# Umrechnung 2D - 3D

Cylinder-to-Flatscreen-Position

* Attach to VR Controller
* Computes the 2D-Position of the corresponding paddle on the flatscreen based on the 3D-position of the controller relative to the cylinder

Player-Position-Handler

* Attach to 2D-Player-Object (for example the paddle in a pong game)
* Gets the 2D-Position of the 2D-Player-object from the Cylinder-to-Flatscreen-Position Script that is attached to the corresponding 3D-player-object

# Pong Game

Ball-Controller

* Attach to Pong Ball
* Makes sure the ball gets teleported to the left side of the screen if it leaves the camera view on the right side of the screen to assure and vice versa to assure seamless movement around the cylinder

Player-Body-Controller

* Attach to Paddle
* The Paddle needs Physics which don’t work if the paddle is moved using the Player-Position-Handler. Therefore the object controlled by the player-position-handler shouldn’t use physics and be invisible in the scene. It will only be needed to tell the paddle where it SHOULD be. This script will then make the visible paddle that uses physics follow the position of the object that is controlled by the player-position-handler.

# Rocket Game

Metorie-Spawner

* Attach to any gameObject
* Spawns meteorites in a set number of positions with a set distance between them. The spawn time and position is controlled by a SpawnInfo Scriptable Object Instance that needs to be separately created and assigned to this script.