

Unity Tutorial for Creating Nina Hewitt's Alpine VR Tour

Prepared by: Emily Ballon, Alpine VR Tour Project Assistant, Mar 2020, Hewitt Biogeography Teaching and Research Lab [<http://blogs.ubc.ca/alpineplants/>]

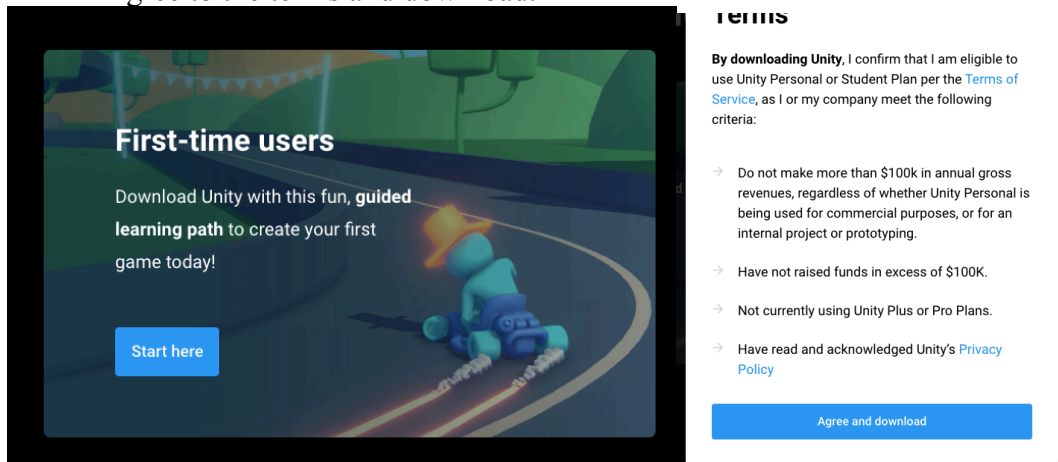


Table of Contents:

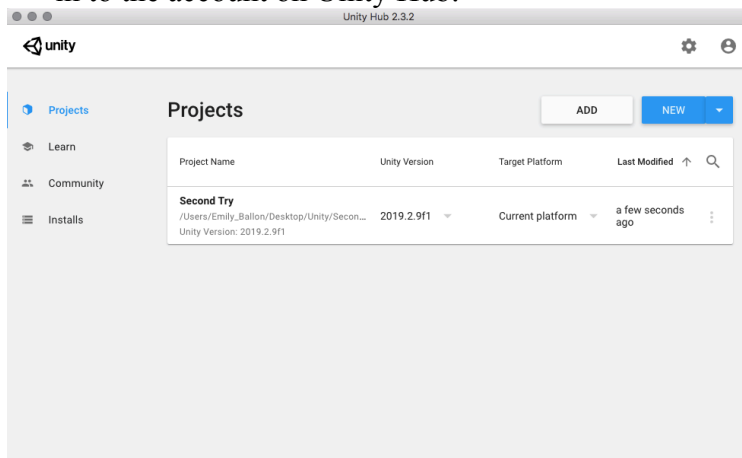
Contents:	Page
Setting up Unity	3
Creating a Project	4
Setting up your scene	5
Useful information to know before adding content	9
Creating spheres	12
Creating sprites	16
Creating forward and back buttons in the spheres	17
Creating pop-up information blurbs	18
Creating multiple choice pop ups	20
Creating hyperlinks	21
Adding videos	22
Building (exporting) the project	23
Uploading	25
Helpful resources	26

Setting up Unity

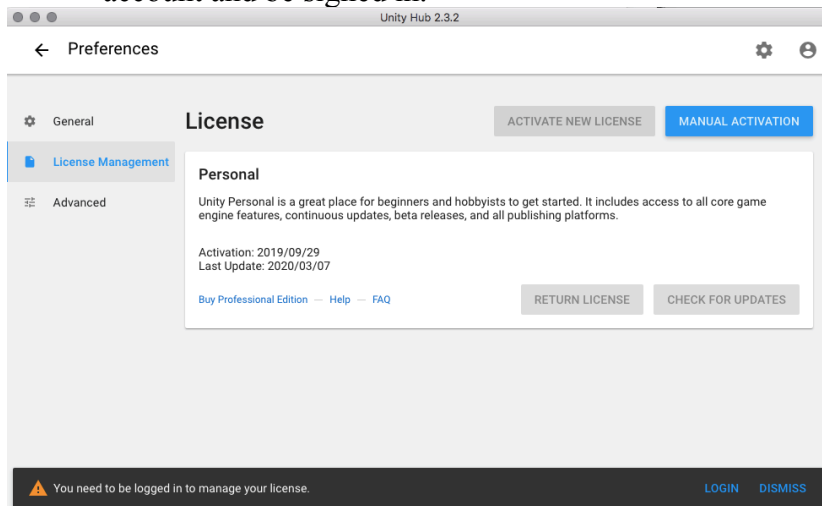
1. Download Unity Personal from the Unity website: <https://store.unity.com/download-nuo>. Agree to the terms and download.



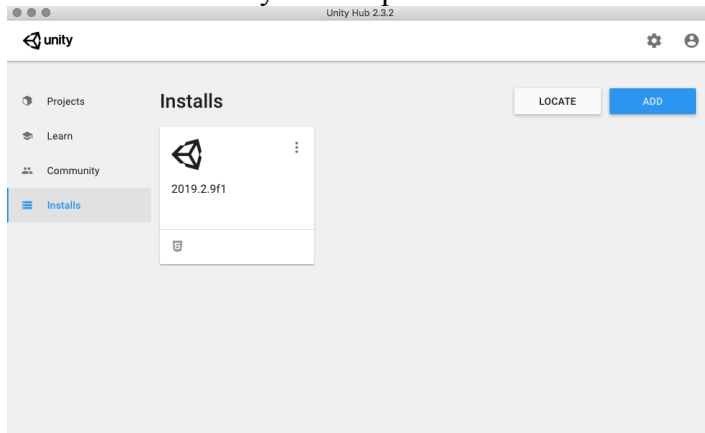
2. Open up Unity Hub. Create an account using the person icon in the top right corner. Sign in to the account on Unity Hub.



3. Click the tool button next to the person icon in the top right to open preferences. Go to license management, and press "activate new license". *You need to have created an account and be signed in.

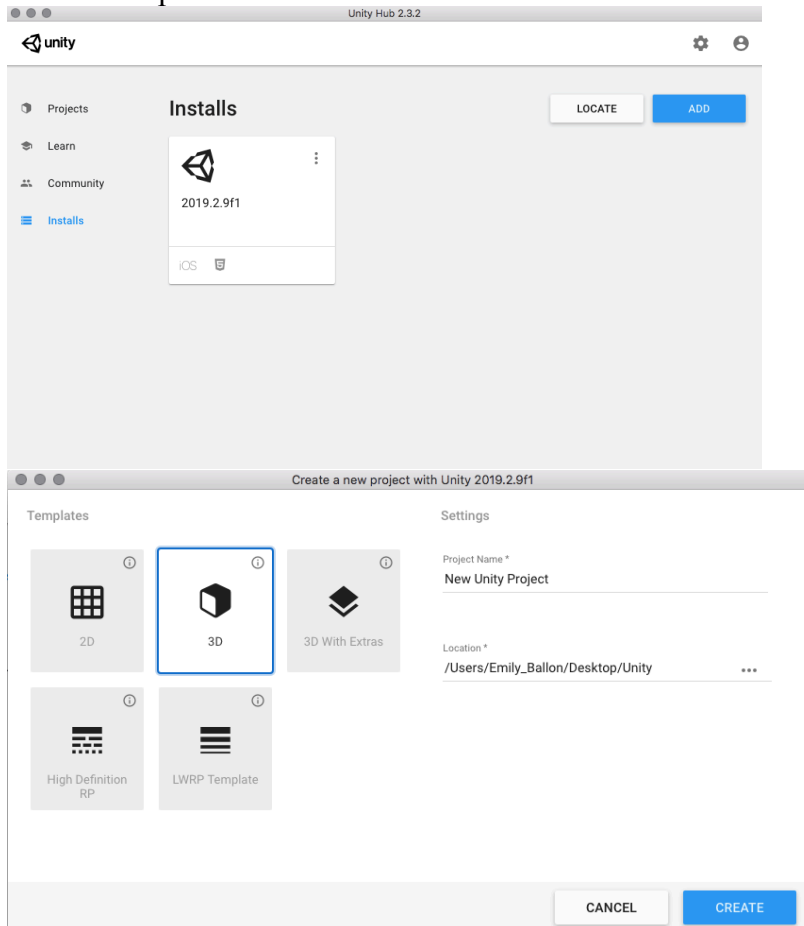


- Using the back button, go back to the Unity Hub project area, and click on installs in the left menu. Click add in the top right and install the most recent version of unity. Add whatever modules are wanted, but none are needed. Can be added later. Now unity is installed on your computer

