

ÜBUNGSBLATT 3

Funktionale Fertigstellung des Spiels.

1. Sobald Sie den Spieler fehlerfrei über das Spielfeld bewegen können, verwenden Sie anstelle der Texte („S“ für Spieler, „G“ für Gegner usw.) auf den Labels der Spielfläche nun Bilder (vom Typ `ImageIcon`). Beachten Sie dabei, dass der Spieler insgesamt 8 verschiedene Blickrichtungen hat (`player1.gif` für Blickrichtung links unten, `player2.gif` für Blickrichtung unten usw.).
 - Die Bild-Icons für die Spielfiguren und Dialogfenster finden sie in Moodle.
 - `System.getProperty("user.dir")` gibt auf Unix- und Windows-Systemen den Pfad zurück, in welchem das Programm gestartet wurde. Liegt das Projekt im Home-Verzeichnis, liefert die Methode folglich „H:\mueller\workspace\projektname“.
Auf Windows-Systemen werden Verzeichnisse und Dateien mit einem „\“ voneinander getrennt; bei Unix ist es ein „/“. Um den Dateitrenner auf dem aktuellen System herauszubekommen, bietet das `java.io.File`-Objekt das statische Feld `File.separator`.

Somit kann ein Unterverzeichnis im Projektordner mit den Icon Bildern angelegt werden, das mit obiger Vorgehensweise leicht zu finden ist.
 - Wenn Sie vorhaben Ihr Spiel als ausführbare JAR-Datei zu kompilieren, so müssen Sie die Bilder stattdessen über den `ClassLoader` laden (siehe dazu *Kapitel Java Web Start im Moodle-Kurs Grafische Bedienoberflächen*)
2. Ersetzen Sie den Text der Navigator-Tasten durch Bilder. Die Icons für den Navigator finden Sie in Moodle. Überführen Sie außerdem die Form des Navigators von einem Rechteck in ein regelmäßiges Achteck (siehe Bild 3.1)
 - Verwenden Sie für letzteres die Methode `setShape()` der Klasse `JWindow`.
 - `setShape()` erwartet als Argument ein Objekt vom Typ `Shape`. Entweder Sie definieren ein solches `Shape` selbst (siehe im Skript das Kapitel: „JFrame Eigenschaften“) oder Sie bedienen sich einer in der Swing-API vorhandenen Klasse, welche Ihnen ein solches Objekt liefert, so z.B. die Klasse `Polygon`.
 - Der Button nimmt in der Regel nicht die exakte Größe der ihm übergebenen Elemente an (in diesem Fall die Größe des Icons), sondern es wird zwischen den platzierten Elementen und dem Rand des Buttons ein gewisser Abstand eingefügt. Dieser Abstand ist hier unerwünscht. Sie beseitigen ihn mit dem Befehl `setMargin(new Insets(0, 0, 0, 0))`.



Bild 4.1: Navigator mit Shape und Symbolen

3. Reagieren Sie nun auf die `GameStatusEvent`-Ereignisse des `DionaRapModels`. Erzeugen Sie ein Dialogfenster für die Anzeige der Spielergebnisse („Game Over“ und „Gewonnen“). Das Dialogfenster hat die Buttons „Neues Spiel“ und „Abbrechen“.

Die Dialogfenster sind durch Aufruf von Methoden der Klasse `JOptionPane` zu erstellen:

- „Game Over“ (mit Icon `gameover.gif`) wird angezeigt, wenn das Spiel verloren wurde (via `isGameOver()`-Methode des Eventobjekts).
 - „Gewonnen“ (mit Icon `gewonnen.gif`) erscheint, wenn keine Gegner mehr auf dem Spielfeld vorhanden sind (`isGameWon()`-Methode).
 - Ein Klick auf den Button „Neues Spiel“ startet eine neue Spielrunde. Das Klicken auf den Button „Abbrechen“ hat dagegen nur zur Folge, dass das Dialogfenster sich schließt.
4. Fügen Sie dem Hauptfenster eine Toolbar hinzu, welche folgende Elemente enthält (siehe Bild 4.2):
- Button „Neues Spiel“: Startet eine neue Spielrunde. Der Button ist nur dann aktiviert, wenn das Spiel gewonnen oder verloren wurde.
 - Anzeige des Punktestandes: Zeigt den aktuellen Punktestand des Spielers an. Das Textfeld soll durch den Benutzer **nicht editierbar** sein.
 - Munitionsanzeige: Soll später den Munitionsvorrat des Spielers als eine Reihe von Munitionsbildern anzeigen. Munition gibt es erst in der Multithreading-Variante. Daher bleibt dieser Bereich vorerst einmal leer.
 - Anzeige des Spielfortschritts: Ein Fortschrittsbalken, der in Prozent angibt, wie viele Gegner bereits besiegt wurden. Wenn also das Spiel gewonnen ist, beträgt der Fortschritt 100%.
 - Button „Settings“: Im Moment ohne Funktion.

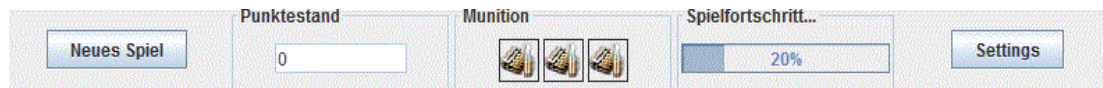


Bild 4.2: Toolbar

Die Panels Punktestand, Munition und Spielfortschritt besitzen einen Rand und eine Überschrift.