

STEIN LUMINARY

K-12

DOCUMENTATION

Marc Frances

Peter Davies

Overview

The Luminary K-12 project aims to implement augmented reality into the Luminary as a way to motivate K-12 students to engage with the art and learn to use their imagination to create new ways to imagine art and the possibilities thereof. This project is the start of many in order to create a way to use the Luminary space for educational and creative growth for students to utilize.

The first project displayed is the Art To Life project, using animations to mimic the pieces of the painting to make the art look like it is coming to life. This process included Animators to make the animations, either by drawing individual frames on a separate layer over the original painting in Photoshop(1), or cutting out pieces from the art, masking the background, and utilizing the Puppet Warp Tool in Photoshop to manipulate the objects on new layers(2). The animators then compiled all animations into one mp4 that was sent to the programmers to apply onto the Vuforia Image Target.



The second project was the Extended Frame project, which allowed for the viewer to see objects outside of the original frame of the object and all the mind to wonder what else could be going on outside of the frame of the painting. This required a 2D artist to create a piece of art that matched the original painting, though outside the borders, to create content for the students to explore.

VIDEO

- <https://drive.google.com/file/d/1KOjWBeUykqRNPE5N7cd0FFRWjwXA4QTF/view?usp=sharing>



Unity

- **Vuforia Version - 10.4.4**
- **Unity Version - 2020.3.11f1**
 - Add Android, IOS, and Universal Windows Platform Build Support in the modules for cross platform.
 - Cross Platform Support (Currently configured for IOS)
- **Mixed Reality Toolkit - 2.7.3.0**
- **XR Plugin Management - ARKit (IOS), ARCore (Android),**

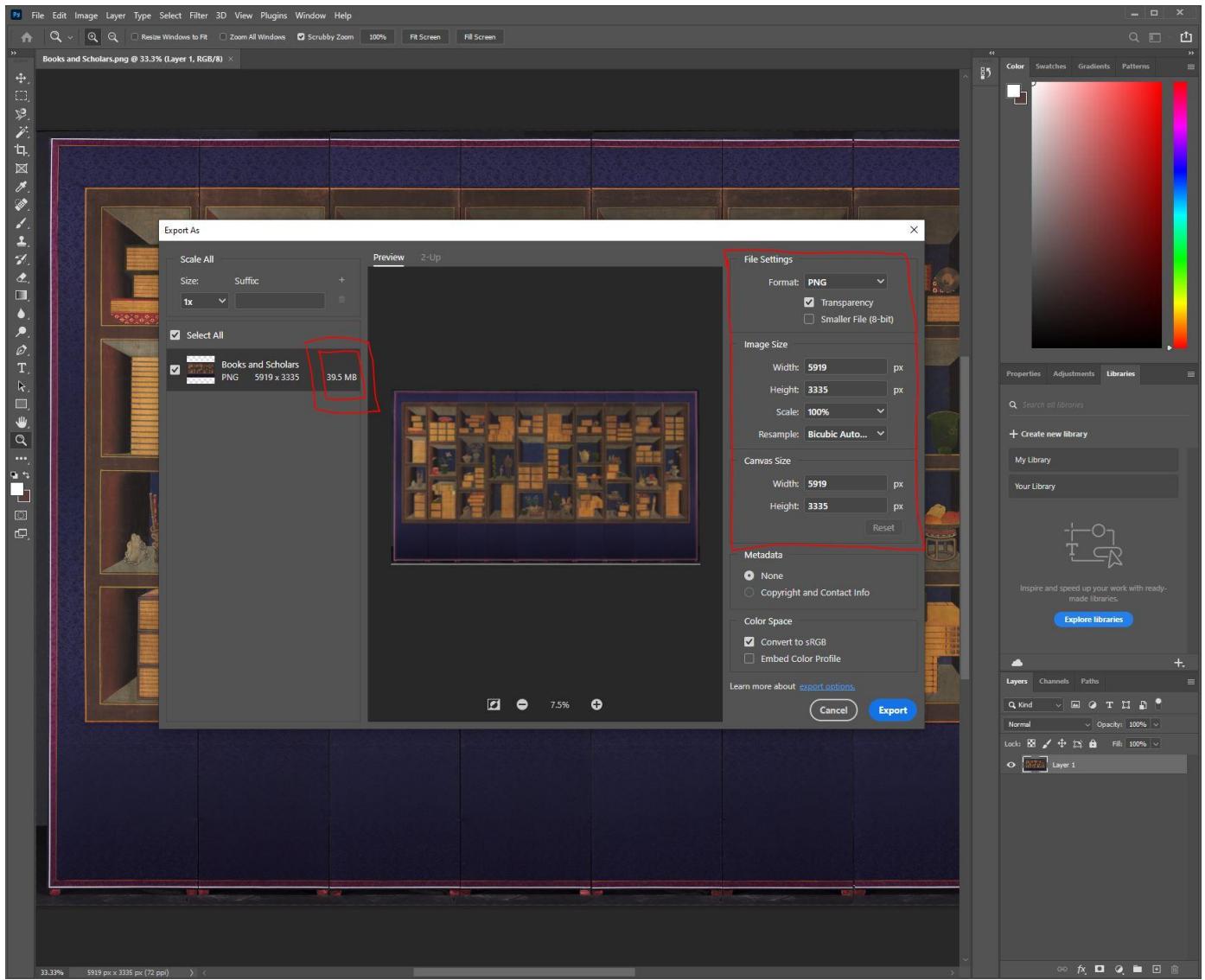
Installing Vuforia Engine

- Great overview from installation to configuration
 - <https://www.youtube.com/watch?v=-QlbDC6wdb8>
- Vuforia Version - 10.4.4
 - IMPORT THIS PACKAGE INTO YOUR PROJECT
 - <https://drive.google.com/file/d/1F1yN0tccXhTcIEIXXR7BC70D6XIK9F1O/view?usp=sharing>