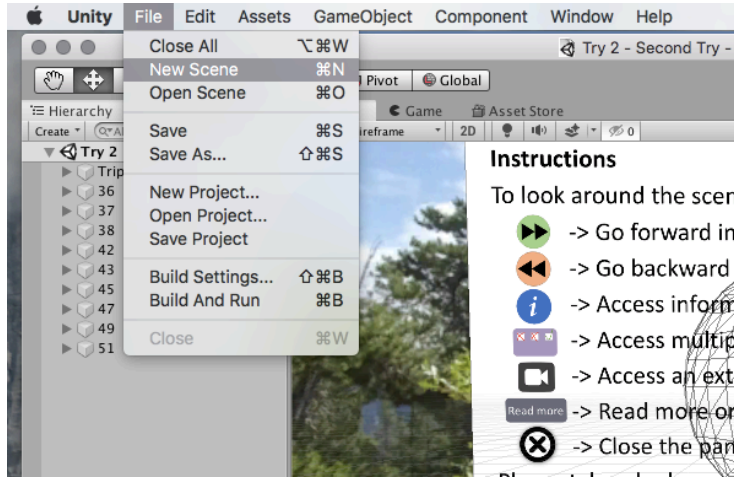


Creating a project

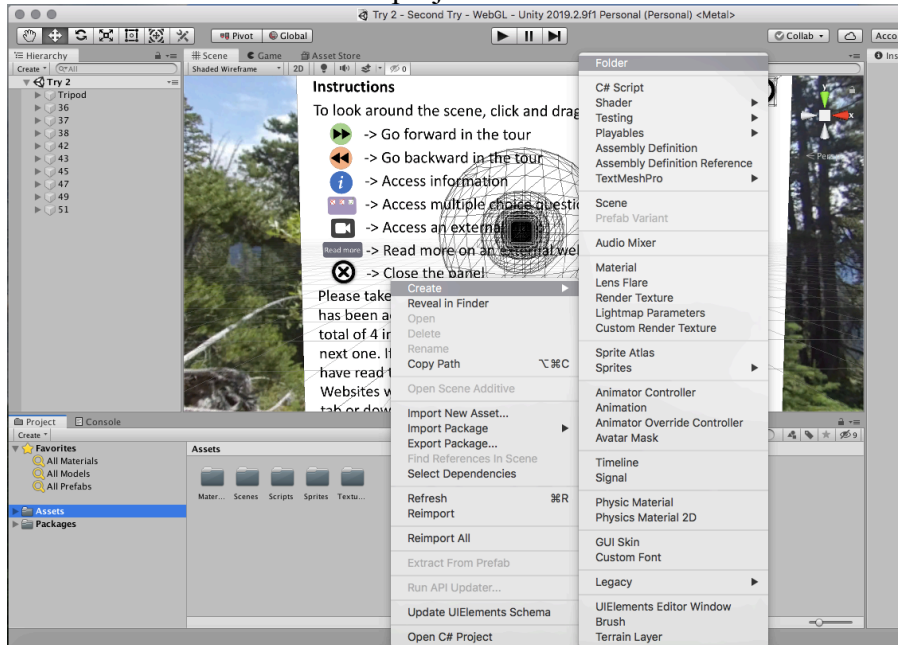
- Navigate to the projects tab in the left menu and click new. Select 3D, name your project, and assign the location to the folder you want to store your entire project in on your computer. Click create.



- Now create a scene to work in. Once your Unity project is open, in the top bar click File > New Scene

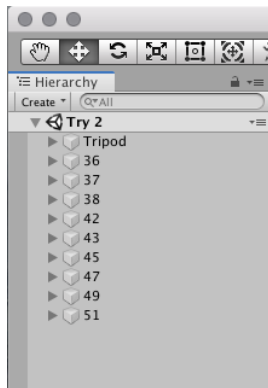


- In the Project tab of the Unity interface (located at the bottom left), select the assets folder. In the assets folder, right click and select Create > Folder. Create 5 folders. Name them Materials, Scenes, Scripts, Textures, and Videos. Place the scene created previously into the Scenes folder (just by drag and drop). *You can create as many folders as you need, it's just a nice way to organize all of the parts of your project. Those 5 are the least amount I used for this project.

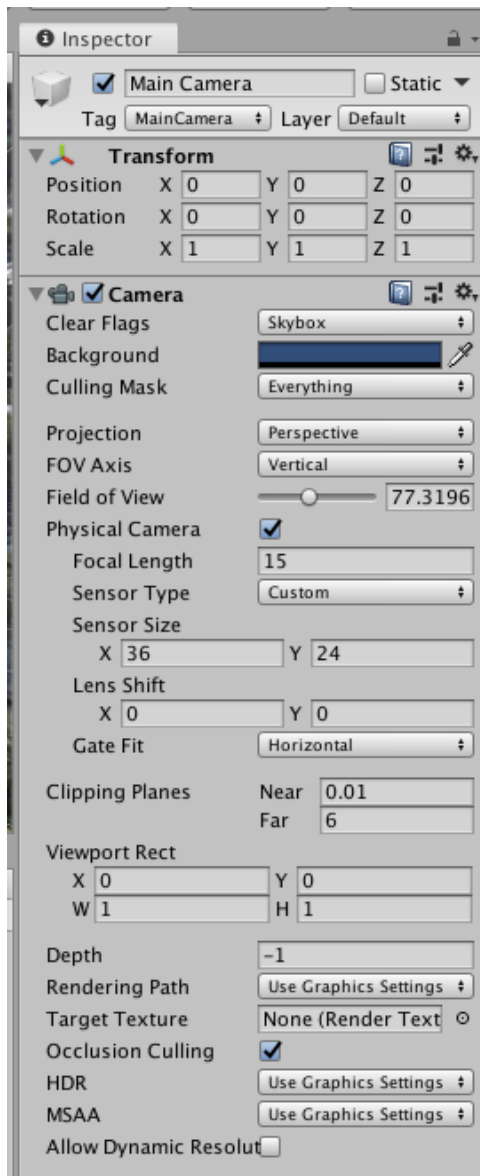


Setting up your scene

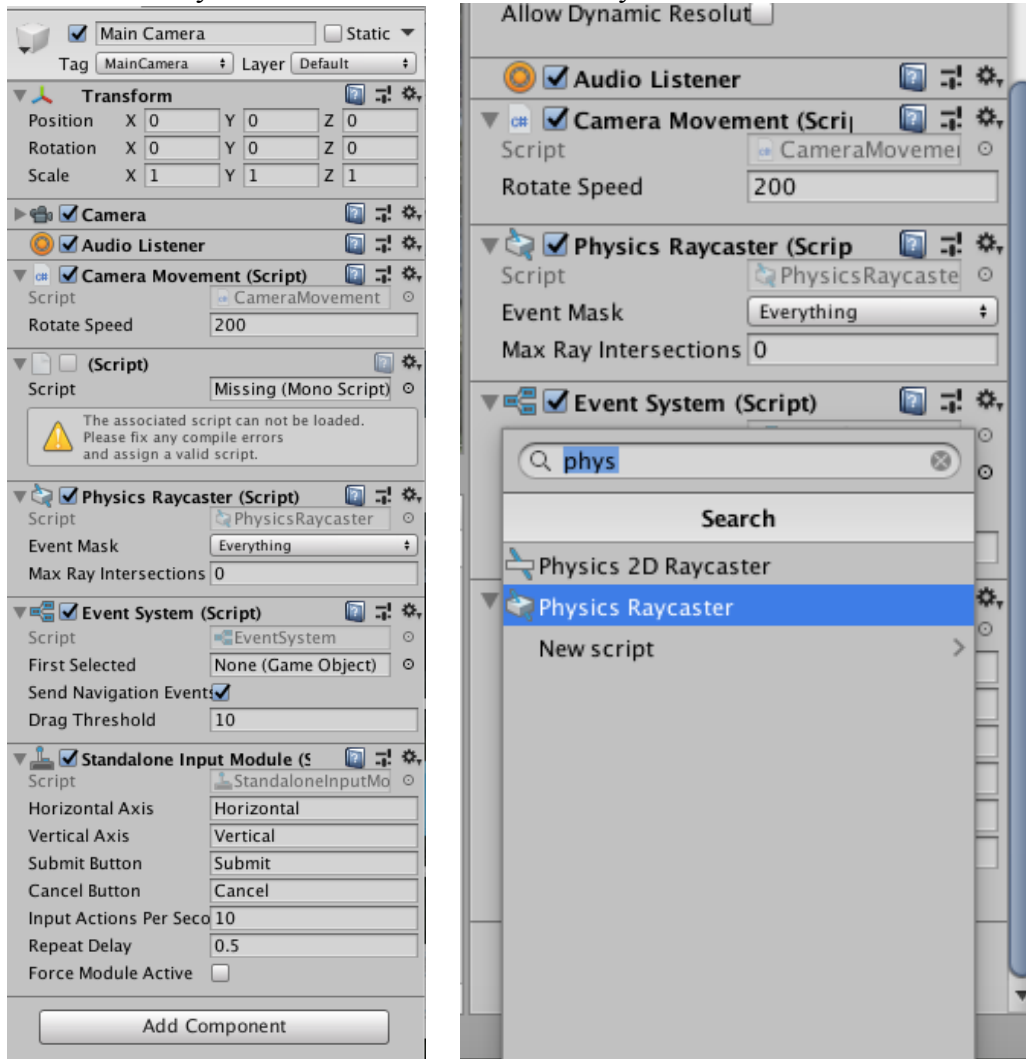
- First, make sure that you are always working in your scene. This is where everything is saved. You can tell if you are in the scene if it is listed under the hierarchy, in the top left area. Mine is called "Try 2"



