



---

## 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 4

Übungsblatt vom: Sonntag, 16.01.2022

Abgabe der Übung: Sonntag, 30.01.2022 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>-SchiffeA4<.x>. So entstehen Projektnamen wie <Gruppennummer>-SchiffeA4.1, 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>-SchiffeA4.1

- 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

- Das Tile (Kästchen), auf dem der Mauszeiger positioniert ist, wird farblich als Zielhilfe hervorgehoben.
- Die Zielhilfe wird durch einen Pfeil ergänzt.
- Runden und Punkte werden mit 7-Segment-Tiles angezeigt.
- Das Java-Style-Guide ist beachtet worden.
- Die Dokumentation des aktuellen Projektes mit Javadoc ist erstellt.
- Die Lösung wurde als gezipptes Eclipse-Projekt mit dem Namen `<Gruppennummer>-SchiffeA4<.x>.zip` exportiert und in Moodle korrekt eingestellt. Zur laufenden Nummer einer Sicherheitskopie `<.x>` siehe **Bearbeitungshinweise** auf Seite 1 der Aufgabenstellung.

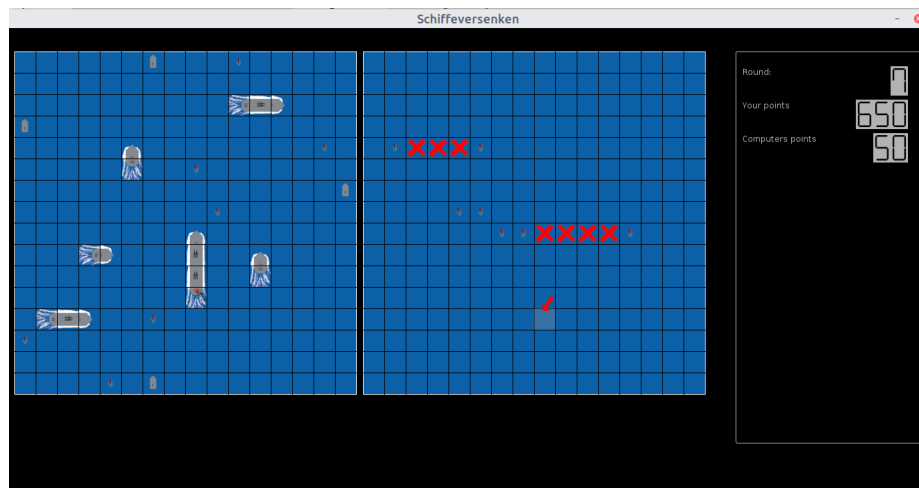


Abbildung 1: Spielanzeige nach Bearbeitung von A4

### Aufgabe 4.1: Zielhilfe

Bauen Sie eine kleine Zielhilfe, die das Feld, über dem sich der Mauszeiger gerade befindet, mit transparenter Farbe darstellt.

- a) Erstellen Sie die neue Methode `renderMouseOver()` in der Klasse `GameFieldRenderer`. Als Parameter sind

- das Canvas `c`,
- `int x` und `int y` als Koordinaten des Mauszeigers,
- `int fieldX` und `int fieldY` als Koordinaten des gegnerischen Spielfeldes

zu übergeben.

- i. Berechnen Sie in der Methode zuerst aus der Spalten- und Zeilenanzahl des gegnerischen Feldes dessen Dimensionen (Breite und Höhe). Die Kästchengröße auf dem Spielfeld können Sie mittels `Constants.TILE_SIZE` abfragen.
- ii. Überprüfen Sie dann, ob sich die Maus über dem gegnerischen Spielfeld befindet und brechen Sie mit `return` ab, falls dem nicht so ist.
- iii. Berechnen Sie als nächstes die Bildschirmkoordinaten des Tiles (Kästchens), über dem sich der Mauszeiger gerade befindet, d.h. die Koordinaten der linken oberen Ecke des Kästchens. Dazu
  - ermitteln Sie zuerst die Feldkoordinaten (Spalte und Zeile) im gegnerischen Spielfeld und berechnen mit diesen die Bildschirmkoordinaten des Tiles.
  - Denken Sie daran, die Bildschirmkoordinaten (`fieldX` und `fieldY`) des gegnerischen Spielfeldes zu berücksichtigen.

