**Software-Entwurf**

**Prüfungsstudienarbeit zur Vorlesung „Java Programmierung“**Prüfer: Prof. Dr. Andreas Berl

**Teilnehmer 1: Michael Mican, 00692390, michael.mican@stud.th-deg.de, AI**

Name, Matrikelnummer, Email, Studiengang

**Teilnehmer 2: Philipp Muhr, 00692629, philipp.muhr@stud.th-deg.de, AI**

Name, Matrikelnummer, Email, Studiengang

**Semester:** SS 2019

**Datum der Abgabe:** 15.05.2019

# ASCIIPinball

**Inhalt:**

* Ein **Mock-Up** mit ausführlicher Erklärung aller Features.
* Ein **grobes Klassendiagramm** mit ausführlicher Erklärung.
* Weitere geplante Ansätze.

**Umfang: ca. 2 Seiten**

1. **Mock-Up**

Ball

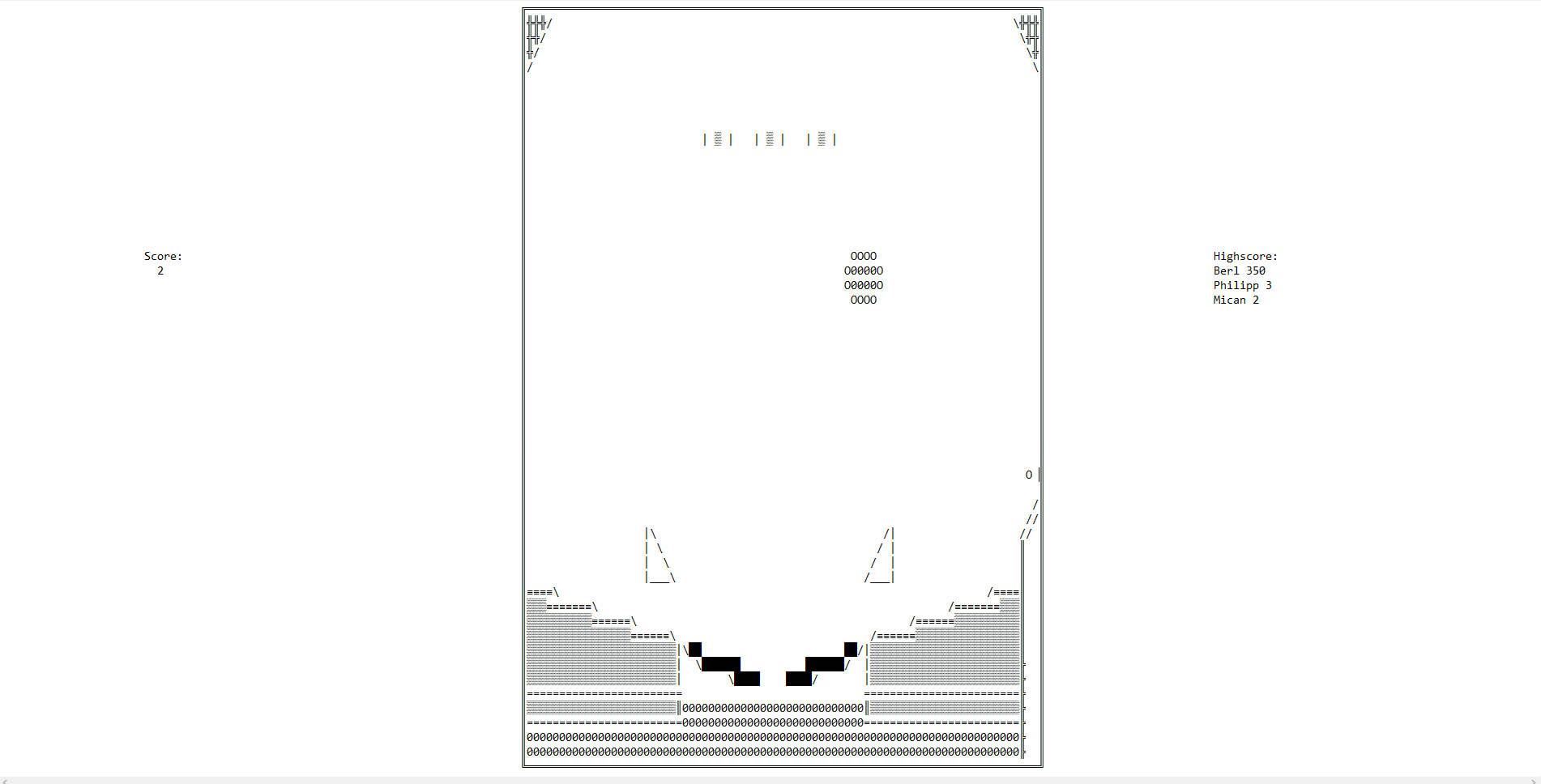
Highscoreliste

Bumper

Point-Door

Aktueller Score des aktuellen Spielers

Klassische Wände



Das Spiel versucht einen einfachen Flippertisch nachzubilden.

