

AltSpaceVR plugin for Unity3D

Documentation

Version: 0.5

Thank you for interest in the AltSpaceVR plugin for Unity3D. This documentation will provide you with detailed information about the plugin. This plugin you should think of as an interpreter. It doesn't translate any C# code from Unity to Aframes, it interprets the gameobject parameters into proper aframe entities.

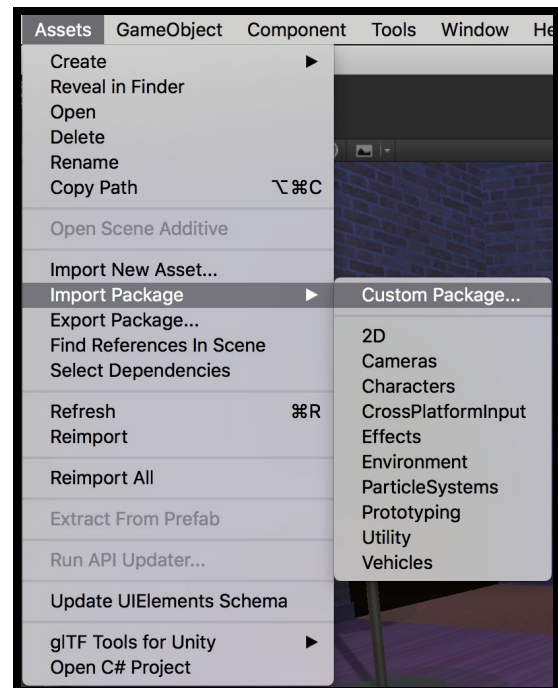
Getting Started

Download the "UnityPackage" from "AltSpaceVR plugin for Unity" Github:

<https://github.com/amadden1990/AltSpaceVR-Unity-SDK>

Open Unity and "Import" the "UnityPackage" into your project by clicking on "Assets > Import Package > Custom Package..." from the top bar. Select the "UnityPackage" from your documents.

You will be prompted with a window that shows all the contents in the "UnityPackage". Make sure everything is selected and click "Import".



Learning the Editor

AltspacVR

AltspacVR

Latest AltspacVR SDK:
`https://sdk.altvr.com/libs/altspac.js/2.9.0/altsp`

Latest AFrame SDK:
`https://aframe.io/releases/0.7.0/aframe.min.js`

Hosting URL

Author's Name

Project Name

Create a new entity.

- a-cube
- a-sphere
- a-cylinder
- a-plane
- n-portal
- n-portal destination
- browser
- enclosure
- resources

Entity Tools

- Clone Selected Entities
- Add Entity Animator To Selected Entities
- Convert Selected to glTF
- Register New Aframe Component

Select project export directory...

Browse

BUILD PROJECT

To open the editor, click on “Window > AltspacVR”

1. Starting from the top is the latest AltspacVR SDK. This is autofilled and shouldn't be changed unless you know what you are doing.

2. **Latest Aframe SDK** - At the release of this version, Aframe 0.7.0 is the most recent compatible version for AltspacVR.

3. **Hosting URL** - Here you would paste the url that you are going to host the project. This is only necessary at the moment for direct references to your GLTF 3D objects, only if you are using the generic GLTF object loader and not Altspac's.

4. **Author's Name** - Type in your name.

5. **Project Name** - Type in your project name.

6. **A-cube** - Create a new cube entity.

7. **A-sphere** - Create a new sphere entity.

8. **A-cylinder** - Create a new cylinder entity.

9. **A-plane** - Create a new plane entity.

10. **N-portal** - Create a new AltspacVR portal.

11. **N-portal destination** - Create a new destination for a existing portal in your scene.

12. **Browser** - Create a new AltspacVR 2D browser enclosure.

