

Virtual Game Simulation: Animate and Control a Character in a Game

Challenge Problem and Resources



Developed by:

The teachers, students, and mentors in the
Gaming Research Integration for Learning Laboratory™ (GRILL™)
Summer 2014

TABLE OF CONTENTS

1. CHALLENGE PROBLEM: ANIMATING AND CONTROLLING A CHARACTER IN A GAME	3
1.1. THE TOOLS	3
1.2. THE CHALLENGE	3
1.3. EXTENSION	3
2. TUTORIALS.....	4
2.1. ANIMATING A MODEL USING AUTO-RIGGER	4
2.1.1. setting up .zip file.....	4
2.1.2. Rigging skanect model	5
2.1.3. Applying the animation.....	6
2.2. ANIMATING IN UNITY: FREE.....	9
2.2.1. setting up .zip file.....	9
2.2.2. Rigging skanect model	9
2.2.3. Setting Up Unity	10

1. CHALLENGE PROBLEM: ANIMATING AND CONTROLLING A CHARACTER IN A GAME

The training or gaming experience can be enhanced by introducing the participant or custom-created character into the simulation or game. In this option, students are challenged to import, animate, and control a real person or custom-created character to function as a character in an off-the-shelf or custom-created game or simulation. This solution would result in a real person or custom-created character that fully functions as a character in a game or simulation.

1.1. THE TOOLS

Examples of valuable tools for this challenge include 3ds Max, Sketchup, Kinect, Skanect, Blender, MeshLab, Mixamo Auto-Rigger, a game engine such as Unity or Source, and computer coding skills.

1.2. THE CHALLENGE

Import, animate, and control a real person or custom-created character to function as a character in an off-the-shelf or custom-created game or simulation. This solution would result in a real person or custom-created character that fully functions as a character in a game or simulation.

1.3. EXTENSION

As an extension of this option, students can streamline the import and animate process so that one could simply take a picture or scan of a new player and have it automatically animated and imported into the game.

2. TUTORIALS

Wright Scholars, in collaboration with educators and the GRILL™ team, created the tutorials described below as possible solutions to solve the challenge problem. At the time of creation these were working tutorials; however, with software updates and changes in technology, additional steps may be required. Teachers are encouraged to communicate any issues, problems, or suggested changes to these tutorials to ensure the dissemination of helpful materials to support challenge problem implementation.

2.1. ANIMATING A MODEL USING AUTO-RIGGER

This tutorial provides a guide for animating scanned 3D models or custom-created characters so that they may be imported into the game engine Unity. Auto-Rigger is utilized as it streamlines the animation process as compared to custom and time-intensive rigging in other 3D processors such as Blender or MeshLab. Suggested software included Skanect, MeshLab, Auto-rigger, and Unity 4.x.

This tutorial includes the following topics:

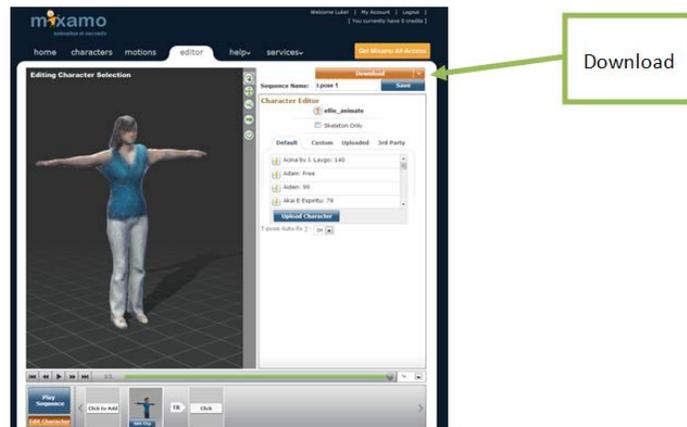
- Setting up .ZIP file (Section 2.1.1)
- Rigging a Skanect file (Section 2.1.2)
- Applying the animation (Section 2.1.3)

2.1.1. SETTING UP .ZIP FILE

- 1) Create a new folder on the desktop to hold the .obj, .png, and .mtl files that will be created in MeshLab.
- 2) Scan a model or load the folder of the scanned model in Skanect and simplify it to 10,000 faces (Mixamo Auto-rigger is free for characters that have fewer than 10,000 faces) and process the model in Meshlab using the following steps:
 - a) *File* → *Import Mesh* and open the .obj file previously saved.
 - b) *Filters* → *Texture* → *Parametrization: Trivial Per-Triangle* → *Apply*. If an error appears, make sure the number of faces is 62,000 or fewer.
 - c) *Filters* → *Texture* → *Transfer Vertex Attributes to Texture* (between 2 meshes). Make sure the *Assign Texture* box is checked. This will save a .png file, which will be used in Unity to give the object color, in the same place as the .obj file.
 - d) *File* → *Export Mesh As*. Change the .ply to an .obj extension and save the mesh. This saved file can replace the previous .obj file that was created.
- 3) Compress the folder by right clicking, selecting *Send to*, and choosing *Compressed (zipped) folder*.

