

SAVE YOUR WORLD! VR

EINE PROJEKTDOKUMENTATION ÜBER EIN FRUSTIERENDES VR PROJEKT

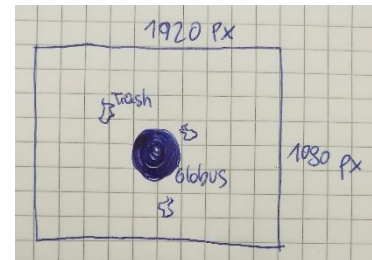
Im Rahmen des MTIN Unterrichts war es unsere Aufgabe ein VR Spiel zu konzipieren und zu programmieren. Da wir uns für eines der verschiedenen zur Verfügung gestellten VR Headsets entscheiden mussten, wählte ich **Google Cardboard** für mein Spiel.

1 DIE PLANUNG

1.1 IDEE

Wie jedes gute Spiel enthält meines einen Globus, der in der Mitte von einem Raum steht und sich dreht. Der/die SpielerIn hat die Aufgabe, die Welt vor der totalen Umweltverschmutzung zu retten.

In dem Raum liegen Müllstücke herum, welche der/die SpielerIn aufheben und entsorgen muss. Jedes Mal, wenn er/sie das tut, wächst ein Baum an der Stelle wo einst der Müll gelegen ist. Wiederholt der/die SpielerIn das mehrmals und entsorgt den ganzen Müll, ist die Welt gerettet und der Globus dreht sich schneller.



1.2 STEUERUNG

1.2.1 FORTBEWEGUNG

Der/die SpielerIn schaut in die Richtung in die er/sie sich bewegen will und drückt gleichzeitig auf den Knopf rechts oben auf der Brille.



1.2.2 INTERAKTION

- Der/die SpielerIn sieht das Objekt an, welches er/sie aufheben will und drückt gleichzeitig auf den Knopf.
- Der/die SpielerIn sieht den Gegenstand an, in welches er/sie das Objekt hineinlegen will und drückt gleichzeitig auf den Knopf.

1.3 ENTWICKLUNGSPLATTFORM

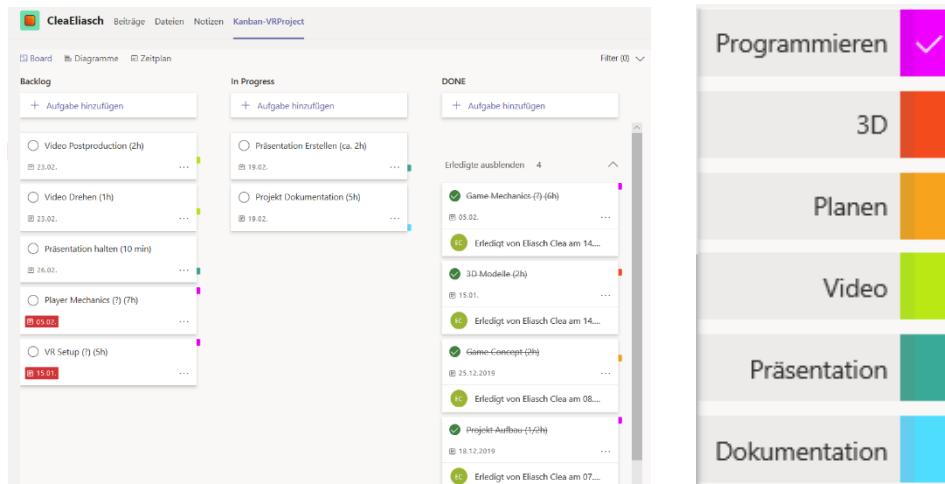
- Windows 10 64 bit
- Unity3D 2019.1.14f
- API Compatibility Level .NET Standard 2.0
- Scripting Runtime Version .NET 4.x Equivalent
- Visual Studio Community 2019 16.4.2