Creating Image Targets With Vuforia

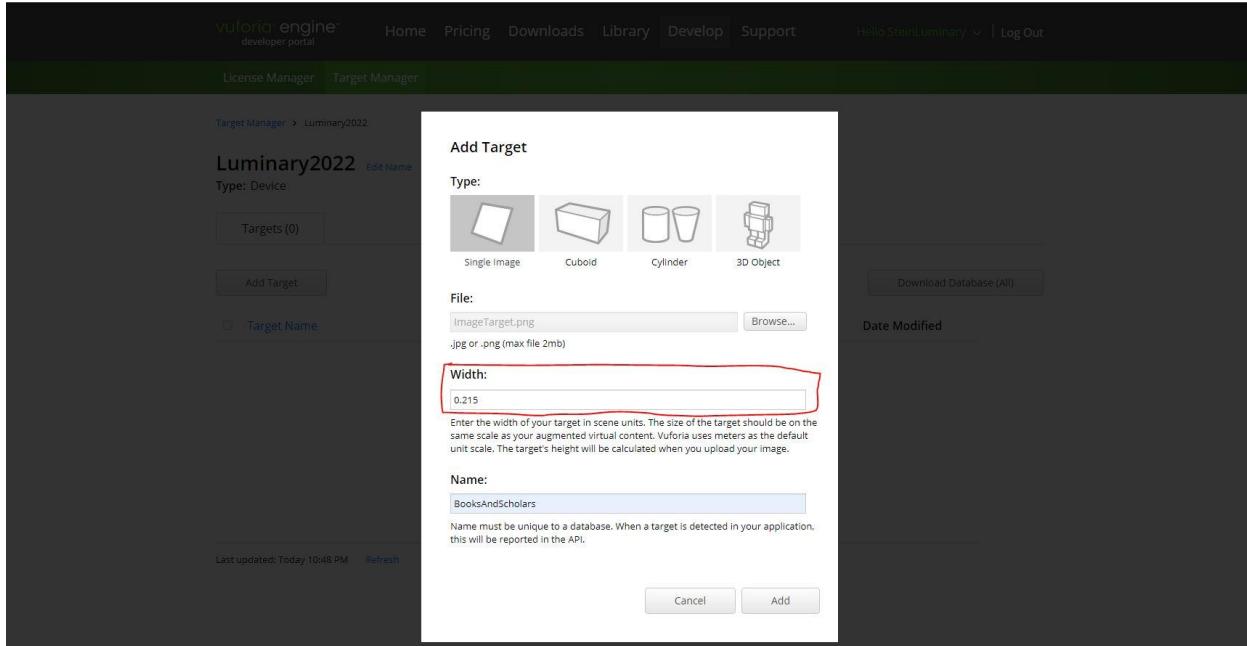
- <https://library.vuforia.com/features/images/image-targets/best-practices-for-designing-and-developing-image-based-targets.html>
- **Must be 8- or 24-bit PNG and JPG formats**
- **Must be less than 2MB in size**
- **JPGs must be RGB or grayscale (no CMYK)**
- -----
- This project uses the Vuforia Engine API to detect image targets in Unity.
 - Login to Vuforia with credentials saved by the team

- Copy the License key and paste it into the Vuforia configuration window in Unity.
- Using Adobe Photoshop to compress image targets into Vuforia formatting.
 - The source images we are receiving will most likely be larger in file size and in resolution.
 - In Photoshop, you can export an image and start playing with settings with a live view of your new file size.



- If you still can't get below **2MB**, you can go into Photoshop, navigate to **Image>Mode>Grayscale**. This will dramatically drop file size without compromising detail.

- You can realistically use Grayscale for everything and preserve a ton of detail since vuforia does not look for color
- Be careful when making them too small. You need to preserve as much detail as you can for the best possible tracking.



- Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses **meters** as the default unit scale. The target's height will be calculated when you upload your image. (for this project, I stuck to a width of **0.215 meters**)

The screenshot shows the 'License Manager' section of a web application. At the top, there are tabs for 'License Manager' and 'Target Manager'. Below the tabs, a heading says 'License Manager' and provides a link to 'Learn more about licensing.' A sub-instruction says 'Create a license key for your application.' There is a search bar labeled 'Search'. A table lists one license entry:

Name	Primary UUID ⓘ	Type	Status ⓘ	Date Modified
SteinLuminary	N/A	Basic	Active	May 02, 2022

At the bottom of the table, there are pagination controls: '25 per page' (with a dropdown arrow), 'Showing 1-1 of 1', and navigation arrows. The status bar at the bottom indicates 'Last updated: Today 8:59 PM' and includes 'Refresh' and 'Log Out' links.

The screenshot shows the 'Target Manager' section of the application. The top navigation bar includes 'Home', 'Pricing', 'Downloads', 'Library', 'Develop', 'Support', and user-specific links 'Hello SteinLuminary' and 'Log Out'. The main content area shows a target named 'Luminary2022' (Type: Device) with one target named 'BooksAndScholars'. On the right, there is a 'Download Database (All)' button. A modal dialog titled 'Download Database' is open, showing '1 of 1 active targets will be downloaded' and 'Name: Luminary2022'. It asks 'Select a development platform:' with options for 'Android Studio, Xcode or Visual Studio' and 'Unity Editor' (which is selected). The 'Download' button is highlighted with a red box. The status bar at the bottom indicates 'Last updated: Today 10:55 PM' and includes 'Refresh' and 'Log Out' links.

