

MIB SoSe 2020

Prototyping interaktiver Medien-Apps und Games

Endabgabe Designdokument

Für das Spiel

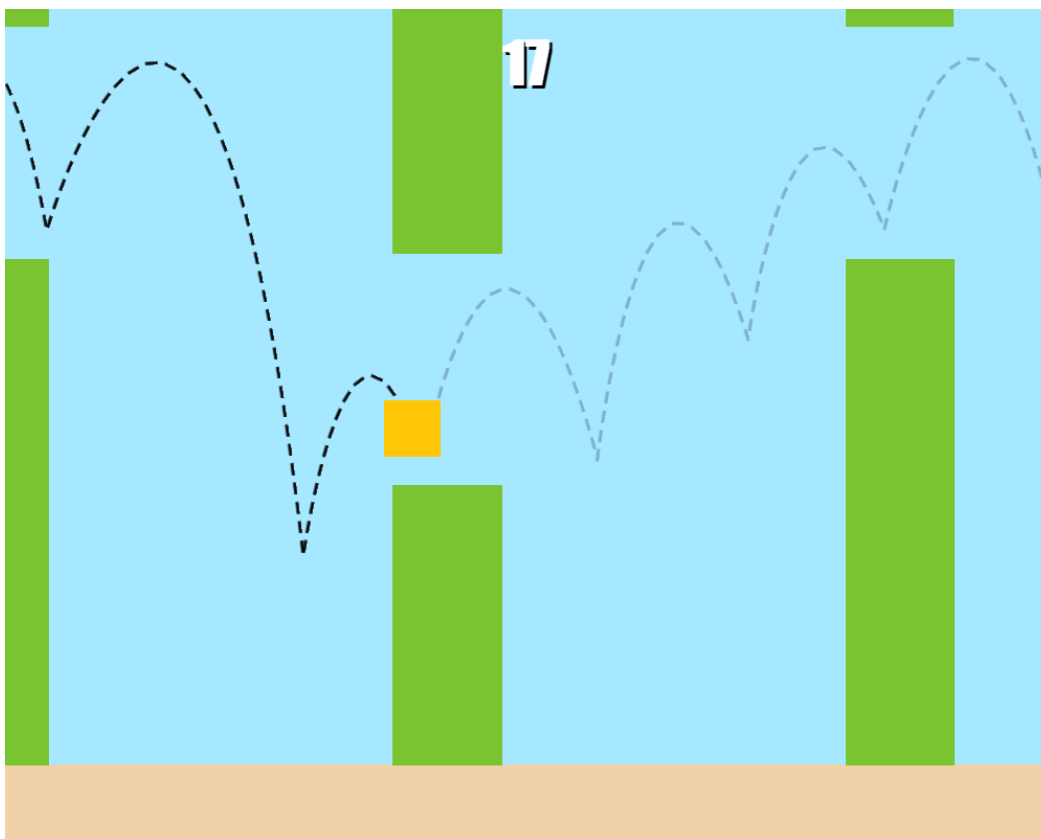
FLAPPY BOX

Eingereicht von: Robin Pittelkow

Matrikelnummer: 257730

Spielerklärung und Spielweise

Das Spielprinzip basiert auf dem 2013 veröffentlichten Mobile-Game „Flappy Bird“. Ziel war es einen Vogel in der Luft zu halten und währenddessen durch Hindernisse hindurch zu fliegen. Die einzige Möglichkeit den Vogel daran zu hindern gegen ein Hindernis zu fliegen war es, den Vogel mit einem Tastendruck springen zu lassen. Da der Vogel durchgehend, der Gravitation verschuldet wieder Richtung Boden fällt, muss der Spieler also ständig, und zum richtigen Zeitpunkt, den Vogel springen lassen, um durch möglichst viele Hindernisse zu springen. Da Level hat kein Ende, da das Spiel dem Endlos-Spiel-Prinzip folgt. Sollte der Spieler gegen ein Hindernis fliegen oder auf den Boden fallen, so ist das Spiel vorbei. Die Punktzahl, die der Spieler bis zu diesem Zeitpunkt erreicht hat, wird dann auf dem Bildschirm angezeigt.



