

## Projektdokumentation Fuck Up VR-Project

### 26.02.2020 - Beginne und Ideenfindung

Ende Februar habe ich in den ersten 3 Stunden Medienproduktion in diesem dritten angefangen, eine grobe Idee für mein Virtual Reality Projekt zu finden. Zuerst hatte ich die Idee, dass der/die Spieler\*in in einem Klassenraum eine Aufgabe erledigen muss, dies habe ich jedoch dann verworfen, da einige Klassenkollegen/kolleginnen aus den vorherigen Gruppen schon ähnliche Ideen umgesetzt hatten und ich nicht wollte, dass es so wirkt, als hätte ich eine Idee geklaut. Aufgrund dessen ist mir dann die Idee gekommen, da zu der Zeit wo wir das Projekt abgeben müssen gerade Ostern ist/gewesen sein wird, eine Ostereier suche in Virtual Reality umzusetzen.

Ich habe mich, in Bezug auf die Target Platform, für die HTC Vive beziehungsweise die VR Oculus Rift entschieden, da ich mit diesen VR Brillen bisher schon die meiste Erfahrung gesammelt hatte, ich die Controls bei diesen zwei Geräten am angenehmsten finde und ich mit Steam VR schonmal ein bisschen gearbeitet habe.

Folgend darauf habe ich dann auf GitHub ein Repository erstellt, auf meinen Laptop gecloned und dann mein Unity Projekt eingerichtet. Ich habe dafür mein Template für 3D Projekte in Unity in meinen Ordner kopiert und dann über Unity Hub geöffnet, da ich wie immer mit der Unity Version 2019.1.14f1 arbeitete. Daraufhin habe ich das Gitignore mit den Informationen für Mac und Windows überarbeitet, sowie das Readme mit allen Informationen aktualisiert, die an diesem Stand des Projektes schon bekannt waren beziehungsweise ausgefüllt werden konnten.

Nach diesen grundlegenden Schritten habe ich angefangen im Assetstore von Unity nach passenden Assets für meine Szene zu suchen. Ich habe zu Beginn eher nach den Modellen gesucht, welche für die Umsetzung meines Projektes essentiell waren. Zu diesen zählten die Ostereier, der Korb, der Globus und auch ein bisschen Umgebung, in welcher dann die Ostereier versteckt werden. Es wurden jedoch noch keine Assets zu der Szene hinzugefügt.

Für diese Arbeit sind alle 3 Stunden Medienproduktion benötigt worden.

### 04.03.2020 - Fertigstellung der Konzeption und Beginn der Raummodellierung

Die 3 Stunden Medienproduktion an diesem Tag habe ich einerseits damit verbracht, die Idee der Ostereier Suche zu finalisieren. Dafür habe ich eine grobe Skizze aus der Vogelperspektive angefertigt, wie die Szene final aussehen sollte, jedoch war diese sehr

minimalistisch. Dazu habe ich notiert, was in der Szene alles passieren sollte, was die Aufgabe der Spieler\*innen ist und welcher Code benötigt wird beziehungsweise was durch Programmierung umgesetzt werden sollte. Weiteres wurden die benötigten Animationen festgelegt und eine optionale Idee angefügt, welche umgesetzt werden sollte, wenn am Schluss noch Zeit übrig bleibt.

