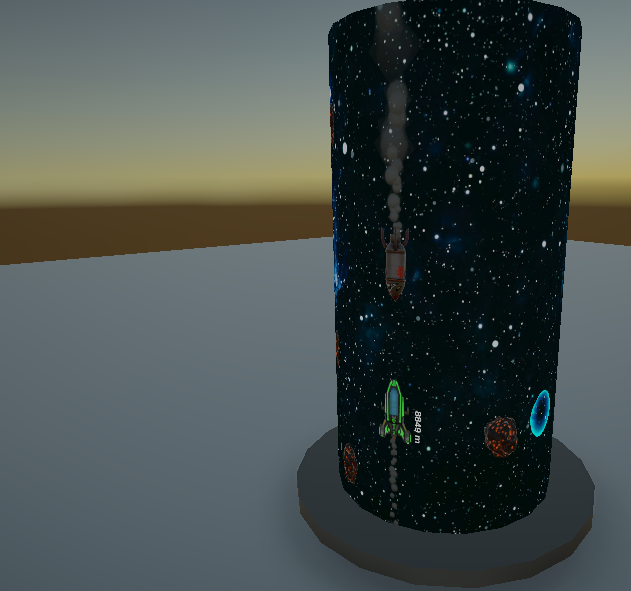
# VR-Prototyp

Um die Spiele mit VR-Controllern testen zu können und besser beurteilen zu können, ob diese Spaß machen und zur Bewegung um die Säule anregen, wurde eine LED-Säule in VR imitiert. Um den Kamera-Output als Textur auf den Zylinder legen zu können, wurde ein spezieller Shader implementiert.



# Spiele

#### Pong

Spielbeschreibung:

Die Spieler spielen sich gegenseitig den Pong ball zu. Je nachdem wo der Ball mit dem Paddle Kollidiert, entsteht eine „goal“ Zone, wo der ball nicht erneut hinein darf. Durch powerups kann die Goal Zone temporär ausgeschaltet werden, sodass der Spieler den Ball einmal herum spielen kann. Dies soll den Gegner verwirren. Es kann andere Powerups geben, die z.B. mehrere Bälle erlauben, oder dem Gegner einen Delay in den Controller einbauen.

Steuerung: Bewegung der X/Z Achse des Paddles per Direkt Ray pointer. Rotation vom Controller übernommen.

Spieleranzahl: 2 – mehrere

Interaktionsweise: Konkurrenz

Bewegung gefördert durch:

* Goal Zone bewegt sich mit dem Spieler um die Säule herum. Je nachdem wieviel Risiko der Spieler eingeht, kann er den Gegner verwirren.

#### Rocket Game

Spielbeschreibung:

Steuerung: Bewegung nur auf der X-Achse durch Rotation des Controllers, minimale zu maximaler Geschwindigkeit durch Rotation des Controllers von 0° bis 90° bzw. 0° bis -90°.

Spieleranzahl: 1 – mehrere

Interaktionsweise: keine

Bewegung gefördert durch:

* Ausweichen vor Meteoriten
* Ausweichen vor Raketen die die Spielerposition verfolgen
* Aufsammeln von Pickups

#### Shooter Game

Spielbeschreibung:

Steuerung: Bewegung direkt durch Closest Position auf der Säule zum Controller, Geschwindigkeit ist allerdings limitiert, Drehung der Zielrichtung durch Rotation des Controllers um 360°.

Spieleranzahl: 2 – mehrere

Interaktionsweise: Konkurrenz

Bewegung gefördert durch:

* Begrenzte Munition und somit Zwang zum suchen von Pickups
* Verstecken hinter Hindernissen auf der Map
* Raketen die Spieler automatisch verfolgen

#### Jump Game

Spielbeschreibung:

Steuerung: Jump-Player springt automatisch, Steuerung nur auf der X-Achse direkt bestimmt durch Closest Position auf der Säule zum Controller

Spieleranzahl: 2 – mehrere

Interaktionsweise: Kollaboration

Bewegung gefördert durch:

* Die Strecke zum nächsten Plattform-Nachladepunkt muss mit begrenzter Anzahl an Plattformen überwunden werden
* Der nächste Plattform-Nachladepunkt muss ggf. erst auf der Säule gesucht werden
* Hindernisse zwingen dazu, größere Sprünge zu vollziehen
* Sich auf festgelegten Strecken bewegende Sägeblätter erzeugen an bestimmten Stellen Zeitdruck

# 3D zu Flatscreen

Im Skript CylinderToFlatscreenPosition kann festgelegt werden, ob die Spielerposition durch Raycast-Direct oder ClosestPosition bestimmt werden soll.