1.4 ZIELPLATTFORM

- Android
- Google Cardboard
- 1920 x 1080 px

1.5 KANBAN BOARD

Im Rahmen der Planung musste jeder/jede SchülerIn ein Kanban Board auf Microsoft Teams erstellen. Dies geschah im Unterricht, und dauerte etwa die ganze Einheit in dieser Woche. Um noch mehr Struktur in mein Kanban Board zu bringen, entschied ich mich dazu die Flag Funktion zu verwenden und den einzelnen Aufgabenbereichen meines Projektes eine Farbe zuzuweisen. Außerdem schrieb ich neben dem Aufgabennamen auch die Stundenanzahl hin, welche ich schätzungsweise dafür brauchen werde. Die Aufgaben, von denen ich noch keine Ahnung hatte wie ich sie lösen sollte, kennzeichnete ich mit einem Fragezeichen (?).



2 DAS SETUP

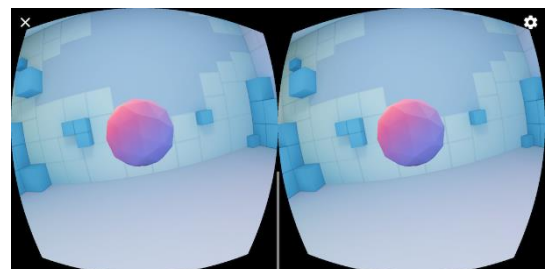
2.1 GOOGLE CARDBOARD VR SETUP

Da ich noch nie zuvor mit Google Cardboard gearbeitet habe und generell bisher sehr wenig Erfahrung mit VR Games machte, entschied ich mich dazu eine Recherche zu betreiben, wie genau ich mein Projekt aufbauen muss und welche Bestandteile nötig sind, um mit diesem Medium problemlos arbeiten zu können.

Dafür erwiesen sich folgende Websites und das Video Tutorial als besonders hilfreich. Diese verwendete ich schlussendlich auch als Basis für den Setup Aufbau.

<https://developers.google.com/vr/develop/android/get-started>
<https://docs.unity3d.com/Manual/android-sdksetup.html>
<https://www.youtube.com/watch?v=OEP7sMwfZnE&list=PLiuDBLn1FK-QOmpA3Vs2JY0DVAj0GfFPn&index=1>

Problemlos konnte ich Google VR installieren und nach den weiteren Setup Schritten (SDKs, NDKs installieren) das von Google zur Verfügung gestellte Testspiel auf meinem Handy mit Google Cardboard ausprobieren.



2.1.1 SETUP PROBLEME

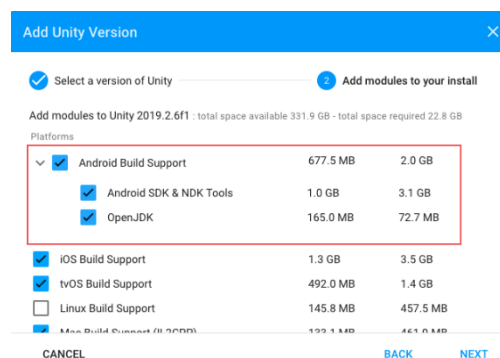
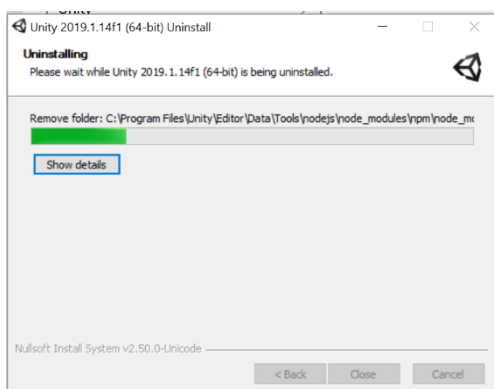
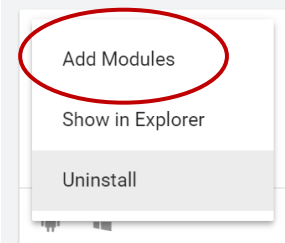
Auch wenn die Tutorials sehr gut sind, war es für mich nicht möglich ihnen auf Anhieb problemlos zu folgen. Wichtig zu bemerken ist, dass ich mitten unter dem Projekt einen großen Fehler Online begangen habe und mir einen Virus auf meinen Rechner holte. Dieser war für mich anfangs nicht bemerkbar bis einiges an Bargeld von meinem Konto verschwand. Aus diesem Grund war ich direkt nach dem Vorfall sehr vorsichtig im Umgang mit dem Internet und mied das downloaden von jeglichen Dateien, da dieser Virus sehr aggressiv und versteckt war.

