

Java-FWP-AI-B6

Robin Prillwitz 00805291

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 entstehen Gegner die den Spieler von allen Seiten berühren. Der Spieler kann diese 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).

Am Ende des Spiels kann der Spieler seinen Highscore mitsamt einem Namen in einer Datenbank speichern. Die "MySQL" Datenbank ist online, darum wird eine funktionierende Internetverbindung benötigt. In dem Credits-Bildschirm kann, mit dem notwendigen Administrator Passwort, die Highscores der remote Datenbank zurückgesetzt werden.

Die Hintergrundmusik des Spiels reagiert auf den jeweiligen Zustand. Im Hauptmenü wird eine ruhigere Version des Arpeggio Leitmotivs gespielt. Zum Spielstart geht diese in den Hauptteil über. Wenn der Spieler stirbt, wird das Outro gespielt. Der Soundtrack stammt aus eigener Komposition. Außerdem haben die Aktionen des Spielers zugehörige Soundeffekte, die passend abgespielt werden.

Die Graphik des Spiels beruhen auf JavaFX. Die meisten graphischen Elemente und Animationen sind durch individuelle Sprites implementiert. 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 JRE von Version 18 oder später.

Da das Projekt in eine eigenständige JAR mitsamt aller Ressourcen (Sprites und Sounds) und aller notwendigen Abhängigkeiten (JavaFX und Freunde) kompiliert werden kann. Außerdem kann durch Maven eine vollwertige Dokumentation mit javadoc erstellt werden.

Weitere Anleitungen zur Kompilation, Steuerung und Tasten und weiteres können in der README.md gefunden werden.

Attribution für benutzte Ressourcen ist in der CONTRIBUTION.md gelistet.

UML Diagramm

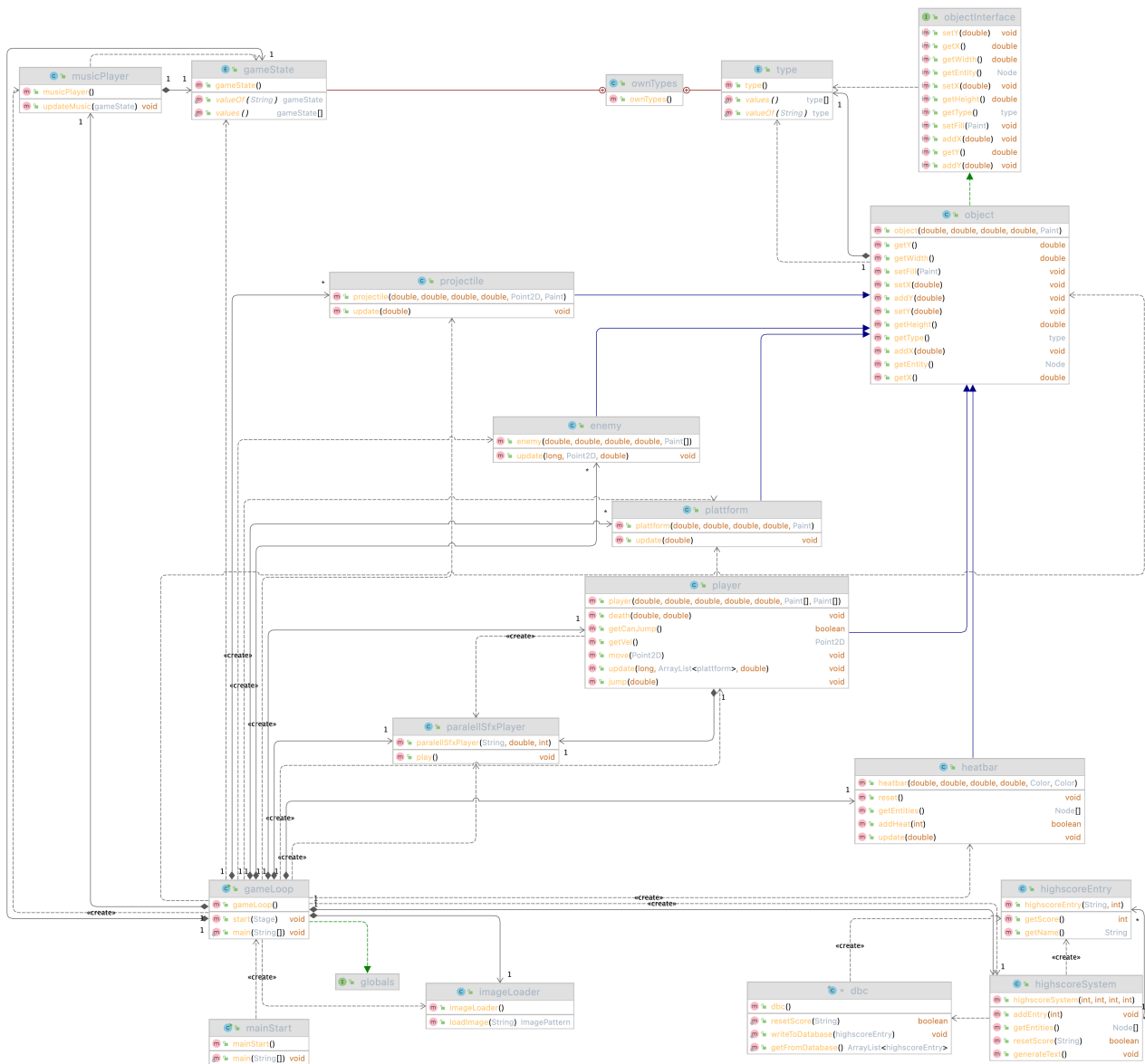


Abbildung 1: UML Diagramm des Projekts. Das Bild befindet sich auch im Haupt-Verzeichniss des Projekts.

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 ParralellSFXPlayer. 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.

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.

Highscore & Datenbankverbindung

Build-System

Musik & Sound