

# Simple Orbit Manager



Unreal Engine 4 - Marketplace Asset



## Simple Orbit Manager

# Documentation

Asset Version	1.2
Update of Asset	16.12.2020
Update of Documentation	16.12.2020
Marketplace Link	<a href="#">Assets by Divivor</a>
Developer and Creator	Divivor   Dennis Kastel
Creator Website	<a href="http://divivor.eu">divivor.eu</a>

# Contents of this Documentation

---

<b>Contents of this Documentation</b>	<b>2</b>
<b>1 Description</b>	<b>3</b>
1.1 Features and Details	3
<b>2 Getting started</b>	<b>4</b>
2.1 Setup	4
2.2 Parameters and Settings	10
2.2.1 Default	10
2.2.2 General Orbit Settings	11
<b>3 Additional Information</b>	<b>12</b>
3.1 File Structure	12
3.2 Changelog	13

# 1 Description

---

The Simple Orbit Manager (in short: SOM) generates Orbits for your Actors and Meshes. Create Orbits manually and attach your Planet or Object to it, or let it generate a random amount of orbits for your Actors or Meshes.

## 1.1 Features and Details

---

- Orbit Movement and Object Rotation
- Oval Orbits and Orbit Offset
- Randomized Spawners for Static Meshes and Actor Classes
- Multiple Orbits supported
- Indicator for Orbits. This can be disabled on the BP Instance.
- 100% Made in Blueprints

## 2 Getting started

---

This section explains how you can add and use the Orbit Managers.

**Pro Tip:** Press *T* in the Editor to disable/enable selection on transparent objects.

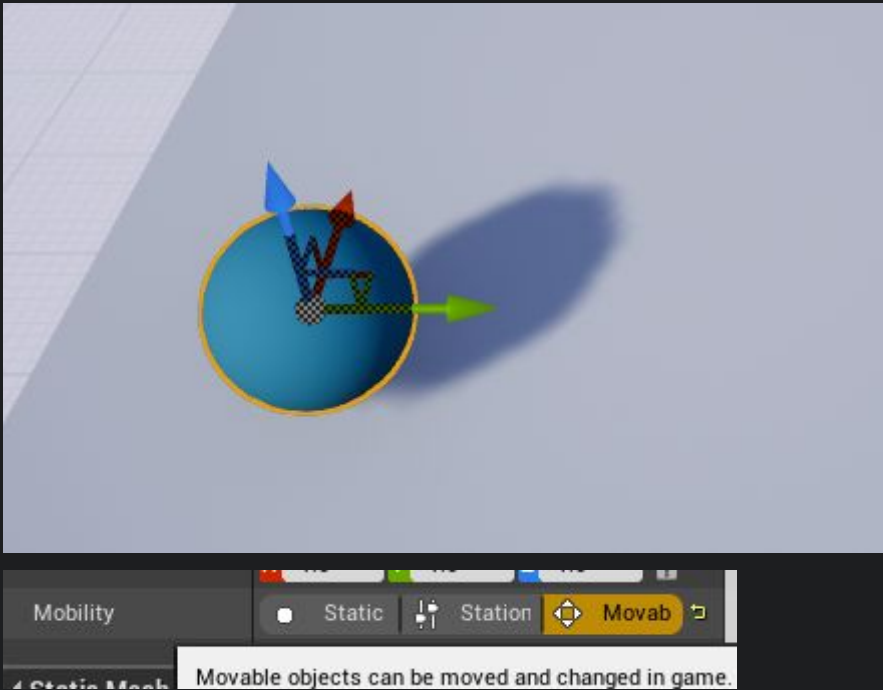
This is useful if you do not want to click on the visual Indicators of the Orbits all the time.

### 2.1 Setup

---

This is how you setup the Simple Orbit Manager in your scene. This tutorial is for creating normal Orbits for already spawned Actors, but can be applied to the Randomized Spawners.

1. Place your Planet/Object which you want to move around the Orbit into the Level. Make sure that the Mobility of your Object is set to Movable if it has the option.



