

The Last Super Bowl - Designdokument

Name: Felix Iwertowski

Matrikel-Nr.: 256535

Studiengang: OMB (7. Semester)

Fakultät: Digitale Medien

Fach: Prima

Professor: Prof. Dipl.-Ing. Jirka R. Dell'Oro-Friedl

Semester: Sommersemester 2022

1. Units and Positions

Generell ist der Game Background auf der Z-Ebene auf 0 und alles was im Spiel sich z.B. berührt wie der Hero und die Plattform sind um 1 nach vorne verschoben damit sie sich von dem Hintergrund abheben. Aufgrund der Json Daten der Kamera, welche mit Rotate arbeitet, ist alles etwas aus dem Focus gerückt und hat somit durchaus nicht zwingend ganzen Werte.

An sich ist der Hero und die erste Plattform auf der er landet, genau mittig und kann somit als Mittelpunkt des CordSystem gesehen werden.

Bei den Einheiten wurde sich aufgrund der Sichtbarkeit für Skaling entschieden, was den Hero um 1 scaled und den Football um 2 damit dieser besser gesehen werden kann. Die Plattformen haben in y-Richtung die selbe Größe wie der Hero und die Borders besitzen auch nur eine enorme Höhe damit man nicht ausversehen aus dem Bild springt.

Die Kamera wurde so hingepackt, dass es aussieht als es 2D aber es handelt sich weiterhin um ein 3D CordSystem

2. Hierarchy

Es gibt einen Haupt-Graphen, namens „Game“. An diesen Graphen befinden sich alle weiteren Elemente angehängt. Dieser besitzt an sich erstmal die Komponenten Light und jeweils die Audio Dateien.

Als Unterknoten führt er zum einen Knoten namens „Borders“, welche beide Absperren links und rechts beinhaltet.

Dann den Unterknoten „Plattform“, welcher wiederum alle 15 im Spiel vorkommenden Plattformen unter sich als Childknoten hat.

Dann hängen noch zwei weitere Knoten Namen „BackgroudTop“ und „BackgroudBottom“ am Game Graphen. Diese sind die Hintergrundelemente (Top ist das Stadion und Bottom die Lava).

Und zu guter letzt gibt es noch den Hero Knoten, welcher aber nur einen Grapheninstanz ist und sich aus dem eigenen Hero Graphen bildet.

Desweiteren gibt es noch den Collectable Graphen welcher der Football ist, dieser wird aber erst im Code an den Game-Graphen angehangen.

3. Editor

So gut wie alles was man sieht wurde direkt im Editor erstellt und an den vorher besagten Game-Graphen angehängt. Mit ausnahmen des Footballs, welche erst später im Code dann an das Game angehängt wird.

Per Code wird also kaum etwas erstellt, sondern nur mit den nötigen Skripten versorgt. Dies ist darauf zurück zu führen, dass z.B. keine weiteren Plattformen geladen werden müssen, wenn man weiter nach rechts gehen würde, da es sich um einen festen Bereich handelt, im welchem sich der Spieler nur aufhalten kann.

4. Scriptcomponents

Die Script-Komponenten wurden grundsätzlich für den Hero und den Football verwendet um die Berührung abzu prüfen.

Generell ist es dadurch einfach, da man nicht einen gesamten Code schreiben muss sondern sich auf kleinere Teile beziehen kann. So kann man den einzelnen Elementen eigene Scripts mitgeben. Auch sehr hilfreich wenn man es einer Klasse wie in dem Fall dem Football mitgibt und diese es dadurch auch wenn sie neu per Code hinzugefügt wird, einfach anwenden kann.

5. Extend

Alles was im Spiel zusehen ist, sind Unter-Nodes von dem Game Node. Zudem gibt es noch eine GraphInstance also eine eigene Klasse namens Hero, welche wiederum dadurch zu Unterklasse des Game Graphen wird.

Zudem wurde ComponentScript extentend, was sehr hilfreich war um weitere Script-Komponenten im Spiel unterzubringen und zu ergänzen.

6. Sound

Beim Sound wurde sich für zwei unterschiedliche Geräusche bzw. Sounds entschieden. So wird beim aufsammeln eines Footballs, ein aus vielen Spielen wie z.B. Mario oder ähnlichem bekannten Münzen Sound abgespielt um den Spieler darzustellen, dass er gerade 100 Punkte verdient hat.

Desweiteren gibt es eine Stadion Atmosphäre im Hintergrund welche im Loop permanent spielt um so dem Spieler das Gefühl zu vermitteln, er sei gerade in einem echten Football Stadion. Im Code werden die beiden Sounds welche Audio-Components des Game Graphen sind, mithilfe der Funktion getcmpAudio aufgerufen, welche sich in der Main.ts befindet.

7. VUI

Das UI ist sehr minimalistisch gehalten, da zum Erreichen des Spielprinzips nicht viel angezeigt werden muss. Es gibt einfach nur eine Punktzahl, welche beim Einsammeln von einem Football sich um den Wert 100 erhöht. Damit hat der Spieler einen Wert, an dem er sich mit anderen messen kann. Sollte der Spieler abstürzen, ist das Spiel direkt fertig, deshalb ist auch keine Lebensanzeige oder Zeitanzeige von Nöten.

8. Event-System

Das Spiel triggert den Endbildschirm, wenn der Hero in die Lava fällt bzw. unter eine bestimmte Höhe von „-5“ fällt.

Dann wird eine Endscore getriggert bzw. angezeigt und das Spiel wird neugeladen, indem der Spieler das Window neu lädt.

9. External Data

Im Ordner „Data“ liegt eine Datei mit dem Namen „testData.json“, diese dient dazu, die festen Variablen für die Kamera festzulegen. In der „Main.ts“ werden diese Daten ausgelesen, in der Funktion `getJsonData`. Die Funktion wird beim Aufbau per `Await Call` aufgerufen und im Anschluss werden damit die Translationen und Rotationen für die Kamera Variablen festgelegt.

Hier könnte man noch andere Sachen einbringen wie z.B. könnte man hier die Anzahl Bälle definieren, welche der Spieler einsammeln kann. Somit könnte man das Spiel schwerer oder einfacher machen.

A. Light

Beim Spiel ist Licht und Schatten kein zentrales Game-Element, weswegen alles gleichmäßig beleuchtet werden kann bzw. muss.

Aus diesem Grund wurde sich im während des Aufbaus im Editor nur für ein Ambient-Light entschieden.

B. Physics

Sowohl der Hero als auch die Footballs, welche in das Spiel geladen werden, haben `RigidBody`s um sich gegenseitig auf Collision zu checken. Die Borders auf den Seiten haben auch feste `Bodys` um den Spieler nicht an den Seiten rauszulassen.

C. Net

/

D. StateMachine

/

E. Animation

/