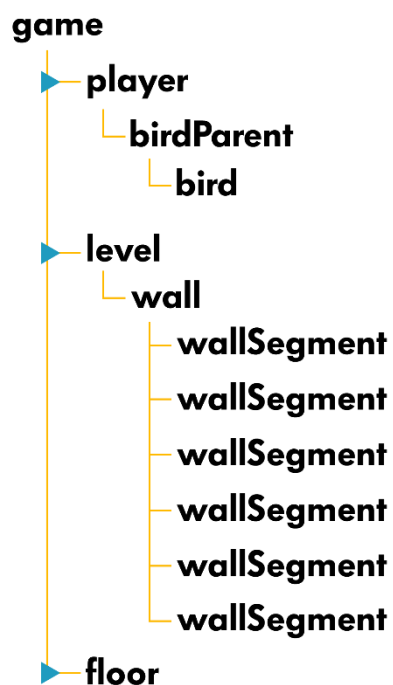
Steuerung

Um den Charakter zu steuern muss lediglich die Leertaste, „W“ oder „Pfeiltaste nach oben“ gedrückt werden. Um das Spiel neu zu starten oder Einstellungen im Einstellungsbereich vornehmen zu können, wird die Maus benötigt.

Hierarchie

Bei der Initialisierung des Spiels wird eine „game“-Node erstellt die als oberstes Parent der Szene dient. Alle weiteren Objekte im Spiel werden dieser Node untergeordnet. Zudem wird eine „player“-, eine „level“- und eine „floor“-Node erstellt. Die „level“-Node werden im Laufe des Spiels alle generierten Hindernisse untergeordnet. So sollte es möglichst einfach gemacht werden auf jedes der im Level enthaltenen Hindernisse zugreifen zu können, um beispielsweise auf jedem Hindernis eine Funktion aufrufen zu können, die untersucht, ob der Player gerade mit dem entsprechenden Hindernis kollidiert. Die Hindernis-Objekte, im Code „wall“ genannt, bestehen aus mehreren Unterobjekten in Form eines Quadrates. Diese werden im Code „wallSegment“ genannt. Diese werden sowohl zur Hindernisgenerierung als auch zur Kollisionsabfrage benötigt. Der Player besteht aus einem „birdParent“ und einem dem „birdParent“ untergeordneten „bird“. Zum besteht unter der „game“-Node eine „floor“-Node, die nur zu visuellen zwecken benötigt wird.

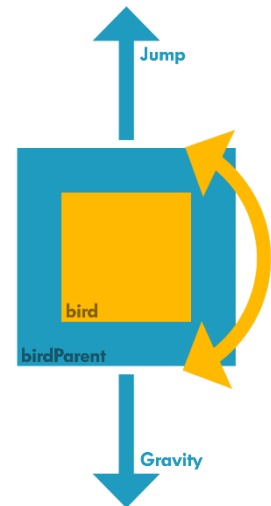
Zur Veranschaulichung:



Funktionsweise der Spielelemente

Player:

Wie im vorherigen Schaubild zu erkennen ist, besteht der Player aus zwei sich untergeordneten Nodes. Die Node „birdParent“ besteht, um die Gravitation zu realisieren. Sie ist also zuständig für die Translation auf der Y-Achse und somit auch für das Springen des Charakters. Die Node „bird“ wird benötigt, um die Rotation auf der Z-Achse zu realisieren. Denn je nachdem ob der Charakter am Fallen oder am Steigen ist, soll er sich der Richtung entsprechend drehen, um wie im originalen Spiel visuell etwas interessanter wirken zu können. Die Parent-Node „birdParent“ wird nicht als sichtbares Objekt gerendert.