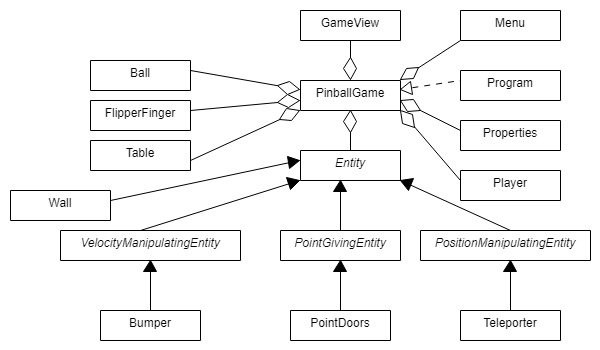
Am Anfang des Spiels hat der Nutzer die Möglichkeit in einem Menü zwischen 1 – 4 Spielern auszuwählen und diesen einen Namen zu geben. Diese wechseln sich der Reihe nach ab, sobald der vorherige Spieler eine seiner drei Runden verloren hat.

Der Ball verfügt über eine Physik und prallt von Objekten ab, außerdem wirkt die Gravitationskraft auf diesen. Es gibt verschiedene Hindernisse, manche versetzen den Ball (z.B. Teleporter-Flächen), andere geben dem Spieler beim „Überrollen“/Kontakt Punkte (z.B. Point-Doors) und wiederum andere stoßen den Ball aktiv ab (z.B. Bumper).

Der Spieler versucht nun den Ball möglichst Punkte bringend über das Spielbrett zu schießen. Hierfür hat er zwei Flipperfinger welche er separat, links und rechts, ansteuern kann.

Im Spiel ist darüber hinaus ein scoreing System enthalten, das den Spieler für das Treffen verschie-dener Objekte (Entities) mit Punkten entlohnt. Diese Punkte entscheiden im Mehrspielermodus am Ende des Spiels über den Sieg. Außerdem werden die besten Scores in einer Highscore-Top-10-Liste mit dem Namen des Spielers verewigt.

1. **Klassendiagramm**



Herzstück des Programms ist die Klasse „PinballGame“ hier werden alle „Einzelteile“ des Spieltisches zusammengeführt. Außerdem sind hier alle aufzurufenden Methoden, welche von der Klasse Programm nur noch in einer while(true) Schleife in der richtigen Reihenfolge zum richtigen Zeitpunkt aufgerufen werden müssen, enthalten. Das PinballGame enthält mindestens einen Ball, welcher seine Bewegung und die wirkende Gravitationskraft verwaltet (nicht die physikalische Kollision! Diese wird von den Entities selbst verwaltet). Die Klasse Table baut die grundlegende Box bzw. das grundlegende Design des Tisches auf und verwaltet die Physik dieser äußeren Wände. Zusätzlich besitzt das Spiel mindestens 2 Flipperfinger. In dieser Klasse werden die Interaktionen des Spielers mit den Flipperfingern, sowie die physikalische Interaktion zwischen diesen und dem Ball gesteuert.

Außerdem besteht das Spiel noch aus verschiedenen Entities. Dies ist eine abstrakte Klasse und gibt Ihren Unterklassen bereits Variablen (wie xpos und ypos), sowie einige Methoden (wie interactWithBall()) für die Unterklassen zur Implementierung vor. Diese Vorgaben werden von den 3 weiteren abstrakten Unterklassen (VelocityManipulatingEntity, PointGivingEntity und PositionManipulatingEntity), je nach Typ der Entities erweitert. Von den abstrakten Klassen erben nun die Klassen, welche die tatsächlichen Hindernisse repräsentieren. Zunächst sind nur Wall (Begrenzende Wände ohne besondere Eigenschaft), Bumper (Hindernis, dass den Ball beschleunigt, wenn er von Ihm abprallt), PointDoors (Scorepunkte gebende Flächen ohne Kollision) und Teleporter (Ball versetzende Objekte) geplant. Dieser Aufbau wurde gewählt, um eine möglichst leichte Erweiterung der vorhandenen Entities zu ermöglichen.

Die Klasse Properties ist eine Klasse mit statischen Variablen, diese wird von der Menu Klasse bei erstmaligen Spielstart gesetzt und dient als Einstellungsdatei (z.B. Einstellung wie viele Spieler spielen gerade am Tisch, oder wie hoch die Gravitation ist).

Wie bereits erwähnt wird die Klasse Menu bei Start des Spieles verwendet. Die Klasse verwaltet eine GUI zur Auswahl der Spieleranzahl.

Die Klasse Player verwaltet die Scores der einzelnen Spieler. Am Ende des Spiels werden die Instanzen dieser Klasse an die Klasse HighScore gegeben, welche sich dann um das Leaderboard kümmert. Außerdem speichert bzw. lädt die Klasse die Top 10 auf der Festplatte bei Spielstart.

1. **Ausblick / In Planung**

Zu Zwecken des einfachen Level Designs ist der Plan eine loadLevel Methode in PinballGame zu implementieren, welche am Anfang des Spiels aus einer auf der Festplatte gespeicherten lvl.txt Datei einen Spieltisch generiert. Somit wird es dem Nutzer möglich seinen eigenen Spieltisch (in einem Texteditor seines Vertrauens) zu entwerfen und diesen anschließend auch zu Spielen. Darüber hinaus sind weitere Entities in Planung, welche potenziell implementiert werden könnten.