

Moorhuhn
Dokumentation

Prima Abgabe
Timo Melnicki
265173
OMB



Steuerung

Nach dem das Spiel startet, kann mit der Linken Maustaste ein Schuss abgegeben werden. Dieser wird auf die Position des Cursors gefeuert. Befindet sich ein Huhn zu dem Zeitpunkt des Mausklicks auf der Position des Cursors, wird dieses getroffen und "stirbt". Der Spieler erhält dafür Punkte, die zum aktuellen Score addiert werden. Ist zu dem Zeitpunkt kein Huhn auf der Position des Cursors ist es ein "Fehlschuss" und es werden keine Punkte addiert.

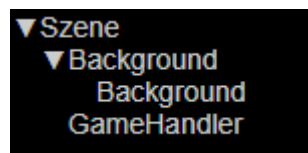
Schafft es ein Huhn die Strecke von einem Bildschirmende zum anderen zu fliegen, verliert der Spieler eines seiner 3 Leben. Bei 0 Leben ist das Spiel beendet und der Spieler hat verloren.

Units und Positions

Die Spielwelt wird um den Ursprung (0) aufgebaut. Anfangs war die Überlegung, die Welt nur im positiven Bereich zu erstellen wurde allerdings verworfen nach der Erkenntnis dass die Hühner auf beide Bildschirmenden geprüft werden müssen und nur ein Abgleich auf negative Werte so keinen Vorteil bietet.

1 beschreibt die Länge und Höhe eines Huhns. Damit kann z.B. die Spielwelt in Hühner berechnet werden. Das ist relevant bei der Höhe da mehrere Hühnchen gleichzeitig spawnen damit es nicht zu klein ist oder bei der Länge, da es beeinflusst wie lange 1 Huhn für die Strecke braucht.

Hierarchy



Die Szene ist das Höchste in der Hierarchy und beinhaltet den Hintergrund und den GameHandler.

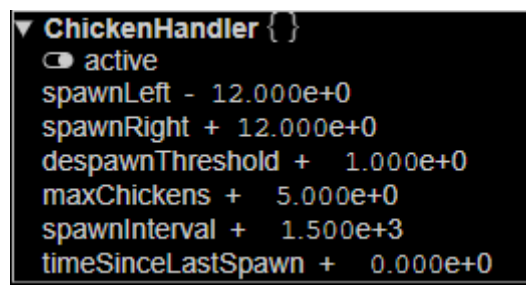
Der Hintergrund beinhaltet die Node mit dem Material und Mesh für den Hintergrund.

Der Gamehandler ist auf derselben Höhe wie der Hintergrund und kann bestimmte Werte der Hühner beeinflussen.

Editor

Kamera, Hintergrund und die Audio Komponente mit der Musik wurden im Editor erstellt. Der “ChickenHandler” befindet sich ebenso im Editor wodurch sich zahlen einfacher anpassen lassen. Es können wichtige zahlen die z.B. aus der größe des Hintergrunds hervorgehen, direkt für den ChickenHandler angepasst werden.

ScriptComponents



Der ChickenHandler ist eine ScriptComponent. Diese hilft dabei den Code übersichtlicher zu gestalten und Werte im Editor anpassen zu können. Sollte z.B. der Hintergrund angepasst werden, kann über den ChickenHandler der Spawnpunkt (links oder rechts) angepasst oder die maximale Anzahl an Hühner erhöht werden. Zusätzlich kann der Intervall angepasst werden sollten die Hühner zu schnell oder zu langsam spawnen.

Extend

Die Klasse “Chicken” und “Player” sind extended Klassen. Für die Klasse “Chicken” wurde ein NodeSprite extended. Die Klasse Spieler ist ein Muatable das extended wurde.

Sound

Es gibt mehrere Sounds in Moorhuhn. Der Erste Sound ist die ambient Musik. Sie dient dazu das Spielerlebnis zu verbessern in dem “Farmmusik” abgespielt wird. Dadurch entsteht das Gefühl man würde sich auf einem Bauernhof befinden.

Der Zweite Sound ist das Schussgeräusch. Die Hauptfunktion dieses Sounds ist es, dem Spieler eine Auditive Rückmeldung zu geben, das auch geschossen wurde und die Spieleraktion stattgefunden hat.

VUI

Das Userinterface beinhaltet 2 Anzeigen. Den Score und die Leben. Diese 2 Elemente sind die Wichtigsten Zahlen die der Spieler für das Spiel benötigt. Der Score zeigt an wie viele Punkte der Spieler bereits gesammelt hat und die Leben zeigen an wann wie weit der Spieler von einer Niederlage entfernt ist.



Event-System

Das Wichtigste Event für das Spiel ist das schießen. Bei jedem Mausklick wird dieses Event durchgeführt. Es dient dazu dem Spieler das Sammeln von Punkten zu ermöglichen und eine Niederlage zu verhindern.

```
//Shot Event  
viewport.canvas.addEventListener("pointerdown", (_event) => {player.pickByRadius(_event)});
```

External Data

```
Moorhuhn > {} config.json > ...  
1  {  
2    "flapForceHorizontal": 0.5,  
3    "flapForceVertical": 4.3  
4  }
```

Die Beiden Variablen “flapForceHorizontal” und “flapForceVertical” werden extern konfiguriert.

Light

War nicht Notwendig

Physics

Die Physics in diesem Spiel beinhalten ausschalten der Kollisionen, da Hühner gegeneinander geflogen sind, wodurch das Spielerlebnis gestört wurde, das wurde mithilfe der “CollisionMask” realisiert.

```
this.rigidBody.collisionMask = 0;
```

Es wurden Forces verwendet, damit die Hühner “fliegen”.

Net

Nicht vorhanden

State Machines

Hühner haben 2 Zustände in denen sie sich befinden können. Entweder sind sie “alive” oder “dead”. Im Zustand “alive” fliegen die Hühner und im Zustand “dead” fallen sie herunter und werden gelöscht.

Animationen

Es wurden 2 Animationen mithilfe von Sprite Sheets erstellt. Die Erste Animation ist das die Hühner fliegen. Die Zweite ist die Todesanimation, wenn die Hühner abgeschossen wurden und herunterfallen.