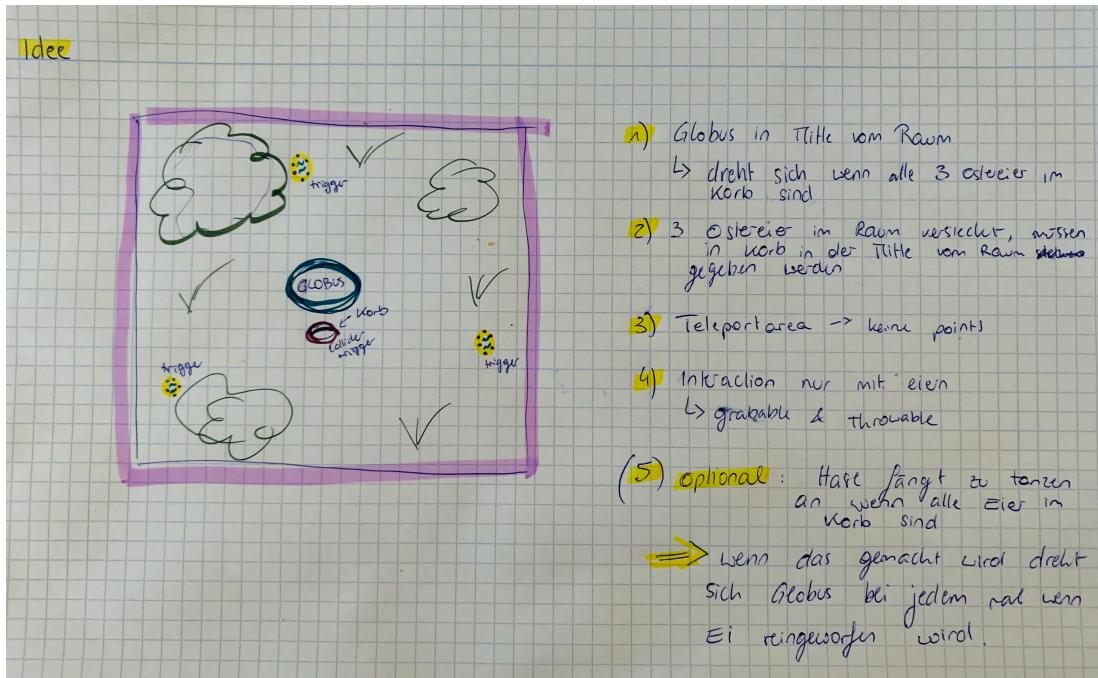


Abb. 1: Skizze mit Stichpunkten des benötigten Inhalts der Szene

Andererseits habe ich in diesen 3 Unterrichtsstunden auch die hauptsächlich benötigten Assets in der Szene platziert und somit den Raum, in dem sich der/die User\*in bewegen kann, grundsätzlich fertiggestellt.

Folgend habe ich dann mein Readme geupdated, hauptsächlich mit dem Bild meiner Skizze und den Links zu den Assets, welche ich bisher in meiner Szene platziert habe.

## 05.03.2020 - weitere Suche nach Assets

Ausserhalb der Unterrichtsstunden habe ich circa noch eine Stunde noch darin investiert, dass ich weitere Assets zusammensuche, die noch benötigt werden könnten. Ich habe unter anderem einen neuen Korb/ Behälter gesucht, in welchem die Ostereier dann platziert werden müssen, um gezählt zu werden.

Bei manchen dieser gefundenen Assets, welche ich von anderen Seiten wie Poly von Google oder Turbosquid runter geladen habe, gab es Anzeigefehler, welche ich leider

nicht lösen konnte. Aufgrund dessen musste ich weiter suchen, bis ich Assets ohne Anzeigefehler gefunden habe.

Danach wurde wieder das Readme geupdated mit den neu dazugekommenen Daten.

### **11.03.2020 - Anfänge mit SteamVR**

In dieser Unterrichtseinheit von 3 Stunden, sowie eine Stunde ausserhalb des Unterrichts, habe ich angefangen, SteamVR in mein Projekt zu implementieren. Dazu habe ich mir für die Anfänge ein Youtube Video (<https://www.youtube.com/watch?v=iJ0oNYIUFJo>) angesehen, um mein Wissen über SteamVR wieder aufzufrischen sowie um keine Fehler zu haben oder etwas vergessen.

Für die Teleport Area hebe ich meine Plane, welche ich schon in meiner Szene platziert habe, einfach dupliziert und das Teleport Area Script hinzugefügt. Zudem wurde ein Player in die Szene implementiert und angepasst, dass dieser, im Gegensatz zu den Objekten im Raum, nicht zu groß ist. Dies ist wichtig, dass das Game realistisch auf die Größe eines Menschen optimiert ist und der User somit nicht irritiert wird.