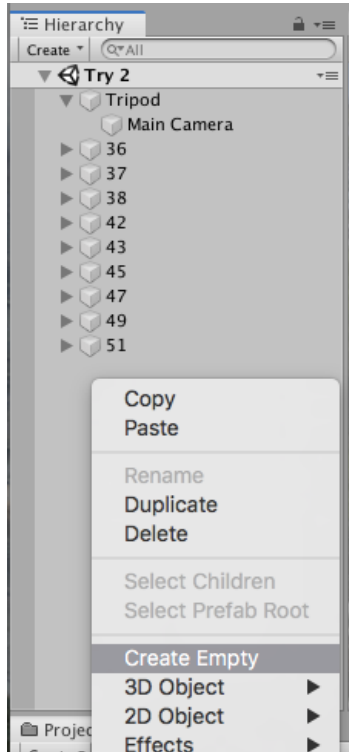
2. In the hierarchy, delete the Directional Light by right clicking and selecting delete
3. Click on Main Camera in the hierarchy, and set the Position to 0,0,0 in the inspector on the right side of the screen. In the inspector under Clipping Planes, set Near to 0.01, and Far to 6.



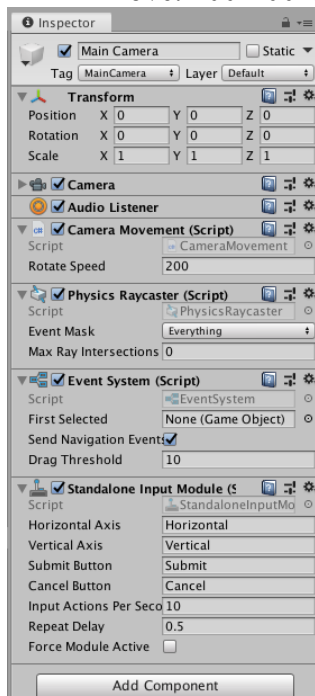
- In the inspector for Main Camera, scroll down to the bottom and click add component. Type “Physics Raycaster” in the text box, and click on it to add it to the camera. Do the same for “Standalone Input Module”. When adding the Standalone Input Module, an EventSystem will be added automatically to the main camera.



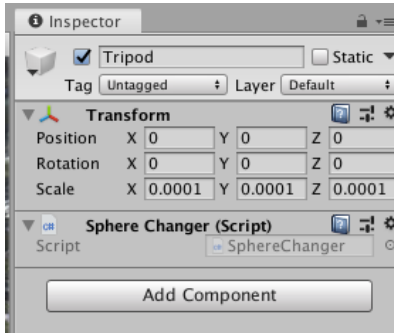
- In the Hierarchy, right click and select Create Empty. This will create a GameObject. Rename the GameObject as Tripod by clicking twice on the name. In the inspector, set the position to 0,0,0, and the scale to 0.0001, 0.0001, 0.0001. Then in the hierarchy, drag and drop the Main Camera onto the tripod.



6. Download the attached scripts from <http://blogs.ubc.ca/alpineplants/vrtour-turoial/> labeled SphereChange.cs, CameraMovement.cs, and BCParks.cs. Place them into the Scripts folder in Assets.
7. Click on the Main Camera in the hierarchy, and add the “CameraMovement” script through the Add Component button at the bottom of the inspector. Set the Rotate Speed under Camera Movement (Script) as desired, for how quickly you want the screen to move. 100-200 is generally good, with increasing values being faster.



8. In the Tripod's inspector, add the "SphereChanger" script with the Add Component button



Useful information to know before adding content

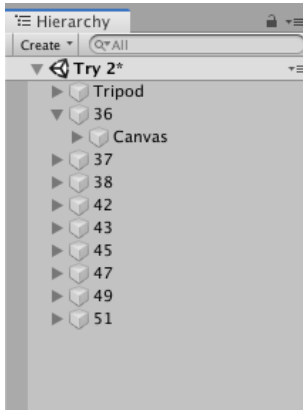
1. To test the project, click the play button at the top of Unity. To make the screen larger, click the Maximize on Play in Game view (the screen will be larger next time you play the project, use the same button to minimize the screen). To exit Game view, hit the play button again. This view enables you to see what a finished project will look like.



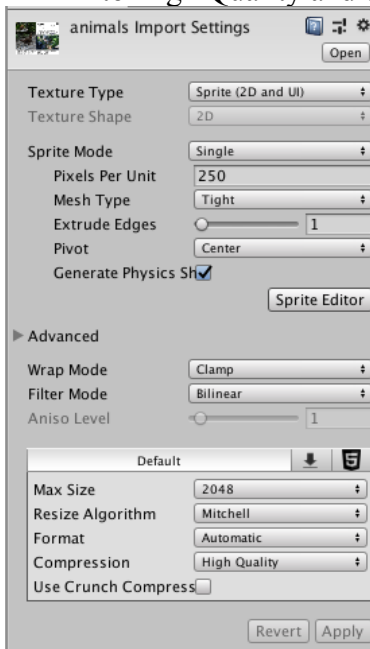
2. Game view is an easy way to manipulate the location of everything within your project and I recommend using it to place things in their correct location, however any changes made in game view will not be saved so be sure to record any adjustments made externally. This also means that you can test things and it will not affect the project at all.
3. Moving around in game view is controlled the same way as in the actual product, with mouse clicks and drags.
4. In edit mode, the buttons in the top left of Unity are helpful for adjusting the scale, position, and rotation of objects, as well as the view of your project. Play around with them until they make sense. Location and sizes of objects can also be manipulated in the Inspector of each object. The easiest way to move around your project is to select the hand button. To move around, click and drag the mouse. Zoom in and out using 2 fingers on the trackpad, rotate the view by holding option and dragging the mouse.



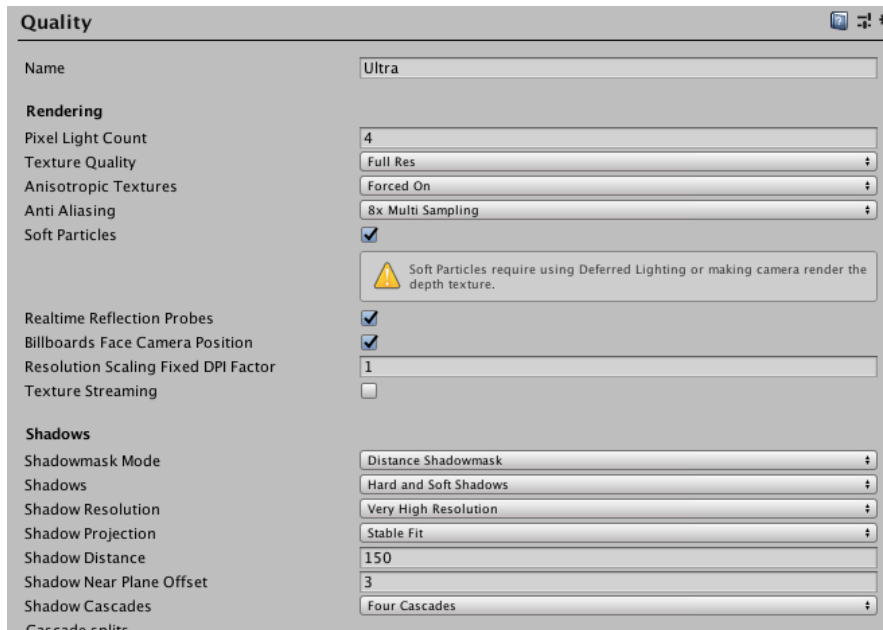
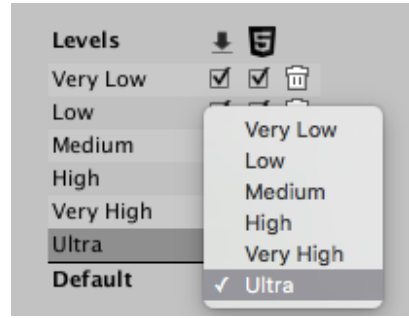
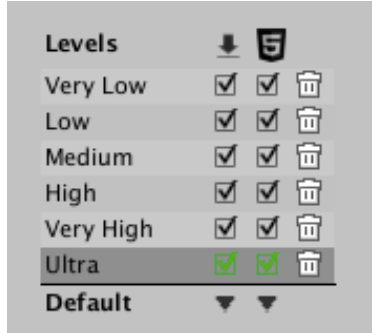
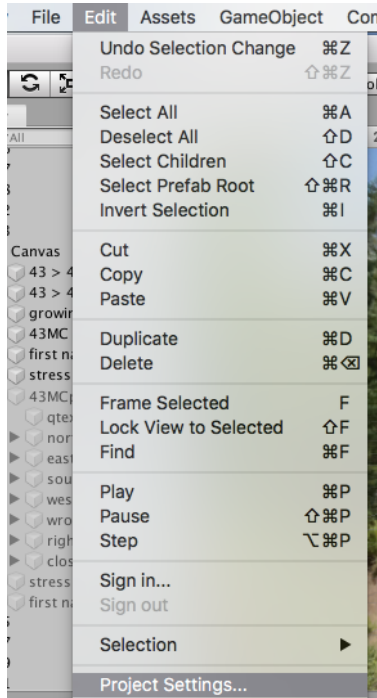
5. The order that objects are seen in the project is the order they appear in the hierarchy. If two objects have the same spatial position, objects at the top of the hierarchy will occur behind those that come lower in the hierarchy. This can be easily manipulated by dragging and dropping objects above or below others in the hierarchy. In the image below, 49 will appear behind 51, if they are at the same position.



