31263 / 32004 Game Programming Lab Week 2

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.

Tasks

Points	Requirements
40%	• Open the "BlankScene" from the Project Window (tip: search Learning the Interface in the Unity Docs) • Right click in the Project Window and create a new C# script called "LoadAssets" • Select the MainCamera from the Hierarchy Window and set its position to (0,0,-5) and its rotation to (0,0,0) in the Inspector Window. • Rename "BlankScene" to "Week2Scene". Press Ctrl+s (Cmd+s for Mac) to save the scene. Name it is "Week2Scene"

50% (P)	• Create a new Sphere primitive from the GameObject menu item.
	• Right Click in the Project Window and create a new material call RedMaterial.
	  ○ Click on RedMaterial and in the Inspector Window change the Albedo to sphere to red = 255, green = 0, and blue = 0, alpha = 255.
	  ○ Select the sphere on the Hierarchy View, then drag RedMaterial to "Materials -> Element 0" in the Inspector Window
	• Make a Prefab out of this sphere by dragging it to the Project Window and rename it to RedPrefab.
	• Repeat the above for a blue sphere:
	  ○ Create a BlueMaterial with an Albedo of (0,0,255,255).
	  ○ Assign it to Element 0 of the sphere (now called RedPrefab and highlighted in blue in the Hierarchy Window to show that it is an instance of a prefab)
	  ○ Create a new prefab called BluePrefab
	• Delete the sphere / any prefab instances from the Scene View / Hierarchy Window.

60% (P)	• Open up the LoadAssets script. • Create a public GameObject variable called redObj of the LoadAssets class (i.e. outside of any method).

	• Save the script and return to the Unity window to auto-compile it. • Create a new empty GameObject, rename it to "LoadManager", set all position and rotation values to 0, set all scale values to 1, and add the LoadAssets script as a component of this game object • Assign the RedPrefab prefab to the redObj variable of the LoadAssets component of the LoadManager game object.
70% (C)	• Create a private GameObject variable called blueObj of the LoadAssets class but make it available to the Inspector View (tip: search SerializeField in Unity Docs). • Open the LoadAssets script again. When the game starts, the LoadAssets component should create a redObj at (2,0,0) and a rotation of zero and a blueObj at (-2,0,0) and a rotation of zero (tip: search Instantiate, Vector3, and Quaternion.identity in Unity Docs). • Save the script, assign BluePrefab to blueObj on the LoadManager gameobject, and press play in Unity to see the results.
80% (D)	• Create a new script called "ConsolePrint" • Edit the RedPrefab and BluePrefab prefabs to have a ConsolePrint component. • For every frame, the ConsolePrint script should print the following "<gameObject name>: i" where <gameObject name> is the name of the game object that the component is attached to while i is an integer that increments every frame and is set to 0 at the start of the game. • Press play in Unity and observe what happens.

90% (HD)	• Change the name of ConsolePrint in the Project Window to "PrintAndHide" and fix any errors that arise.
	• In PrintAndHide make the member variable "public Renderer rend".
	  ○ Select RedPrefab in the Project Window, then drag it onto its own PrintAndHide.Rend reference.
	  ○ Repeat the above for BluePrefab.
	• Create a new Tag called "Red" and assign the RedPrefab Tag to this.
	• Create a new Tag called "Blue" and assign the label of BluePrefab Tag to this.

100% (HD)	• In the PrintAndHide script
	  ○ If the Tag of the game object is "Red" and i = 100, deactivate the game object.
	  ○ If the Tag of game object is "Blue" and i = a random integer between 200 and 250 which is generated when the game starts, disable the Renderer component of this object.
	• Press play in Unity and observer what happens, especially in the console.
	• CLEAN, EFFICIENT, ELEGANT CODE!

Submission

1. Complete the 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 midnight of Sunday of that week**.
6. Failure to follow any of these could result in a 0% mark for that week.