Leider kam ein Problem mit der Teleport Area auf. Diese wurde immer wieder pink und dies konnte nur behoben werden, wenn die Plane mit dem Script oben gelöscht und dann wieder eingebaut wurden. Jedoch war die Teleport Area immer wieder Pink, wenn das Projekt neu geöffnet wurde. Dadurch, dass einige Klassenkollegen/kolleginnen aus vorherigen Gruppen auch dieses

Problem hatten, habe ich zuerst bei Ihnen nach Lösungswegen gefragt, jedoch hatte von ihnen keiner dieses Problem lösen können. Nach weiterer Recherche konnte leider auch keine Lösung gefunden werden. Da die Area jedoch, so bald ich im Spiel auf Play gehe, wieder normal aussieht und nicht pink, habe ich mich entschlossen, es einfach dabei zu belassen, da es trotz der pinken Farbe ganz normal funktioniert hat. Falls es für den optischen Eindruck des Spieles störend ist, kann es zum



Abb. 2: Zwischenstand des Projektes

Schluss bei der Fertigstellung einfach ein letztes mal neu erstellt werden.

Weiteres wurden in diesen Stunden auch den Ostereiern das Interactable und Throwale Script angefügt, jedoch konnte es noch nicht in Virtual Reality getestet werden.

Zum Schluss wurde das Readme mit dem bisherigen Stand des Games aktualisiert, dazu wurde ein Screenshot aus der bisher bestehenden Szene angehängt. Dabei entstanden zuerst Fehler bei der Anzeige im Readme in Github, also das Bild wurde einfach nicht angezeigt. Ich dachte zuerst, dass beim Pfad etwas nicht stimmt oder ich mich beim Namen vertippt habe, jedoch lag der Fehler einfach darin, dass der Screenshot Ordner in meinem Projekt nicht mitgepusht wurde, weswegen Github den Screenshot logischerweise auch nicht finden konnte. Nach dieser Erkenntnis konnte das Problem gelöst werden.

### **18.03.2020 - Test im 2D Debug Modus**

Dadurch, dass wir nicht mehr in die Schule konnten und somit auch der Zugriff auf die HTC Vive oder die VR Oculus Rift nicht möglich war, musste das Projekt nun im 2D Debug Modus auf die Funktionalität getestet werden. In ca 4 Stunden Arbeit wurde das folgende gemacht:

Ich habe zuerst gegoogelt, ob es irgendwelche wissenswerten Dinge über den 2D Debug modus zu wissen gibt beziehungsweise, was in diesem Modus möglich war, da ich damit noch nie gearbeitet habe. Dies habe ich deshalb gemacht, da ich ja feststellen musste, ob mein Konzept auch im 2D Debug Modus umgesetzt werden kann. Dabei habe ich leider nicht sehr viele Informationen gefunden, weswegen ich einfach mal herum probiert habe. Ich habe darauf hin getestet, ob ich mich überhaupt in der Szene bewegen kann, was auch mit der Tastatur sehr simple funktioniert hat. Es kam jedoch zu dem Problem, dass meine Ostereier einfach durch den Boden gefallen sind, woraufhin ich einen Box Collider auf diese gegeben habe. Somit wurde dieses Problem gelöst.

Folgend, da ich die von SteamVR bereitgestellten Scripts den Ostereiern ja vorher schon angefügt habe, konnte ich testen, ob ich die Ostereier aufheben und mich damit in der Szene bewegen kann. Dies funktionierte optimal, ich konnte mich in der Szene mit den Eiern herumbewegen und in den vorgesehenen Kübel werfen. Dabei ist mir jedoch aufgefallen, dass der Kübel zu klein beziehungsweise die Ostereier zu groß waren und sie einfach wieder aus dem Kübel herausfielen. Daraufhin habe ich den Kübel vergrößert und einen Box Collider auf den Kübel gegeben, welchen ich der Form des Kübels grob angepasst habe. Der Box Collider wurde direkt auf Is Trigger gestellt, da im Hintergrund ja

die Anzahl der eingeworfenen Eier mitgezählt wird und bei allen gefundenen Eiern eine Animation ausgeführt wird.

Während ich das Projekt immer wieder während des Arbeitens auf GitHub gepusht habe, ist mir aufgefallen, dass die Packages auch immer mitgepushed werden, obwohl es in meinem Gitignore steht. Dieses Problem konnte jedoch sehr schnell behoben werden, da ich einfach einen Schreibfehler hatte. Leider ist mir dieser erst sehr spät aufgefallen, weswegen ich einige Zeit mit Recherche zu diesem Problem verschwendet habe.

