**Arbeitsbericht Projektarbeit PC-Spiel Flappy Bird**



Autor: Mentor:

Simon Balcke Herr Röhrig

**Inhaltsverzeichnis**

Index Titel Seite

1. Deckblatt 1
2. Inhaltsverzeichnis 2
3. Protokoll 13.06.2022 3
4. Protokoll 14.06.2022 4/5
5. Protokoll 15.06.2022 6
6. Protokoll 20.06.2022 7
7. Protokoll 27.06.2022 8
8. Protokoll 28.06.2022 9
9. Protokoll 01.07.2022 10

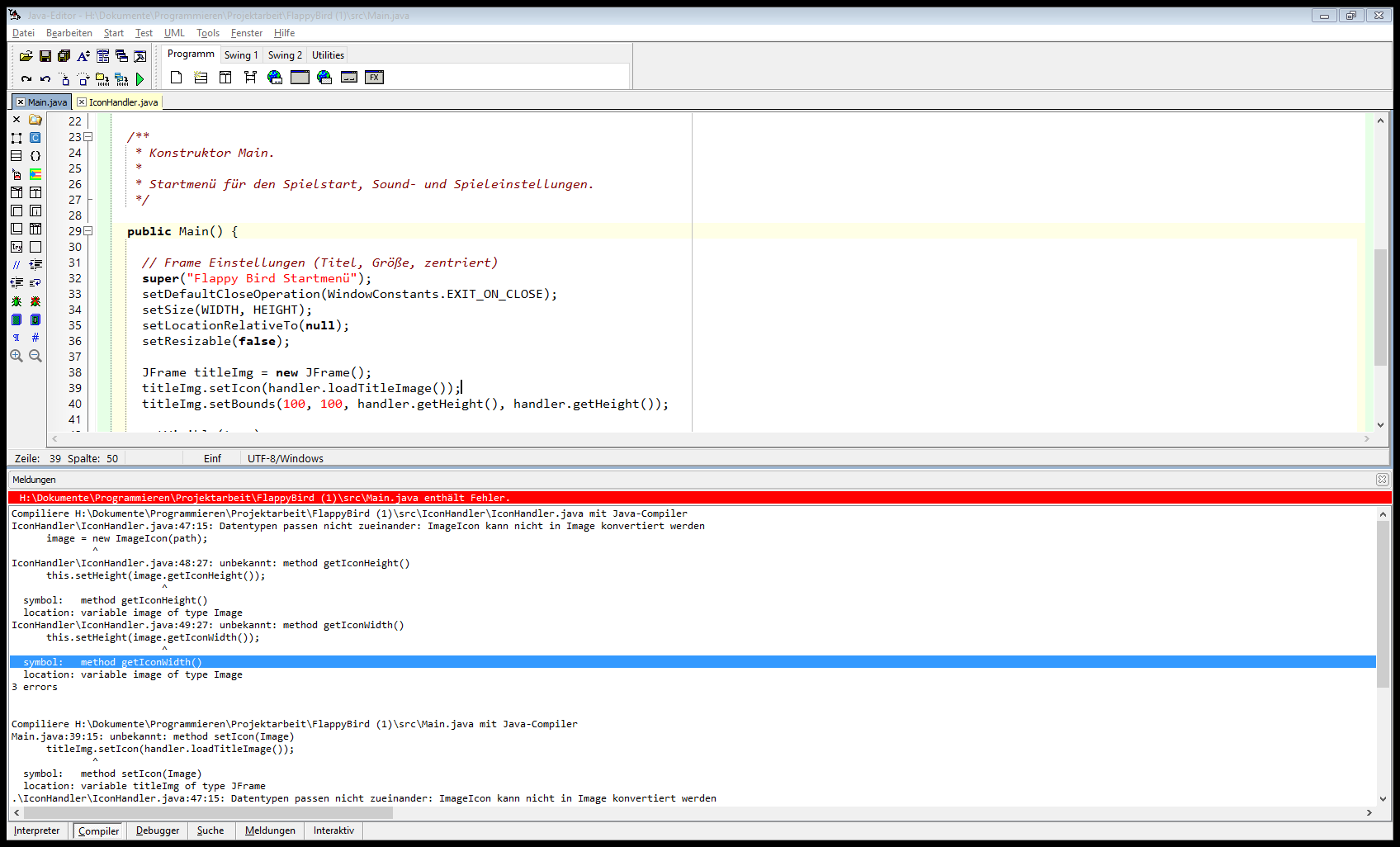
**Protokoll 13.06.2022**

Am 13.06.2022 wurde dem Projekt eine Klasse „IconHandler“ hinzugefügt, mit dessen Hilfe verschiedene Images von dem resources Ordner geladen und im kompletten Projekt auf beispielsweise JLabels gesetzt werden können.