Mit ClosestPosition erfolgt die Berechnung der X-Position auf dem Flatscreen durch prozentualen Winkel des Controllers in der realen Welt zu 360° und Berechnung der X-Koordinate auf dem Flatscreen durch selben prozentualen Anteil an der Flatscreenbreite. Die Y-Position und Z-Rotation werden eins zu eins vom Controller in der realen Welt übertragen.

Mit Raycast-Direct wird statt der Position des Controllers die Position auf der Säule, auf die der Controller zeigt genutzt.

Die Bewegung der Spielerfigur in der Spielwelt erfolgt durch den FlatscreenPlayerTransformHandler.

# Physics

Mithilfe des Scripts FlatscreenPlayerTransformHandler kann keine Physiksimulation genutzt werden, da dieses das Transform direkt manipuliert. Soll Physik genutzt werden, wird ein GameObject mit dem FlatscreenPlayerTransformHandler benötigt, um die Position vorzugeben. Die Spielerfigur kann dann mit dem PlayerPhysicsBodyTransformHandler diese Position unter der Nutzung von Physik verfolgen.

# Game Menu

#### Quick Menu

Durch Bewegung des Controllers zu einer Position nah an der Säule kann ein Quick Menu aktiviert werden. Dieses enthält Buttons für Continue, Restart und SwitchGame. Sobald das Quick Menu aktiviert wurde, erscheint ein Pointer, mit dem diese durch Berührung mit dem Pointer aktiviert werden können.

#### Switch Game Menu

Im Switch Game Menu werden die Spiele aufgelistet. Durch Berühren der StartButtons für die jeweiligen Spiele mit dem Pointer können die Spiele gestartet werden.

# Button Press Detection

Um Button Presses zu detektieren, wird der PrimaryButtonWatcher benötigt. Der Controller, dessen Button Presses detektiert werden sollen, kann mittels HashCode im PrimaryButtonWatcher festgelegt werden. HashCodes können z.B. mit Debug-Messages herausgefunden werden. Wird ein Button gedrückt, wird das primaryButtonPress-event im PrimaryButtonWatcher aufgerufen.

# Aufbau der Szene

Die Szene ist in mehrere „Generische“ Ordner unterteilt, und soll dem Entwickler in zukunft ermöglichen weitere Spiele für die 360° Säule zu entwickeln.

### General Game Setup

In dieser Gruppe befindet sich die übergeordnete Logik, also alles was der Spieler in jedem Spiel braucht, aber nicht sehen kann. Das heißt die 2D Representation vom Canvas mit dem Orthogonalen Kamerasystem und das Spielmenü. Hier muss der Entwickler nur das GameManager Skript auf dem Parent (General World Setup) anpassen.

### World (3D)

Diese Gruppe beinhaltet die Welt, wie sie der Spieler theoretisch in VR sehen würde. Also eine Plane als Boden und die Virtuelle 360° Säule. Mit dieser Welt wird beim testen interagiert.

Hier muss nichts angepasst werden, nur die Referenz „CalculationTargetCylinder“ ist ggf. interessant, da diese zur Berechnung in weiteren Skripten verwendet wird.

### XR Interaction Setup

Da das Spielsystem so Framework-Agnostisch wie möglich geplant ist, kann das XR Framework vom Entwickler bestimmt werden. Um zwar zum testen die Oculus Quests, aber produktiv die Vive Tracker benutzen zu können, wird hier das Open XR Interaction Toolkit empfohlen. Hiermit muss nichts im Prefab verändert werden, nur müssen später die Referenzen zu den jeweiligen Controllern in den Controller Managern eingetragen werden.

### 3DControllerManager

Die Controller Manager Prefabs folgen der Referenz (XR Controller Origin) vom XR Interaction Toolkit. Es wird die „stabilized“ Version der Controller Transforms empfohlen, da diese das Zittern der Hände herausfiltert, aber dennoch keine Große Verzögerung erzeugt.

### Games

Hier liegen die jeweiligen Spiel-Szenarien. Im GameManager müssen dafür Referenzen erstellt sein (ggf. umänderung im Code um n-Menge an Spielen laden zu können; Array etc.)

Um eigene Spiele mit der Codebase programmieren zu können, am besten einmal in die bereits vorhandenen hineinschauen.