Um die Plane nicht mehr verlassen zu können, habe ich Wände um meine Szene gebaut, dies hat jedoch nicht funktioniert. Ich wollte die Wände durchsichtig haben, weswegen ich zu Beginn einfach den Mesh Renderer bei den Wänden ausgeschalten habe. Dies hat aber auch nicht funktioniert, weswegen ich daraufhin ein Material erstellt habe, welches im Renderingmode Transparent und im Alphakanal vom Albedo auf 0 eingestellt ist.

Leider hat man danach immer noch durch die Wände gehen können, jedoch habe ich mich dann dazu entschlossen, es einfach zu lassen, da dies am 2D Debug Modus liegt und es zu aufwändig wäre, dieses Problem zu beheben, noch dazu ist es nicht relevant für die Umsetzung des Games. Weiteres ist das Spiel ja eigentlich für VR ausgelegt und die Teleport Area begrenzt den Bereich, in dem man sich bewegen kann, sowieso.

Zum Schluss dieser Arbeitsstunden habe ich ein Script erstellt und schon einmal angefangen zu recherchieren, wie man es programmiert, dass eine Animation startet, wenn alle Ostereier im Korb sind. Weiteres wurde noch das Readme aktualisiert.

## **13.04.2020 - Erstellung der Programmierung und Implementierung von Feinheiten**

Durch die Abgabe des Diplomprojektes und die benötigte Pause in den Osterferien wurde die Weiterarbeit am Projekt etwas nach hinten verschoben.

An diesem Tag habe ich angefangen, in meinem EggCollector Script zu arbeiten.

Angefangen habe ich mit der Erstellung meiner Globusanimation, wobei ich jedoch realisiert habe, dass bei dem Globus Asset, welches bereits in meiner Szene war, die Weltkugel kein separater Teil ist, weswegen ich es nicht direkt animieren konnte. Um dieses Problem zu lösen habe ich einfach eine Sphere erstellt, eine Map von der Erdkugel darauf gegeben und diese dann einfach über die andere Erdkugel, welche am Globus fixiert war, platziert.

Diese neu erstellte Sphere habe ich dann mit dem Animator in Unity animiert und es kam zu einem kleinen Problem. Da eine Kugel bei einem Globus ja etwas geneigt ist, habe ich

das bei meiner Sphere auch gemacht, jedoch hat dann die Animation sehr eigenartig ausgesehen also es hat sich sehr eigenartig um die eigene Achse gedreht. Dieses Problem habe ich dann so gelöst, dass einfach ein Empty Game Object erstellt wurde namens Globus, worin ich dann die Sphere gezogen habe und dann auf dem Empty Game Object die Neigung eingestellt habe. Dies hat dann wie gewünscht funktioniert. Ich habe also die Neigung nicht direkt auf der Sphere selbst gemacht, auf der Sphere war nur die 360 Grad Rotation.

Darauf hin habe ich eine OnTriggerEnter Methode in meinem EggCollector Script erstellt, dadurch sollten die Ostereier gezählt werden, die in den Pot geworfen werden. Dies ist durch den Is Trigger gestellten Box Collider auf dem Pot möglich.

In dieser Methode habe ich dann definiert, dass sich der Globus immer einmal um die eigene Achse dreht beziehungsweise, dass eben die vorhin angefertigte Animation ausgeführt wird, wenn ein Ei reingeworfen wird. Dabei kam jedoch das Problem auf, dass ich die Animation vom Globus nicht beim Script in der Console anfügen konnte, da es ein Clip ist und keine Animation. Dies konnte jedoch dann dadurch gelöst werden, indem ich einfach ein zweites Inspector Fenster geöffnet habe. In dem einen Inspector habe ich das Fenster vom Pot gelockt, darauf ist ja auch das EggCollector Script, und im anderen Inspector habe ich dann die Sphere mit der Animation darauf geöffnet. Von da wurde dann die Animation einfach in die globeAnimation des EggCollector Scripts im Inspector gezogen.