- Download the database. You should receive a Unity package. Save this for later

License Manager Target Manager

License Manager > SteinLuminary

SteinLuminary

Edit Name Delete License Key

Please copy the license key below into your app

```
Ayokn7j////////AAABmR8OCEmcAk2visiL+UQgrxFMseOLuPfSa13ebwAnPs1KnSluS/uNmYULzj3Pr9ixtYBekSkB2AfCu75%JKCWLau
j9997Ra4hKqIVWV799e6+1sqgh+2wf5D3ceylRNy6nhUOmZ7YlmcAyhV11ZNg0pxrrM06tfP4Cz7DCgFh1/u0Opjp2nH17A8gX/QF+
1sknpD24Cm7d13v3K0k6z7F8tch:2yMt1z8D1WA9+vmlNSTM16L59XpRGW9SX0btz+xQTHq9nZ60g1Z0Q3BnwgM7TJV/x64nbHaWsqF
MGa77022w2MUIPLC9kf2VSnDetFl1yPmcSt1UtWaKdHZyRcyD/a8xVldxD8p/FoxifeKTGZ
```

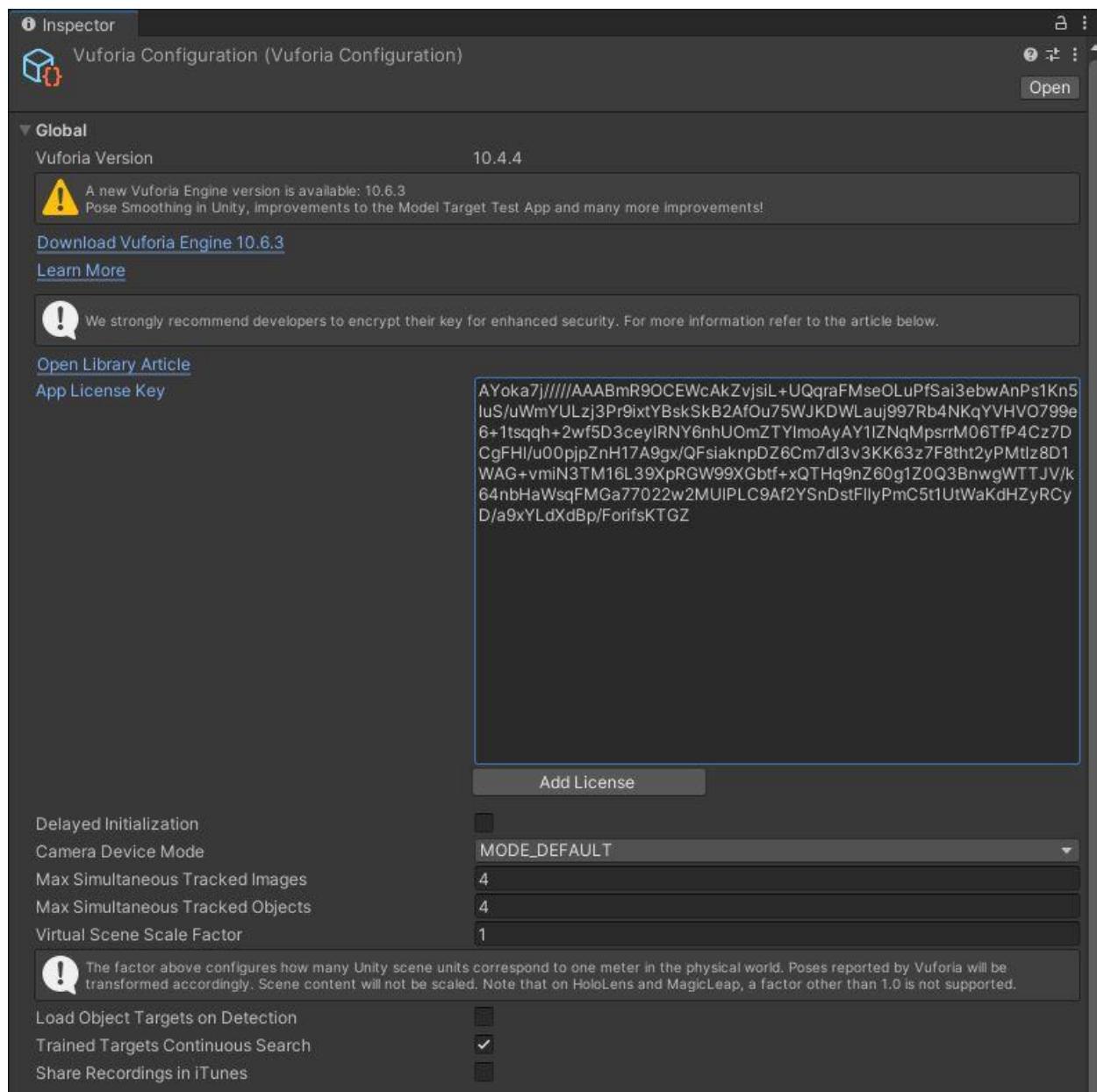
Plan Type: Basic
Status: Active
Created: May 02, 2022 20:59
License UUID: 2b6094f72b7c446086a76289ef338d04

Permissions:

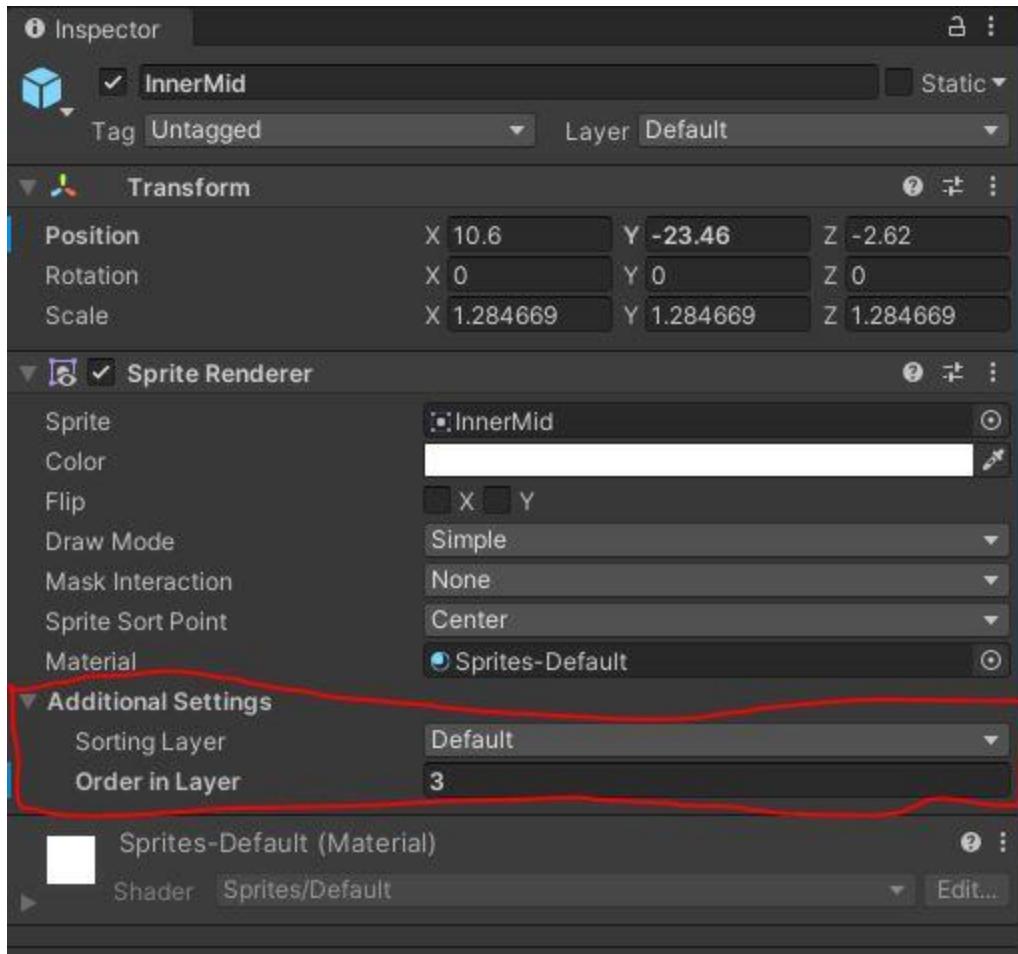
- Advanced Camera
- External Camera
- Model & Area Targets
- Watermark