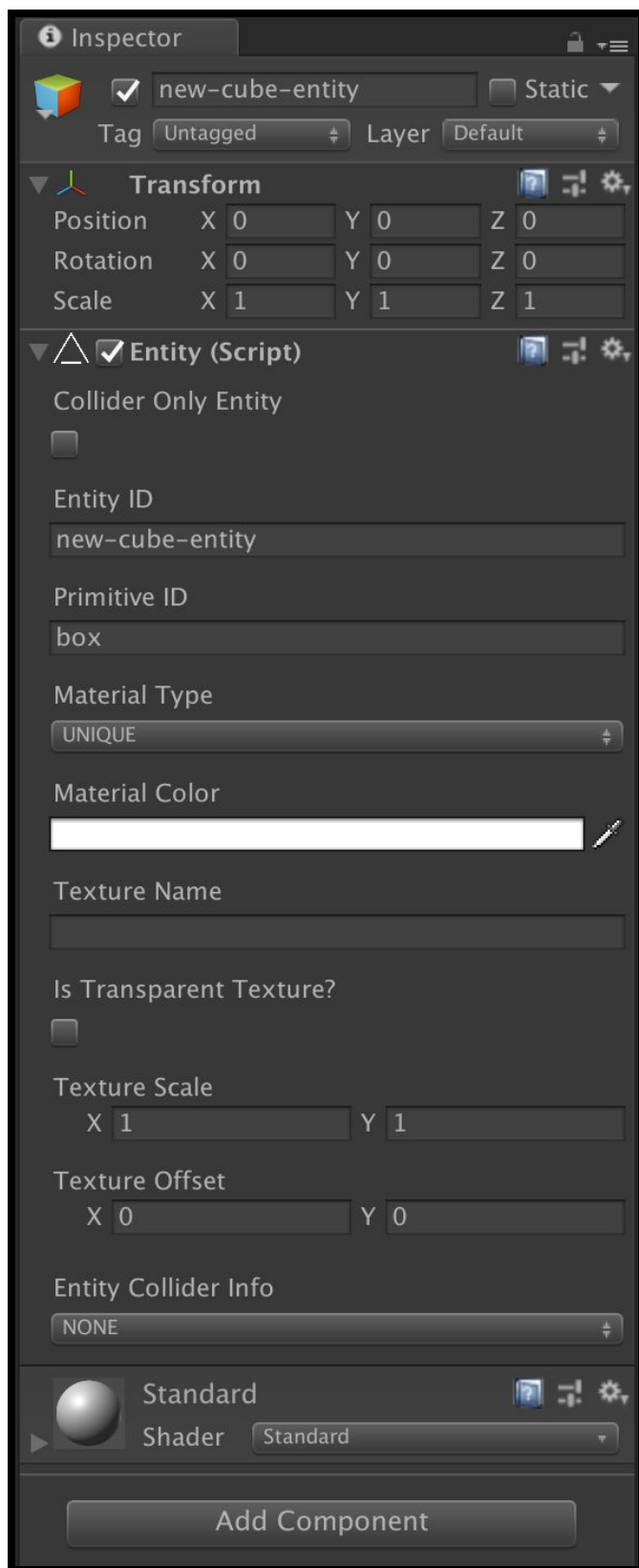
13. **Enclosure** - Create a new AltspacVR 3D browser enclosure.

14. **Resources** - Create a new AltspaceVR native resource. Choose between “Effects, Interactables, and Objects”.
15. **Clone Selected Entities** - Select one or more entities and clone them. This differs from the native Unity duplicate shortcut. When each entity is cloned with this button, a new material is applied to the newly created entity.
16. **Add Entity Animator To Selected Entities** - Select one or more entities and add a tween animator component to them.
17. **Convert Selected to glTF** - Select one or more gameobjects, not entities, and add a gltf component to them. This gameobject cannot have a entity component attached. Select between aframe’s generic glTF object loader or AltspaceVR’s glTF object loader.
18. **Register New Aframe Component** - Select one or more entities and add the “RegisterAframeComponent”. This will allow you to write javascript to add custom functionality to the selected entities.
19. **Browse** - Select the directory you wish to save your project.
20. **Build Project** - Build the project.

The Entity Component

The entity component is the translator that translates the Unity values into Aframe code. This needs to be added to every primitive shape being exported with the overall project. The entity component comes with a variety of parameters:

- **Collider Only Entity** - This checkbox allows you place gameobjects around your scene that acts as a collider once exported. The mesh will not be visible upon export. The mesh in the editor is only visual representation of what the collider looks like. **This feature is only available for a-cube entities.**
- **Entity ID** - This is the name of your entity.
- **Primitive ID** - This is the name of the primitive you are creating. Do not change the default value unless you know what you are doing.



- **Material Type** - There are three material types; Unique, New Mixin Material, & Use Mixin Material.

- **Unique** - Create a brand new material.

- **New Mixin Material** - Create a brand new material that can be referenced by other entities.

- **Use Mixin Material** - Type in the Mixin ID to use an available Mixin Material.

- **Mixin ID** - The id applied to the mixin material. This is set by a New Mixin Material, and referenced by entities that set their Material Type as Use Mixin Material.

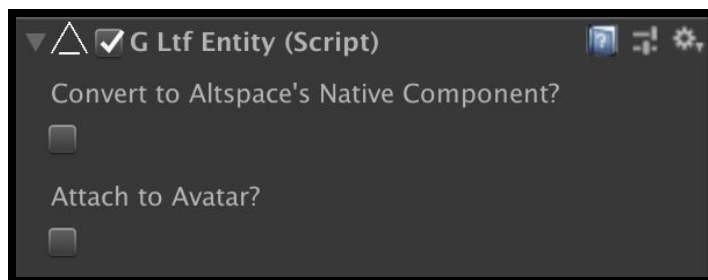
- **Material Color** - The color of the material.

- **Texture Name** - The full name of the texture you are applying to the entity. This includes the correct spelling, and the correct file extension. **This is case-sensitive.**

- **Is Transparent Texture?** - Is the applied texture transparent?

- **Texture Scale** - The texture scale of your texture.
- **Texture Offset** - The texture position offset.
- **Entity Collider Info** - Contains four colliders:
 - Mesh
 - Box
 - Capsule
 - Sphere
- **Entity Collider Type** - Contains three types:
 - **Object** - Collide with other objects, the environment, and the cursor.
 - **Environment** - Collide with everything object does and the avatar.
 - **Hologram** - Collide with other holograms and the cursor.