Wall:

Jede „wall“-Node das gespawnt wird besteht aus 6 einzelnen „wallSegment“-Nodes. Bei der Erstellung des Hindernisses wird ein Segment-Objekt nach dem anderen erstellt und um eine Einheit auf der Y-Achse verschoben. Eine Zufallszahl wird benutzt um eine Zahl zwischen 1 und 5 (einschließlich) zu generieren. Anhand dieser Zahl kann während der Wall-Generierung in der „for“-Schleife die Erstellung des entsprechenden Segments unterbunden werden. In diesem Fall wird das Segment mit der Zufallszahl und das Segment mit der Zufallszahl + 1 ausgelassen. Somit gibt es 5 verschiedene Möglichkeiten, an denen eine Öffnung in dem Hindernis sein kann.

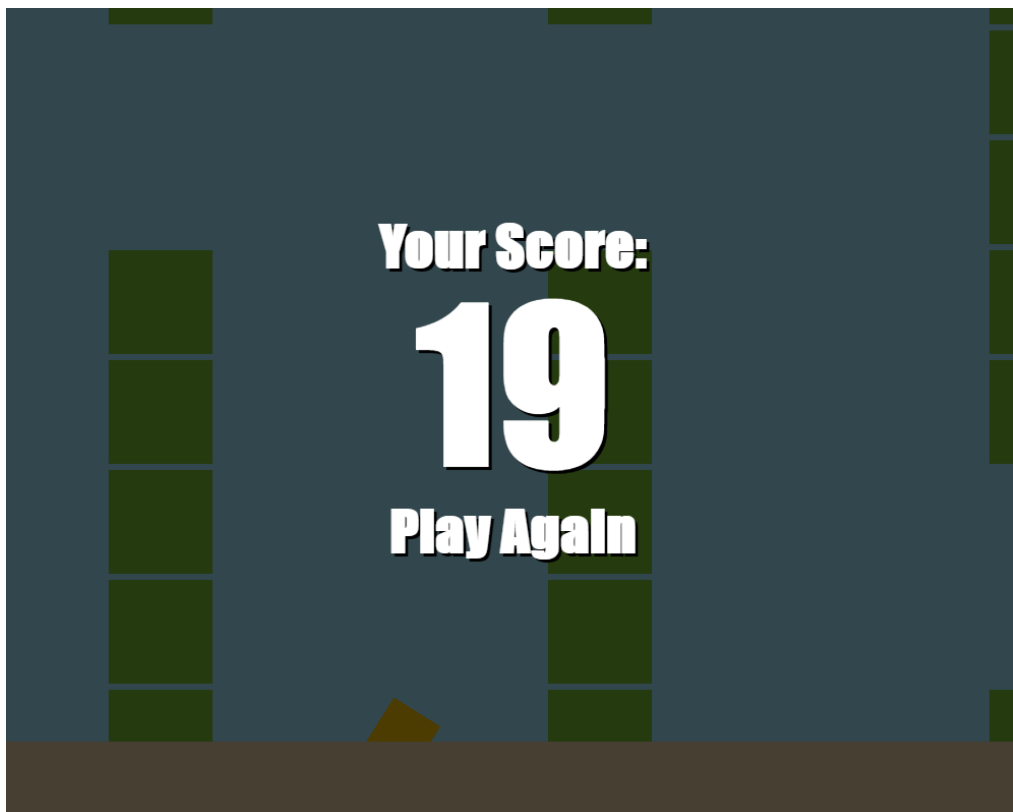
```
let hole: number = Math.floor(Math.random() * 5) + 1;

for (let i: number = 0; i < 8; i++) {
  if (i !== hole && i !== hole + 1) {
    let segment: WallSegment = new WallSegment("Wall");
```



GUI

Sollte der Spieler gegen ein Hindernis fliegen oder den Boden berühren, so endet das Spiel. Zudem wird ein End-Game-Screen eingeblendet, auf der der Spieler die erreichte Punktzahl ansehen kann. Darunter hat er die Möglichkeit das Spiel neu zu starten. Um die Schrift des End-Screens besser erkennen zu können, wird der Rest des Spiels, der noch im Hintergrund zu erkennen ist, etwas mit einer HTML-div abgedunkelt.



Zudem ist am oberen Bildschirmrand über dem Canvas ein kleines Einstellungsmenü, mit dem der Sound eingestellt werden kann. Der Sound kann sowohl stumm geschaltet werden als auch mit einem Slider in der Lautstärke eingestellt werden.

