Technische Dokumentation

Beat Läufer

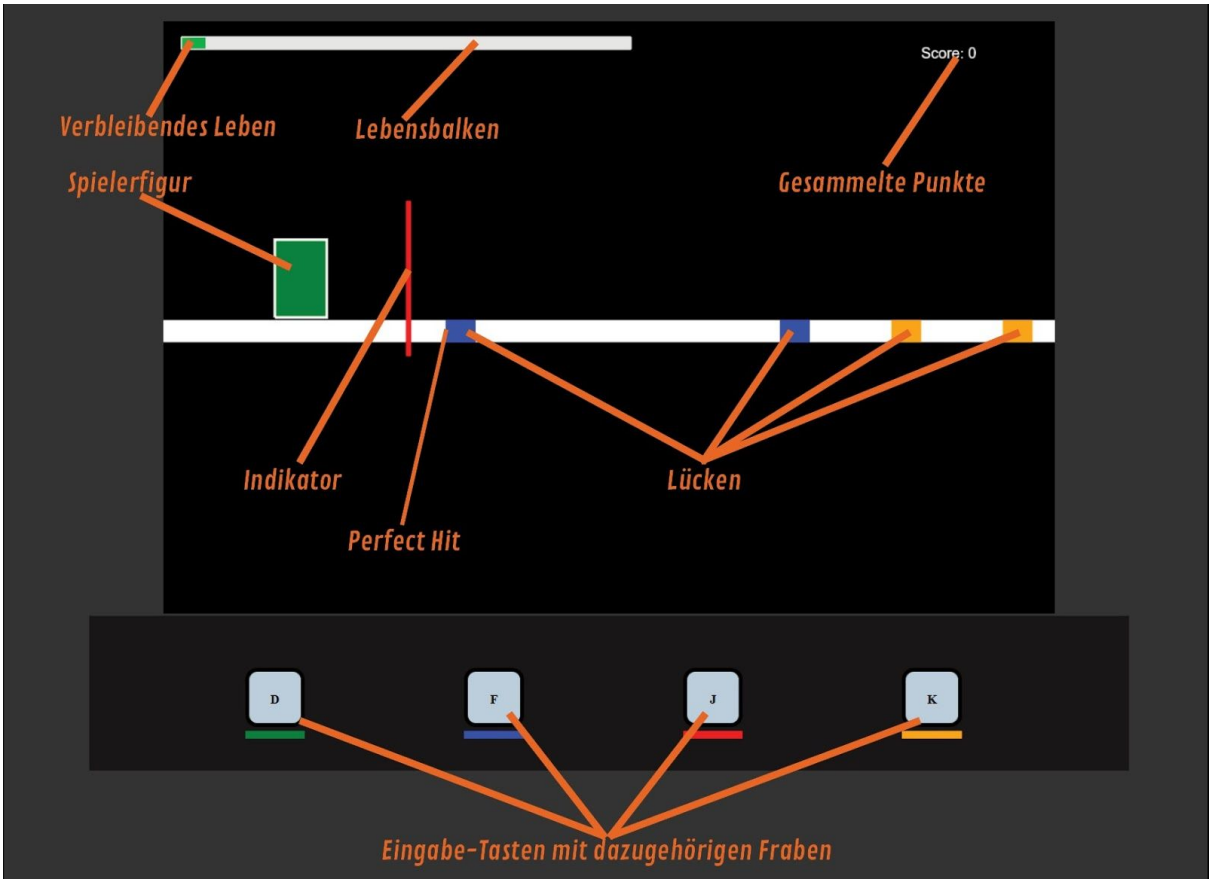
Max Hammer • 2300038
Waldemar Goßmann • 2285410

HAW Hamburg
Fakultät Design, Medien und Information
Media Systems WiSe 2018/2019
Audio-Video-Programmierung
Prof. Dr. Andreas Plaß

Beschreibung

Unser Projekt **“Beat Läufer”** ist ein rhythmus-basiertes Spiel.
Der Spieler produziert Töne und Beats, verbindet diese mit dem Sound der Software und somit entsteht ein vollständiger Song.