Hierbei traten zunächst NullPointerExceptions auf, welche einer falschen Anordnung des resources Ordners zu Grunde lagen.

Bei der weiteren Implementierung trat ein Compile-Fehler auf, welcher durch einen unpassenden Datentyp zustande kam.

Screenshot:



**Protokoll 14.06.2022**

Am 14.06.2022 wurde das Problem mit der NullPointerException und dem CompileFehler behoben.

Die Images werden nun auf dem JFrame in einem JLabel angezeigt, jedoch an der falschen Position.

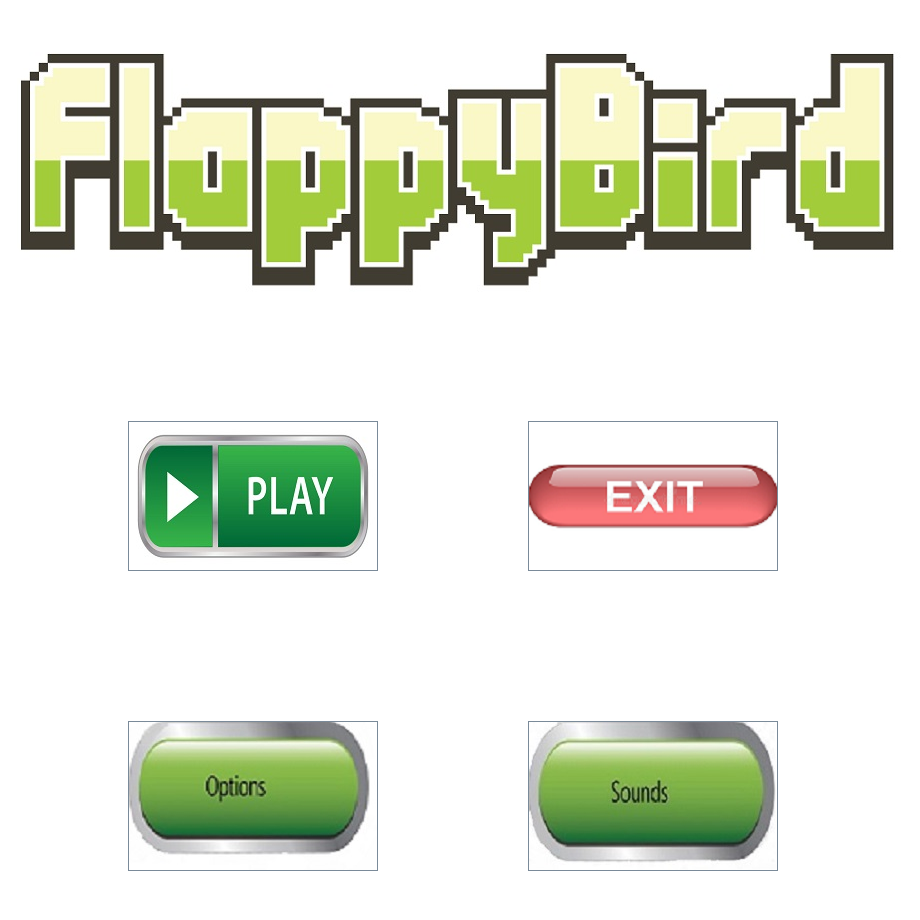
Screenshot JFrame:



Nach der Überarbeitung der Klasse Main wurde das oben genannte Problem durch das Auslassen eines Containers behoben. Stattdessen wurden die JLabels direkt dem JFrame hinzugefügt.

Zudem wurde das Menü durch die Implementation von 4 JButtons erweitert.

