

ASSIGNMENT 3

INTRO, PURPOSE AND GOAL

1. To learn to control timing stuff so you can control exactly what happens and when

HAND-IN

Hand in before next week's Wednesday lecture, by pasting the URL to your github project next to your name in this sheet:

<https://docs.google.com/spreadsheets/d/1rekEFkgqSKu9matm9qia9eLW68W9Ph06gJlogIDCJ68/edit#gid=0>

RESSOURCES

1. Juice it or Loose it: <https://www.youtube.com/watch?v=Fy0aCDmgngxg>
2. Invoke / InvokeRepeating: <https://unity3d.com/learn/tutorials/topics/scripting/invoke>
3. Coroutines: <https://unity3d.com/learn/tutorials/topics/scripting/coroutines>
4. Timers: <https://docs.microsoft.com/en-us/dotnet/api/system.timers.timer?view=netframework-4.7.2>
5. Time.deltatime: <https://docs.unity3d.com/ScriptReference/Time-deltaTime.html>
6. Transform and position: <https://docs.unity3d.com/ScriptReference/Transform.html>
7. Another way to make stuff move: <https://docs.unity3d.com/ScriptReference/Vector3.MoveTowards.html>
8. Tweening manually: $x += (\text{target} - x) * 0.1$
9. Lerp & SmoothDamp: <https://docs.unity3d.com/ScriptReference/Vector3.Lerp.html> / <https://docs.unity3d.com/ScriptReference/Vector3.SmoothDamp.html>
10. Tweening library: DoTween or other. <https://assetstore.unity.com/packages/tools/animation/dotween-hotween-v2-27676>
11. Animation: <https://unity3d.com/learn/tutorials/s/animation>
12. Triggers and ontriggerenter: <https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnTriggerEnter.html>
13. Rotations degrees and radians: <https://docs.unity3d.com/ScriptReference/Transform.Rotate.html>
14. GetComponent / RequiredComponent / addcomponent: <https://docs.unity3d.com/ScriptReference/GameObject.AddComponent.html> / <https://docs.unity3d.com/ScriptReference/GameObject.GetComponent.html> / <https://docs.unity3d.com/ScriptReference/RequireComponent.html>
15. GameObject.Find: <https://docs.unity3d.com/ScriptReference/GameObject.Find.html>

ASSIGNMENT

GREEN (MINIMUM REQUIREMENTS)

1. In your flight simulator game you made in assignment 2, you must make the following improvements:
 - a. Use a coroutine to make the powerups sometimes move, and sometimes stand still for a while. The movement of the powerup should not be physics(Rigidbody/force) based.
 - b. Use a tweening library to animate the score counter in the UI every time a point is scored. The animation could be a rapid expansion and deflation of size, or a shake, or both. This shows the user that something has happened in the UI.

2. In your racing car game from assignment 2, make a timer, so that the player can run out of time. The player gains more time by driving over the ramps and hitting invisible triggers on the ramps. The countdown timer must be visible on the screen.
3. Make the enemies in your third person shooter rotate around themselves.
4. Prep your project for GIT and upload to Github. paste URL here:
<https://docs.google.com/spreadsheets/d/1rekEFkgqSKu9matm9qia9eLW68W9Ph06gJloglDCJ68/edit#gid=0>

YELLOW (GO FOR IT)

1. Make your code safer and more durable and less error prone, by checking if required components are available or using RequiredComponent tag, and by checking for null if you search for GameObjects.
2. Try to use the Unity Animator instead of tweening other ways

RED (CHALLENGE YOURSELF)

1. Don't tween using a library, make your own static class which can take any GameObject as a parameter for one of its many methods, and tween it. Use Lerp and MoveTowards and other relevant methods to make multiple tween types available.