History:
License Created - Today 20:59

- Find the App License Key and copy it.



- After installing Vuforia and configuring your App License Key. Import the unity package you downloaded from the portal.



- Sprite Layering in Unity is important for this project. It's similar to layers in Photoshop. It's just a parameter you can edit in the component of the sprite.

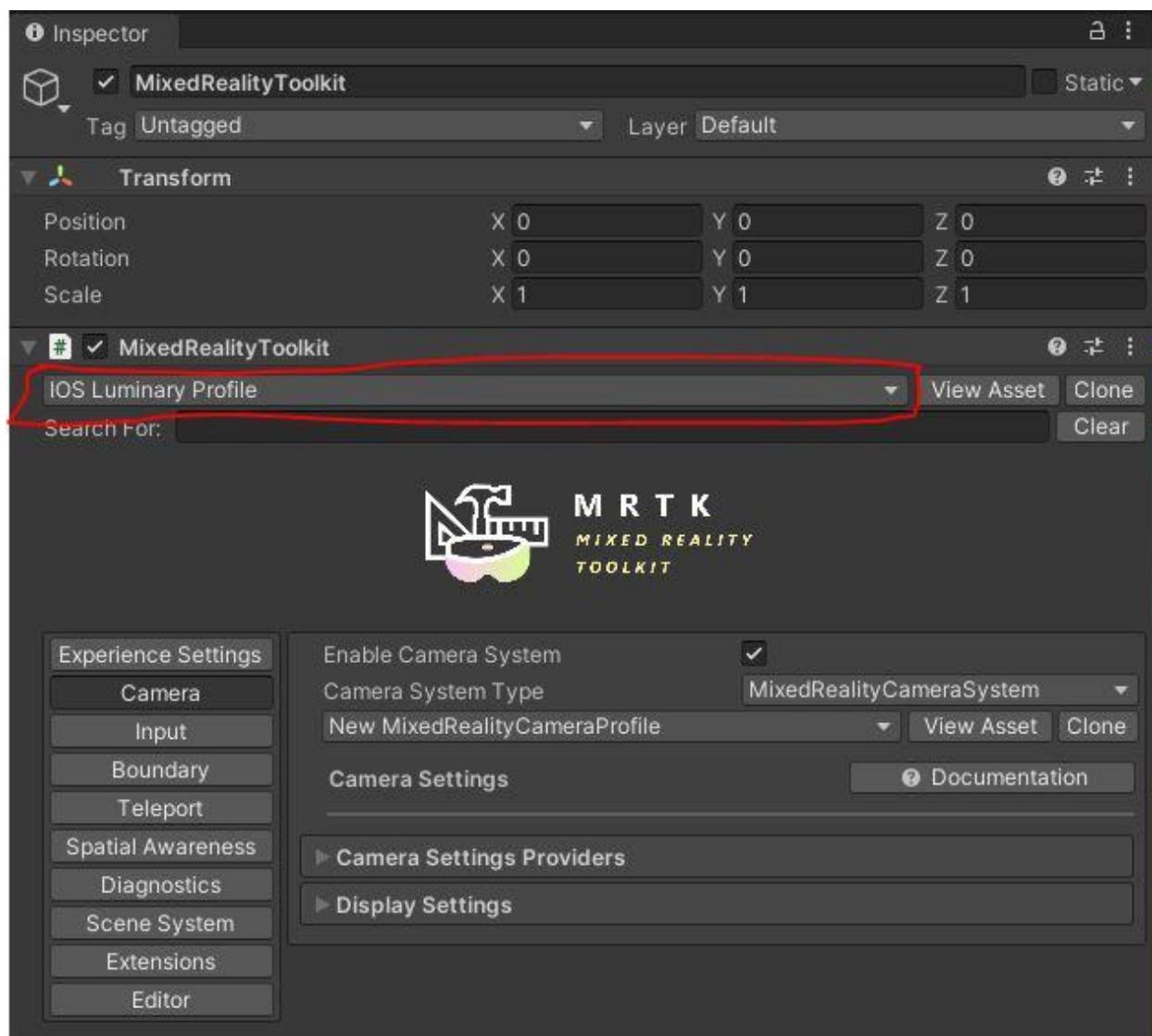


- Sprite layering example

Implementing MRTK For Interactions

- The ZIP folder and GitHub should have the completed projects with dependencies included. In the case of upgrading MRTK for the future, use the **Mixed Reality Feature Tool**

