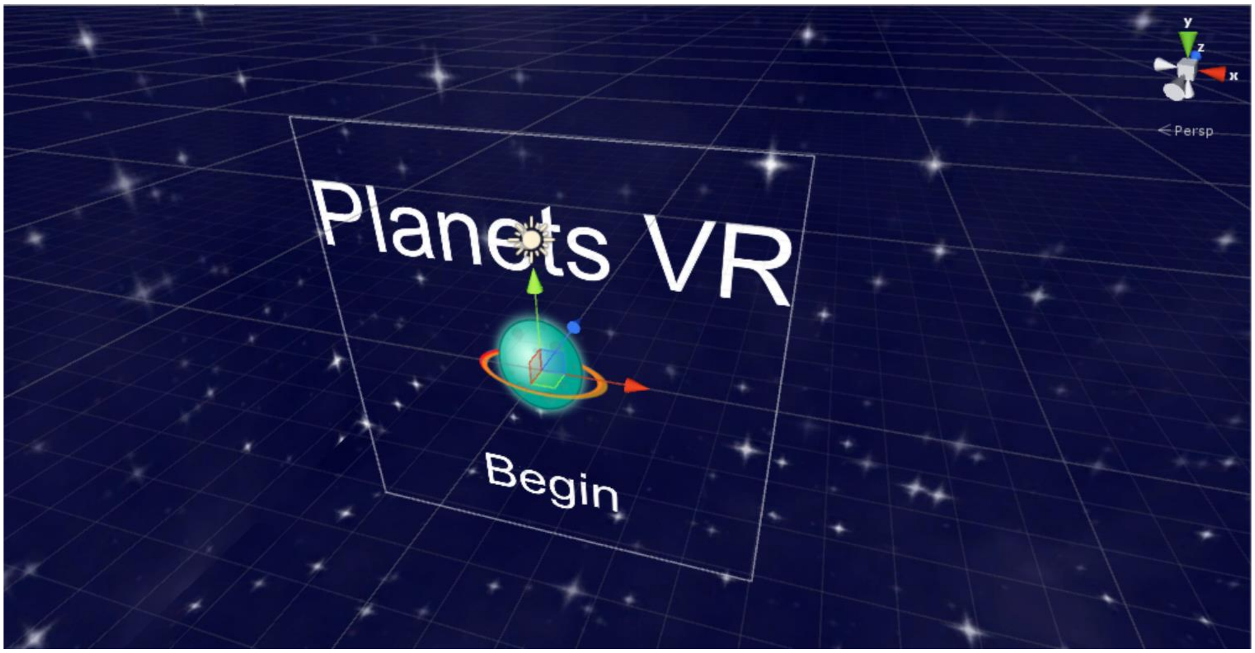
LAB #4 – PROJECT NOTES

BUILDING A USER INTERFACE IN UNITY

This is a companion document to the Lab #4 document that will cover step by step how to go through the process of creating a user interface using the built-in Unity GUI system. We will be continuing our work on the Space VR mini-app, and adding in a new menu for our experience.

After completing this lab, we will have two scenes in our project: One that contains our planets, and another that will be our main menu. For today's lab, you may either continue your project from last week if you completed the additional exercises, or use the included Week 4 project files.



USER INTERFACES

Reflect back on what we covered in our second week about how to design and interact with virtual environments. We will put those into action today by implementing our own UI in the Unity Editor for our application by creating a main menu and adding a text object to our planets.

CREATING A MAIN MENU SCENE

The first thing that we are going to create is a main menu for our scene. This will give us a place where the user can land and get acclimated to our experience and launch the main scene.

1. Click File > New Scene and name it UI_Scene

2. Following the instructions from last time, add your skybox to the scene that you created.
3. Add the GoogleVR viewer camera to your scene following the instructions from the last lab.

Think about the elements that you would expect when first launching an application. We will be adding in three elements to our main menu: a title, an icon, and a button to launch our experience's main scene.

CANVASES AND EVENT SYSTEMS

All UI elements in Unity require a Canvas object to be their parent. You may have more than one Canvas in your scene, but you cannot have elements such as text or buttons without a Canvas.

1. Right click in the Inspector window and select UI > Canvas to add a Canvas element to your scene
2. There are multiple types of canvas render modes you can choose from in a Unity scene:
    a. Screen Space – Overlay
    b. Screen Space – Camera
    c. World Space

   For VR, we'll want our canvases to be tied into a world location, so change the Canvas > Render Mode to World Space by selecting it from the drop down menu.

When you add in a canvas to your scene for the first time, Unity will also automatically include an Event System game object. We'll use this in the next lab to handle the interactions and events in our scene.

