

Technische Hochschule Deggendorf

Fakultät Angewandte Informatik
(Bachelor-Studiengang Angewandte Informatik)

JAVA Programmierung FWP - 2D Plattformer

Projektarbeit
für das Fach
Javaprogrammierung AI-B6-FWP
SS2022

vorgelegt von:

Robin Prillwitz (00805291), 84326 Falkenberg

am: 30. Juni 2022

Erstprüfer:

Inhaltsverzeichnis

Abkürzungsverzeichnis	i
Glossar	i
1 Allgemeine Übersicht	1
1.1 UML-Diagramm	2
2 Eigene Beiträge	3
2.1 Highscore & Datenbankverbindung	3
2.2 Build-System	3
2.3 Musik & Sound	3
Eidesstattliche Versicherung	4

Abkürzungsverzeichnis

JAR	Java Archive.
JRE	Java Runtime Enviornment.
UML	Unified Modelling Language.

Glossar

Javadoc	Java's standard Dokumentationsmethode.
JavaFX	Java Graphikbibliothek.
Maven	Build System.
MySQL	Relationale Datenbank.

1 Allgemeine Übersicht

Domino's Pizza® dominiert die Dystopie Düsburgs. Kann Pizza Hut® den Ruf der runden Fressalien retten? Begib dich zur Auslieferung an die Front und beweise deine italienischen Wurzeln!

Das Spiel *Pizza Hut 2077* ist ein Sidescrolling Jump'n'Run. Das Spiel besteht aus einer simplen 2-Dimensionalen Welt. Diese ist mit zufällig generierten Plattformen gefüllt. Das konkrete Ziel des Spiels besteht lediglich in dem Highscore. Während des Ablaufes muss der Spieler Gegnern ausweichen und sich von Plattform zu Plattform hangeln. Der Spieler kann diese Gegner durch das Werfen von Pizzen eliminieren. Die Feuerrate der Pizzen ist jedoch limitiert. Das Spiel ist zu Ende, wenn der Spieler von einem Gegner berührt wird oder aus der Welt fällt (bzw. nicht auf einer Plattform landet). Alle dem Genre üblichen Funktionen sind wie zu erwarten realisiert.

Am Ende des Spiels kann der Spieler seinen Highscore mitsamt einem Namen in einer Datenbank speichern. Die *MySQL* Datenbank ist online gehostet, darum wird eine funktionierende Internetverbindung benötigt. In dem Credits-Bildschirm kann, mit dem notwendigen Administrator Passwort, die Highscore Tabelle der Datenbank zurückgesetzt werden. Das Einfügen von neuen Highscores zum Spielende verläuft ohne Passwort. Die Dateneingabe hat jedoch nur Rechte zum Erstellen neuer Daten und ist auch gegen SQL-Injections gehärtet.

Die Hintergrundmusik des Spiels reagiert auf den jeweiligen Zustand. Im Hauptmenü wird eine ruhigere Version des Arpeggio Leitmotivs gespielt. Zum Spielstart geht diese mit über eine Brücke in den Hauptteil über. Der Hauptteil wird während des Spielverlaufs geloopt. Wenn der Spieler stirbt, wird der Hauptteil durch das Outro unterbrochen. Der Soundtrack stammt aus eigener Komposition. Außerdem haben die Aktionen des Spielers zugehörige Soundeffekte, wie beispielsweise beim Springen oder bei dem Werfen von Pizzen, die passend abgespielt werden.

Die Graphik des Spiels beruhen auf *JavaFX*. Die meisten graphischen Elemente und Animationen sind durch individuelle Sprites implementiert. Einfache Animationen wurde durch das Abspielen verschiedener Sprites nacheinander realisiert. Alle Sprites stammen aus eigener Produktion. Interaktive Elemente wie Text und Eingabefelder nutzen jedoch *JavaFX* Widgets.

Das Projekt wird mithilfe von *Maven* kompiliert. Zur Ausführung ist eine funktionierende Instanz davon nicht notwendig, lediglich eine *Java Runtime Environment (JRE)* von Version 18 oder später. Da das Projekt in eine eigenständige *Java Archive (JAR)* mitsamt aller Ressourcen (Sprites und Sounds) und aller notwendigen Abhängigkeiten (*JavaFX* und Freunde) kompiliert werden kann. Außerdem kann durch *Maven* eine vollwertige und interaktive Dokumentation mit *Javadoc* erstellt werden. Die gesamte Codebase ist auch in der *Javadoc* Konvention kommentiert.

Anleitungen zur Kompilation, Steuerung und Tasten und weiteres können in der *README.md* gefunden werden. Attribution für benutzte Ressourcen ist in der *ATTRIBUTION.md* gelistet.