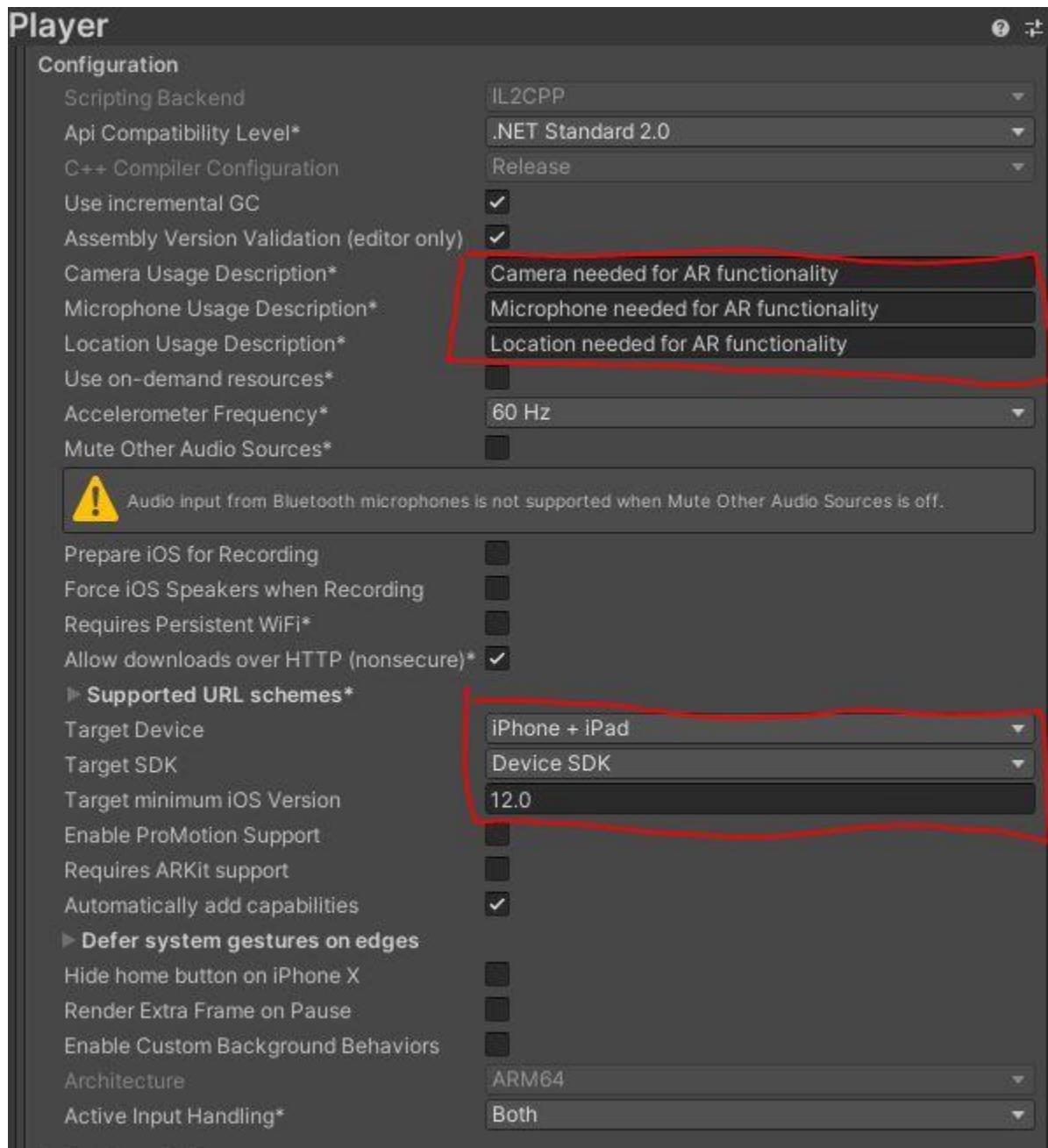
- <https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/welcome-to-mr-feature-tool>
- This tool will allow you to install MRTK to any Unity project. Makes life very easy and is necessary for the cleanest updates going forward.
- The only implementation of MRTK is the use of buttons and description panels for the art.
 - You can do such much more with the toolkit, but this tool at least allows interactivity to be cross platform with included hand tracking on certain devices



