6.08.2019

31263 / 32004 Game Programming Lab Week 3

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
50% (P)	Making a 2D Sprite Sheet • Go to www.piskelapp.com and click Create Sprite • In the first frame (the default window that open up) create a 11x11 pixel box ○ Tip: You can see the cursor's pixel position in yellow on the bottom right • Create or duplicate 4 more frames ○ In each frame, indent the edge by one pixel ○ The preview of the animation in the top right of the screen should look like the box is pinching in on the sides ○ See the Status-50Percent image to see what this should look like. • On the right-hand side, select Resize ○ Set width and height to 128 ○ Tick "Resize canvas content" ○ Click the "Resize" button • On the right-hand side, select Export ○ Select the PNG tab ○ Change to 3 columns and 2 rows ○ Click download under "Spritesheet file export"

  - Rename the downloaded file to BoxPinch

60%
(P)

Importing the Sprite Sheet into Unity

- Open the Unity activity for this week, make sure Week3Scene is open.
- Create a new folder in the Project Window called Sprites
- Drag in the BoxPinch PNG file into this folder
- Select the newly created BoxPinch asset
  - The Inspector Window will show the Import Settings – that is, how this sprite is imported from the PNG file into a usable and compressed asset to be used in Unity.
  - Make sure the Texture Type is Sprite (2D and UI)
  - Sprite Mode = Multiple
  - Click the "Sprite Editor" button
  - In the window that opens, the first dropdown on the top left should be Sprite Editor
  - Click Slice in the top left, leave all of the settings in the dropdown menu and click the Slice button
  - You should see slight white boxes around each of the frames of your animation, indicating that these are each individual sprites now. You can select each box and adjust them if you want, but for this simple sprite sheet the automatic process should have worked well.
  - Exit out of the Sprite Editor window (and Apply changes if it asks), then click the arrow next to your BoxPinch sprite in the Project Window to see all the individual sprites of the animation.

70%
(C)

Creating an Animation Clip from the Sprite Sheet

- Create a new folder called Animations
- Open the "Animation" window and attach the tab somewhere on your Unity layout where it has plenty of horizontal space
- Drag the BoxPinch sprite into the Hierarchy View
  - It should instantiate in the Hierarchy Window as the name of the first sprite (e.g. BoxPinch_0). Rename it to "Box"
  - Make sure it has the default transform settings by clicking the cog symbol next to the Transform component name in the Inspector Window and click "Reset"
- Select the Camera, set it to Orthographic and Size 1
- With Box selected, the click "Create" in the Animation Window
  - Name it "PinchAnim" and save it to the Animations folder
  - This also creates an Animator Controller in the same folder. Rename this to "SimpleAnimator". If you select Box again, you will see that this Animator has been attached as a component in the Inspector Window.
- With either Box selected in the Hierarchy Window or SimpleAnimator selected in the Project Window, make sure PinchAnim is selected in the drop-down on the top left of the Animation window
  - Select all of the BoxPinch sprites (e.g. BoxPinch_0, BoxPinch_1, etc.) and drag them into the animation timeline
  - If you press the play button in the top left of the animation window, you can see

	that the animation plays very quickly. This is because the sprites have been added at 1/60th of a second intervals (i.e the animation frame changes once every 0.017 seconds or 17ms). ○ Change this so that there are 10 samples (intervals) in a second. That is, each frame lasts for 0.1 seconds (or 100ms) before switching, and the entire 5 frame animation lasts for 0.5 seconds
80% (D)	Creating a 2 State Animator Controller • Double click the SimpleAnimator asset to open it in the Animator Window ○ This view is known as a "State Machine" – boxes are states and there are arrowed lines that connect the states. At any one time, the animator exists in a specific state and is playing the animation associated with that state. ○ It should have 4 existing states - "Any State", "Entry", "Exit", and "PinchAnim". Entry is called as soon as the game starts, it then transitions to PinchAnim, which loops its animation continuously. ○ If you select the Box gameobject and press the main play button, you will see in the Animator Window a blue bar under PinchAnim that corresponds with the playing of the animation. ○ You can find out more at https://docs.unity3d.com/Manual/class-AnimatorController.html . • Create a new empty state in SimpleAnimator ○ Call it "Reverse" ○ Assign the PinchAnim to the Motion field. • Create a new transition from the PinchAnim state to the Reverse state. ○ Edit this transition so that as soon as the PinchAnim state finishes, it immediately switches to the Reverse state. ○ There should be no blending of the animations (e.g. no transition period where both animations are playing) • Do the same for a transition going back the other way ○ If you select Box and press play, you should see in the Animator Window that the state machine switches back a forth between the two states, each one play for half a second before transitioning back to the other. ○ However, because they have the same animation clip, if you look at the Scene or Game View windows, the Box gameobject looks like it is animating the same way as before (when it was only playing PinchAnim). • Change the Reverse state to play the animation backwards ○ The result should be that when you press play, the Box gameobject pinches in and then grows backout again, and then repeats. ○ Hint: it is a single value change
90% (HD)	3 State Animator with Animation created in Unity • In the Activity folder for this week, there is a script called "PlayRotation". Import this into Unity ○ All this script does is wait for the player to press the Spacebar key and then sets a parameter called "RotateParam" on a referenced Animator component.

	• Create a new animation of the Box rotating 90 degrees around the z-axis. This should take 0.5 second to complete the rotation • Create a new state in the SimpleAnimator to play this animation when the following conditions have been met: ○ The Reverse state has just completed its animation ○ The “RotateParam” trigger parameter has been set. • It should then continue on by transitioning to PinchAnim • Hints: Some things that may be useful to look into ○ Creating a new Animation Clip ○ Animation Parameters and Transition Conditions ○ Transitions Priority List ○ Try the “Apply Root Motion” property ○ See Status-90Percent and Status-100Percent
100% (HD)	Playing Basic Audio • There are two audio clips in the Activity folder for this week’s lab. Import them into your project. • Set-up two Audio Sources ○ One to play the background music at 50% volume, on loop, when the game starts ○ The other to play whenever the Spacebar is pressed. Modify PlayRotation.cs for this.

Submission

When you complete the activity to the grade threshold that you want, you then need to:

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 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.