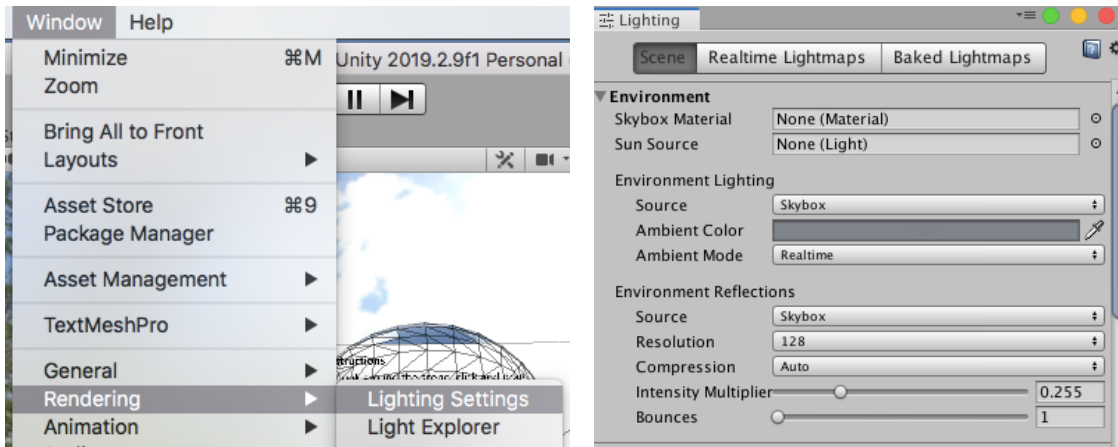
6. Make sure to name all objects added to the hierarchy with individual and identifiable names. There will be many different objects by the end and it is good to have things organized.
7. To increase quality of sprites, go to the Inspector of the sprites, increase the compression to High Quality and click apply.



8. To increase the quality of the tour, go to Edit > Project Settings. Click the Quality tab on the left. Click the down arrows next to default (there will be as many columns as platforms you have set up, this will make sense later) and change the quality to Ultra. Set Texture Quality to Full Res, Anti Aliasing to 8x Multi Sampling, and Shadow Resolution to Very High Resolution.

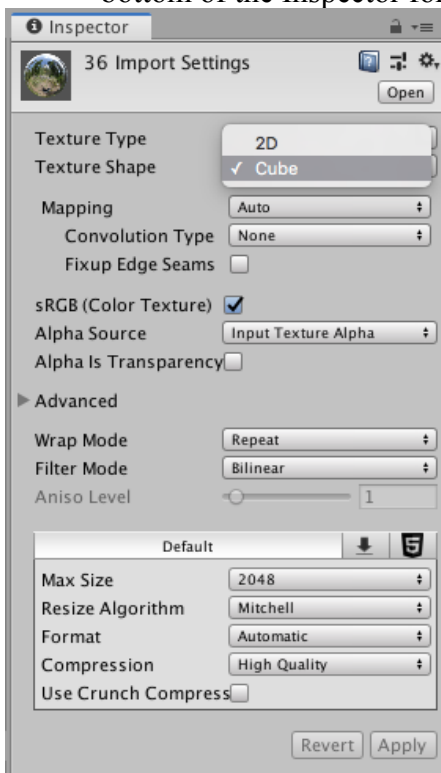


- Another way to increase the quality of the tour is by going to Window > Rendering > Light Settings. Go to the Quality tab on the left and scale the Intensity Multiplier down to about 0.25.

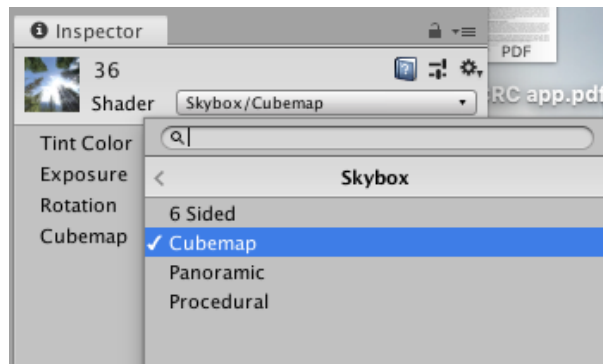
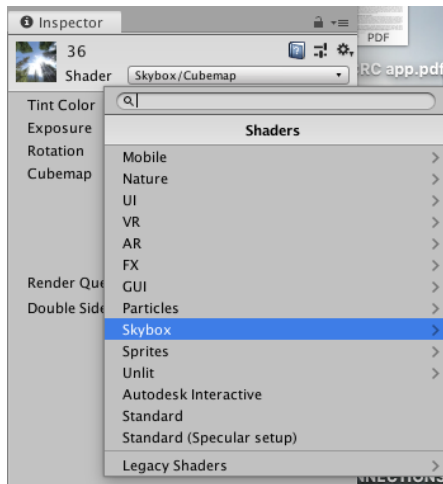
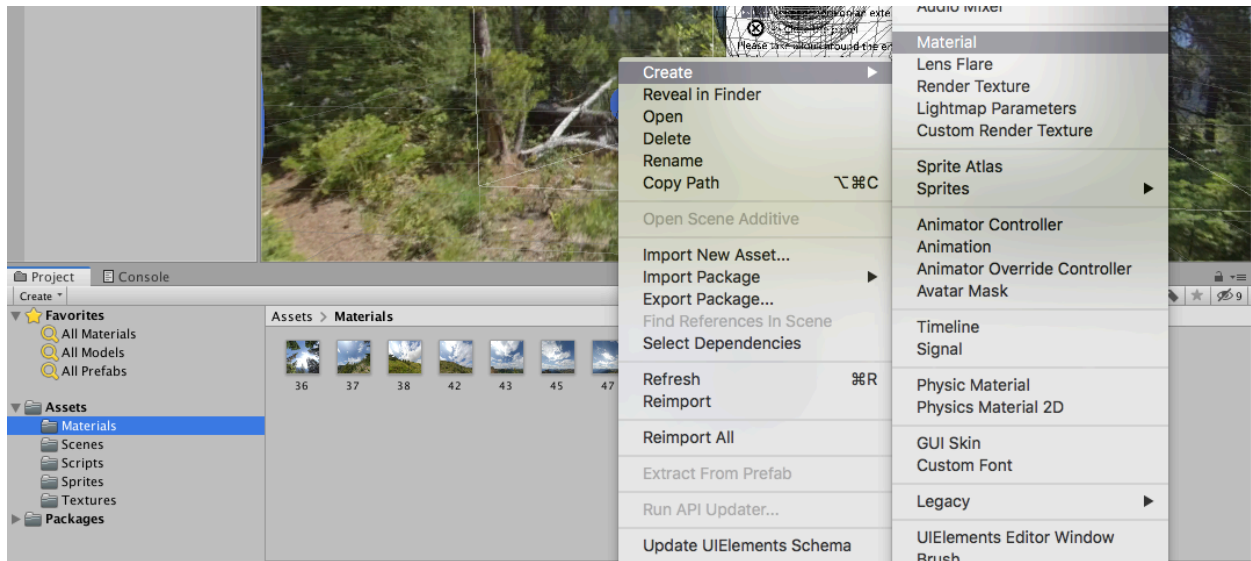


Creating Spheres

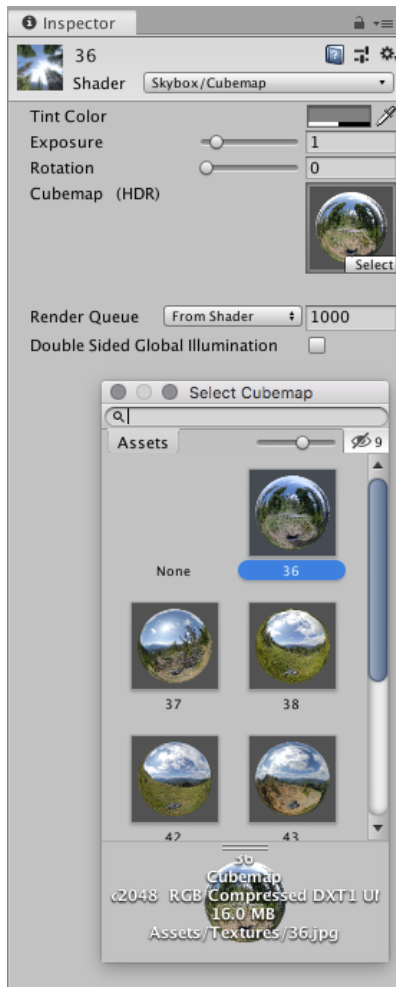
1. Outside of Unity, export your 360 images as 2D images with a 2:1 ratio in JPEG or PNG format. Drag the images into the Textures folder in Unity. Click on each image, and in the Inspector change the Texture Shape to cube in the drop down. Click Apply at the bottom of the Inspector for all changes to be saved.