**Tipp:** Am besten machen Sie sich dazu eine Skizze mit den verschiedenen Koordinatenangaben.

- iv. Stellen Sie mit `c.setColor()` eine semitransparente Farbe ein (alpha z.B. 0.3) und füllen Sie das betreffende Kästchen auf dem Spielfeld mit `c.fillRect()`. Zu den Methoden siehe Abschnitt **Canvas** im Programming manual.
- b) Implementieren Sie zusätzlich noch einen Zeiger, der auf das transparente Feld zeigt. Zeichnen Sie dazu das Tile "arrow.down" an einer leicht nach rechts oben versetzten Position (siehe Abb. 1).
- c) Zu guter Letzt muss die neue Methode `renderMouseOver()` noch in der `render()`-Methode der Klasse `GameView` für das gegnerische Spielfeld aufgerufen werden:
  - i) Analysieren Sie dazu zunächst die Methode `render()`, um die benötigten Parameterwerte zu ermitteln.
  - ii) Ergänzen Sie dann die neue Methode direkt nach der Anweisung zum Zeichnen des gegnerischen Spielfeldes.

## Aufgabe 4.2: Schwebender Zeiger (Optional)

Optional können Sie einen schwebenden Zeiger implementieren.

- a) Erstellen Sie dazu eine Variable `animationOffset`, deren Wert sich nach folgender Rechenvorschrift ermitteln lässt:  

$$\sin\left(\frac{\text{Systemzeit}}{\text{Geschwindigkeitskonstante}}\right) * \sqrt{Kachelgroesse}.$$
 Als Anfangswert für die Geschwindigkeitskonstante lässt sich 250 empfehlen.
- b) Zeichnen Sie nun das Tile "arrow.down" an der Position.

$$\begin{pmatrix} Feldposition_x \\ Feldposition_y \end{pmatrix} + \frac{Kachelgroesse}{2} * \begin{pmatrix} 1 \\ -1 \end{pmatrix} + animationOffset * \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

### Tipp:

Zur Realisierung von Vektoroperationen lassen sich die Funktionen `multiply()` und `add()` der Klasse `Vec2` nutzen (siehe Programming manual). Die aktuelle Systemzeit können Sie mittels `System.currentTimeMillis()` erfragen. Hilfreiche Mathematische Funktionen befinden sich im Package `Math`.

## Aufgabe 4.3: 7-Segment Anzeige

Die Methode `draw7segNumberAt(Canvas c, int x, int y, int number)` in der Klasse `GameView` macht nicht gerade das, was sie beschreibt, zeigt bei den Runden und den Punkten keine Zahlen im 7seg-Format.

- a) Implementieren Sie die Methode jetzt so, dass sie mit der Hilfe der Tiles "7seg.0" bis "7seg.9" wirklich eine Anzeige im 7seg-Format leistet( siehe Abb. 2). Die benötigten Tiles haben Sie in der Klasse `LoadingScene` bei der Bearbeitung des Aufgabenblattes 2 bereits geladen.

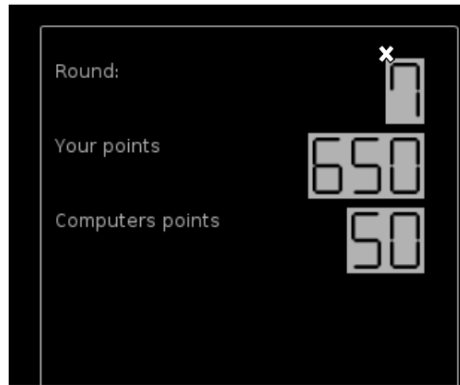


Abbildung 2: Punkt- und Rundenanzeige in 7seg-Format, das weiße Kreuz markiert die Übergabeposition des Punktes (x/y)

- i) Die Parameter der Methode geben die Zeichenfläche (c) und die Position (x, y) der Kachel für die letzte Stelle der darzustellenden Zahl (number) als Bildschirmkoordinaten an.
- ii) Die Breite einer Kachel (Tile) beträgt 26 Pixel, ihre Höhe 44 Pixel.
- iii) Die Tiles für die einzelnen Ziffern sollen an der korrekten Position gerendert werden, so dass die gesamte Zahl zusammengebaut dargestellt wird. Da die Tilezahlen schwarz sind, müssen Sie zunächst ein helles Rechteck zeichnen lassen, auf dem die schwarze Tilezahl dann sichtbar werden kann.

**Tipp:** Um einen passenden Algorithmus zu entwerfen, erinnern Sie sich an das in den Programmier-Übungen besprochene Stellen- (bzw. Positions-)System von Zahlen. Bestimmen Sie die Anzahl Stellen der übergebenen Zahl und bauen diese dann aus den Tiles zusammen.

## Aufgabe 4.4: 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 4.5: Packen und Hochladen

Exportieren Sie Ihr Projekt als zip-Datei `<Gruppennummer>-SchiffeA4<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, 30.01.2022 um 23:30 Uhr durchzuführen.