Danach wurde im Script definiert, dass die Animation des Hasen beginnt, wenn 5 Eier in den Pot geworfen wurden. Noch zu der Animation wird auch eine Audiodatei abgespielt und ein WinningText erscheint über dem Pot. Eigentlich habe ich in meinem Konzept nur geschrieben, dass sich der Globus rotiert wenn man alle Eier in den Pot geworfen hat, jedoch habe ich mich dann dazu entschieden, nach jedem Osterei „eine kleine Belohnung“ zu machen und am Schluss, wenn alle gefunden worden sind, gibt es dann quasi eine Tanzparty mit dem Osterhasen.

Um ein schummeln zu vermeiden, also das man theoretisch ein gefundenes Ei immer wieder in den Pot wirft bis man das Spiel gewonnen hat, wird das Interactable Script nach kollidieren mit dem Collider deaktiviert.

```
//cheaten nicht ermöglichen, Eier nicht mehr aus Pot nehmen können  
coll.GetComponent<Interactable>().enabled = false;
```

Abb. 3: Ausschnitt aus dem EggCollector Script wodurch Schummeln nicht mehr möglich ist

Nachdem die Programmierung eigentlich fertiggestellt wurde, habe ich eine Skybox zu meiner Szene hinzugefügt, um das Endresultat des Games etwas zu verschönern. Aus dem selben Grund habe ich dann auch den Post Processing Component geadded um auf einem Empty Game Object dann das Post Process Volume Script angefügt, einen Layer Namens Post Processing erstellt und diesem dem Empty Game Object angehängt. Während des Spielens ist immer wieder ein Error aufgetreten „ArgumentException: Object at index 0 is null“, jedoch konnte ich nicht herausfinden, wieso dieser Error auftritt. Ich habe die Lösung dieses Errors dann nach hinten verschoben, da das Game trotzdem einwandfrei funktioniert und es somit nicht höchste Priorität hatte.

Zu Beginn der Game-Entwicklung war vorgesehen, nur 3 Eier in den Pot werfen zu müssen, ich habe es jedoch dann auf 5 erhöht, da sonst das Game sehr schnell vorbei wäre.

Während des immer wieder Testens des Games kam schließlich ein weiteres Problem auf. Mir ist aufgefallen, dass, wenn man ein Osterei nicht von oben in den Collider fallen lässt, sondern quasi im Pot los lässt, wird das Osterei doppelt gezählt. Leider konnte ich dieses Problem nicht lösen, das einzige was ich umgestellt habe, ist dass in der OnTriggerEnter Methode nicht mehr eggCount == minEggs steht, sondern >=, dass wenn das letzte Ei versehentlich so wie oben genannt reingeworfen wird, man trotzdem gewinnen hat. Grundsätzlich sollten die Ostereier einfach von oben in den Pot fallen gelassen werden, dann kommt es auch zu keinem Problem, komplett behoben konnte es jedoch nicht werden.

Es ist kurz überlegt worden, eine Start und End Scene zu machen und dann ein SceneLoader Script zu erstellen. Diese Idee habe ich jedoch dann wieder verworfen, da es ja eigentlich ein VR Game ist und Buttons sowie Text in VR immer etwas schwierig ist. Ich habe jedoch schon grob eine Start Scene erstellt, werde sie aber nicht in mein Game einbauen.

```
//wenn 5 Eier im Pot dann epic party
if (eggCount >= minEggs)
{
```

Abb. 4: Ausschnitt aus dem EggCollector Script des If Statements

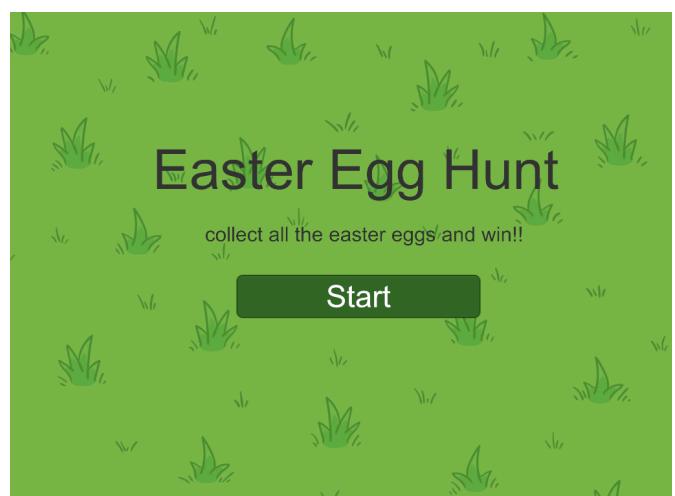


Abb. 5: Grob Angefertigte Start Scene

Dies alles oben genannte konnte in circa 5 stunden umgesetzt werden.

