

FX-Umsetzung der Spielebibliothek

Sven Eric Panitz
Hochschule Rhein-Main

In diesem Papier wird die Umsetzung der Spielerahmenbibliothek in Javas FX-Bibliothek beschrieben

1 Einbinden der Bibliothek

Es gibt bisher keine kluge und automatische Einbindung der Bibliothek in die verschiedenen Realisierungen. Die Quellen müssen manuell zusammen in ein Projekt kopiert werden.

Wir gehen davon aus, dass ein Spiel in einer Entwicklungsumgebung (eclipse, IntelliJ) entwickelt werden soll. So wird der einfachste Weg sein, in der Entwicklungsumgebung Projekt anzulegen. Dann die Quelltexte der Rahmenbibliothek in das Projekt zu kopieren. Dann die Quelltexte der JavaFX-Realisierung in das Projekt zu kopieren. Bild- und Klang-Dateien können in eclipse in den Quelltextordner `src` kopiert werden. Zum Testen, ob das Projekt korrekt aufgesetzt wurde, empfiehlt es sich, das in der Bibliothek enthaltene Spiel `SimpleGame` zu kompilieren und zu starten.

Als `SoundTool` in der JavaFX Umsetzung wird die Klasse für Klänge aus der Swing-Umsetzung verwendet. Daher ist die Klasse

`name.panitz.game.framework.swing.JavaSoundTool` aus der Swing-Umsetzung auch zu den Quelltexten zu kopieren.

Es spricht nichts dagegen, in dem Projekt nicht nur die Bibliothek der FX-Realisierung sondern auch die Klassen der Swing-Realisierung zu kopieren und in dem Projekt sowohl das Spiel für swing als auch für javafx zu kompilieren und zu starten.

Um ein Spiel in der FX-Umsetzung zu starten, ist eine Unterklasse der folgenden Klasse mit dem korrekten super-Aufruf für das eigene Spiel zu implementieren:

`name.panitz.game.framework.fx.GameApplication.`

2 Die FX-Umsetzung

2.1 Zeichnen der Komponenten

```

1 package name.panitz.game.framework.fx;
2
3 import name.panitz.game.framework.GraphicsTool;
4 import name.panitz.game.framework.GameObject;
5 import javafx.scene.canvas.GraphicsContext;
6 import javafx.scene.image.Image;
7 import javafx.scene.text.Font;
8
9 public class FXContextTool implements GraphicsTool<Image>{
10     GraphicsContext gc;

```

```
11
12 public FXContextTool(GraphicsContext gc) {
13     this.gc = gc;
14 }
15
16 @Override
17 public void drawImage(Image img, double x, double y) {
18     gc.drawImage(img, x, y);
19 }
20
21 @Override
22 public void drawRect(double x, double y, double w, double h) {
23     gc.setLineWidth(1);
24     gc.strokeRect(x, y, w, h);
25 }
26
27 @Override
28 public void fillRect(double x, double y, double w, double h) {
29     gc.fillRect(x, y, w, h);
30 }
31
32 @Override
33 public void drawOval(double x, double y, double w, double h) {
34     gc.setLineWidth(2);
35     gc.strokeOval(x, y, w, h);
36 }
37
38 @Override
39 public void fillOval(double x, double y, double w, double h) {
40     gc.fillOval(x, y, w, h);
41 }
42
43 @Override
44 public void drawLine(double x1, double y1, double x2, double y2) {
45     gc.strokeLine(x1, y1, x2, y2);
46 }
47
48
49 @Override
50 public void drawString(double x, double y, int fontSize, String
    fontName, String text){
51     gc.setFont(new Font(fontName, fontSize));
52     gc.fillText(text, x, y);
53 }
54
55 @Override
56 public Image generateImage(String name, GameObject<Image> go){
57     Image image = new Image(getClass().getClassLoader().
    getResourceAsStream(name));
58     go.setWidth(image.getWidth());
59     go.setHeight(image.getHeight());
60     return image;
61 }
```

```

61
62 @Override
63 public void setColor(double r, double g, double b) {
64     gc.setFill(new javafx.scene.paint.Color(r, g, b, 1));
65     gc.setStroke(new javafx.scene.paint.Color(r, g, b, 1));
66 }
67 }