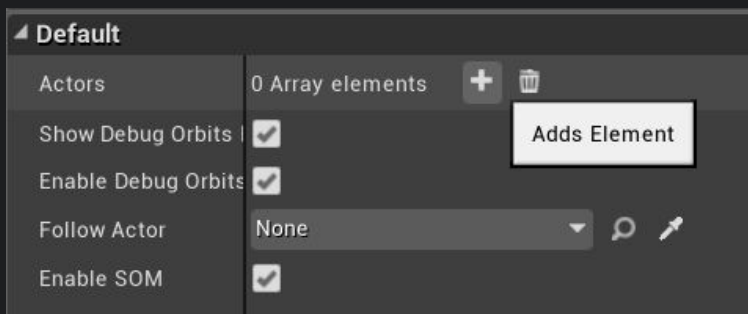
2. Place an Instance of the BP\_SOM\_ActorOrbit into your Level. It can be found under SimpleOrbitManager -> Blueprints -> BP\_SOM\_ActorOrbit.



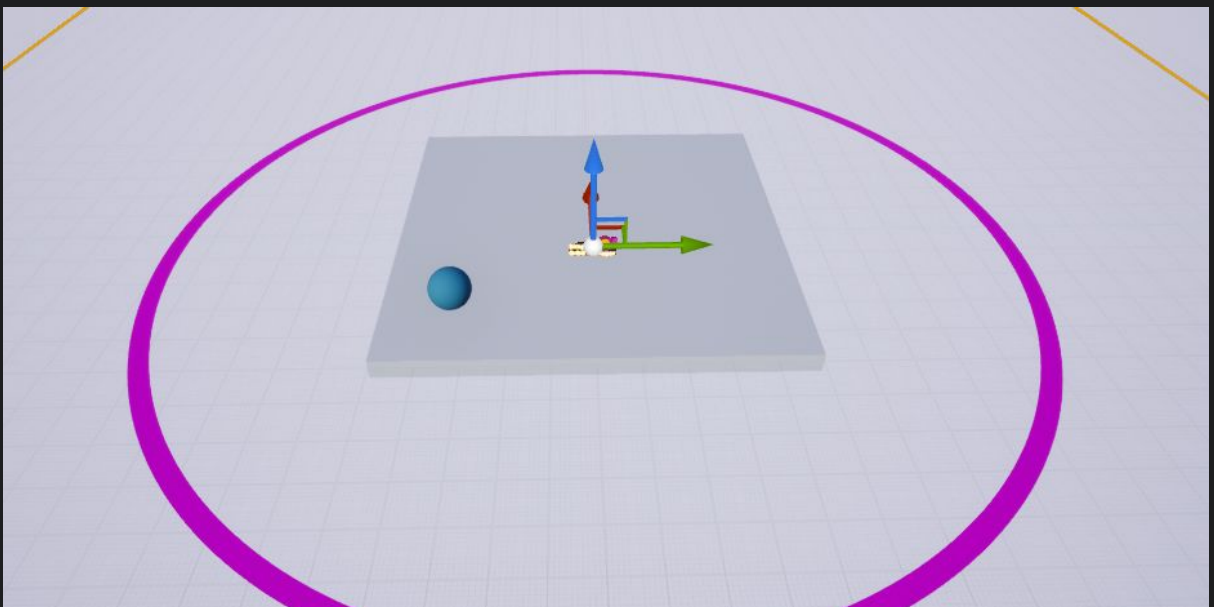


You should see the Logo of SOM at the Location where you have placed it.

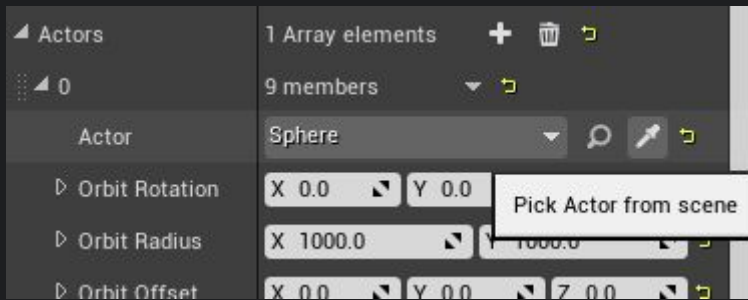
3. Select the SOM Instance and add a new entry into the “Actors” Array.



