31263 / 32004 Game Programming Lab Week 7

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%	Scene Setup • Create a cube gameobject called "floor" at the origin of the scene, with scale (5,0.5,5) and a material color of (204,0,204). • Create a cube gameobject called "box" at position (0,0.5,0), with scale (0.5,0.5,0.5) and a material color of (255,255,0). • Move the main camera to position (0,3,-4) and rotation (45,0,0). • Create an empty gameobject called TweenController • Create scripts called Tweener and InputManager in the scripts folder and attach both to TweenController.
50% (P)	The Tween Class • Create another script called Tween and remove the " : Monobehaviour" statement at the top to make it a normal C# class. ○ This means Update and Start will never be called, so delete them as well. • In Tween, create the following auto-properties (tip: these properties will be publically accessible but can only be set privately. See "Properties" on MSDN C# docs): public Transform Target { get; private set; } public Vector3 StartPos { get; private set; public Vector3 EndPos { get; private set; }

	public float StartTime { get; private set; }
	public float Duration { get; private set; }
	• In Tweener, create a private member variable Tween activeTween.

60% (P)	<u>Tween Constructor and Adding Tweens to the Tweener</u> • In InputManager: ○ Create a serialized private gameobject member variable called "item" and assign the "box" to it in the inspector. ○ Create a private Tweener variable called "tweener" and use GetComponent<>() in Start() to assign it. • In Tween, create a constructor that assigns each of the properties. • In Tweener, create a public method called void AddTween(Transform targetObject, Vector3 startPos, Vector3 endPos, float duration). ○ If the activeTween is null, then instantiate it with the following line: activeTween = new Tween(targetObject, startPos, endPos, Time.time, duration);
70% (C)	<u>Adding Tweens based on Input</u> • In InputManager, detect when the following keys are pressed down and pass the following values to AddTween, along with the transform of "item" and the that transform's current position as the startPos parameter: ○ A - endPos = (-2.0f,0.5f,0.0f) and duration = 1.5f ○ D - endPos = (2.0f,0.5f,0.0f) and duration = 1.5f ○ S - endPos = (0.0f,0.5f,-2.0f) and duration = 0.5f ○ W - endPos = (0.0f,0.5f,2.0f) and duration = 0.5f
80% (D)	<u>Updating the Tweens and Moving Tween Targets</u> • In the Update() of Tweener ○ If the activeTween.Target is further than 0.1f units away from the activeTween.EndPos, then lerp towards the EndPos (tip: see Vector3.Distance and Vector3.Lerp) ▪ Calculate the fraction of the linear interpolation by time with the following: float timeFraction = (Time.time - activeTween.StartTime) / activeTween.Duration; ▪ The lerp should be calculated from StartPos to EndPos, not current position to EndPos. ○ If the activeTween.Target is less than or equal to 0.1f units away from activeTween.EndPos, then set activeTween.Target.position = activeTween.EndPos and null the activeTween variable.
90% (HD)	<u>Cubic Interpolation</u> • Rather than linear interpolation, convert it to cubic easing-in interpolation. ○ See http://easings.net for examples ○ See http://gizma.com/easing for interactive example and equations ○ Hint: this only requires one extra line of code at most! Look at your equation for calculating your time fraction.

100% (HD)

Multiple Tween Targets

- Make sure you look at Status-100Percent, press spacebar a few times, and then quickly press the w,a,s,d keys in any order (spam them!)
- In Tweener:
  - Comment out the activeTween declaration and add a private List<Tween> activeTweens.
  - Create a public method called TweenExists(Transform target)
    - If this transform already exists as a target in the activeTweens list, return true. Otherwise, return false.
  - Change AddTween() to return a bool.
    - Call TweenExists() with the target transform.
    - If TweenExists() returns false, then instantiate a new tween and add it to the list and return true.
    - If TweenExists() returns true, then AddTween() should return false.
  - In Update(), loop through all elements of the activeTweens list, updating all of their positions.
    - Once a tween has completed, remove it from the activeTweens list.
- In InputManager:
  - Create a private List<GameObject> itemList member variable and in start add “item” to it.
  - On spacebar, clone the “item” game object and put the clone at (0,0.5,0) and add the clone to the itemList.
  - When any of the movement keys are pressed:
  - Loop through the itemList and attempt to add a new tween.
    - If AddTween() returns true, the break out of the loop.
    - If AddTween() returns false for all items then do nothing.
- 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.