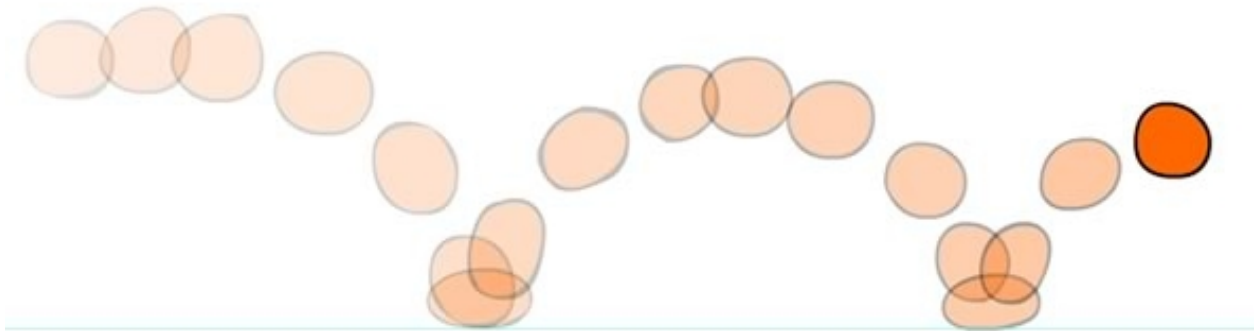


ASSIGNMENT 2: ANIMATIONS IN UNITY

CS 248 Winter 2017-2018

Due Date: Monday, 29 January by 6:30 pm



Introduction The goal of this assignment is to become familiar with the multiple methods of creating animations in a Unity game. Thus, you will create animations using 2 different techniques: (1) programmatically, and (2) either using the built-in Unity animation editor or with a third party tool such as Maya/Blender. You will then integrate these animations into the basic framework of a multi-scene game, using a small amount of scripting.

Background When building a game, it is very common to combine animations created in different ways. For example, when animating a character moving from point A to point B in a world, this is typically achieved using two different types of animation working together at once: programmatic (scripted) and pre-generated (using something like Maya).

For example, in an FPS game, a “walk cycle” (2 second animation loop of the character walking in place) is generated using a 3rd party tool like Maya. However, to actually make the character move through physical space in the game, you typically must do this programmatically (using nonlinear easing to make the motion look more realistic, as discussed in class). When the character moves through space, his walk cycle animation plays at the same time, giving the illusion of him actually walking around.

This combination is necessary because it would be impossible to use a 3rd party tool (like Maya) to create animations of all possible places the character could walk ahead of time, and conversely, it is very, very challenging to try and specify in code a realistic animation of all the limbs of a character moving together in tandem.

Implementation

For this assignment, you will be required to create 3 scenes containing the following:

1. Main Menu Scene:

- (a) Two Buttons - one to go into Demo Scene 1 and another to go into Demo Scene 2
- (b) An animation using a script (iTween is highly suggested)
 - i. For example, you can do a 2D animation in a background menu image, or maybe an animation of the two buttons moving

2. Demo Scene 1:

- (a) At least two objects
 - i. Each object should be playing a different animation loop imported from Maya/Blender/etc, or built using Unity's animation editor
- (b) Two cameras
 - i. One using perspective projection and another using a top-down orthographic projection
 - ii. Input of some kind to switch between cameras (e.g. pressing the spacebar)

3. Demo Scene 2:

- (a) Interactive Graphics Of Any Kind
 - i. Examples:
 - A. Fireworks that are created on mouse click (or touch, if using mobile)
 - B. Objects that chase towards your mouse position
 - C. RTS-style click command to move objects around
 - D. Checkerboard with pieces that can be dragged, dropped and snapped to a grid

Notes:

- In both demo scenes 1 & 2, input of some kind should return the user to the Main Menu scene (e.g. pressing the "Escape" key or an onscreen button).
- You should not use Unity's physics engine to simulate motion or character animations, as these are the topics of the next two homework assignments.
- You should not use Unity's prefabs for 1st and 3rd person controllers.

Additional Idea (Optional): Adding skyboxes to these scenes will make everything look much nicer. See the link in Resources for a tutorial on how to create them.