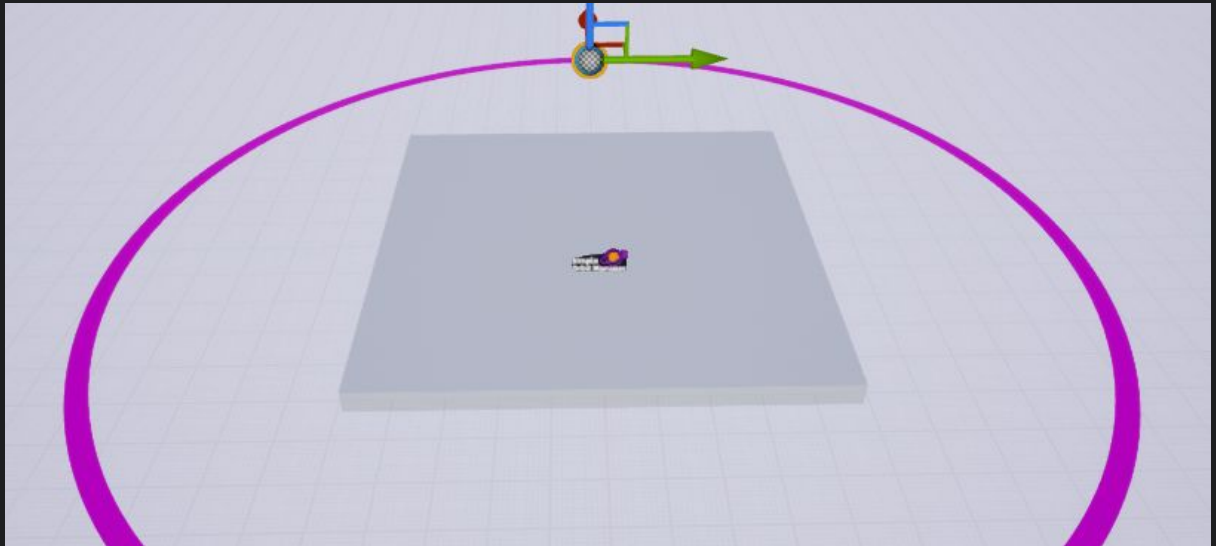
You should now see a circle indicator for the Orbit Radius.



- Now set the Actor Value to your Planet/Object which you want to move around.



It should automatically snap to the 0 Degree Position of the Orbit.



- Simulate your Level and you should see your Object Move around the Orbit, but it is very slow at the moment. Increase the Movement Speed of the Orbit to around 10 and simulate again

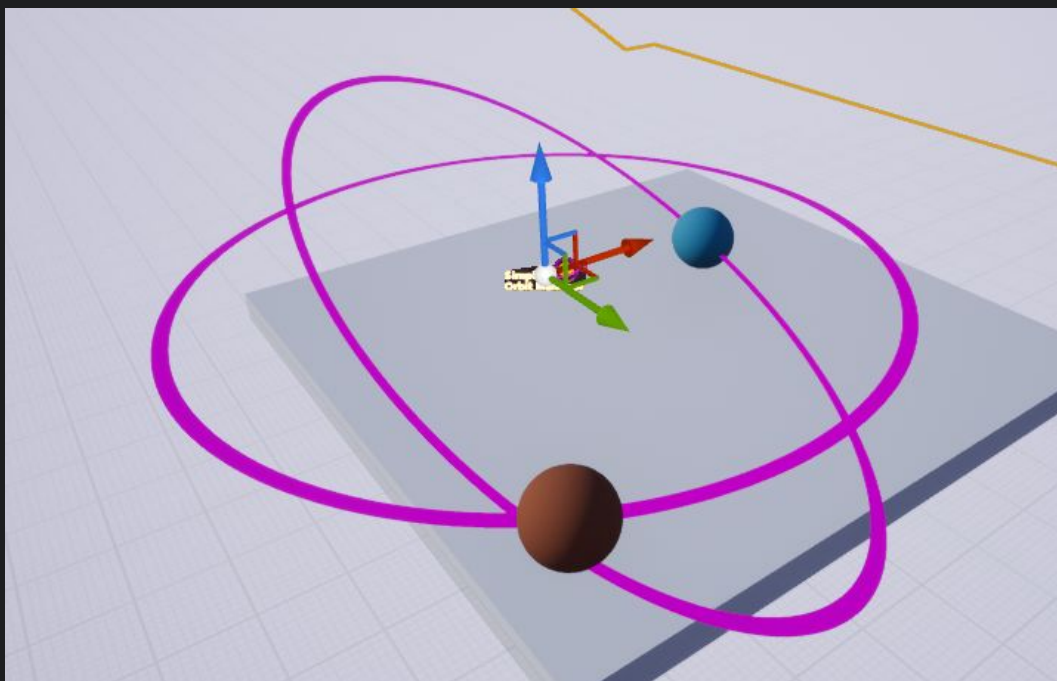


You should now see it fly around the orbit faster.