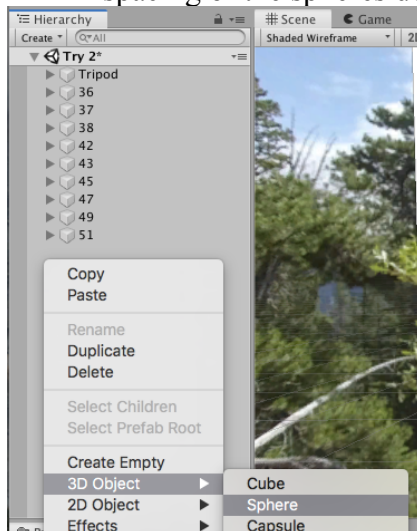
2. Enter the materials folder, right click in the folder and select Create > Material, to create a material. Create as many materials as you have 360 images. For each material created, click on the material and in the inspector next to Shader, click on the box and select Skybox > Cubemap.



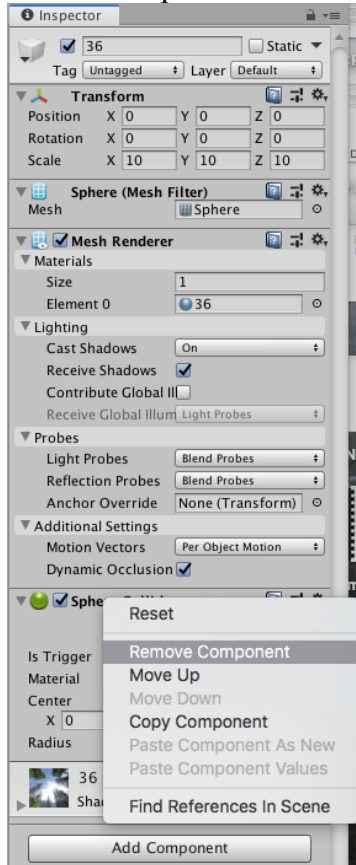
3. Assign a 360 image to the material by clicking on the box next to Cubemap (HDR) and select the 360 textures. Make sure to name each material to keep track of the image that it contains.



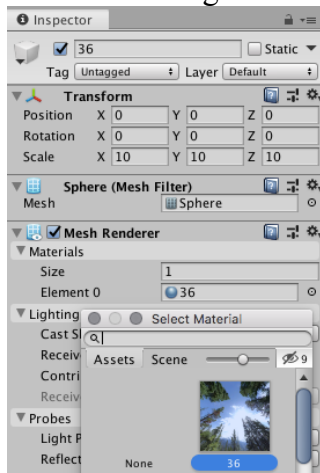
4. Create a sphere in the Scene by right clicking in the empty space of the hierarchy and selecting 3D object > sphere. Create as many spheres as there are materials. In the Inspector, set the scale for all the spheres at 10,10,10. Set the first sphere's position to 0,0,0 Space the following spheres at position 0,0,20, then 0,0,40 etc. The scale and spacing of the spheres does not really matter, it is just important to be consistent!



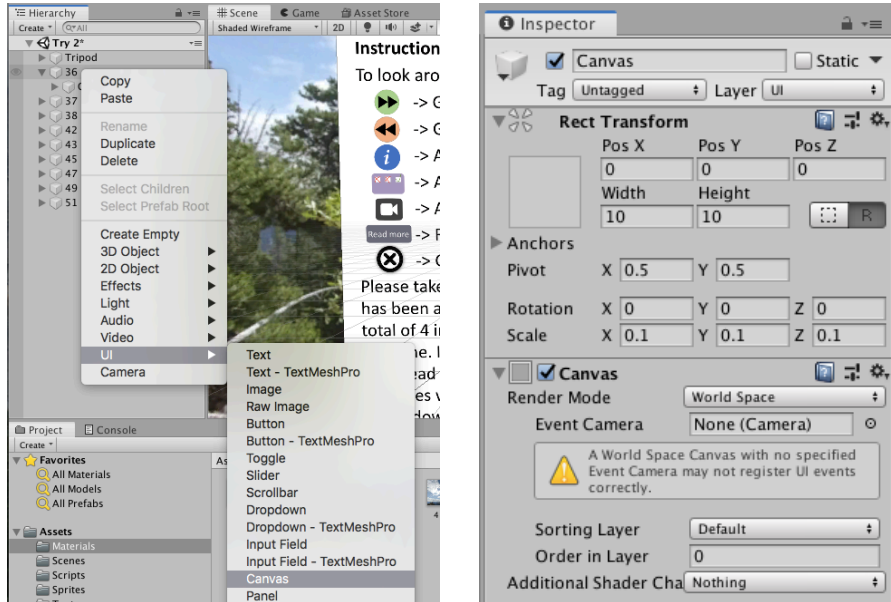
- For each sphere, in the Inspector remove the Sphere Collider component by right clicking on Sphere Collider and selecting Remove Component.



- Assign a 360 image to a sphere by clicking on the sphere in the Hierarchy, then in the sphere's Inspector, clicking on the circle to the right of Element 0 under Mesh Render, and selecting the correct image. Alternatively, you can drag the correct image from the materials folder onto the bar next to Element 0 in the sphere's Inspector. Make sure to name your spheres correctly. In case you change the background of the world to a 360 image, you can change it back to grey by choosing in the top bar Window > Rendering > Light Settings then assigning None to the Skybox Material by clicking on the small circle to the right.



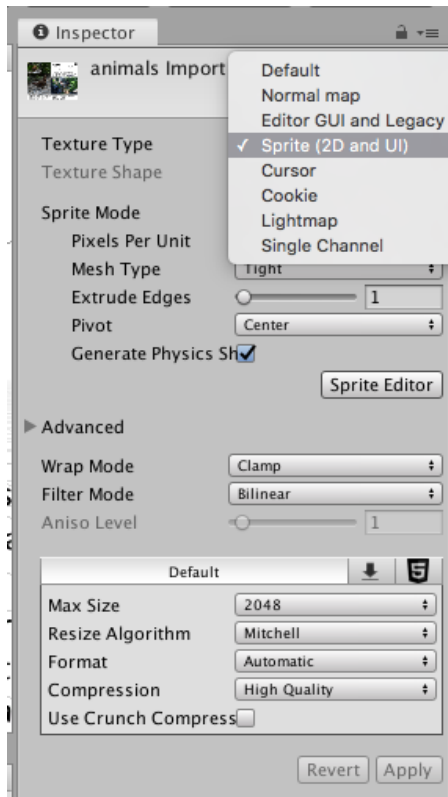
- To add interactive content to the spheres, a Canvas is necessary. Add this by right clicking on the sphere in the hierarchy, selecting UI > Canvas. Set the Canvas position to 0,0,0, the height and width to 10,10, and the scale to 0.1,0.1,0.1. Same as with the scale of the spheres, these are values that I have chosen. It does not matter what you chose, but make sure to be consistent. For each Canvas, change the Render Mode of the Canvas to World Space in the Inspector. Create a Canvas for every sphere that you will be adding content to.



- The tour will start wherever the camera is (in this case 0,0,0) so make sure that this is where the first sphere is located.