Deswegen erwies sich das downloaden der nötigen SDKs und NDKs als ein großes Problem, an dem ich einiges an Zeit verschwendete. Anfangs versuchte ich den Download zu umgehen und die Files von einer bereits auf den Schulrechnern existierenden Unity Version auf meinen Rechner zu kopieren. Dies dauerte etwa 3h aber funktionierte trotz aller Anstrengung nicht.

Da anscheinend kein Weg drum herumgeführt hat, kaufte und installierte ich so schnell wie möglich ein anderes, hoffentlich besseres, Virenprogramm auf die Hoffnung eines sauberen Rechners.

Laut den Tutorials sollte das installieren der benötigten Dateien ganz einfach über den Unity Hub funktionieren. Dort sollte es den Reiter „Add Modules“ geben, welcher bei mir jedoch NICHT zu finden war. Deswegen deinstallierte und installierte ich Unity Hub noch einmal, was jedoch nichts änderte. Mein Plan war es danach, Unity zu deinstallieren und wieder zu installieren, jedoch kam ich gar nicht so weit. Denn nach dem deinstallieren, war sinnloser Weise die gleiche Unity Version noch immer installiert, und der Reiter „Add Modules“ erschien auf einmal gefühlt aus dem Nichts. Ehrlichgesagt hatte ich keine Ahnung warum das so war, vielleicht hatte ich die gleiche Version zweimal installiert (?) oder Sonstiges. Aber ich war einfach froh das es ENDLICH funktionierte und ich die nötigen Setup Schritte ausführen konnte.

Installs



3 DAS SPIEL PROGRAMMIEREN

Nach dem wohl am meisten frustrierenden Schritt des Setups, wagte ich mich an das tatsächliche erstellen meines Spieles in Unity.

3.1 ERSTE SCHRITTE

Als erstes baute ich mit Basic Objekten (Cubes) meine Szene auf und änderte die Reference Resolution auf die gewünschten 1920 x 1080px. Außerdem änderte ich ein paar Parameter in den Settings (Product Name, Company Name, ...)



Die ersten Schritte, wie das Erstellen des „Spielers“ (Kamerarig) und die Fortbewegungen löste ich mit YouTube Tutorials

Fortbewegung mit Google Cardboard in Unity:

<https://www.youtube.com/watch?v=Qgpb5QzhuOg&list=PLiuDBLn1FK-QQompA3Vs2JY0DVAj0GfFPn&index=2>

Erste Schritte mit Google Cardboard in Unity:

<https://www.youtube.com/watch?v=OEP7sMwfZnE&list=PLiuDBLn1FK-QQompA3Vs2JY0DVAj0GfFPn&index=1>

3.1.1 POINTER PROBLEME

Ein Problem was mir sofort während den ersten Schritten auffiel war, dass der Pointer nicht richtig funktionierte. Wenn sich der/die SpielerIn umsah, verschwand der Pointer auf manchen Flächen oder verzerrte sich (Vor allem beim extremen Hinauf oder Hinter schauen). Anfangs dachten ich und Herr Gruber, dass dies nur ein Anzeigefehler in Unity war, doch beim ersten ausprobieren meiner Szene auf meinem Handy verhielt sich der Pointer genau gleich komisch.

