positionOnOpponentsField` in der Konsole anzeigen und fertigen Sie sich eine Skizze des Feldes mit dieser Mausposition an. Die Koordinatenwerte `tileX` und `tileY` werden schon zur Kontrolle in der Konsole angezeigt.

- b) Wenn der Spieler etwas getroffen hat, soll er Punkte bekommen. Nutzen Sie hierzu `POINTS_FOR_HIT` aus der Klasse `Constants`. Zusätzlich sollen, wenn das getroffene Schiff gesunken ist, weitere Punkte (`POINTS_FOR_SHIP_SUNK` ebenfalls aus der Klasse `Constants`) hinzukommen und es muss die Methode `handlePossibleGameEnd()` aufgerufen werden.

Hinweis: Verwenden Sie die Methode `addPlayerPoints()` aus der Klasse `GameModel` und beachten die erste Anweisung in der Methode `clickedAt()`.

- c) Ergänzen Sie in der Methode `rewardAgentForDestroyingPlayer()` das Verhalten für den Computergegner, wenn dieser ein Schiff vollständig getroffen hat.

Hinweis: Verwenden Sie hier die Methode `addAiPoints()`.

- d) Implementieren Sie nun zu guter Letzt die private Methode `handlePossibleGameEnd()`. Gehen Sie hierfür wie folgt vor:

- i) Um ein mögliches Spielende festzustellen, müssen Sie über sämtliche Schiffe des Spielers und des Computergegners iterieren. Nutzen Sie dazu die Methode `getCopyOfShipListAsArray()` aus der Klasse `GameField`, die Ihnen eine Kopie der Schiffsliste liefert.
- ii) Legen Sie eine neue private Hilfs-Methode an, der Sie ein Schiffsarray übergeben und die `true` liefert, wenn alle Schiffe gesunken sind.
- iii) Sollten sich keine schwimmenden Schiffe mehr auf dem betreffenden Spielfeld befinden, muss die Methode `endGame(boolean playerWon)` aufgerufen werden. Der Parameter `playerWon` muss dabei `true` sein, wenn der Spieler gewinnt, `false`, falls der Computer siegreich ist.

Hinweis:

Mit Hilfe von `this.getModelInstance().getHumanPlayerField()` und `this.getModelInstance().getComputerPlayerField()` bekommen Sie außerhalb der Methode `clickedAt()` Zugriff auf die entsprechenden Feldinstanzen.

Tipp: Optional können Sie mit der `AtomicBoolean` Klasse von Java arbeiten.

Aufgabe 3.3: Java-Dokumentation

- a) Alle von Ihnen ergänzten Klassen und Methoden sind mit einem JavaDoc-Kommentar zu versehen.
- b) Die Kopfdokumentationen der von Ihnen veränderten Klassen sind anzupassen.
- c) Zeilenkommentare sind geeignet einzufügen.
- d) Erstellen Sie für den aktuellen Projektstand die komplette Javadoc-Dokumentation.

Aufgabe 3.4: Packen und Hochladen

Exportieren Sie Ihr Projekt als zip-Datei <Gruppennummer>-SchiffeA3<.x>.zip und laden diese ins Moodle hoch. Sie **bestätigen die Erklärung zur Eigenständigkeit**. Zur laufenden Nummer einer Sicherheitskopie <.x> siehe Abschnitt **Bearbeitungshinweise**.

Die Abgabe ist bis Sonntag, 16.01.2022 um 23:30 Uhr durchzuführen.