Screenshot:



**Protokoll 15.06.2022**

Am 15.06.2022 wurde die Klasse FlappyBird.java erstellt, welche die Funktionalität für das Hauptspiel und die grafische Oberfläche enthält. Zudem wurde die Klasse IconHandler.java um 2 Methoden erweitert, welche 2 neue Images lädt und zurückgibt. Ebenfalls wurde für die Dokumentation Beschreibungen für Methoden, Konstruktoren und die Klasse im Quellcode hinzugefügt.

Screenshot:

Ein Bild, das Text, Screenshot, Elektronik, Anzeige enthält.

Automatisch generierte Beschreibung

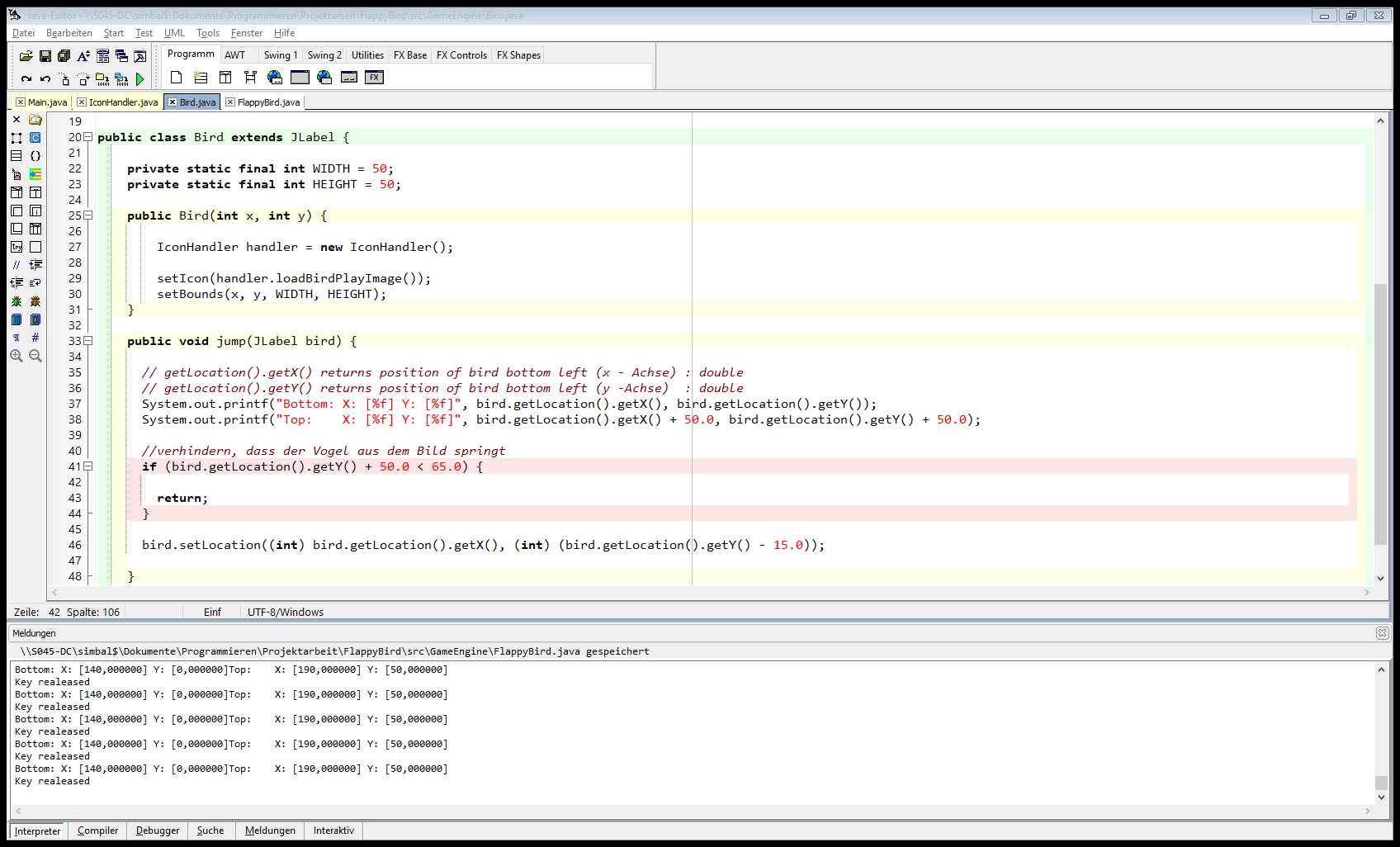
**Protokoll 20.06.2022**

Am 20.06.2022 wurde die Klasse Bird.java um die Methode jump erweitert, welche dafür sorgt, dass der Vogel um eine bestimmte Anzahl an Pixeln entlang der y-achse nach oben verschoben wird.