2.1.2. RIGGING SKANECT MODEL

1. Go to www.mixamo.com and select Create Free Account if you do not have one already.
2. Go through the steps to sign up, select Upload File and select the .zip file. The model should appear on the next screen with textures applied. Make sure it is oriented so that the character is facing forward.



3. Click Rig, and drag the colored markers to their appropriate body parts and click Preview.
4. Click Finish and then click the Apply animation box. A collection of animation boxes should display on the screen. Scrolling over the boxes will give a preview of the animation. (you will only be using the free animations.)
5. Find the box titled T-pose and select it.
6. The model should now appear in the T-Pose under the Editor tab. Change the name of the model under Sequence Name and click on the down arrow in the orange download box. Select Character Only – Free.
7. Select Checkout on the next window that pops up. A Downloads tab will appear on the right side of the screen. Highlight the model by clicking it.
8. In the dropdown menu, change the pose to T-Pose.
9. At the bottom of the Downloads tab, change the format to FBX for Unity (.fbx) and select Download.
10. Open Unity and create a new project. Go to Assets > Import New Asset... and find the .fbx file under downloads and select Import. The model and an .fbm folder should appear in the Assets folder under the Project tab. The .fbm folder contains the texture.
11. Drag the model into the Scene and double click it in the Hierarchy tab to zoom to it.
12. Select the object in the Project tab.
13. Select the Inspector tab (on the right side when using the default layout), and there will be three tabs inside of it: Model, Rig, and Animations. Click on the Model tab and change the Scale Factor to 1 and select Apply.

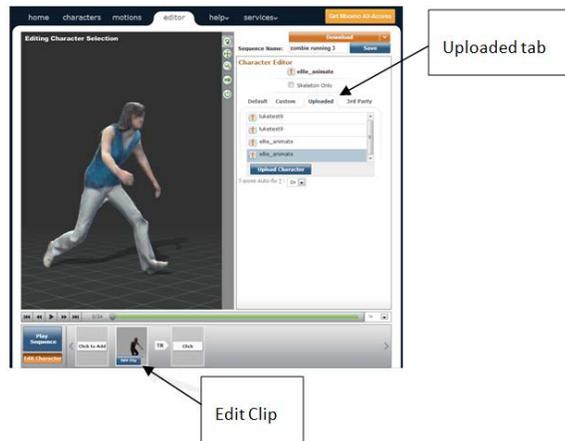
14. Go to the Rig tab and change the Animation Type to Humanoid. The Avatar Definition should be set to Create From This Model. Click Apply.
15. Go to the Animations tab and Uncheck Import Animation. Click Apply.

2.1.3. APPLYING THE ANIMATION

1. In Mixamo, select the Motions tab. In this tutorial, we will be using zombie running, but the process is the same for any animation.

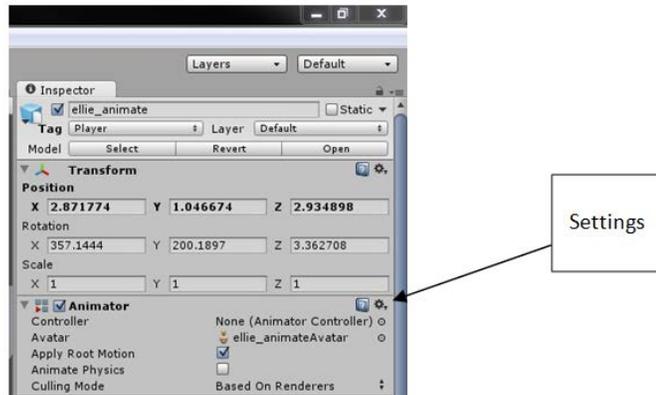
Note: Unfortunately, there are a limited amount of free animations so instead of walk and run, use zombie walking and running.

2. If the character does not appear in the preview, select the blue *Edit Character* box in the bottom panel and the character should be under the *Uploaded* tab in the right panel. Click it

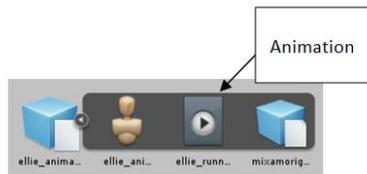


and then select *Edit Clip* in the sequence box at the bottom of the page.