- Let's change the Orbit Size/Radius. You can create Oval Orbits bei setting different radius valius.

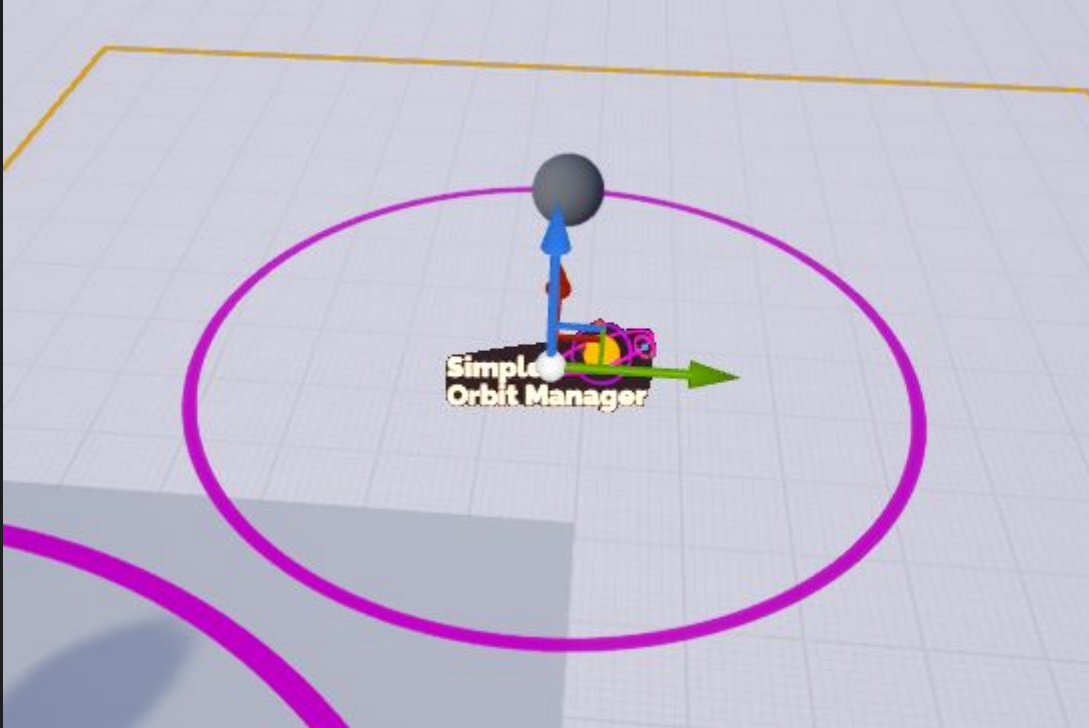


- Now we are adding a second Orbit to this SOM Instance. Simply add a new Entry to the Actors Array and adjust the settings and add another Planet Object.



You can also change to rotation of the orbits to create some variation.

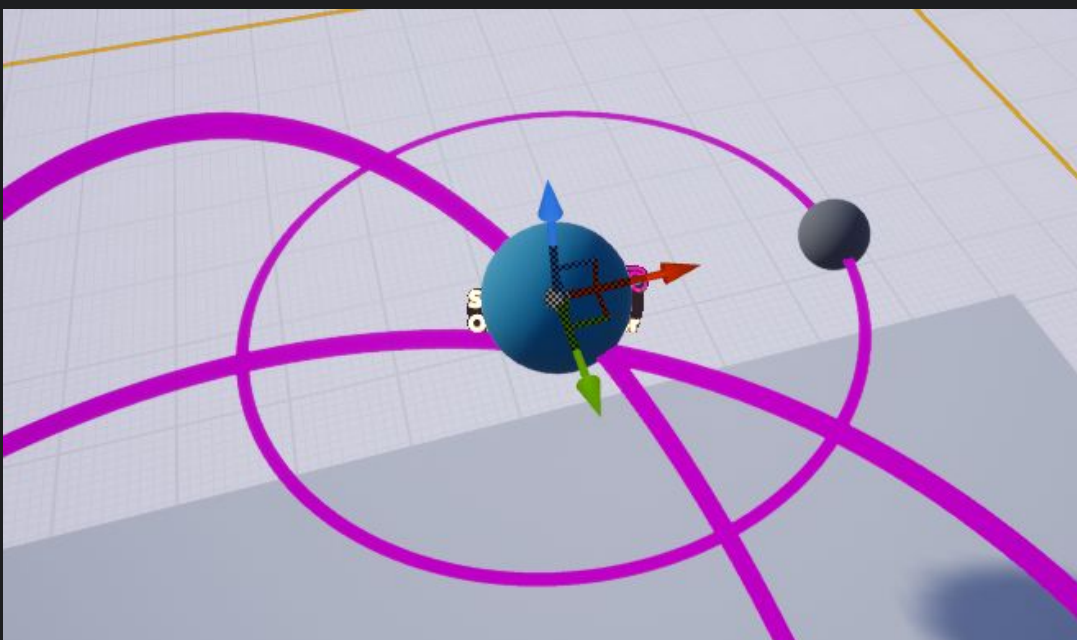
- Let's add a Moon to the Blue Planet. For this we have to add a new BP\_SOM\_ActorOrbit Instance into the Level. Add Moon Actor to it and edit the settings to make it look good as a Moon Orbit.