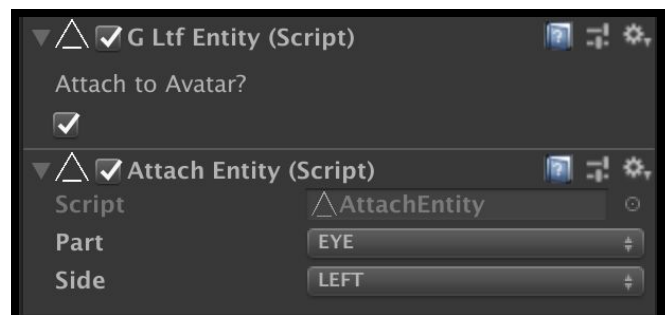
Custom 3D Objects



If you want to export a custom model that you created using either Unity's Probuilder or from a 3rd party program like 3d Max, Maya, Blender, etc, you will want to convert the selected gameobject to a gltf object. Once you add the GLTF Entity Component to your custom 3d object, you are given

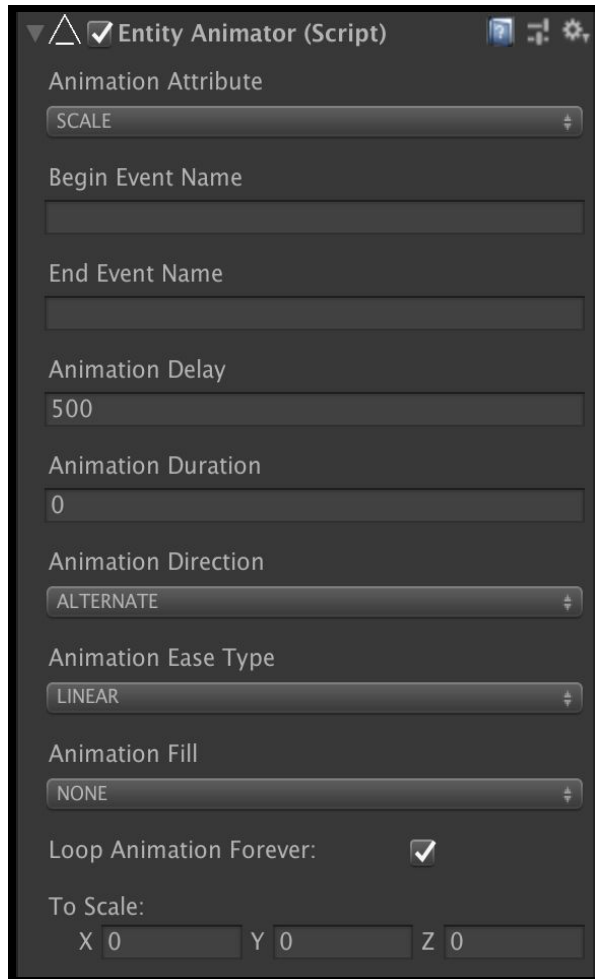
two boxes:

1. Convert to Altspace's Native Component
 - Will use Altspace's gltf object loader if checked. If left unchecked your object will use the default Aframe gltf object loader.
2. Attach to Avatar - Defaults the object to use the Aframe gltf object loader as well



as provides you with a list of body parts to attach the 3d Object to.

Animation Tweens



If you want to add any animation tweens to you entities (primitive only at the moment) you will need to add a Entity Animator to your primitive entity. To do this, select the intended entity and click “Add Entity Animator To Selected Entities” from the AltspaceVR window. Here you are able to animate the position, rotation, scale, color, and the visibility. For instructions about the animator refer to the link below:

<https://aframe.io/docs/0.7.0/core/animations.html>

Register New Aframe Components

This is one of the best new features added to the plugin. Here you are able to incorporate custom javascript snippets or files and register them as aframe components. After you are able to apply them to your entities within the scene (primitives only at the moment). To read up on registering aframe components click on the link below.

<https://aframe.io/docs/0.7.0/core/component.html#register-a-component>

At add this feature to an entity, select the intended entity and click on the “Register New Aframe Component” from the AltSpaceVR window. After doing so you are presented with a couple of fields and components:

1. **Register Component Name** -

This is the name of the component that the entity will use to access your custom component.

2. **Javascript File** - Attach a

javascript file you wish to add to your entity. You can import javascript file with a “.js” extension but Unity will throw an error, but you can ignore it. If you want to avoid the error text you may also import your javascript files as a “.txt” file.

3. **Load Button** - Load the Attached Javascript file onto the entity.

4. **Clear Button** - Clear the

Javascript file from the entity as the text from the text area below.

5. **The Text Area** - This is the text area that will populate when the load button is pressed after the javascript file has been serialized into the editor. This space also doubles as a in-editor text editor. You are able to write code directly into the area without loading a javascript file if you so choose.