Zudem wurde verhindert, dass der Vogel den Gültigkeitsbereich überschreitet und somit aus dem JFrame verschwindet. Zur besseren Kontrolle der Ergebnisse wurden die Koordinaten des JLabels (Vogels) nach jedem Keyevent auf der Konsole ausgegeben. Sowohl der unterste linke Punkt, als auch der oberste rechte Punkt. Diese Daten werden nach der Vervollständigung des Projektes entfernt.

Des Weiteren wurde die Klasse FlappyBird.java um einen KeyListener durch die Implementation des Interfaces erweitert. Nach einigen Fehlschlägen wurde das erwartete Ergebnis erreicht.

Screenshot:



**Protokoll 27.06.2022**

Am 27.06.2022 wurde die Klasse Pine.java in die Klasse FlappyBird.java eingebunden. Zudem wurde mittels Multithreading die Verschiebung des Vogels nach unten (Gravitation) und die Verschiebung der Röhren nach links gewährleistet.

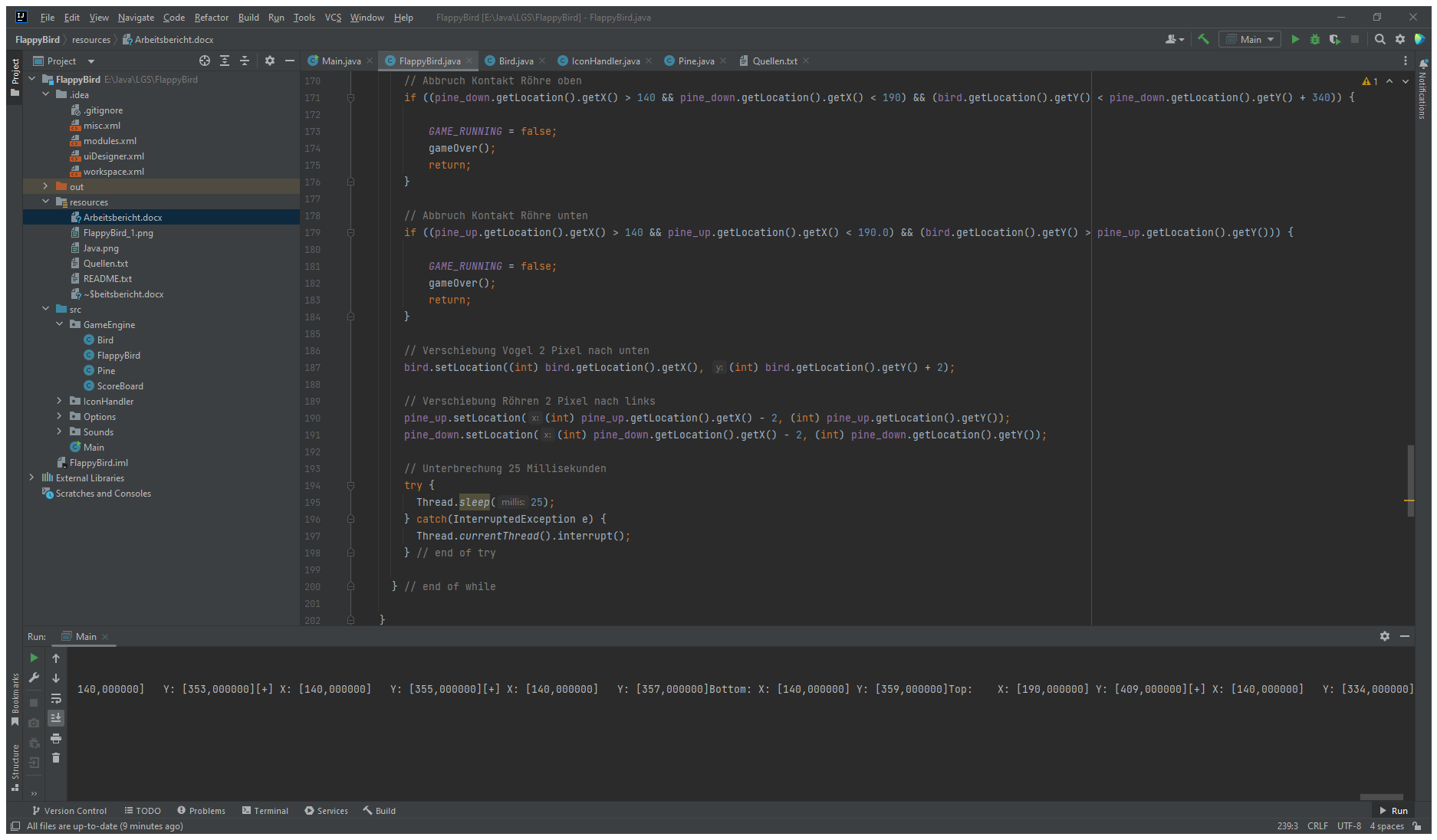
Ebenfalls wurde implementiert, dass das Spiel gestoppt wird, sobald der Vogel eine Röhre oder den Boden berührt. Anschließend wird ein JLabel mit der Aufschrift „Game Over“ angezeigt und die Verschiebung durch den Thread gestoppt.