## 14.04.2020 - Finale Änderungen und Fertigstellung des Projektes

In die finale Arbeit an diesem Projekt habe ich schlussendlich noch ungefähr 1 1/2 Stunden investiert, um ein für mich optisch zufriedenes Endergebnis zu erreichen. Angefangen habe ich damit, einen neuen Wald zu suchen, da die bisher in der Szene platzierten Bäume eine Art Backface Culling auf den Blättern hatten, wodurch die Baumkrone von der Ansicht des Players nicht sichtbar war, nur von der Vogelperspektive aus. Deshalb wurde ein neues Asset gesucht und in der Szene platziert. Danach habe ich die 5 Ostereier neu versteckt, hierbei kamen kurz ein paar Probleme auf, da die Terrain des Waldes sehr uneben wars, ich konnte jedoch dann eine relativ ebene Fläche finden, wo ich dann alles plazierte. Daraufhin habe ich das Post Process Volume nochmal neu angepasst. Ich habe das Color Grading noch etwas verändert und habe Bloom sowie Depth of Field hinzugefügt. Final habe ich dann noch alle nicht benutzten Assets, Scenes, Scripts, Materials usw gelöscht, um mein Unity Projekt etwas aufzuräumen und nicht unnötig viele Daten auf GitHub zu pushen. Daraufhin habe ich das Readme mit den Links zu den neuen Assets, die Lessons learned und den Project State aktualisiert. Weiteres wurde ein Screenshot von der finalen Game Ansicht aus meinem Unity Projekt in meinem Readme hinzugefügt und somit das Projekt abgeschlossen.

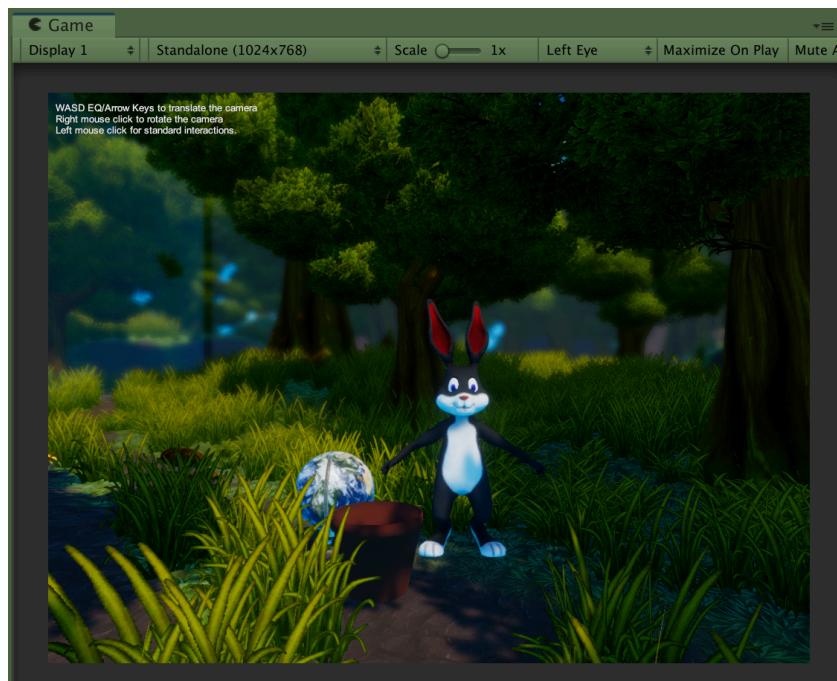


Abb.6: Finaler Screenshot des Games in Unity