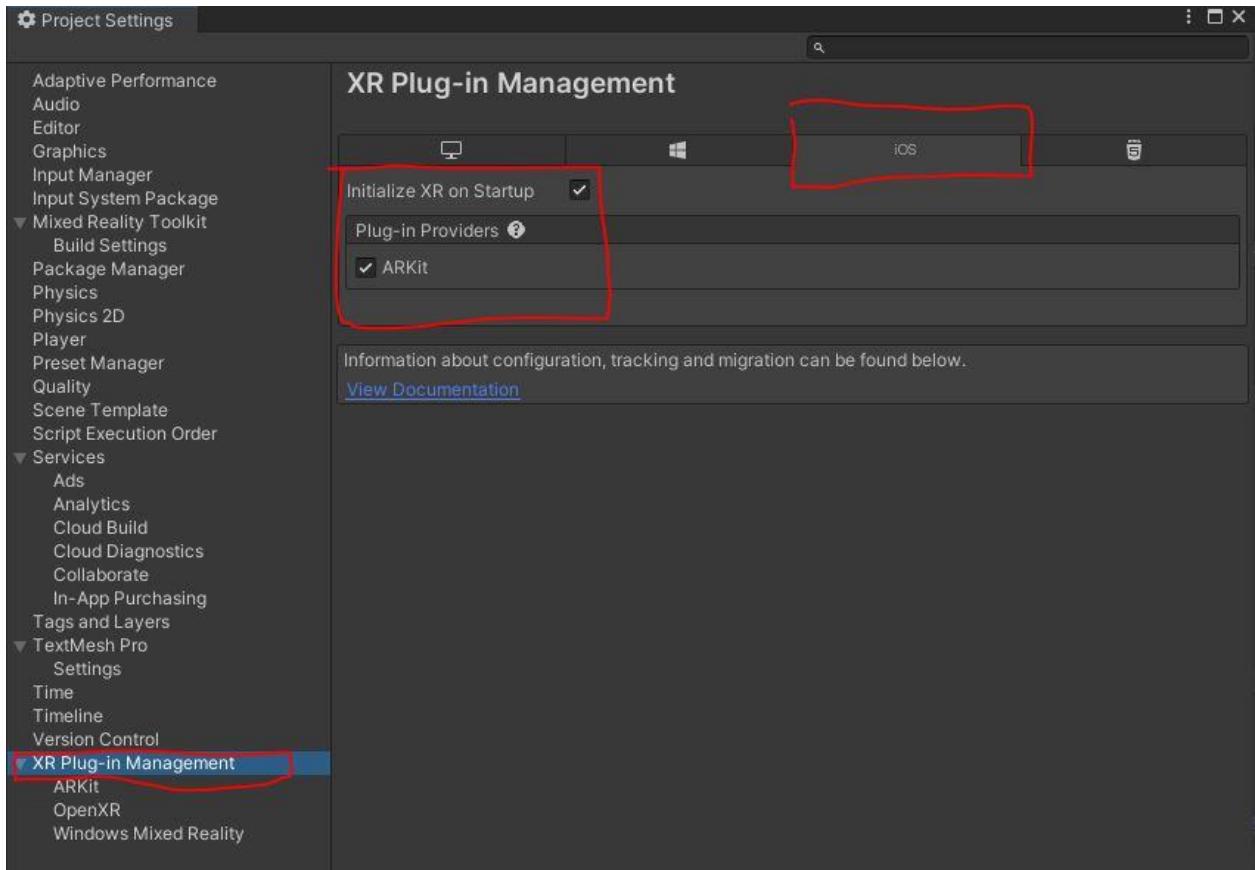
- MRTK Buttons within the Hierarchy are pretty self explanatory. If you know how buttons work in Unity, you should be able to see how the description panel spawns in and out.

Building The Game

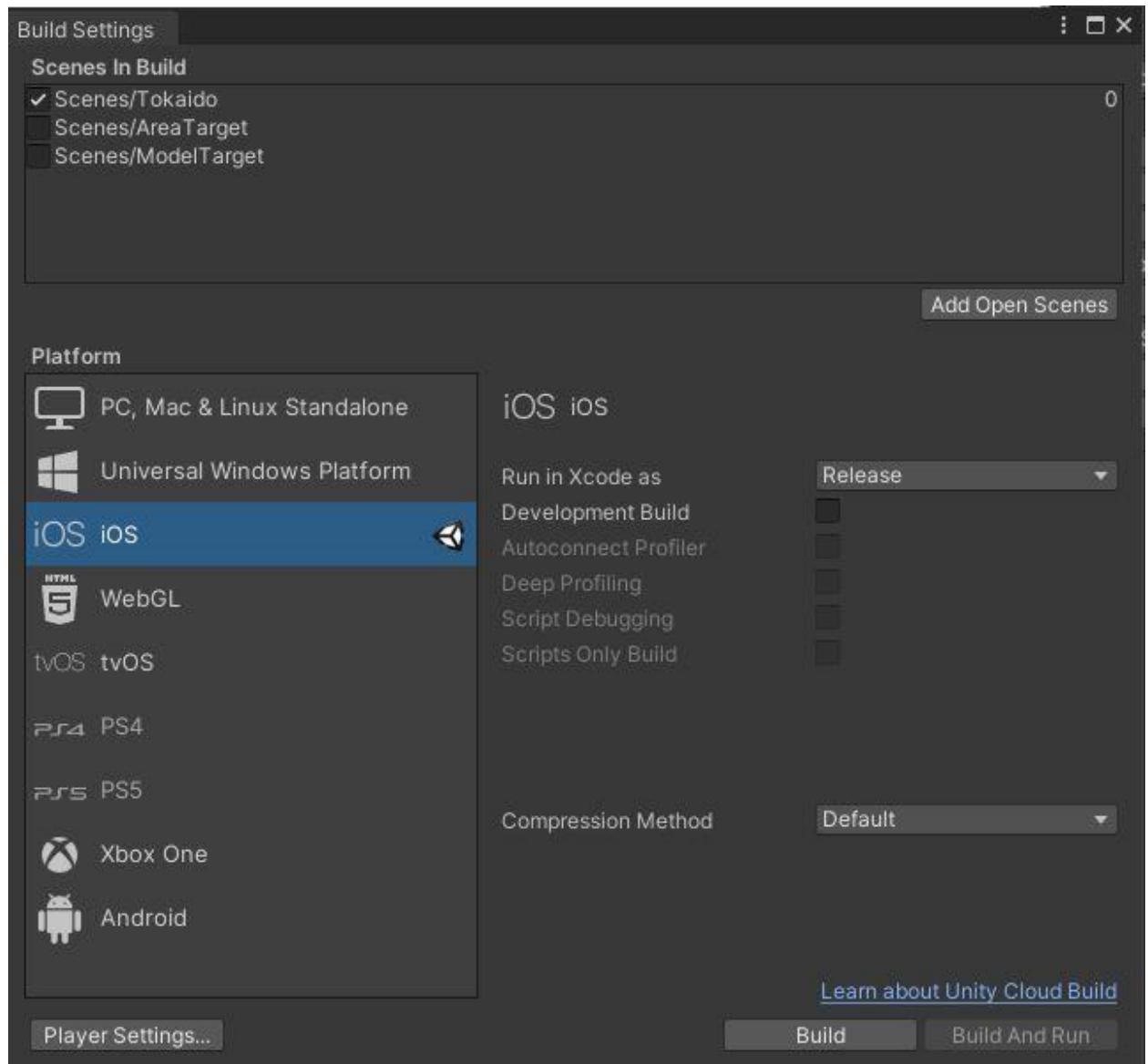
- Requirements
 - For IOS you will need a MAC with XCode installed. You will Also need a Free Apple Developer License which you can easily get by following this tutorial
 - <https://9to5mac.com/2016/03/27/how-to-create-free-apple-developer-account-sideload-apps/>



- Make sure your player settings for IOS look like this!