Dasselbe Problem hatte Carolin, welche auch mit Google Cardboard arbeitete. Deswegen fragte ich sie um Rat und sie hatte tatsächlich einen Lösungsansatz, da sie auf das gleiche Problem gestoßen ist. Wenn man einen Cube zu dem Pointer hinzufügte, funktionierte dieser Einwandfrei ohne Probleme.



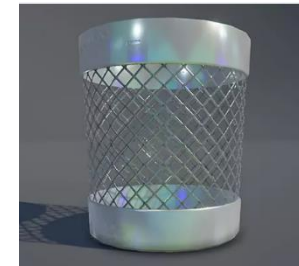
3.2 3D MODELS

Anschließend suchte ich mir aus dem Unity Asset Store und von der Website free3D.com passende 3D Modelle raus:

- Globus
- Mülleimer
- Plastikbecher
- Baum



Das Einfügen der 3D Modelle funktionierte perfekt. Ich hatte das Glück, dass bei dem Globus automatisch ein Spin Script dabei ist. Somit musste ich mich nicht mehr um die Drehung des Modells kümmern und konnte mich sofort mit dem Programmieren der einzelnen nötigen Funktionen widmen. Zum Programmieren arbeitete ich nur mit EINEM Plastikbecher und Baum im Raum.



3.3 INTERAKTION MIT DEN OBJEKTEN

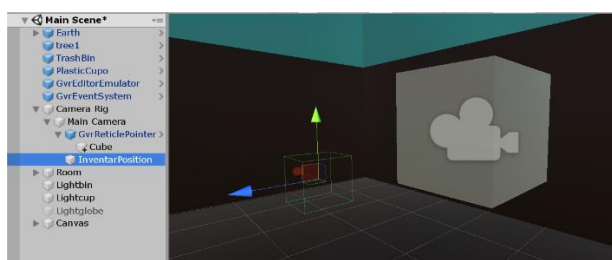
Ein wichtiger Bestandteil meines Spieles, ist das Aufheben und Hinlegen der Plastikbecher. Als Hilfestellung benutze ich wieder ein YouTube Tutorial.

Aufheben/Loslassen von Objekten mit Google Cardboard

<https://www.youtube.com/watch?v=wWSYT-Yrr9E>

```

SpinFree.cs PlayerGrab.cs PlayerWalk.cs
Assembly-CSharp
1 using UnityEngine;
2
3 [Version]
4 public class PlayerGrab : MonoBehaviour {
5     public GameObject plasticCup;
6     public GameObject inventory;
7     public GameObject tree;
8
9     bool inHands = false;
10    Vector3 binPos;
11    Vector3 treePos;
12
13    [Version]
14    void Start()
15    {
16        binPos = new Vector3(-1.26f, 0.1f, 0.02f);
17        treePos = new Vector3(2.14f, 0.186f, 1.93f);
18    }
19
20    [Version]
21    void Update()
22    {
23        if (Input.GetButtonDown("Fire1"))
24        {
25            if (!inHands)
26            {
27                plasticCup.transform.SetParent(inventory.transform);
28                plasticCup.transform.localPosition = new Vector3(2.14f, 0.186f, 1.93f);
29                inHands = true;
30            }
31            else if (inHands)
32            {
33                this.GetComponent<PlayerGrab>().enabled = false;
34                plasticCup.transform.SetParent(null);
35                plasticCup.transform.localPosition = binPos;
36                tree.transform.localPosition = treePos;
37                inHands = false;
38            }
39        }
40    }
41
42 }
  
```



Das Objekt, welches man aufhebt, landet im Inventar. Die Position von dem Inventar legte ich mit einem Cube fest (InventoryPosition). Dort „Spawnen“ die Objekte sobald man sie aufhebt.