```

Listing 1: src/name/panitz/game/framework/fx/FXContextTool.java

2.2 Das Spielfeld

```

1 package name.panitz.game.framework.fx;
2
3 import javax.sound.sampled.AudioInputStream;
4
5 import javafx.animation.AnimationTimer;
6 import javafx.scene.canvas.Canvas;
7 import javafx.scene.canvas.GraphicsContext;
8 import javafx.scene.image.Image;
9 import name.panitz.game.framework.GameLogic;
10 import name.panitz.game.framework.GraphicsTool;
11 import name.panitz.game.framework.KeyCode;
12 import name.panitz.game.framework.SoundObject;
13 import name.panitz.game.framework.swing.JavaSoundTool;
14
15 public class FXScreen extends Canvas {
16     GameLogic<Image, AudioInputStream> logic;
17
18     private JavaSoundTool soundTool = new JavaSoundTool();
19
20     @SuppressWarnings("deprecation")
21     public FXScreen(GameLogic<Image, AudioInputStream> gl) {
22         this.logic = gl;
23         this.setWidth(logic.getWidth());
24         this.setHeight(logic.getHeight());
25
26         setFocusTraversable(true);
27
28         GraphicsContext gc = this.getGraphicsContext2D();
29         GraphicsTool<Image> gct = new FXContextTool(gc);
30         logic.paintTo(gct);
31
32         setOnKeyPressed((ev) -> [
33             logic.keyPressedReaction(KeyCode.fromCode(ev.getCode()).getCode())
34         ]);
35         ev.consume();
36     ];
37
38     setOnKeyReleased((ev) -> [

```

```

38         logic.keyReleasedReaction(KeyCode.fromCode(ev.getCode()).getCode
39         ());
40         ev.consume();
41     });
42     timer.start();
43 }
44
45 AnimationTimer timer = new AnimationTimer() {
46     private long lastUpdate = 0;
47
48     @Override
49     public void handle(long now) {
50         setFocused(true);
51         logic.move();
52         logic.doChecks();
53
54         if (now - lastUpdate >= 28_000_000) {
55             GraphicsContext gc = getGraphicsContext2D();
56
57             gc.clearRect(0, 0, getWidth(), getHeight());
58             logic.paintTo(new FXContextTool(gc));
59             lastUpdate = now;
60         }
61
62         for (SoundObject<AudioInputStream> so : logic.getSoundsToPlayOnce
63         ()){
64             so.playSound(soundTool);
65         }
66         logic.getSoundsToPlayOnce().clear();
67     }
68 };
69
70 }

```

Listing 2: src/name/panitz/game/framework/fx/FXScreen.java

2.3 Die Einstiegsklasse

```

1 package name.panitz.game.framework.fx;
2 import javax.sound.sampled.AudioInputStream;
3
4 import javafx.application.Application;
5 import javafx.scene.Scene;
6 import javafx.scene.canvas.Canvas;
7 import javafx.scene.image.Image;
8 import javafx.scene.layout.BorderPane;
9 import javafx.scene.layout.FlowPane;
10 import javafx.stage.Stage;

```

```
11 import name.panitz.game.framework.Button;
12 import name.panitz.game.framework.GameLogic;
13
14 public class GameApplication extends Application {
15     Canvas canvas;
16     GameLogic<Image,AudioInputStream> game;
17     public GameApplication(GameLogic<Image,AudioInputStream> game) {
18         canvas = new FXScreen(game);
19         this.game = game;
20     }
21
22     public void start(Stage stage) throws Exception {
23         canvas.setWidth(game.getWidth());
24         canvas.setHeight(game.getHeight());
25         BorderPane bp = new BorderPane();
26         bp.setCenter(canvas);
27         FlowPane fp = new FlowPane();
28         for (Button b : game.getButtons()) {
29             javafx.scene.control.Button fxbutton = new javafx.scene.control.
30             Button(b.name);
31             fxbutton.setOnAction((ev) -> b.action.run());
32             fp.getChildren().add(fxbutton);
33         }
34         bp.setBottom(fp);
35         stage.setScene(new Scene(bp));
36         stage.show();
37     }
38 }
```

Listing 3: src/name/panitz/game/framework/fx/GameApplication.java