

31263 / 32004 Game Development

Lab Week 5

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 and 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%	<ul style="list-style-type: none">• Open the project folder in Unity. This is similar to the solution from last week so it should look familiar. Open Week5Scene if it is not already open.• Create an empty GameObject from the GameObject menu and call it "ParentObj" and set clear its position and rotation by selecting the gear icon of the Transform component and clicking "Reset".• Apply the PrintAndHide component to the ParentObj instance and change the tag to "Red".
50% (P)	<ul style="list-style-type: none">• Create a Cube primitive from the GameObject menu.• Scale it to (1.5,0.75,1), reset the position and rotation, and apply the RedMaterial to it.• Add the cube as a child of ParentObj by dragging it onto the ParentObj in the Hierarchy Window.• Associate the cube's Renderer with the ParentObj's PrintAndHide.rend variable in the Inspector Window.• Override the prefab reference of RedPrefab in the Project Window so that the prefab has this parent-child relationship. Do this by dragging ParentObj in the Hierarchy Window to RedPrefab in the ProjectWindow and accepting any warning.

	<ul style="list-style-type: none"> Do the same for the BluePrefab by changing the cube's material to BlueMaterial, the tag Blue, and replacing the BluePrefab in the Project Window. Delete the ParentObj and its child from the Hierarchy Window.
60% (P)	<ul style="list-style-type: none"> Modify the LoadAssets script so that when redObj is initiated it is placed at (2,1,0) and the blueObj is at (-2,-1,0). Create a new C# script called InputManager and attach it to the MainCamera. In InputManager, make a private member variable that is an array of Transforms called "transArray". When the game starts this script should: <ul style="list-style-type: none"> Declare transArray as an array of size 2. Assign the Transform components of redPrefab and bluePrefab to this array (tip: search GameObject.FindWithTag(...)) Important: InputManager should do its starting actions in Start() and LoadAssets should do its starting actions in Awake(). Consider why this is (see Week 4 Lecture) and take note of it!
70% (C)	<ul style="list-style-type: none"> In Update() of InputManager, detect whether the player has pressed the "W" (tip: see Input.GetKeyDown(...)) <ul style="list-style-type: none"> When pressed, the redObj should be rotated by 45 degree around the z-axis and the blueObj by -45 degrees around the z-axis. (tip: Search Transform.Rotate(...)) Does this without creating any new variables
80% (D)	<ul style="list-style-type: none"> Go to "Edit->Project Settings->Input" and edit the Fire1 axes to be bound to the "Q" key. In Update() of InputManager, detect whether the player has pressed the Fire key (tip: see Input.GetButtonDown(...)) <ul style="list-style-type: none"> When pressed, the redObj and blueObj should swap their vertical positions but keep their horizontal positions. (tip: search Transform.position and see Status-100Percent executable to see the correct behaviour) This flip-flopping should happen every time Fire1 is pressed and should regardless of what their starting position are.
90% (HD)	<ul style="list-style-type: none"> In InputManager, when the Fire1 key is released: <ul style="list-style-type: none"> Get the rend variable from the PrintAndHide component on the redObj. Change the color to be a random value between (51.0,0.0,0.0) and (255.0,0.0,0.0). (tip: see Transform.GetComponent<>(), Material.color, and the class Color – note that new Color(...) takes values between 0.0f and 1.0f for standard colors). Repeat for blueObj but for the range (0.0,0.0,51.0) and (0.0,0.0,255.0). Print both of these new values to the console in the form "Red: <color>" by passing the material.color variable directly to the debug statement. In PrintAndHide, comment out all the console printing.
100% (HD)	<ul style="list-style-type: none"> In InputManager, when the "E" key is pressed: <ul style="list-style-type: none"> If redObj and blueObj have a PrintAndHide component, then remove it.

- If redObj and blueObj do not have a PrintAndHide component, then add it and ensure that the gameobject it is added to is active.
- In PrintAndHide, when the component is added, it should:
 - Assign the rend variable if it is not already assigned.
 - Ensure that the renderer referenced by rend is enabled.
- Pressing play, then “E”, then “Q” will cause a NullReferenceException error in the console
 - Prevent the error such that pressing Play, “E”, then “Q” will still swap the vertical positions of the cubes but not change the cube colors.
- 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.