Die Spielfigur bewegt sich von links nach rechts über den Bildschirm. Auf der rechten Bildschirmhälfte erscheinen Lücken in unterschiedlichen Farben.
Jede Farbe ist einer Taste zugewiesen und beim Drücken wird ein bestimmter Beat oder Ton erzeugt.
Der Spieler muss nun die passende Taste drücken, wenn der Indikator, der sich vor der Spielfigur befindet, über einer Lücke ist und bevor diese die Spielfigur erreicht.
Wenn die richtige Taste genau am Anfang der Lücke gerückt wird, ist der Spieler genau im Rhythmus des Songs und erreicht somit einen “Perfect Hit”.Je näher der Spieler an diesem “Perfekt Hit” ist, je öfter er also im Rhythmus bleibt, desto mehr Punkte bekommt er. Die Punkte des aktuellen “Treffers” werden ihm angezeigt, sowie auch seine bisher gesammelten Punkte.
Am Ende des Liedes wird Ihm eine Bewertung angezeigt. Diese beinhaltet die Gesamtpunktzahl sowie die Häufigkeit der gesammelten Arten von “Treffern”.
Der Spieler verliert kontinuierlich “Leben” während des Spiels. Mit jedem erfolgreichen “Treffer” füllt er dieses wieder auf. Sollte sein “Leben” auf Null sinken so endet das Spiel. Dies bestraft den Spieler wenn er zu sehr außerhalb des Rhythmus liegt. Sein aktuelles “Leben” wird ihm in einem Lebensbalken angezeigt.



Insgesamt stehen 3 Level und 2 Lieder über das Hauptmenü zur Auswahl. Jedes Level hat eine unterschiedliche Schwierigkeit und zwischen einer und vier verschiedenen Tasten die verwendet werden müssen. Je höher die Schwierigkeit, desto höher der Beat-Per-Minute des Liedes, die Geschwindigkeit, in der die Lücken auf den Spieler zukommen, die Anzahl der Tasten sowie die Lückengröße.

Technische Umsetzung

Das Hauptmenü besteht aus einer einzelnen Seite, welche mit Overlays arbeitet um zum Beispiel ein Optionsfenster anzuzeigen. Die Buttons werden hier beim Hovern mit leichten Transition-Effekten animiert . Entscheidet man sich in der Levelauswahl für ein Lied, wird man zu diesem weitergeleitet und die URL erhält den Namen des gewählten Levels als Anhang (zB. "*game.html?WeWillRockYou*").

Menü-Button Hover Animation

Die URL wird zu Beginn des Levels in der Setup-Funktion überprüft und je nach Inhalt wird eine unterschiedliche Level-Datei geladen. Diese Level-Datei beinhaltet die Audiodateien, die für dieses Level geladen werden sollen, die Positionen und Farben der Lücken, die Angaben zur Schwierigkeit des Levels, sowie die Scroll-Geschwindigkeit und Größe der Lücken.

Nachdem die Level-Daten geladen und die Lücken in einem Array gespeichert wurden, wird der Gameloop gestartet. Der Spieler und die Plattform mit den Lücken werden auf dem Canvas gezeichnet und jeden Frame wiederholt. Die Position der Lücken wird dabei jedes Mal in der CalcFloor-Funktion neu berechnet und angepasst. Die CalcNextGap-Funktion berechnet zusätzlich welche Lücke die Nächste zu treffende Lücke ist und wann es zu spät ist diese noch zu treffen.