1.1 UML-Diagramm

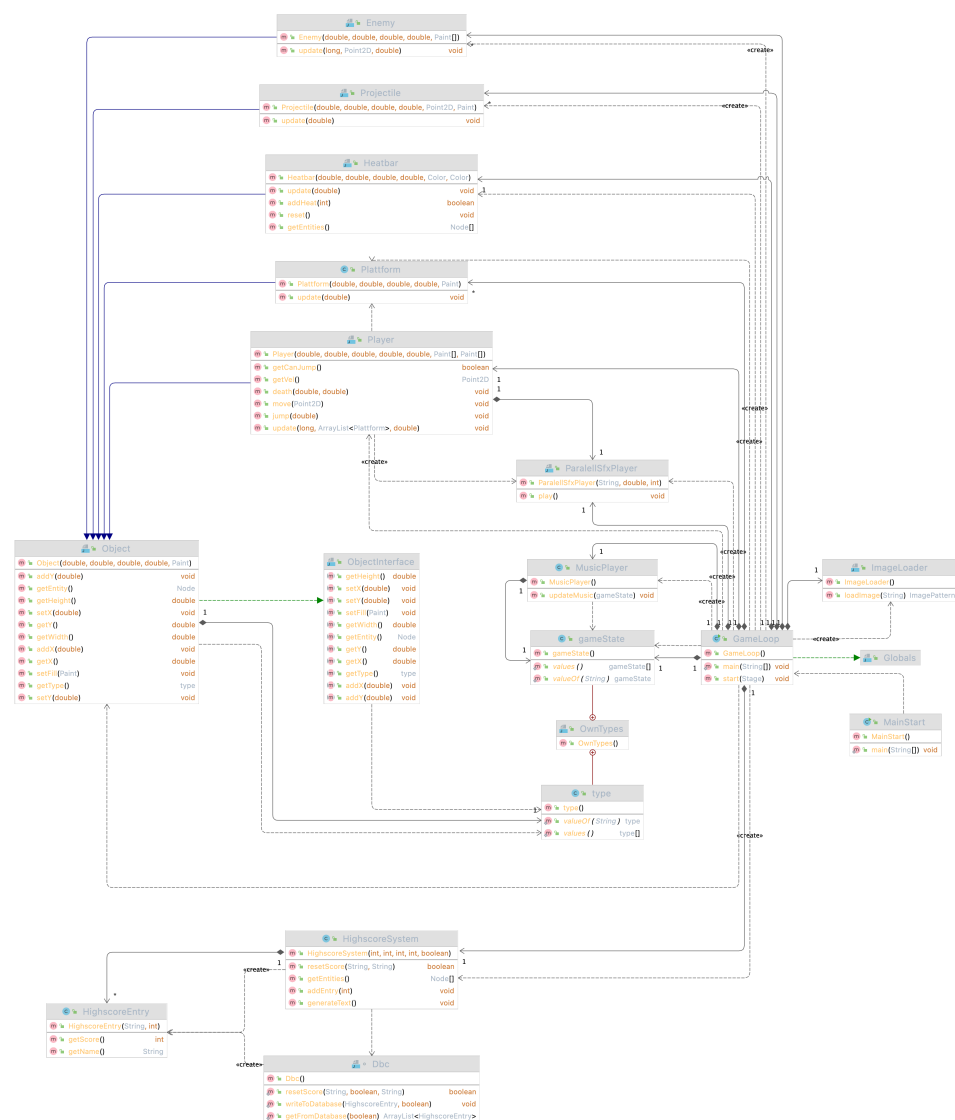


Abbildung 1: *Unified Modelling Language (UML)*-Diagramm des Projekts. Das Bild befindet sich auch im Haupt-Verzeichniss des Projekts.

Das ist Abbildung 1 dargestellte *UML* Diagramm gibt den gesamten Inhalt des implementierten Packets an. Äußere Abhängigkeiten werden nicht aufgeführt. Der Großteil spielt sich in und um der Klasse `GameLoop` ab. Die relevanten Teile der Core-Gameplay-Schleife und weiteres werden darin erstellt und aktualisiert. Alle im Spiel auftretenden Elemente erben von der Klasse `Objekt`, welche das `ObjectInterface` implementiert. Musik wird durch die `MusicPlayer` Klasse gehandhabt und Sound durch den `ParraellSFXPlayer`. Bilder und Graphiken werden durch den `ImageLoader` geladen aber von den Verbrauchern selbst verwaltet. Das Highscore System mitsamt der Datenbank Verbindung wird in `HighscoreSystem`, `HighscoreEntry` und `DBC` (DataBase Connection) implementiert. Die Klasse `MainStart` ist eine helferklasse um die *JavaFX* Applikation zu initialisieren.

2 Eigene Beiträge

Viele Teile des Programms, vor allem Teile der größeren Klassen, wie von GameLoop sind durch Kollaboration entstanden. Kein Teil wurde ausschließlich in Isolation entwickelt. Daher überdecken sich einige Teile möglicherweise.

2.1 Highscore & Datenbankverbindung

2.2 Build-System

2.3 Musik & Sound

Eidesstattliche Versicherung

Ich, Robin Prillwitz, 00805291

(Vorname, Name, Matr.-Nr.)

versichere an Eides Statt durch meine Unterschrift, dass ich die vorstehende Arbeit selbständig und ohne fremde Hilfe angefertigt und alle Stellen, die ich wörtlich oder dem Sinne nach aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe, mich auch keiner anderen als der angegebenen Literatur oder sonstiger Hilfsmittel bedient habe.

Ich versichere an Eides Statt, dass ich die vorgenannten Angaben nach bestem Wissen und Gewissen gemacht habe und dass die Angaben der Wahrheit entsprechen und ich nichts verschwiegen habe.

Die Strafbarkeit einer falschen eidesstattlichen Versicherung ist mir bekannt, namentlich die Strafandrohung gemäß § 156 StGB bis zu drei Jahren Freiheitsstrafe oder Geldstrafe bei vorsätzlicher Begehung der Tat bzw. gemäß § 163 Abs.1 StGB bis zu einem Jahr Freiheitsstrafe oder Geldstrafe bei fahrlässiger Begehung.

84326 Falkenberg, 30. Juni 2022

Ort, Datum

Robin Prillwitz

Unterschrift

Kolophon

Dieses Dokument ist ein $\LaTeX 2_{\epsilon}$ (2022-06-01) Dokument der KOMA-Script Klasse. Alle eigenen Zeichnungen sind mit TikZ gesetzt. Kompiliert wurde es mit Xe \LaTeX und biber 2.17 mithilfe von T \LaTeX Live auf macOS 12.4 am 30. Juni 2022.