We'll want to modify a few things about our Canvas so that we can see it from our main camera:

1. Select the Canvas element in the Hierarchy
2. Under the 'Rect Transform' element, set the following attributes:
    a. Position: X, Y, Z = 0
    b. Width and Height: 10
3. Under 'Canvas Scalar (Script)' element, set the 'Dynamic Pixels Per Unit' attribute to 10

CREATING A MENU

Now that we have our Canvas element in place, we'll want to start adding objects to it! The first thing that we'll do is put in a text element that will be our title.

1. Select the Canvas in the Hierarchy and make sure it's highlighted
2. Right click while the Canvas is highlighted and click UI > Text to add a Text object to the Canvas
3. Click on the Text object in the Hierarchy and change the following attributes in the Inspector:
    a. Position: X = 0, Y = 1.64, Z = 0
    b. Width: 100, Height: 50
    c. Scale: X, Y, Z = 0.1
4. Under the Text (Script) attribute:
    a. Text: Give your application a name!
    b. Font Size: 20
    c. Horizontal and Vertical Overflow: Overflow
5. Click the Play button to view your title!

Once we've added our title, we're going to go ahead and give our application an icon. The first thing we'll want to do is bring in an image to use as our icon and convert it into a sprite:

1. Find an image that you'd like to use (check licenses before using in your project!) or create your own icon.
2. Import your icon by right clicking in the Assets panel and choosing Import New Asset and locating your file. For best results, use a .png file that has a transparent background.
3. Select your newly imported file in the Assets panel to open the Inspector window
4. Under 'Texture Type', select 'Sprite (2D and UI)
5. Click the Apply button

Now that we have a sprite to use, we're going to go ahead and add it as an image to our menu:

1. With the Canvas selected, right click and add UI > Image
2. Select the image and set the following attributes in the Inspector:
    a. Position: X = 0, Y = 0.1, Z = 0
    b. Width: 3, Height: 3 – *though if you choose a non-square image, this may need to be different!*
3. Under the Image (Script) box, select the small circle to the right of the Source Image box and locate your new sprite in the Assets tab. You should see your sprite appear on your menu!

Lastly, we want to add a button to our scene so that we can have the user click on it to launch the main scene.

1. With the Canvas selected, right click and add UI > Button
2. Select the button and set the following attributes in the Inspector:
    a. Position: X = 0, Y = -3.24, Z = 0

    b. Width: 10, Height: 2

3. Select the small circle to the right of the Source Image box and select None (Sprite)
4. Click the color to bring up the color selector and set the Alpha value to zero so that the button background is transparent
5. Expand the Button object in the Inspector and click on the Text object that is a child of the button
6. Change the text to “Begin” and set the overflow options to Overflow, as we did above
7. Change the text color to one that is visible against your skybox

Test your menu’s appearance by running your experience! Now that we’ve finished here, we’ll move on to the next stage of our application by adding UI elements to our planets in the other scene.

ADDING UI ELEMENTS TO 3D OBJECTS

Switch over to your other scene by double clicking on it in the Asset window – make sure to save your main menu scene first! We’re going to add a UI to our planets to display the name and some information about the planet.

1. Select one of the planets in your main scene by choosing the root object (top most parent) in the Inspector.
2. Right click to create a new Empty Game Object and name it ‘PlanetCanvas’.
3. On the PlanetCanvas object, right click and add UI > Text. We can skip adding the Canvas manually, as Unity will automatically add one to parent our Text object
4. Select the Canvas object in the Hierarchy and set the following attributes in the Inspector:
    a. Render Mode: World Space
    b. Position: X = 0, Y = -.5, Z = 0
    c. Width and Height: 2
    d. Scale: X, Y, and Z = 0.5
    e. Dynamic Pixels Per Unit: 100
5. Rotate the Canvas to face your camera
6. Select the Text item that is a child of the Canvas we just added to our planet and set the following attributes:
    a. Position: X, Y, Z = 0
    b. Width and Height: 2
    c. Scale: X, Y, and Z = 0.1
    d. Set your Text to whatever planet you’ve selected
    e. Change your Text settings for both overflow options to Overflow
    f. Select the Best Fit checkbox and make the minimum font 0
7. Press Play to view your scene and make any adjustments necessary!

You can turn your PlanetCanvas into a prefab object to make it easier to add text elements to other planets.

1. In the Asset folder, right click and Create > Prefab. Name the Prefab PlanetCanvas.
2. Drag your PlanetCanvas from the Hierarchy into the prefab. You can now use the PlanetCanvas prefab in your project!

NEXT TIME

For our next lab, we will begin learning about scripting in C# for Unity development and add in behaviors to our scenes to handle navigational elements and showing/hiding the UI that we have added to our planets when we “click” on them with our headset trigger.

Before you come to the next lab, finish this week’s lab by adding UI text to the rest of the objects in your scene. Alternatively, download the zip folder for Week 5 before attending the next lab.