Design Dokument

Floppy Bird

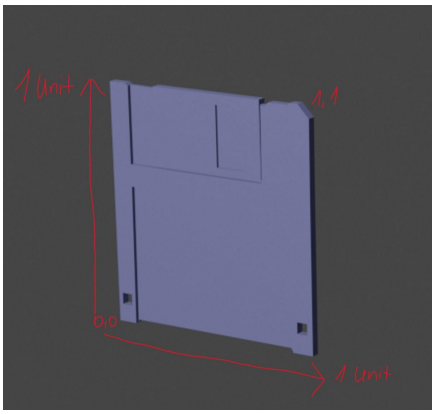
Violetta Pyralov
263005
Prima
WiSe 22/23

0. Idee

Die Idee stammt von dem Spiel Flappy Bird. Bei Floppy Bird spielt man eine Floppydisk, die nicht die Tubes treffen darf, da sonst alle Daten auf der Floppydisk verloren gehen.

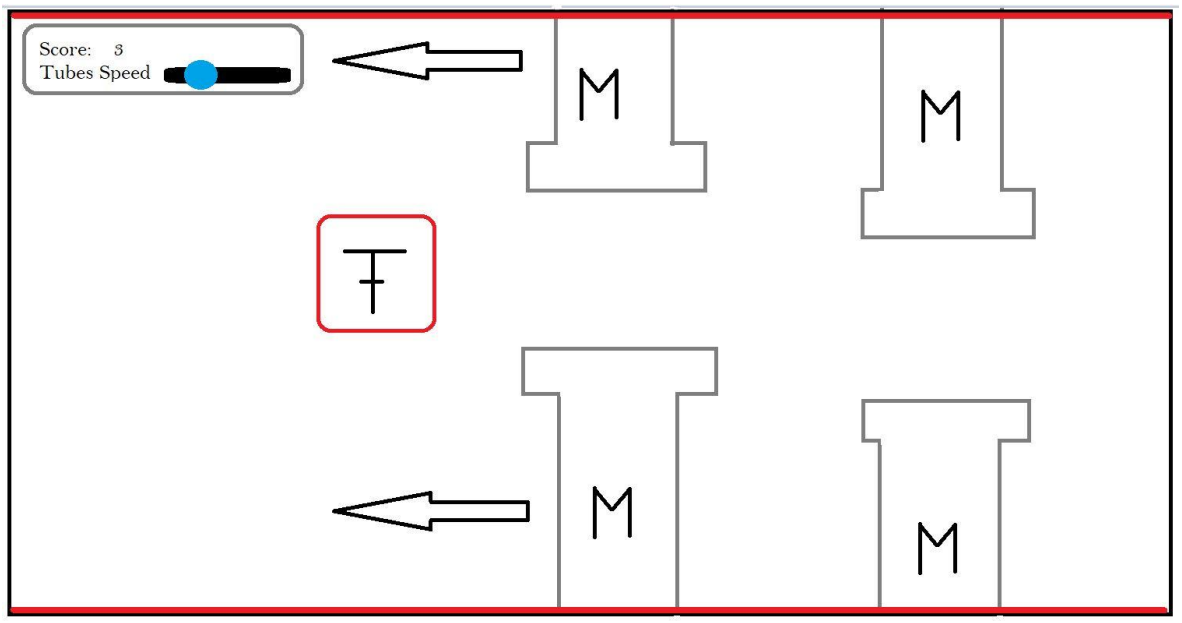
Steuerung
Leertaste drücken, um den Floppy Bird nach oben zu bewegen.

1. Units und Positionen



Der Floppy Bird (FB) ist eine Unit hoch und breit.
Die Tubes sind einen FB breit und mehrere FB hoch, um das Verschieben in der y-Achse ohne Lücken zu realisieren.

Layout



Die Tubes bewegen sich in -x-Richtung. FB bewegt sich nur in beide y-Richtungen.
Die Tubes und FB sind mit einem Rigidbody ausgestattet. Im Code wird überprüft, ob

der FB über einen bestimmten Wert kommt, sollte FB über oder untern diesen Wert liegen ist das Spiel zu Ende. Im VUI kann der Spieler seinen Score sehen und die Geschwindigkeit der Tubes anpassen.

