Basic User Interface Creation

NOTE: Setup instructions should be viewed before moving to this step.

Student Goals

- Learn the details involved in the creation and manipulation of basic UI elements in Unity
- Create a personalized sample version of the Main Menu scene using these instructions
- Be able to build other UI's in Unity using the logic learned in these instructions

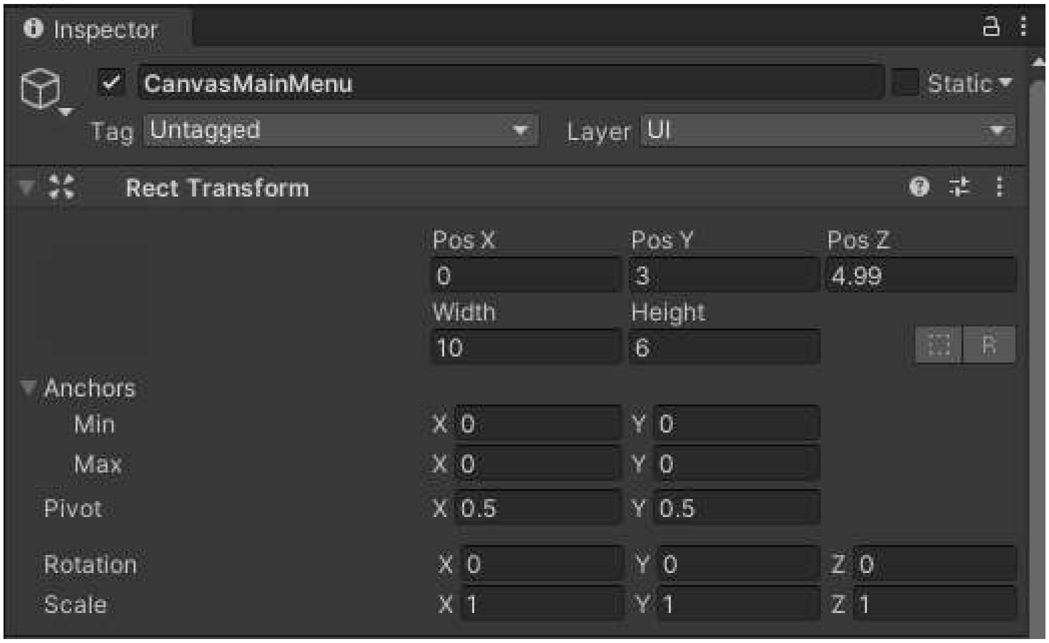
Instructions

Naviagate to the HelloCardboard scene

Canvas

1. Right-click on a blank spot in the hierarchy and select Canvas from the UI menu. A canvas is a holder for UI elements in a scene similar to how an art canvas is a holder for a painting. You can double click on the canvas object (or any object) in the hierarchy to get a better view of it in the scene window.
2. In the inspector for the canvas object, under the canvas component, change the render mode from screen space to world space. This is necessary so that the canvas can be placed anywhere we like in the scene rather than just taking up the whole screen like a pause menu would in a traditional none VR video game.
3. Still in the inspector for the canvas object, remove the graphic raycaster component by right-clicking the component title and selecting remove component from the menu. Add the GvrPointerPhysicsRaycaster and GvrPointerGraphicsRaycaster components. Do **NOT** change the settings of these components. They are necessary so that the reticle can detect the objects and respond to them in game. Also, do **NOT** alter the Canvas Scaler component.
4. Using the Pos X, Pos Y, Pos Z, Width, and Height elements of the Rect Transform component, move the the canvas wherever you would like it to be in the scene. For instance, to allow for the best view, we chose to place our canvas just barely in front of one of the walls of the CubeRoom.
5. For this particular project, do **NOT** change the name of the canvas object from its default. In later instructions, the default name will need to be the same across multiple scenes to make certain actions universal.

A sample canvas can use the following rect transform settings; however; you are free to use whatever position and dimensions you prefer. Your canvas name should be "Canvas"; this will differ from the images as Canvas already existed in our scene.



☑ CanvasMainMenu □ Static

Tag Untagged Layer UI

Pos X 0 Pos Y 3 Pos Z 4.99

Width 10 Height 6

Anchors

Min X 0 Y 0

Max X 0 Y 0

Pivot X 0.5 Y 0.5

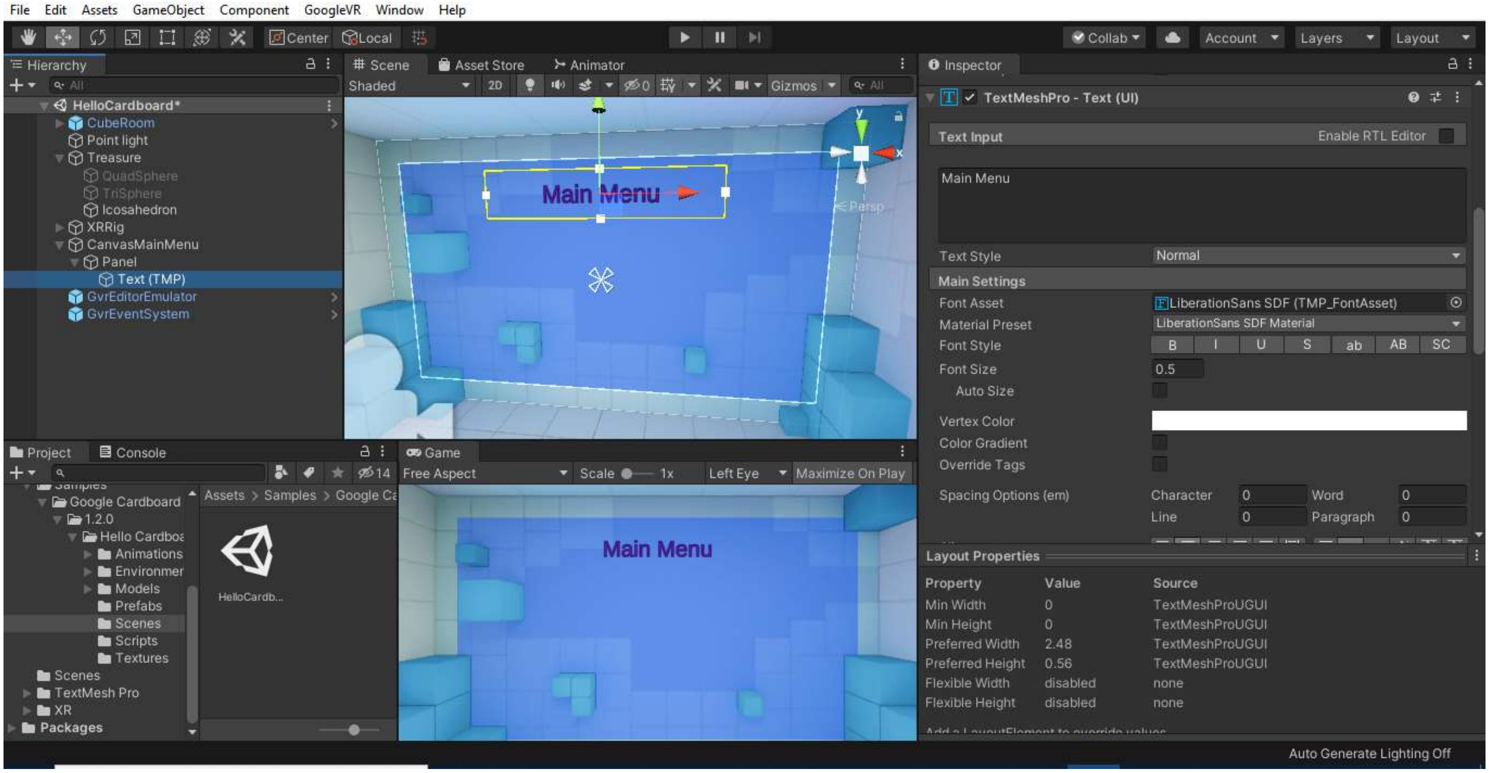
Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

Panel

1. Right-click on the canvas object in the hierarchy and select panel from the UI menu. Ensure that panel is a child of the canvas object.
2. In the inspector for the panel object, click on the shape box in the rect transform component and select center middle. This allows us to move the panel wherever we like instead of just having it be dependent on the canvas object. Next, similar to what you did for the canvas, use the rect transform to set the position and size of panel. You can place it wherever you want. It does not have to be on top of the canvas; however, that is where we placed ours for the HelloCardboard scene.
3. Make sure the image component is checked and set the source image of the panel to None (sprite) by clicking the target looking identifier on the source image bar and selecting None from the menu that comes up. Make sure that the Raycast Target and Maskable boxes are checked in the image component.
4. Set the color of the panel to whatever you like. To get the exact color we used, use color code 0045FF with an alpha value of 100.
5. Next, add Text - Text Mesh Pro as a child of panel from the UI menu. Set the position, color, and actual text to whatever you like. We chose to use the text as an identifier of the scene by setting our text to say Main Menu and positioning it at the top of the panel.

This is what ours looks like up to this point; however, your project will differ in the position, color, and text you selected.



TextMeshPro - Text (UI)

Text Input — Enable RTL Editor

Main Menu

Text Style: Normal

Main Settings

Font Asset: LiberationSans SDF (TMP_FontAsset)
Material Preset: LiberationSans SDF Material
Font Style: B I U S ab AB SC
Font Size: 0.5
Auto Size
Vertex Color
Color Gradient
Override Tags
Spacing Options (em): Character 0, Word 0, Line 0, Paragraph 0

Layout Properties

Property	Value	Source
Min Width	0	TextMeshProUGUI
Min Height	0	TextMeshProUGUI
Preferred Width	2.48	TextMeshProUGUI
Preferred Height	0.56	TextMeshProUGUI
Flexible Width	disabled	none
Flexible Height	disabled	none

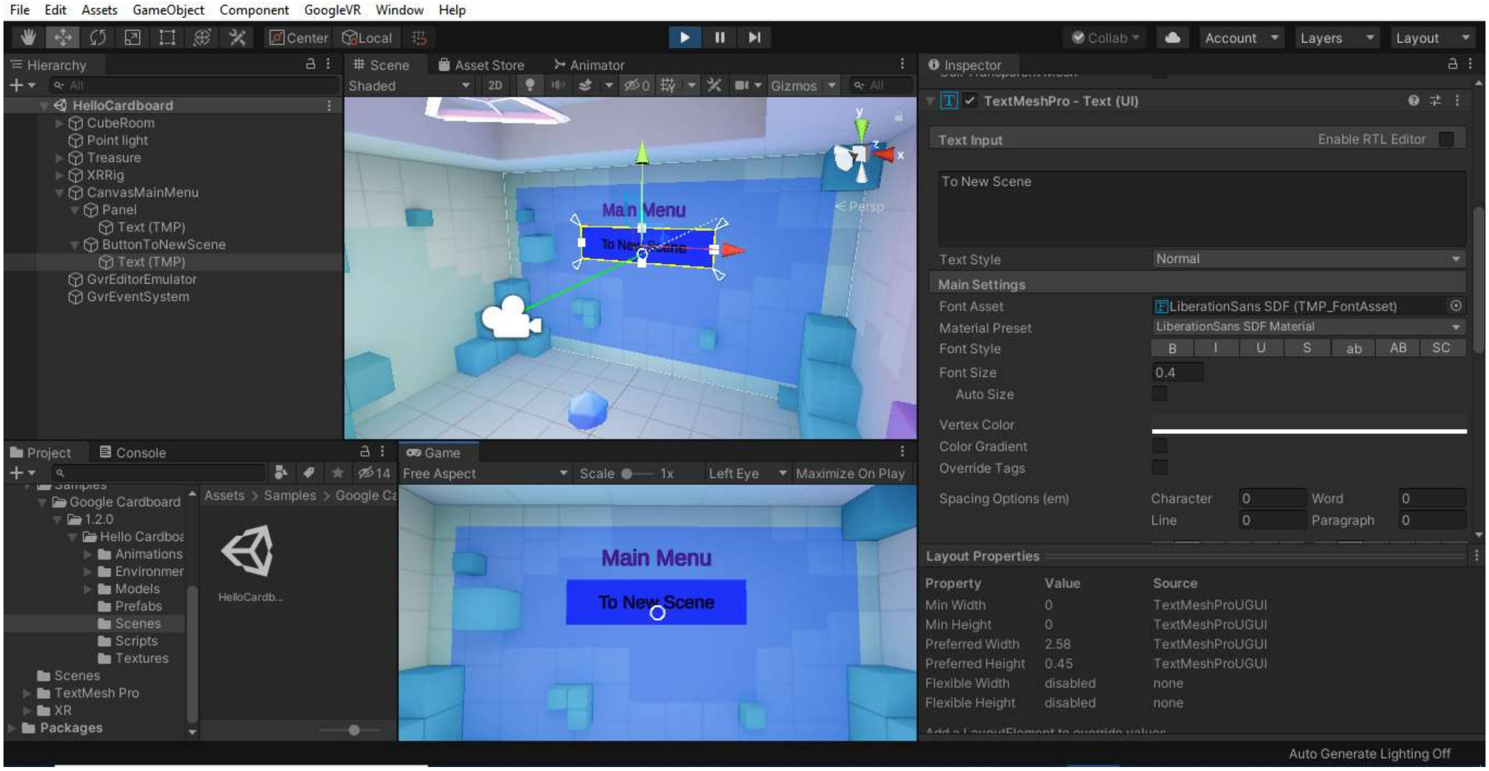
Add a LayoutElement to override values

Button

1. Add Button - Text Mesh Pro as a child of canvas but not of panel. It is recommended that you give your button a distinct name to distinguish it from other buttons that may be added in the future.
2. Follow steps 2, 3, and 4 of the Panel instructions with respect to the button. To copy our exact color for the button, use color code 8651FF with alpha 200.
3. In the inspector for the button, make sure the button component is checked. Make sure interactable is checked so that the button can be clicked, transition is set to color tint, and Target Graphic is set the button image. Target Graphic can be set by dragging the button object from hierarchy to the Target Graphic bar.
4. Leave the normal color as the original white. Set the highlighted and pressed colors to whatever you like. We used color code 1951FF with alpha 255 for the highlighted color and color code D951FF with alpha 255 for the pressed color.
5. Text (TMP) should have been automatically added as a child of the button object upon its creation. Select the child text object from the hierarchy and set the position and actual text to whatever you want. (*NOTE: The color will remain the same as previous text unless you change the material preset of the text or create a new material to apply to the text. Due to the challenge that both the options pose, they will not be covered in this tutorial.*)

At this point, you can test the project by pressing play and seeing how the button and reticle react when you hover over the button and press the button. The button has no function assigned to it at this point.

Action will be assigned to the button in later tutorials. Below is an example of what your HelloCardboard scene could look like at this point in game mode.



TextMesh Pro - Text (UI)

Text Input — Enable RTL Editor ☐

To New Scene

Text Style: Normal

Main Settings

Font Asset: LiberationSans SDF (TMP_FontAsset)

Material Preset: LiberationSans SDF Material

Font Style: B I U S ab AB SC

Font Size: 0.4

Auto Size: ☐

Vertex Color

Color Gradient: ☐

Override Tags: ☐

Spacing Options (em): Character 0, Word 0, Line 0, Paragraph 0

Layout Properties

Property	Value	Source
Min Width	0	TextMeshProUGUI
Min Height	0	TextMeshProUGUI
Preferred Width	2.58	TextMeshProUGUI
Preferred Height	0.45	TextMeshProUGUI
Flexible Width	disabled	none
Flexible Height	disabled	none