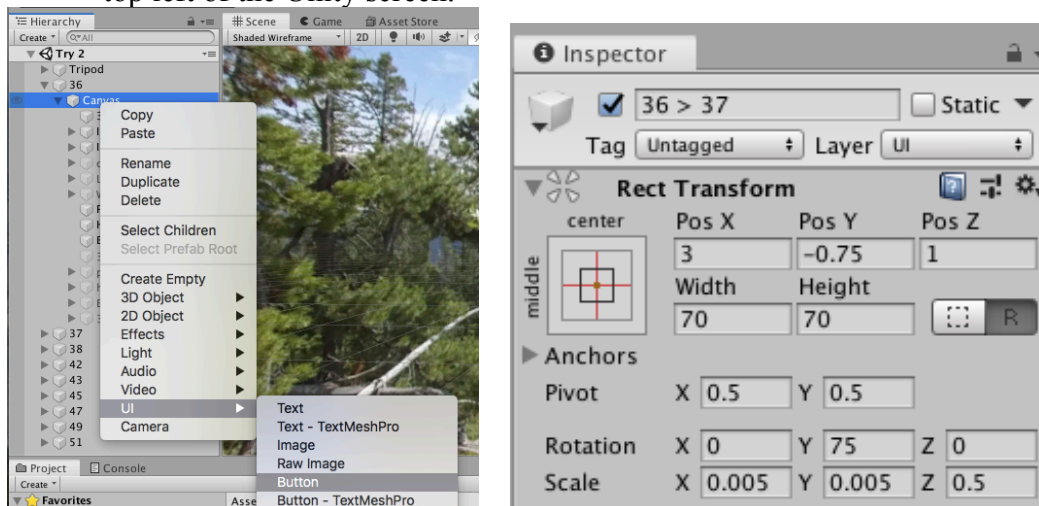
Creating Sprites

- Sprites are images that can be used to customize buttons and panels. To create a sprite, drag a 2D image in JPEG or PNG format from your computer into the Sprites folder in Assets.
- Select the image and in the Inspector change the Texture Type to Sprite (2D and UI) then click apply at the bottom.

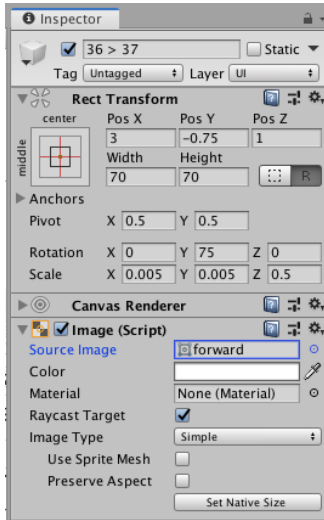


Creating forward and back buttons in the spheres

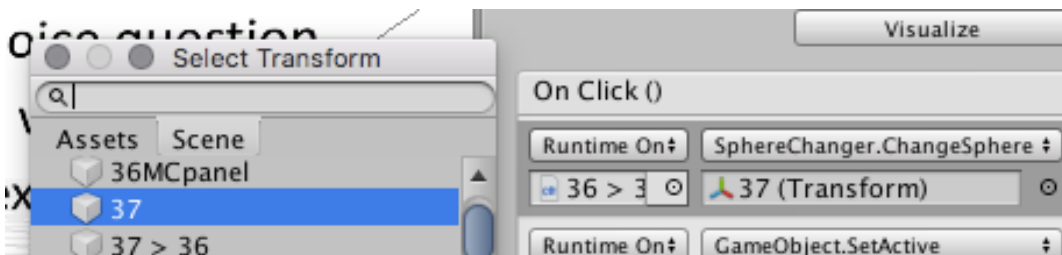
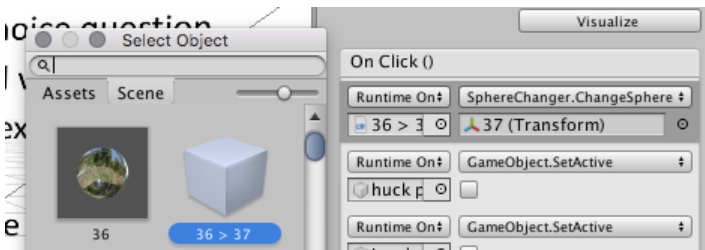
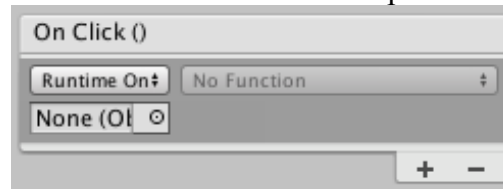
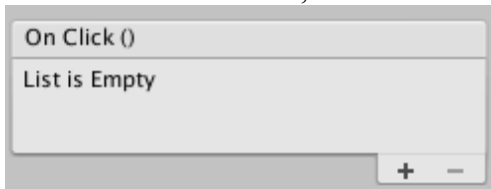
1. In the hierarchy, right click on the canvas in the sphere that you would like to add buttons to. Select UI > Button. Adjust the x,y,z, height + width, scale, and rotation of the button to place the button in the correct position. This can also be done using the buttons in the top left of the Unity screen.



2. To change the image of the button, assign a sprite to the Source Image in the Inspector. Delete the text if undesired by right clicking on it in the Hierarchy and choosing Delete. You can also change the colour/opacity of the button in the Inspector next to Color.

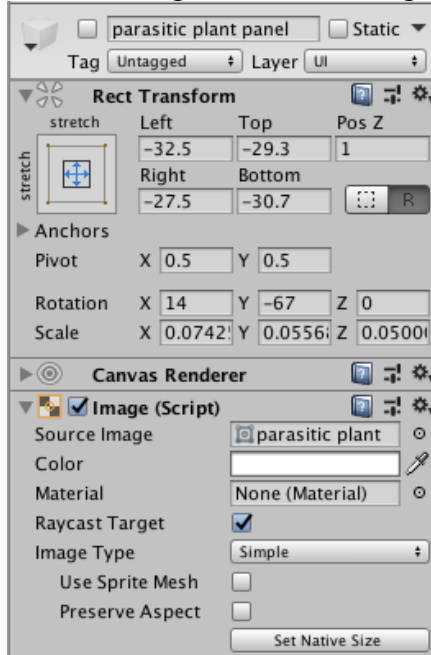


3. Select the button and in the Inspector, choose Add Component, and add the “SphereChanger” script.
4. Under On Click () box in the Hierarchy, click the plus in the bottom right corner. Three boxes will pop up. Click on the small circle in the bottom left box and click the Scene tab. Chose the desired button that you would like to perform the function. Then in the top right drop down, click No Function > SphereChanger > ChangeSphere (Transform). In the bottom right box, chose the sphere that you would like the scene to move to by clicking on the small circle to the right and choosing the correct sphere from the Scene tab. In this case, the button is called “36 > 37” and the correct sphere is called “37”.

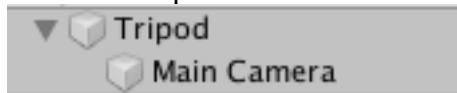


Creating pop-up information blurbs

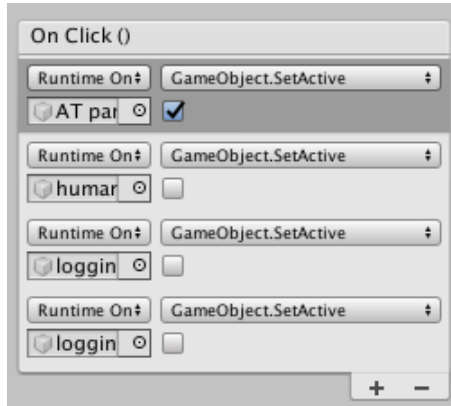
1. Create a button the same way as above (Step 1 of Creating forward and back buttons in the spheres)
2. Right click on the button, select UI > Panel.
3. Change the colour, image, scale, location and size of the Panel as desired in the Inspector.



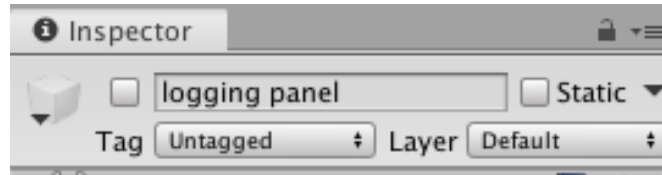
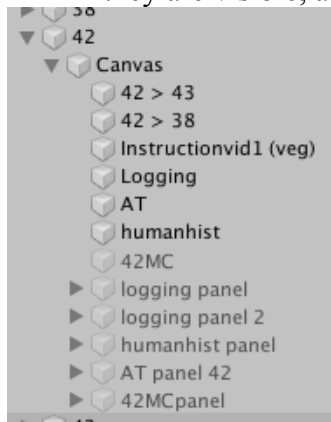
4. If the Panel is a child of the button (meaning that it is attached to the button as a drop-down option in the hierarchy), changing the scale and rotation of the button will affect the size and rotation of the panel. It is easiest to adjust the panel when it is a child of the button. I recommend putting the button in its correct location first, adjusting it, then adding the panel. In the image below, the Main Camera is a child of the Tripod, for an example.



5. Add a button to the panel that will be used to close the panel. Make sure that it is a child of the panel.
6. To enable a button to open/close a panel, click on the button's inspector, and then the + sign in the On Click () area, as before. As before with the forward and back button, assign the button you would like to perform the function. In the drop-down menu, select GameObject > SetActive (bool). Check the box if you want the button to open something, and leave it unselected if you want the button to close something. This can be used to open or close multiple panels at once. In the image below, if the button is clicked, the bottom 3 objects will close and the top object will open.