2. Hierarchie

World

- Tubes
  - TubeContainer
    - Tube Oben
    - Tube Unten
    - PunktCollider
- FloppyBirdBody
- Background

Die “Tubes”-Node enthält “TubeContainer”. Innerhalb des Containers befindet sich jeweils ein Tubespaar mit einem Collider zwischen den zwei Tubes. Sobald man diesen Collider passiert, erhält man einen Punkt. Die Tubes werden im Code erzeugt

Floppy Bird besteht aus einem Body und zwei Flügeln. FB wird im Editor erstellt.

Die Collisionborder für oben und unten werden im Editor erstellt.

Der Background wird im Code erstellt, da sich dieser ebenfalls konstant bewegt.

3. Editor

Tubes:
Werden durch den Code erstellt. Sie sind zufällig in der Höhe platziert und bewegen sich konstant in die negative x-Richtung. Das Spiel endet erst, sobald FB eine Tube trifft.

Floppy Bird:
Floppy Bird ist statisch und bewegt sich nur in beide y-Richtungen. Seine Flügel sind mit Joints verbunden.

Kamera:
Die Kamera ist im Editor platziert, da diese ebenfalls statisch ist.

4. ScriptComponents

ScriptComponents wurden für Tubes (`ContinousTubeMovement.ts`) und den Floppy Bird (FloppyBirdPlayer) genutzt.

Für die Tubes:
In diesem Skript wurde die Bewegung der Tubes erstellt sowie das Löschen von den Tubes, wenn diese einen bestimmten Wert in x-Richtung überschritten haben.

Für Floppy Bird:
In diesem Skript wurde die Steuerung, die Collision und der Rest nach Spielende erstellt.

5. Extend

Die Klasse `Tube.ts` extended die `f.Node`, damit die Tubes vom Code aus in die Szene geladen werden können.

6. Sound

Im `PlaySoundManager.ts` werden die 3 verschiedenen Sounds initialisiert.
Der Wing Sound wird beim Drücken der Leertaste abgespielt. Beim Passieren von einem Tube-Paar wird der Point Sound abgespielt, um darauf hinzuweisen, dass ein Punkt hinzugefügt wurde. Wenn man unten aus dem Spielfeld fällt, wird der Collision Sound abgespielt, um deutlich zu machen, dass das Spiel zu Ende ist.

7. VUI

Im Interface werden dem Spieler die bereits überwundenen Tubes angezeigt (Score). Es ist ebenfalls möglich, die Geschwindigkeit der Tubes zu regulieren, indem man die Leiste in eine Richtung zieht. (`UIManager.ts`, `TubeSpeedUIHandler.ts`)

8. Event-System

Es werden Fudge-Events verwendet, um auf Collisions zu reagieren, z.B. die Collision vom Player zu den Tubes oder den Punkt-Collidern. In `FloppyBirdPlayer.ts` wird z.B. die Funktion `collisionHandler` als Callback-Funktion für den Rigidbody des Spielers registriert.

9. External Data

Eine JSON-Datei wird als Quelle von externen Daten verwendet, mit denen man einige Parameter des Spiels anpassen kann. Die Geschwindigkeit der Tube kann zudem im User-Interface angepasst werden.

A. Licht

Jedes Objekt sollte die gleiche Beleuchtungsstärke besitzen, es sollen ebenfalls keine Schatten zu sehen sein, deswegen wurde das Ambient Light ausgewählt für die Beleuchtung der Szene.

B. Physik

In dem Punkt Physik wurden alle drei Punkte (collisions und forces) umgesetzt. Die Collisions wurden benutzt, um eine Collision von Floppy Bird und den Tubes zu überprüfen.

Forces wurden benutzt, um den Floppy Bird mit einem Impuls in y-Richtung zu bewegen. Auf Floppy Bird wirkt die Gravitation, die ihn nach unten zieht.

E. Animation

Animationen wurden bei jedem zweiten Tube-Paar eingesetzt. Dabei werden die Tubes in der y-Achse konstant verschoben und bieten ein etwas schweres Hindernis.