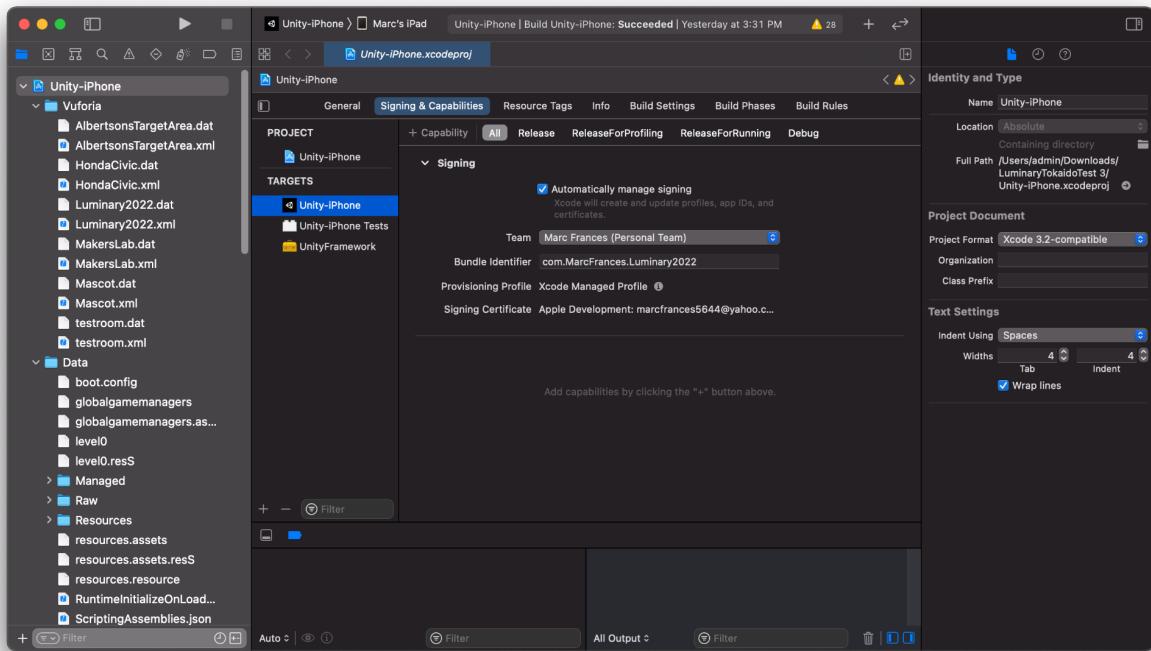
- Make sure this matches in your Project Settings



-Build and you should get an Xcode File.

Downloads				
Name	Size	Kind	Date Added	
↳ LuminaryTokaidoTest 3	--	Folder	Yesterday at 2:06 PM	
↳ Unity-iPhone	--	Folder	Yesterday at 2:06 PM	
↳ Unity-iPhone.xcodeproj	411 KB	Xcode Project	Yesterday at 2:06 PM	
↳ Unity-iPhone Tests	--	Folder	Yesterday at 2:06 PM	
↳ UnityFramework	--	Folder	Yesterday at 2:06 PM	
↳ UnityData.xcassets	--	Folder	Yesterday at 2:06 PM	
↳ process_symbols.sh	373 bytes	Shell Script	Yesterday at 2:06 PM	
↳ MapFileParser.sh	804 bytes	Shell Script	Yesterday at 2:06 PM	
↳ MainApp	--	Folder	Yesterday at 2:06 PM	
↳ Libraries	--	Folder	Yesterday at 2:06 PM	
↳ LaunchScreen...nePortrait.png	128 bytes	PNG image	Yesterday at 2:06 PM	
↳ LaunchScreen...andscape.png	128 bytes	PNG image	Yesterday at 2:06 PM	
↳ LaunchScreen...ne.storyboard	3 KB	Interface...ocument	Yesterday at 2:06 PM	
↳ LaunchScreen...ad.storyboard	3 KB	Interface...ocument	Yesterday at 2:06 PM	
↳ LaunchScreen-iPad.png	128 bytes	PNG image	Yesterday at 2:06 PM	
↳ Info.plist	3 KB	Property List	Yesterday at 2:06 PM	
↳ Frameworks	--	Folder	Yesterday at 2:06 PM	
↳ Data	--	Folder	Yesterday at 2:06 PM	
↳ Classes	--	Folder	Yesterday at 2:06 PM	
↳ LuminaryTokaidoTest8.zip	457.7 MB	ZIP archive	Yesterday at 2:02 PM	
↳ LuminaryTokaidoTest 2	--	Folder	Yesterday at 1:38 PM	
↳ LuminaryTokaidoTest7.zip	457.7 MB	ZIP archive	Yesterday at 1:33 PM	
↳ LuminaryTokaidoTest	--	Folder	Apr 25, 2022 at 12:05 PM	
↳ LuminaryTokaidoTest6.zip	404.1 MB	ZIP archive	Apr 25, 2022 at 12:03 PM	
↳ ModelTarget_IOS	--	Folder	Apr 19, 2022 at 9:21 PM	

- (On Mac) Right click on the file that contains the XCode Project, click **New Terminal at Folder**
 - Add these commands in order to have a successful XCode build!
 - chmod +x MapFileParser.sh
 - chmod +x process_symbols.sh
- Go into XCode!



- Connect an Apple Device to your Mac and make sure you have your Apple Developer License connected the Mac and your Device.
 - Because it's so difficult to work with apple, you then have to verify the app through your general settings on the device you installed the app on.
 - In Settings, go to **General>Device Management>Your Development Account>Verify App.**
 - That will verify your app and it should be good to go. Keep in mind, you will need an internet connection on your apple device to Verify the app because yes!
 - **DISCLAIMER**
 - With your free developer account, you can only sideload 3 apps across all your registered devices. So be sure to delete before you install. Sometimes it Appends, sometimes it doesn't. I haven't looked into that one! Just keep that in mind in case your build fails.
- Everything should hopefully work!