Ein Problem, welches ich derzeit noch habe, ist, dass man das Objekt im Inventar „loslässt“ sobald der/die SpielerIn losgeht. Da der Gehmechanismus auch durch das Betätigen des Knopfes aktiviert wird. Im Code erkenntlich, „spawnt“ das Objekt automatisch in den Mülleimer (binPos) sobald man wieder losgeht. Eigentlich wäre es geplant gewesen, dass man das Objekt durch das Ansehen des Mülleimers und gleichzeitiges Drücken des Knopfes hineinlegt. Ich fand leider kein passendes Tutorial, konnte Carolin nicht fragen, da ihre Spielmechanismen anders funktionieren und selbst hatte ich auch

keinen Plan wie das funktioniert. Deswegen „spawnt“ der Becher in der derzeitigen Testversion automatisch in den Mülleimer.

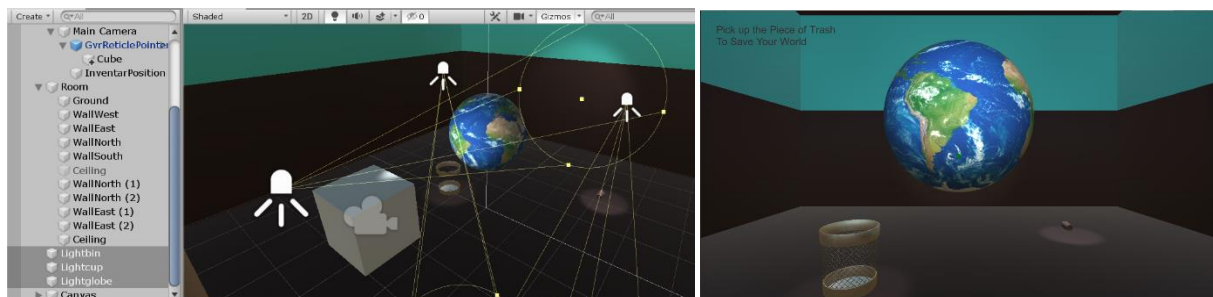
Außerdem „wächst“ der Baum auch nicht wirklich aus dem Boden, sondern erscheint an der treePos (Position des Baumes) sobald der Becher im Mülleimer liegt.

Trotz dieser Einschränkungen bin ich dennoch selbst zufrieden mit dem was ich bis jetzt erreicht habe, trotz der Startschwierigkeiten und meinem Unwissen was VR angeht.

3.4 LICHSETUP UND TEXT

Der letzte Schritt für mich war es 3 Spotlights und einen Text einzufügen, welcher erklärt was zu tun ist.

Die 3 Spotlights platzierte ich auf den Globus von vorne und auf den Plastikbecher sowie auf den Mülleimer von oben.



4 DAS ENDERGEBNIS

Wie bereits oben erwähnt, bin ich trotz des unfertigen Projektes recht zufrieden mit meiner Arbeit, da ich zwar nicht MEIN geplantes Ergebnis erreichen konnte aber immerhin allen Anforderungen an das Projekt, welche von der LEHRKRAFT gestellt wurde, gerecht geworden bin (ein sich drehender Globus; der/die SpielerIn kann sich fortbewegen; der/die SpielerIn kann mit einem Objekt interagieren; das Spiel funktioniert in VR und lässt sich auf dem definierten Device spielen; das Projekt wurde mitdokumentiert).

4.1 Zeitaufwand Vergleich

Generell schätzte ich den Arbeitsaufwand ganz gut ein und plante mir meistens genug Zeit ein für die einzelnen Tasks. Das einzige was sehr viel länger gedauert hat als anfangs geplant, war das VR Setup. Durch die Probleme bei dem SDKs und NDKs Download, war ich gleich von Anfang an ziemlich weit hinten. Doch dies konnte ich gut aufholen, da ich dann weniger Zeit für die 3D Modelle brauchte. Dank den Tutorials fiel mir der Programmerteil auch etwas einfacher als erwartet. Wobei man auch festhalten muss, dass ich ja schlussendlich nicht alle Funktionen des Spieles umsetzen konnte so wie ich sie eigentlich geplant hätte.

Das Kanban Board half mir nicht wirklich, da ich generell die Zeitplanung von meinen Projekten anders handhabe (Excel Tabelle mit Kalenderwochen). Trotzdem war es interessant damit zu arbeiten.