To add this SOM Instance to the Blue Planet we will have to set the Follow Actor of the SOM Instance to the Blue Planet Actor.



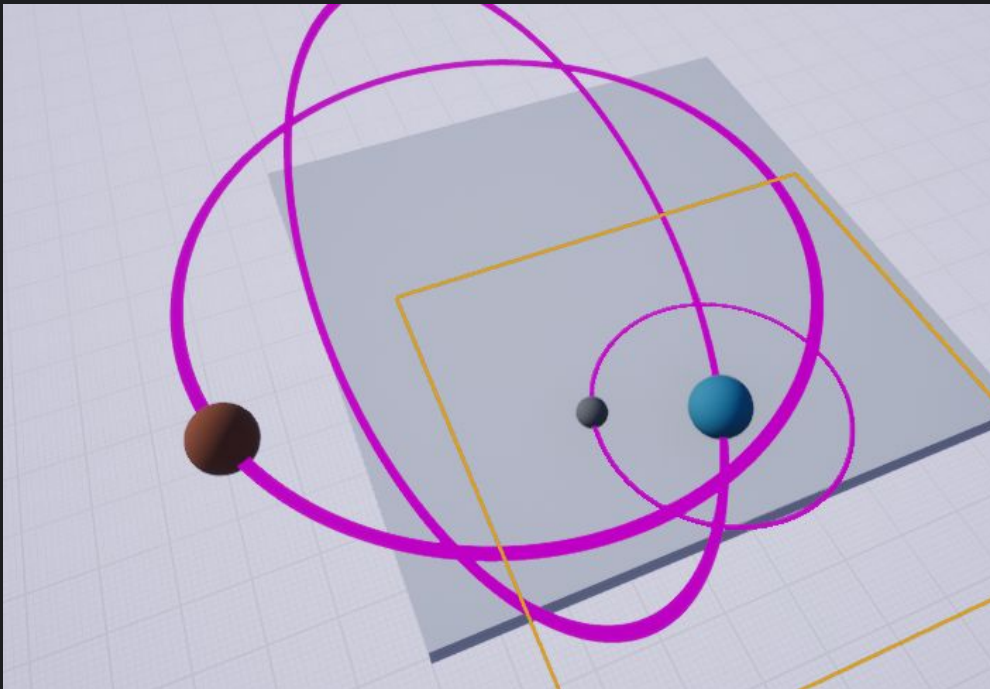
This will snap the Moon SOM Instance to the Blue Planet.



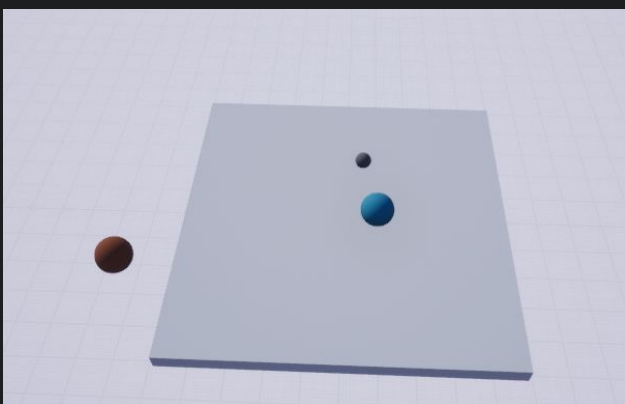
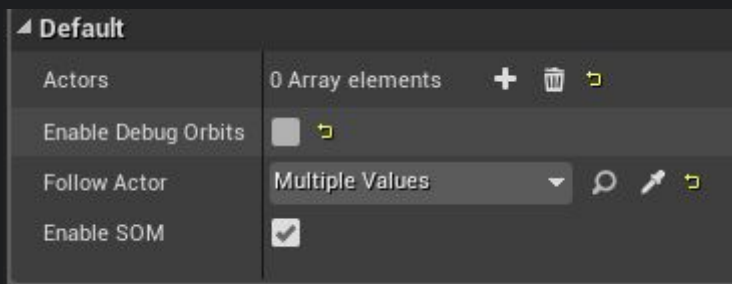