Photoshop Tips and Tricks

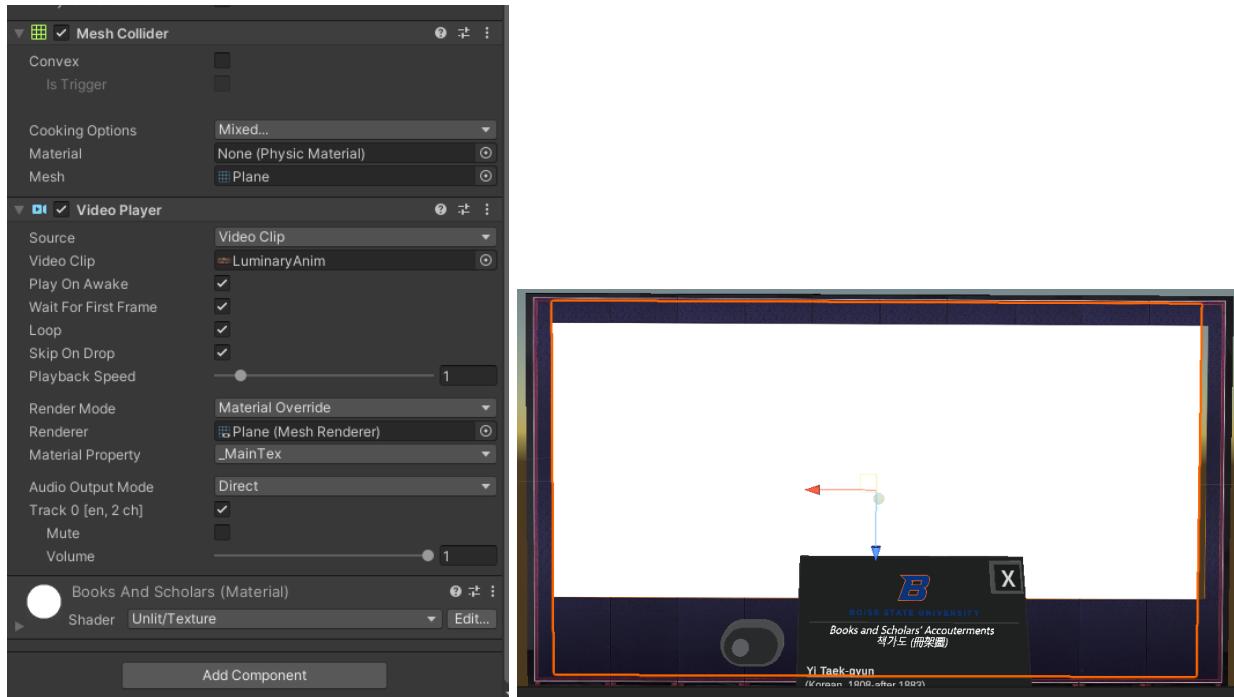
- The first method for animation involved drawing individual frames on separate layers over the original painting in Photoshop(1).
- The second method involved cutting out pieces from the art, using the Patch Tool to fill the hole left in the background, and utilizing the Puppet Warp Tool in

Photoshop to manipulate the objects on new layers(2). By duplicating the object onto 5-6 layers, manipulating each layer with the Puppet Warp Tool, and utilizing the Animation Timeline in Photoshop, the pieces could be animated without causing distortions where they overlap onto the painting.

- Another method is exporting each of the individual pieces you've cut out in photoshop. These can then be turned into sprites which can be individually animated in Unity. This also opens the doors to interactivity since each sprite is separate and can possibly be tapped on.

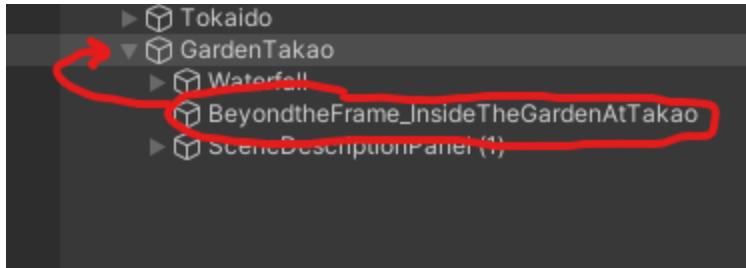
Placing Video Player Into Scene for Animations

- Add an asset that is your mp4 video to the Assets tab
- Create a plane 3D Object, just leave it with a standard texture
- Add Mesh Renderers and Colliders to the object if not already applied
- Add a Video Player Component in the Inspector, placing your desired mp4 as the “Video Clip”
- Make sure to check the boxes of “Play on Awake,” “Wait for First Frame,” “Loop,” and “Skip on Drop”

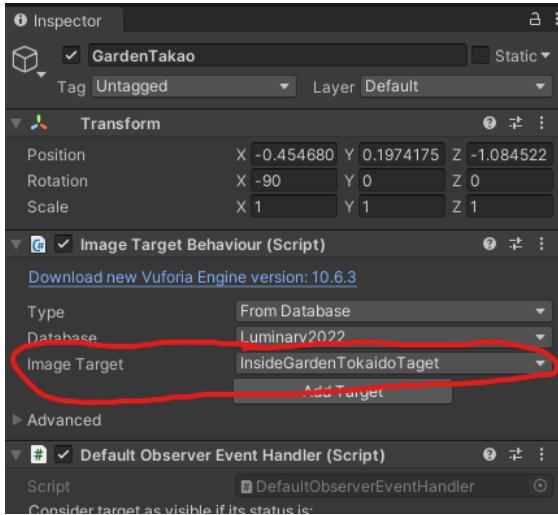


Beyond The Frame Sprite Implementation

- Import the image into Unity, ensuring that the object is being imported as a Sprite
- Attach the sprite as a child of the Image Target in the Hierarchy



- In the Image Target Object, Under the “Image Target Behaviour (Script)” Component, apply the “Image Target” to be the sprite of your desired image



- Finally, attach an Unlit Material to your screen. This will brighten up the screen as it won't be affected by light or shadows.