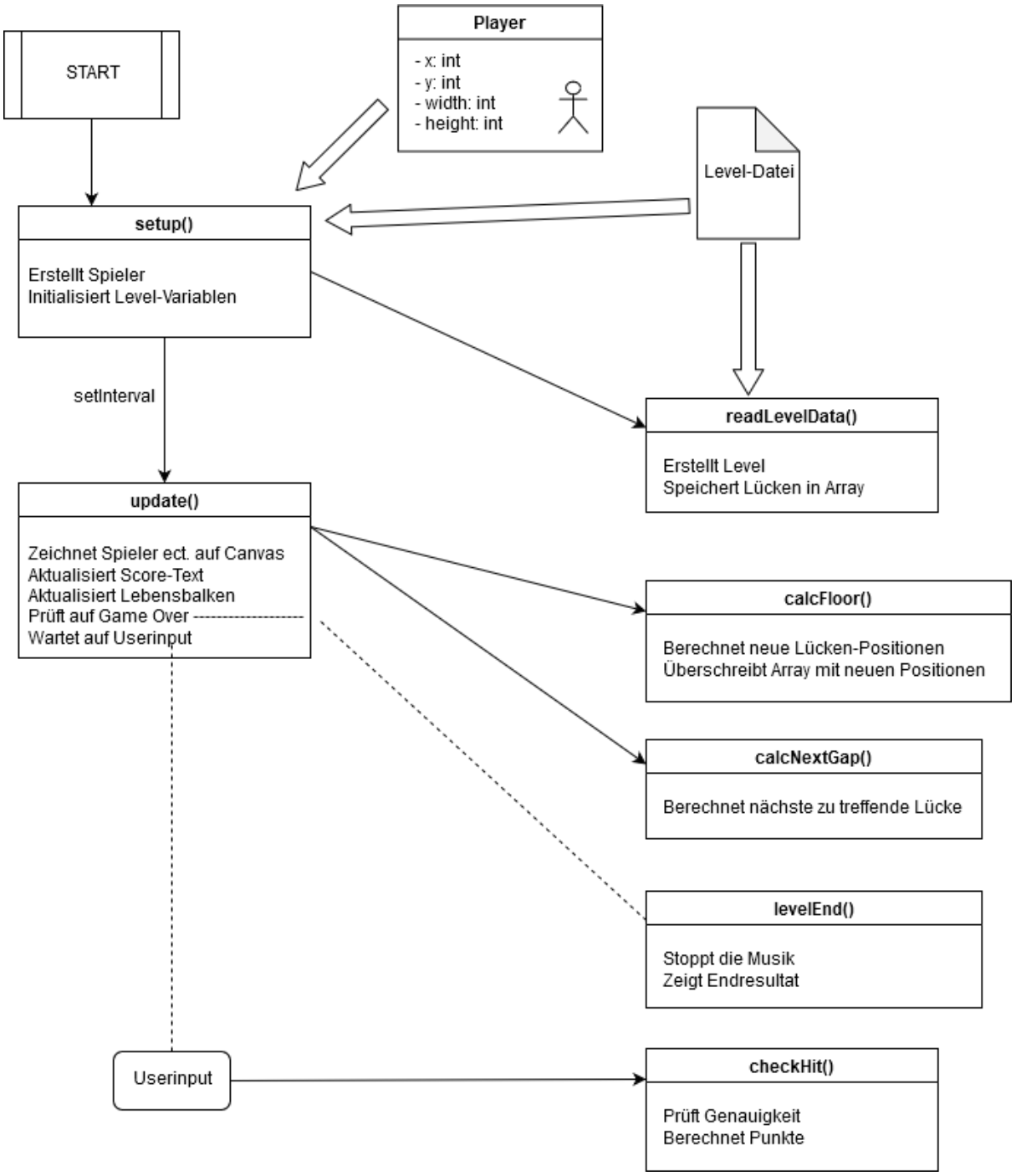
```
//Audio Datein
const lvl2BS = "sounds/Heathens/Heathens Background.mp3";
const lvl2Sound1 = "sounds/Heathens/Heathens Drum1.mp3";
const lvl2Sound2 = "sounds/Heathens/Heathens Hat1.mp3";
const lvl2Sound3 = "sounds/Heathens/Heathens Kick1.mp3";
const lvl2Sound4 = "";
//Level Daten (ScrollSpeed, BackgroundDelay, GapWidth, FirstNoteDistance)
const lvl2SS = 20;
const lvl2BD = 0.5;
const lvl2GW = 40;
const lvl2FND = 2140;

//Farbe und Position der Lücken
data2 = {"gaps":[
"red-0",
"red-796",
"red-1592",
"red-2388",
"red-3184",
"red-3980",
"red-4776",
"red-5572",
"red-6368",
"red-7164",
"red-7960",
```

Level-Datei

Während des Spielens warten die Keylistener auf den Input des Spielers. Beim Drücken der belegten Tasten wird der dazugehörige Sound abgespielt und die CheckHit-Funktion wird aufgerufen. Die CheckHit-Funktion überprüft ob die korrekte Taste gedrückt wurde und berechnet wie groß der Fehler war. Je näher am perfekten Hit, desto geringer der Fehler. Aus dem Fehler wird anschließend die Punktzahl berechnet, die der letzte Hit dem Spieler gebracht hat. Außerdem wird hier bei korrekten Hits der Lebensbalken ein wenig gefüllt.

Wenn der Spieler das Ende des Levels erreicht oder die Lebenspunkte des Spielers auf Null fallen, wird die LevelEnd-Funktion aufgerufen. Diese stoppt die Musik und lässt ein Overlay erscheinen auf dem das Endergebnis angezeigt wird. Hier hat der Spieler die Möglichkeit es noch einmal zu versuchen oder ins Hauptmenü zurückzukehren und ein anderes Level zu spielen.



Ablauf-Diagramm