## Documentation of the Simple Orbit Manager

Simulate the Level again and you should see the Moon Orbiting around the Blue Planet, while the Blue Planet is moving around its own Orbit.



9. Lastly we will deactivate the Indicator Circles during Playing/Simulating. Disable “Enable Debug Orbits” and simulate again. The Circles should now have vanished.



You are now ready to use the Simple Orbit Manager in your Projects. For more information on the Parameters, read the Tooltips inside the Unreal Engine.

## 2.2 Parameters and Settings

---

This part of the documentation covers all Parameters and values that can be edited inside the details panel of the SOM Blueprints.

### 2.2.1 Default

---

- **Enable Debug Orbit**  
By enabling this option the debug orbits/rings will be visible during runtime. Make sure to disable this option, otherwise your players will see the rings during the game.
- **Follow Actor**  
This option allows the SOM Actor to follow another object. This is useful for orbits around other moving objects, for example Moon Orbits.
- **Enable SOM**  
This option enables all movement functionality.
- **Actors**  
Contains data for orbits inside the BP\_SOM\_Actors object. Similar to “Orbit Data”.
- **Static Mesh**  
The Mesh that is being used for objects within the BP\_SOM\_MeshOrbit.
- **Collision Type**  
Sets the Collision Type for the spawned Instanced Static Meshes. Used in BP\_SOM\_MeshOrbit.
- **Orbit Data**  
Contains orbit data for BP\_SOM\_MeshOrbit.
- **Cast Shadow**  
Can enable or disable Shadows on BP\_SOM\_MeshOrbit.
- **Override Materials**  
Can override materials of the Meshes on BP\_SOM\_MeshOrbit.

## 2.2.2 General Orbit Settings

---

This contains parameter information that can be applied to all SOM Actor types. The minimum and maximum options of the Random Spawners generate a random value in range for the according parameter type.

- **Orbit Rotation**  
The Rotation of the orbit itself.
- **Orbit Radius**  
The radius of the orbit. There is a X and a Y option for the radius, if you set them to the same value you will get a circle. If you use different values for X and Y you can make elliptic/oval orbits
- **Orbit Offset**  
With this option you offset the center of the orbit, relative to its current rotation.
- **Start Angle**  
This specifies the starting angle of your objects on the orbit.
- **Movement Speed**  
The movement speed around the orbit in degrees per second.
- **Default Rotation**  
The default rotation of the Object that is being placed in the orbit.
- **Rotation Speed**  
Additional rotation speed of the moved object in degrees per second. For example a rotation asteroid that is also moving around an orbit.
- **Rotate relative to Orbit**  
This option transforms the rotation of the object to be relative to the orbit.
- **Default Scale**  
The default scale of the moved object.

## 3 Additional Information

---

### 3.1 File Structure

---

- | **SimpleOrbitManager**
  - | **Blueprints**
    - | - **Core**
      - Contains the Core Blueprint Files of this project
  - | **Demo**
    - | - **Blueprints**
    - | - **Materials**
    - The demo folder contains examples for the orbits and a demo map. You can simply delete the “Demo” Folder if you don’t need the contents of it.
  - | **Materials**
    - Contains all the used materials for the visual Indicators.
  - | **Meshes**
  - | **Textures**
    - Currently only contains the Logo Image.

## 3.2 Changelog

---

### 1.2 (December 2020)

- Restructured Files
- New Documentation
- Added options to change materials and shadows on Mesh Spawners
- Changed absolute positions to relative positions

### 1.1 (May 2019)

- Added a "Show Debug Orbits in Editor" functionality

### 1.0 (January 2019)

- Original Release