Für die Dokumentation wurde für jede Methode und jeden Konstruktor eine java documentation verfasst, welche die Verwendung der Methode, dessen Parameter und falls vorhanden den Rückgabewert beschreibt.

Unter anderem wurde der Quellcode durch Refactoring optimiert und besser lesbar gemacht. Aufgrund dessen wurde die Existenz einiger Packages beendet und die eine enthaltene Klasse in das Package „GameEngine“ mit eingebunden.

Damit ein Spielfluss entsteht, wurde außerhalb der angezeigten Oberfläche ein weiteres Röhrenpaar erzeugt, welches in das Bild kommt, sodass immer mindestens ein Röhrenpaar im Bild zu sehen ist.

Screenshot:



**Protokoll 28.06.2022**

Am 28.06.2022 wurde die Funktionalität für die Spieloptionen implementiert und in das Hauptmenü mit eingebunden.

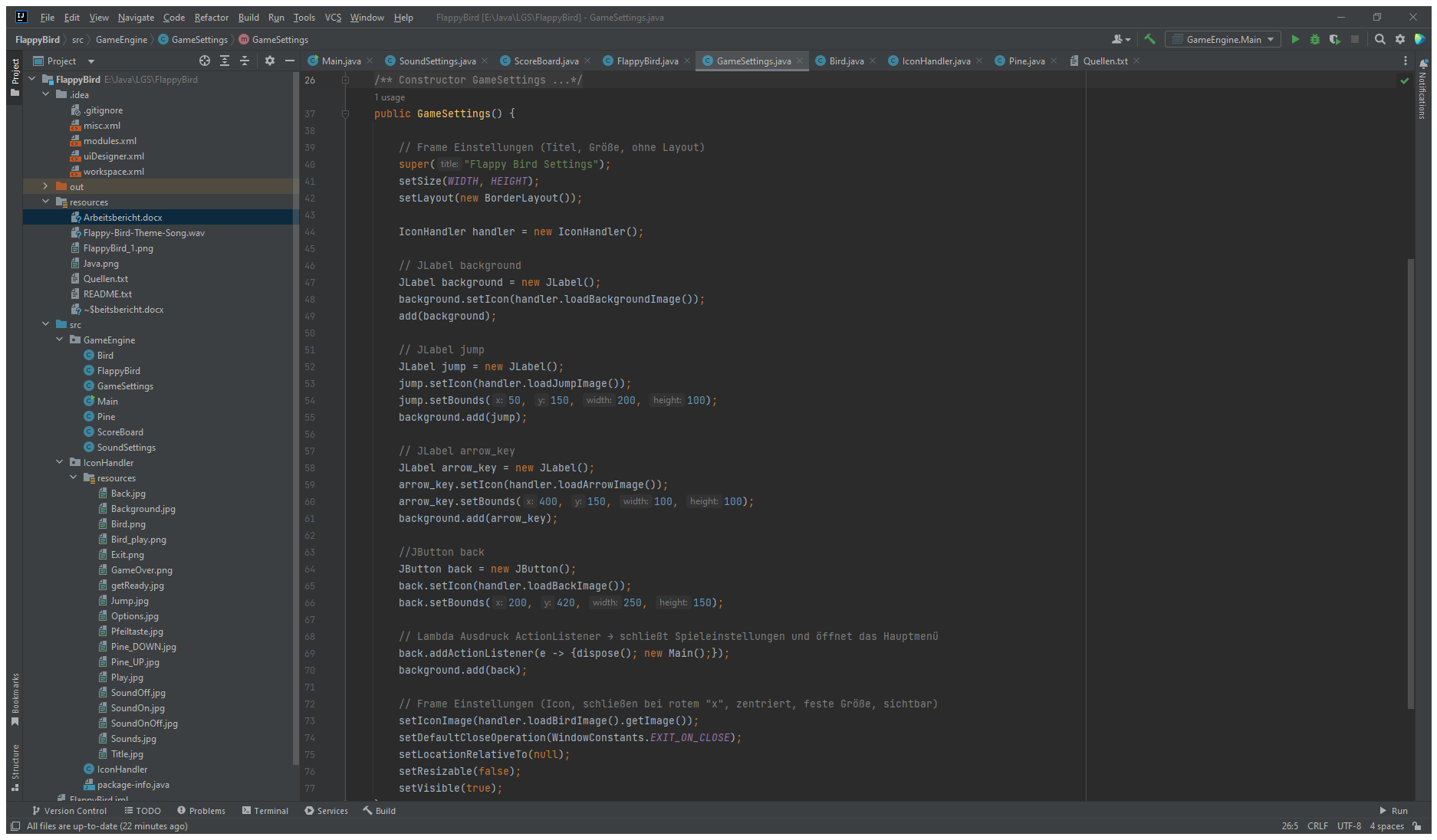
Hierzu wurde lediglich ein neues JFrame mit dem Flappy Bird Hintergrund erzeugt und die Knopfbelegung (In diesem Falle die Pfeiltaste nach oben) in einem JLabel dargestellt.

Mit einem „Back“ Knopf ist es möglich zum Hauptmenü zurückzukehren und das Spiel zu starten oder zu beenden.

Für die schönere Darstellung wurden die Bilder mit dem Programm Paint überarbeitet und an den Hintergrund angepasst.