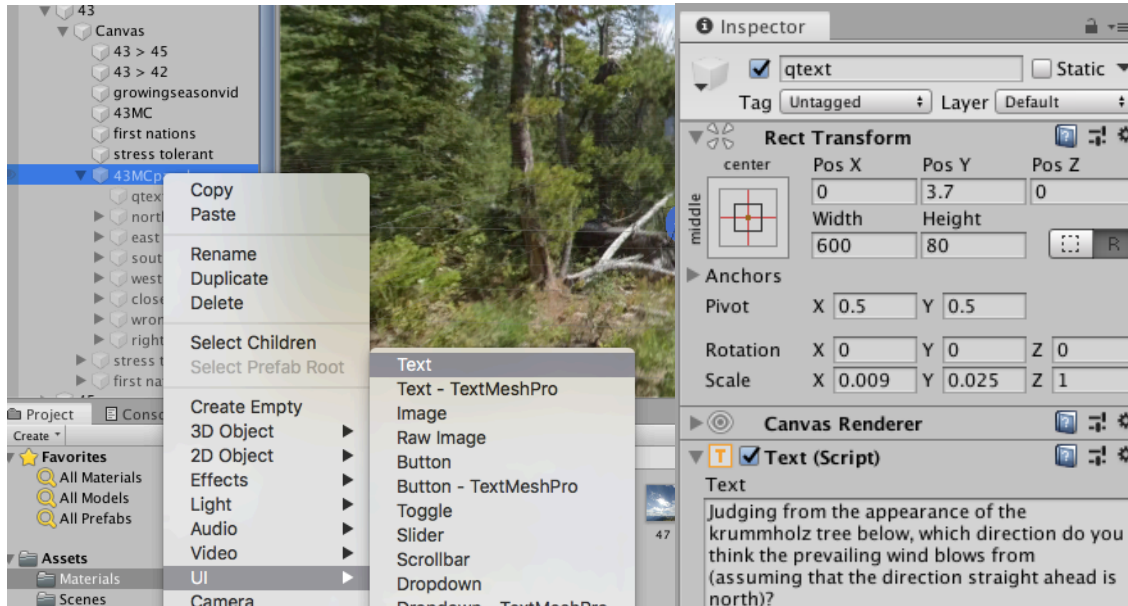
- Once the panel has been completed, in the hierarchy drag the panel off of the button so that it is no longer a child of the open button (but it is still a child of the canvas). In the Inspector for the panel, unclick the top left button next to the name to hide the panel until it is opened. Make sure that in the hierarchy, the panel is lower than the button so that the button does not obscure the panel. In the image below, the buttons are in black writing as they are visible, and the grey writing is the panels as they are invisible.



- If you would like to change the position of a panel later, I recommend making it a child of the button again, then moving it by repositioning the button.

Creating multiple-choice pop ups

- Create a button the same way as above.
- Create a panel that can be opened and closed with the button the same way as above.
- Add text to the panel by right clicking on the panel in the Hierarchy, selecting UI > Text. Scale and move the text to the top of the panel as you would with any other object. Make sure the text is a child of the panel. This text will serve as the question. The text can be changed in the text box in the inspector.



4. Create as many buttons as there are multiple choice options, add them to the panel, and label each button as a different option. Don't forget a close button. In this example, the options are north, east, south, west. Wrong and right are 2 more panels that will display the answer when each button is clicked.



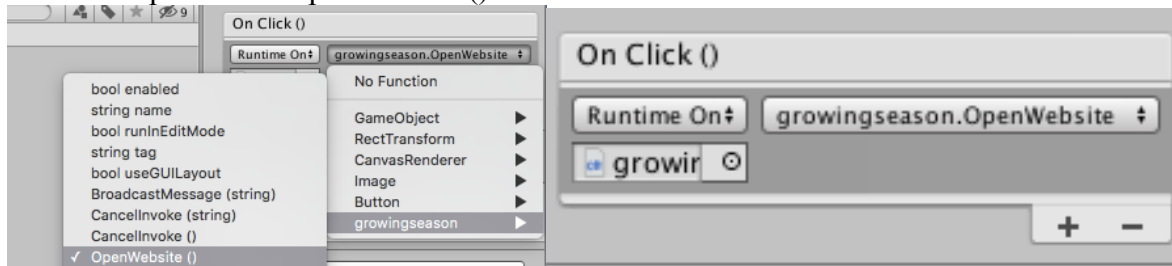
5. For each multiple-choice option/button, create another panel the same way as above. These panels will be smaller than the original panel, and a child of the original panel. The panel can be used to display whether the answer is correct or incorrect with text. Enable the buttons to open and close the answer panels in the On Click () area as before.
6. Make sure to place buttons and panels in the correct order of the hierarchy so that nothing is obscured. This includes the original question panel, as well as the panels that display the right or wrong answer.
7. This whole part can be very confusing. Make sure that your buttons are connected to the right panels, and that everything has close buttons (the question, right, and wrong panel).

Creating hyperlinks

1. Create a button the same way as above.
2. Drag the hyperlink script onto the button in the hierarchy, or use the Add Component method. *Note, there are two different scripts for whether the project will be run on the web or in a browser. For a desktop build, use the part of the script that has

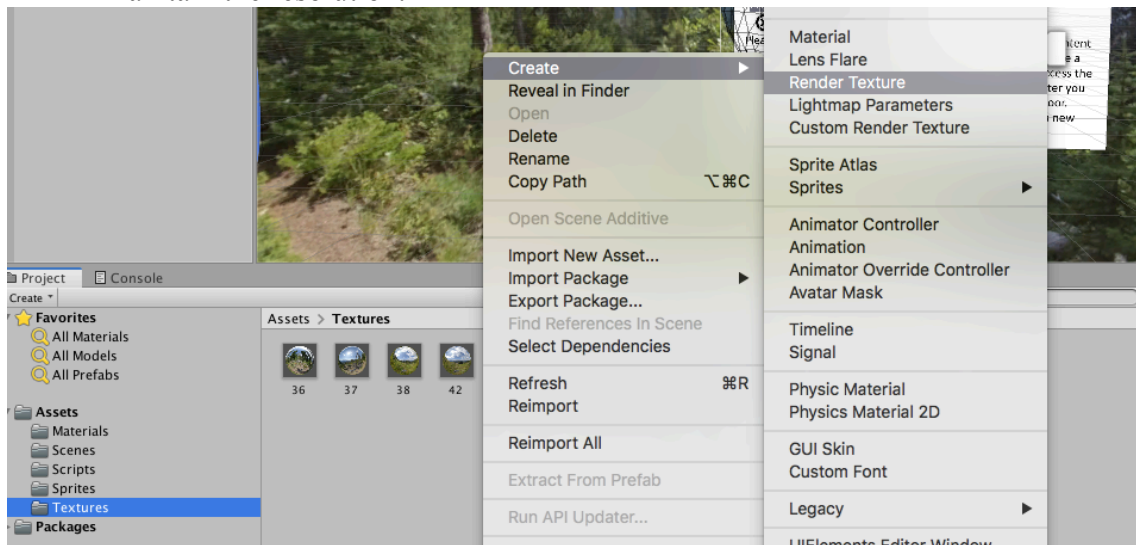
Application.OpenURL and delete the other line of code. For a web build, use the line that has Application.ExternalEval, and delete the other line of code. Replace the URL with your link. Make sure only to replace the underlined area (this is the URL). Ensure that the name of the script is the same as the name next to public class in the script. For URL included in the tour will need a unique script.

3. In the inspector of the button, press the plus sign near the On Click () box. Choose the button as the bottom left option, then in the top right as the function, choose the hyperlink script name > OpenWebsite ().

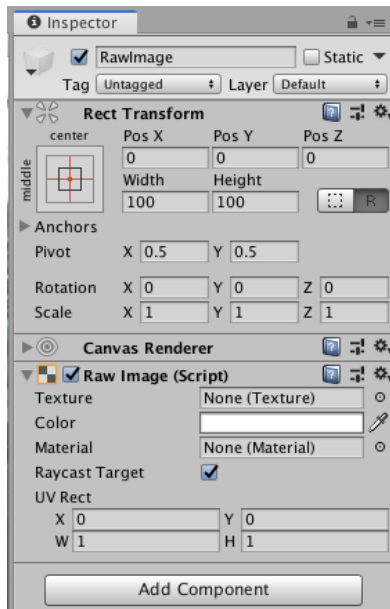


Embedding videos

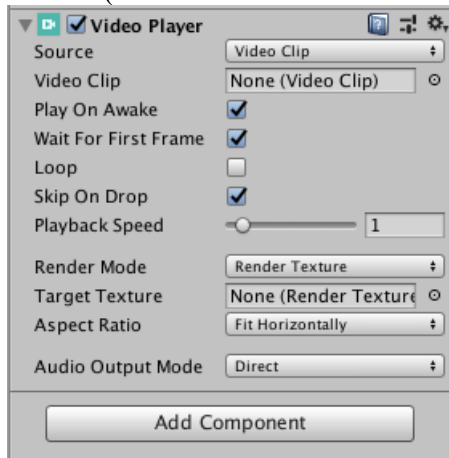
1. Drag all your videos into the Video folder in Assets.
2. Create a button the same way as above.
3. Create a Panel that you can open with the button, the same way as above.
4. In the Texture folder in Assets, right click, select Create > Render Texture. In the Inspector, set the size of the render texture to the same dimensions as the video to maintain the resolution.