Grading Students will bring their laptops and demo their scenes during the grading session on Monday, Jan 29 in Gates 205, 208, and 210. The grading session lasts from 4:30 pm to 6:30 pm. Students may come any time within that window, go to any of the TAs in the 3 offices, and do not need to stay for the entire grading session. The assignment will be graded out of 10 points according to the following criteria:

- 2 points: Student animates a main menu title (or something more complex) using a script.
- 4 points: Student has at least 2 animations created using a third-party modeling tool (Maya/Blender/etc), or using the built-in editor.
- 1 point: Student has 2 cameras in Demo Scene #1 that can they can switch between.
- 2 points: Student has some sort of interactive graphics in Demo Scene #2.
- 1 point: Student has a way to switch through all 3 required scenes.

Documentation & Resources

iTWEEN

For your scripted animations, it is *highly* recommended to use iTween, an extremely popular and free library that makes animating an object in code very easy.

Code Example:

```
iTween.MoveBy(GameObject.Find("MainGameTitleImage"), iTween.Hash("y", -7, "easeType", "easeInOutExpo", "loopType", "none", "delay", 0.0, "time", 2.0));
```

The first parameter is the object to move, and the second parameter is a hashmap, with keys like:

- *y*: Move the selected object -7 units in the 'y' direction.
- *easeType*: Use the inOutExponential curve.
- *loopType*: Don't loop this animation.
- *delay*: Don't delay – play this animation right now.
- *time*: This animation should take 2.0 seconds to complete.

This one-liner would typically be placed in a script component on a GameObject. If for example, you wanted the animation to occur on scene start, it would be placed inside the C# method "Start()", inside of a script component.

- iTween, a widely used tool for animation in Unity (available for free from the Unity asset store): <http://itween.pixelplacement.com/index.php>
- Interactive graphical tool that illustrates various easing types (helpful for iTween): http://www.robertpenner.com/easing/easing_demo.html
- List of easing types available in iTween: <http://answers.unity3d.com/questions/53837/itween-list-of-easetypes.html>

MAYA/BLENDER/ETC

After you've created your animations using a 3rd party tool, the best way to make them compatible with Unity is to export them in the .fbx format (or if using Blender, you can just import the .blend file). Unity can easily read this format, and shouldn't lose any key frames or other valuable information. Assigned texture references might be broken, but re-attaching them is as simple as dragging and dropping the material onto your animated object inside of Unity (just like in assignment 1).

- Intro to animating with Maya: <https://www.youtube.com/watch?v=HSTRBRq3WqQ>
- Exporting Maya to .fbx: <https://www.youtube.com/watch?v=wL9ngU2oGMk>

- Intro to animating with Blender: <https://www.youtube.com/watch?v=n0VspDU0ErE>

- Taking an animation (.fbx) and importing it into Unity (Starting at 1m36s): <https://www.youtube.com/watch?v=J7Bncm2KGMo&t=1m36s>

BUILT-IN UNITY ANIMATION EDITOR

If you choose not to use a 3rd party tool, your animations can be created using the built-in Animation editor. It is highly recommended you use a 3rd party tool, as it is probably easier in the long run, and allows for much more complex and natural looking animations. That being said, this tool does allow for animations to be generated natively for any object in your scene, especially objects with a mesh. A good way to start might be to import the mesh you used in assignment #1, and then start adding animation to that. The animation editor is similar to the After-Effects animation video we watched in class (and also very similar to Maya's animation editor). After selecting your object, you can animate attributes on its components, notably the position/rotation/scale of the transform component.

Here's a link to a tutorial on how to create animations in Unity:

- This tutorial is for 2D buttons, but animating is the same for 3D models (part 1 of tutorial is unnecessary)
<http://www.raywenderlich.com/79031/unity-new-gui-tutorial-part-2>

LINKS

Some additional links are listed here for reference:

- Intro to scripting with Unity (starts @ 3:35)
<http://unity3d.com/learn/tutorials/projects/roll-a-ball/moving-the-player>
- Detect if mouse clicked (for example, on an image acting as a button...) :
<http://docs.unity3d.com/ScriptReference/MonoBehaviour.OnMouseDown.html>
– Note that the object being clicked on needs to be a GUIElement or a Collider
- Great work-around to allow GUIElements & iTween to Work Together:
<http://answers.unity3d.com/questions/25922/using-itweenmoveto-with-guitextures.html>
- Tutorial for creating Skyboxes in Unity: <https://www.youtube.com/watch?v=fSMcTLUorAo>
- Official tutorials, documentation, and other resources: <http://unity3d.com/learn>
- Unity's great tutorial on creating a basic interactive scene (Roll-A-Ball):
<http://unity3d.com/learn/tutorials/projects/roll-a-ball>