Zudem wurde die Projektstruktur nochmals verändert, sodass nur noch 2 Packages (GameEngine und IconHandler) vorhanden sind. Die Main Klasse, welche die Main-Methode enthält, wurde in das Package GameEngine verschoben. Andernfalls wäre es nicht möglich gewesen eine Instanz der Klasse Main zu erzeugen und zum Hauptmenü zurückzukehren.

Screenshot:



**Protokoll 01.07.2022**

Am 01.07.2022 wurde die Klasse SoundSettings.java geschrieben und die Funktionalität des Abspielens von Hintergrundmusik hinzugefügt. Mit Hilfe eines neuen Threads wird die Musik dauerhaft in einem Loop abgespielt. Zudem enthält die Klasse die Funktionalität eines neuen JFrames für die grafische Oberfläche der Soundeinstellungen. Diese wird mittels Knopfdrucks im Hauptmenü erzeugt und angezeigt.

Die Methode, welche die Musik abspielt, lässt sich ohne Erzeugung einer Instanz aufrufen. Innerhalb dieser Methode werden einige Exceptions abgefangen, um einen Abbruch des kompletten Spieles vorzubeugen.

An diesem Tag wurde das Programmieren abgeschlossen und die Dokumentation verfeinert.

Zu der Dokumentation zählt dieser Arbeitsbericht, das Pflichtenheft, die Methodenbeschreibung und die Spielanleitung.

Dies beendet den Abschnitt der Protokollierung des Arbeitsverlaufes.

Screenshot:

