31263 / 32004 Game Programming Lab Week 10

Getting Started

1. Download the corresponding week's zip file from the Lab section of UTSOnline.
2. Unzip the project folder and open it in Unity. If there are any warnings about difference in versions, just continue. If this causes any red errors in the console once the project opens, notify the tutor.
3. Within the Weekly folders there are image and executable files starting with "Status…". These files give you a preview of what is expected for each point percentage below.
    a. If you are on Mac and the Status-100Percent App file won't run, hold control and click (right click) then select Open, if there is a security warning, acknowledge it an press open again.
    b. If you are running the Windows executable on the lab computers, you need to copy the entire executable folder into Windows(C:)/Users/<student number>/AppLockerExceptions. From there you can double run the .exe file.

Tasks

Points	Requirements
40%	• Create a new Scene called "StartScene". Make sure both StartScene and Walking Scene are in the build settings. Open StartScene. • Click on the "Main Camera". In the Inspector Window, change the "Clear Flags" to "Solid Color" and the "Background" to (0, 0, 0, 0) • Go to "Window->Lighting->Settings", under "Scene->Environment->Skybox Material" press the circle button and select "None". • Go to "GameObject->UI->Canvas". Double click the new Canvas in the Hierarchy Window to focus on it. • In the Scene Window, right click the coordinates gizmo in the top right of the window (that shows where the x, y, and z axis of the world are pointing) and select "Back"
50% (P)	• With the Canvas selected, go to GameObject->UI->Button • Select the newly created button and rename it to RoundButton in the Hierarchy. ○ In the Inspector, under the Rect Transform component, make sure Pos X and Pos Y are 0, Width = 110 and Height = 110.

	○ In the Inspector, under Image->Source Image click the circle button and click the "Button_theme17_round0" sprite texture. ○ Drop the RoundButton down to see the Text game object and click on it. ▪ Change the text to say "Start Game". ▪ Change the Font to "ComicRelief" and the Font Style to Bold ▪ Change the Color of the Text to pure white. ▪ At the bottom of the Inspector, click "Add Component" and type "Shadow" to add the in-built Shadow component. ▪ Change the Shadow Distance Effect to be x = -1.5 and y = -1.5 ▪ In the Rect Transform of the Text, set Bottom = 10
60% (P)	• Go to GameObject->UI->Image and rename it Border ○ Make sure the Border is a child of the RoundButton. ○ Change the Source Image of this Image to be "circle_border_md" and the Color to be (255,255,128,255) ○ In the Rect Transform component of the Border, set Pos X = 0, Pos Y = 3, Width = 105, Height = 105. • Save a prefab from the RoundButton. • Create another image and rename it Background ○ Make sure the Background is higher than the RoundButton in the Hierarchy Window (not a parent, just higher in the list under the canvas). This will make the Background image appear behind the RoundButton images ○ In the Background's Rect Transform, select the Anchor Presets gizmo which looks like a square with red target line in it and the words "middle" and "center" around it. ▪ In the small window that opens Hold shift and alt and click the bottom right most option to center the image and stretch it in both directions ○ Click Add Component and type "Gradient Effect" ▪ Set the "Top" and "Bottom" values to colors that you like.
70% (C)	• Create a new Script called UIManager. ○ Create a method called public void LoadFirstLevel() ○ In this method, load the WalkingScene while preserving the game object that this component is attached to. ○ Create an Empty GameObject called Managers and attach the UIManager to it. • On the RoundButton gameobject, under the Button component, click the + symbol on the " On Click ()" list ○ Drag the Managers gameobject onto the "None (Object)" slot. ○ From the drop down list that says "No Function", select UIManager->LoadFirstLevel".

	• Press play to test.

80% (D)	• Save this StartScene and open the WalkingScene
	• Create a UI Canvas and focus the Scene View on it.
	  ○ Drag the RoundButton prefab onto the Canvas.
	  ○ Make sure that the On Click() list of the RoundButton is empty.
	  ○ Create a new tag called "QuitButton" and assign it to the RoundButton's tag.
	  ○ Change the text of the button to "Quit"
	  ○ Change the Width and Height of the RoundButton to 60
	  ○ Change the Width and Height of the Border to 55
	  ○ Click the Anchor Presets gizmo for the RoundButton, hold shift and alt, and set the anchor to be Top-Right
	• Go back to the UIManagerScript:
	  ○ Import the UnityEngine.UI package with "using" keyword
	  ○ Create a public QuitGame() method which simply stops play mode in the editor by using UnityEditor.EditorApplication.isPlaying = false;
	  ○ Create the method public void OnSceneLoaded(Scene scene, LoadSceneMode mode)
	  ▪ If the scene in the parameters has the build index of the WalkingScene then find the GameObject with the tag "QuitButton", get the Button component.
	  ▪ With the Button component, add the QuitMethod() as a listener of the onClick property (hint: see Button.onClick)
	  ○ In LoadFirstLevel(), set OnSceneLoaded(…) as a C# delegate of SceneManager.sceneLoaded (hint: see SceneManager.sceneLoaded)
	• Test this by starting in the StartScene, clicking the "Start Game" button to load the WalkingScene, and then clicking the "Quit" button to stop the game.

90%	• In the WalkingScene, create another Canvas and rename it PlayerCanvas
	  ○ Make PlayerCanvas a child of the Player GameObject
	  ○ Change the Canvas "Render Mode" to "World Space"
	  ○ Change the Canvas Rect Transform to Pos X and Pos Y = 0 and Width and Height = 5
	• With the PlayerCanvas selected, create a new image and rename it HealthBar.
	  ○ Set Pos X = 0, Pos Y = 2.2, Width = 0.8, Height = 0.1, and Source Image "Fill_Round_lg"
	• Duplicate HealthBar, rename it InnerBar, and make it a child of HealthBar.
	  ○ Change the color to (0,255,0,255) and make the width 0.75 and height 0.05
	  ○ Change the "Image Type" to "Filled", "Fill Method" to "Horizontal", "Fill Origin" to

	"Left" and "Value" to 1 ○ Create a new Tag "PlayerHealthBar" and set it as the tag for InnerBar. • In UIManager: ○ In OnSceneLoaded, after getting the button component, also get the Image component of the InnerBar gameobject. ○ Also get the Transform of the Player GameObject by searching for the tag "Player" ○ Both of the above should be stored in private member variables. ○ In Update, if the Player GameObject is not null then: ▪ As the player moves away from the center x point of the world (i.e. x = 0), the players "health" should reduce. If the player is at X = 0, their health should be 1.0f, at x = +/-5 or greater their health should be 0.0f (hint: use Mathf.Abs, Mathf.Clamp, and normal ratio division of the player's x position) ▪ Use the above value to set the fill value of the InnerBar (hint: see Image.fillAmount). ▪ If the player health is <0.5f and the health bar is not red, then make it red (hint: see Image.Color and Color.red) ▪ If the player health is >0.5f and the health bar is red, then make it green. ○ When play is pressed from the StartScreen, the players health bar should be full and green at the centre of the screen and reducing and turning red as the player moves away from the centre.
100% (HD)	• Notice that the health bar is not facing the camera and that makes it hard to read. Turn the health bar into a "billboard" by having the health bar always facing the camera's backwards direction. ○ The health bar should always appear horizontal on the screen, like normal GUI text, but should be in position above the character's head (see Status 100%) ○ In both Status100% and your activity, use the J and L keys to rotate the camera. • When you press play, you may see some popping occasionally. This is because the health bar is a child of the player gameobject, so if the rotation of player gameobject (done in CharacterMovement) is done after the UIManager Update, then the inherited rotation overrides the camera facing rotation. ○ Fix this by doing the camera facing rotation in LateUpdate. • CLEAN, EFFICIENT, ELEGANT CODE!

Submission

When you complete the activity to the grade threshold that you want, you then need to:

1. Complete the “Status-StudentSubmission.txt” file in the highest level of the project folder.
2. Remove all other “Status-…” files and folders to reduce the size of your project.
3. Zip the entire project folder.
4. Re-name the zip file to **“[student ID]-LabWeek[week number].zip”.**
5. Submit the zip file to UTSOnline for the associated link for this week in the Lab **before Monday 9am of the following week**.
6. Failure to follow any of these could result in a 0% mark for that week.
7. You will also **demo your submission to your tutor** at the **start of the following week's lab.**

If you finish the activity early, show it to your tutor before you submit it on UTSOnline so they can help you make some final corrections and mark it at that time.