5. Right click on the panel in the Hierarchy, select UI > Raw Image. In the inspector, set the height and width of the Raw Image to the same dimensions of the video to maintain resolution. Under Raw Image (Script), set the Texture of the Raw Image to the render texture created above. The raw image should be a child of the panel.



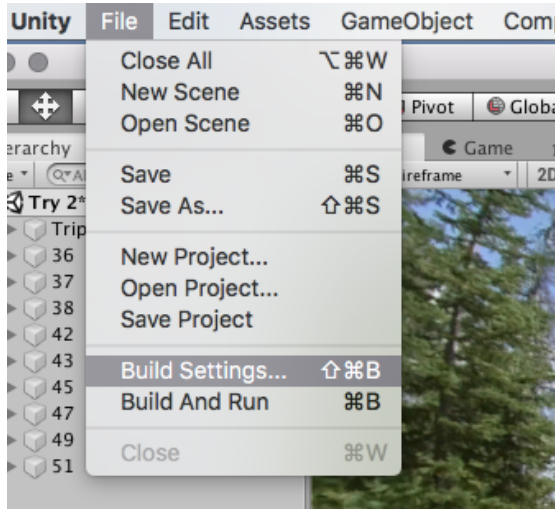
6. Select the panel that is housing the raw image. In the Inspector, click Add Component and search for Video Player. In the Inspector of the Panel, assign the desired video to Video Clip under the Video Player tab, and assign the previously created render texture to Target Texture under the Video Player tab. Make sure that Play On Awake is checked (this starts the video as soon as you click the button and the panel is opened).



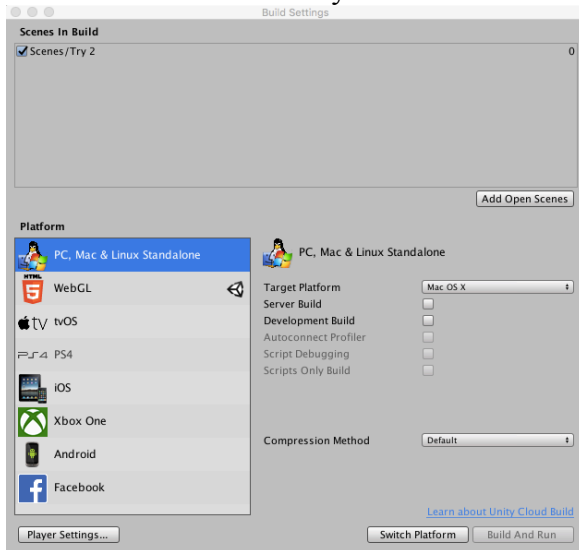
7. Add a close button at the bottom of the panel that closes the panel (and all of its children).
8. *Note: If the videos are large files, adding videos straight into the tour makes the build a very large file, which can make it hard to find websites that will support it. In the VR tour build, the videos were instead uploaded to Youtube and linked as a URL.

Building (Exporting) the project

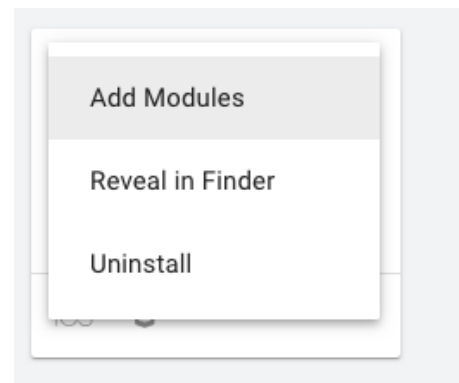
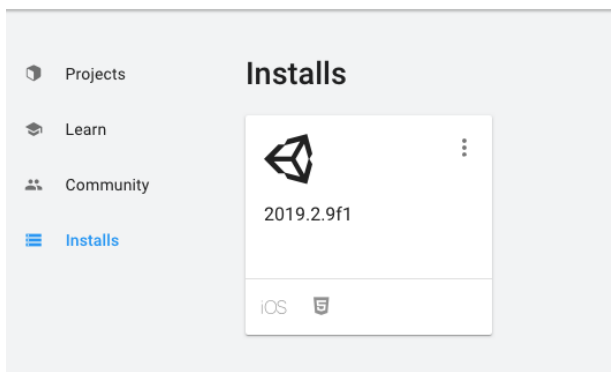
1. In the top bar, select File > Build Settings.

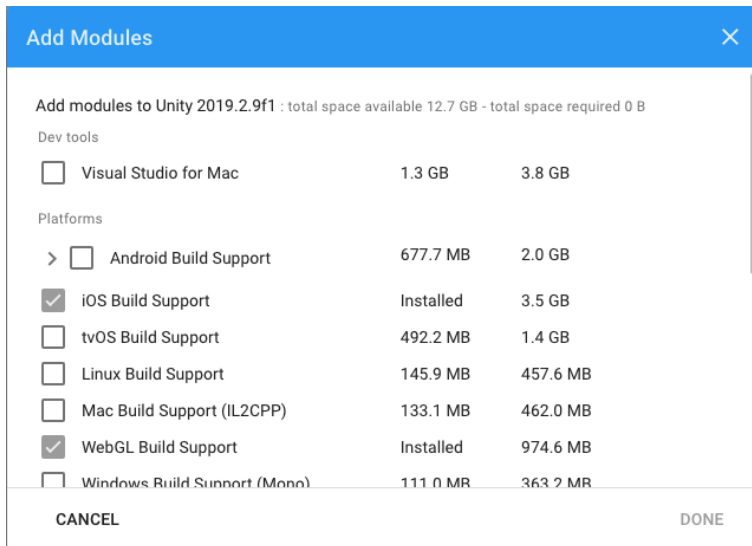


2. Select the platform that you would like to build the file for. The options are in the bottom left area. Choose the one you want and then click Switch Platform.



For some you will need to download the module. Do this by going to the Unity Hub, clicking Installs on the left menu, click the three dots in the top right of your Unity install, select Add Modules, then check the box next to the support you want and click Done.





3. At the top, click Add Open Scenes to add your project to the build
4. Click Build and save the file to your computer.

Uploading the tour onto Wordpress via simmer.io or itch.io

1. Create an account on simmer.io or itch.io (we used simmer.io for this upload).
2. For simmer:
 - a. On the homepage, click the upload icon in the top right.
 - b. Drag your WebGL build folder directly into the purple box on the upload screen, wait for file to upload, fill out the boxes and click save game.
 - c. Fill out all other boxes in the editor page, and save at the bottom.
 - d. Embed the game into wordpress by copying the code that starts with <iframe and ends with iframe> available in the Embed Your Game window or by copying the code at the bottom of the game in the <> icon.
 - e. Paste the code in the text part of an edit Wordpress page.
3. For itch:
 - a. On the homepage, click the down arrow in the top right. Choose Upload a New Project under CREATE from the dropdown menu. Give your project a title, URL, Classification, etc. Under Kind of Project, choose HTML.
 - b. On your computer, export the WebGL build as a zip file. Under Uploads, upload the zip and click This file will be played in the browser.
 - c. For Embed Options, choose Embed in page, and Manually set size. Set the height and width of the project as desired.
 - d. Under Frame options, check the Fullscreen button.
 - e. Choose the rest of the settings as desired.
 - f. Click Save and view page.
 - g. To embed the game, click Edit project at the top of the game window. Choose distribute along the top bar next to edit project in the next page. Choose Embed game from the left menu options. Copy and paste the code that starts with <iframe and ends with iframe> into the Wordpress text area.
4. For both itch and simmer, be aware of file size restrictions. The tour we are uploading is ~130MB.

Helpful resources used to learn all of the above:

Information about how to set up the basics of the project. *Note that none of them were followed exactly for building the tour, and rather for learning how to use Unity and taking different pieces of information:

<https://www.youtube.com/watch?v=2q9wc9Y35wg>

<https://medium.com/game-development-stuff/how-to-make-a-360%C2%BA-image-viewer-with-unity3d-b1aa9f99cabb>

<https://tutorialsforvr.com/how-do-i-build-vr-app/>

<https://tutorialsforvr.com/creating-virtual-tour-app-in-vr-using-unity/>

Code for rotating photospheres:

[http://www.andrewnoske.com/wiki/Unity - Panorama Viewer](http://www.andrewnoske.com/wiki/Unity_-_Panorama_Viewer)

Creating hyperlinks:

<https://www.youtube.com/watch?v=XBKLZAqPA1g>

Adding videos straight into the tour:

<https://www.youtube.com/watch?v=OdVGyV3rTGs>

Uploading to Simmer and embedding in Wordpress:

<https://www.youtube.com/watch?v=3Jpqvt6FGQM>

Uploading to itch:

<https://www.youtube.com/watch?v=H2HCXVAbe0w&t=147s>

If anything is unclear or you want to know how to do something else, just google it. There is so much information available on the internet relating to Unity.