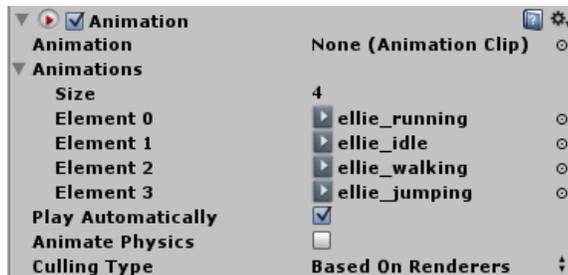
3. Here, you can adjust some of the animation settings. Make sure *In Place* is on (just for walk and run), name the animation, click the down arrow in the *Download* box, and select *Full Animation Sequence with Character* then select *Checkout*. The *Downloads* tab will display.
4. Select the animation and change from *Include Skin* to *Skeleton Only*. At the bottom, change the download format to FBX for Unity (.fbx) and select *Download*.
5. Go back to the Unity project and import the animation by going to *Assets*→*Import New Asset...* and go to animation located under downloads. An animation icon (blue cube) will display in the *Project* tab.
6. Select the animation icon, there are three tabs inside of the *Inspector* tab.
 - a. Select the *Model* tab, change the *Scale factor* to 1, and click *Apply*.
 - b. Select the *Rig* tab, change the *Animation Type* to *Legacy*, and click *Apply*.
 - c. Select the *Animations* tab, change *Anim. Compression* to *Off*, change *Wrap Mode* to *Loop* and click *Apply*.



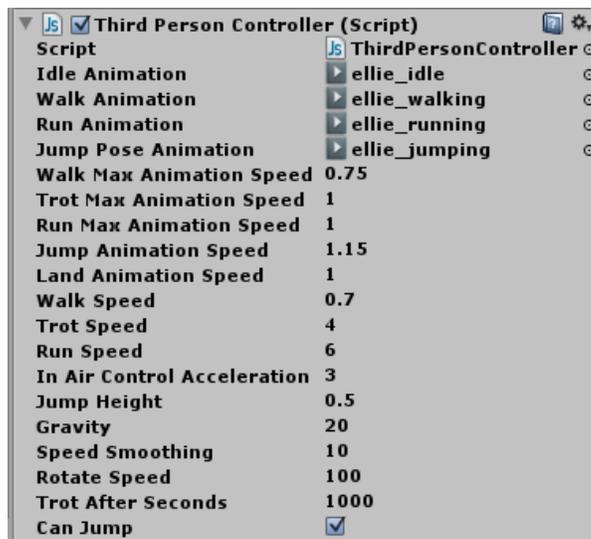
7. Repeat steps 1-6 for each of the movements.
8. Go to Assets > Import Package > Character Controller and select Import.
9. Select the model in the Hierarchy tab and go to:
 - a. Component → Physics → Character Controller
 - b. Component → Scripts → Third Person Controller
 - c. Component → Scripts → Third Person Camera: Component → Miscellaneous → Animation
10. Each of these components will appear in the Inspector tab. Find the Animator (different from Animation) component and delete it by clicking on the settings icon in the upper right hand corner and selecting Remove Component.
11. In the Project tab, expand the animation icon and find the animation.



12. Drag this animation into the Animation Component in the Inspector tab. Change the Culling Type to Based on Renderers.



13. Next, drag each animation into the *Third Person Controller* script where the specific animation is indicated.
14. Create a platform for the character to walk on. First, go to *Terrain*→*Create Terrain*, then go to *Terrain*→*Set Resolution* and change the *Terrain Width* and *Terrain Length* to 50. Move the character so that it is positioned above the terrain.
15. Create a light in the game so the character is visible. Go to *GameObject*→*Create Other*→*Directional Light*.



16. Click the play icon above the *Scene* window. The character can be controlled with the arrow keys or a,s,d,w (hold shift to run, hit space to jump).
17. Adjust the following settings as needed:
 - a. *Third Person Controller*: Walk Animation Speed, Run Animation Speed, Walk Speed, Run Speed, Rotate Speed, Trot After Seconds (set to high number so this will not occur)
 - b. *Third Person Camera*: Distance, Height, Angular Max Speed
 - c. *Character Controller*: If the character appears to start above the ground when play is pressed, the Y center may need increased to compensate.

2.2. ANIMATING IN UNITY: FREE

With the 3D scans from Kinect, you can create animated characters in Unity. This tutorial details an alternative, completely free way to animate a figure in Unity. Recommended software included MeshLab, Mecanim Tutorial, and Unity.

This tutorial includes the following topics:

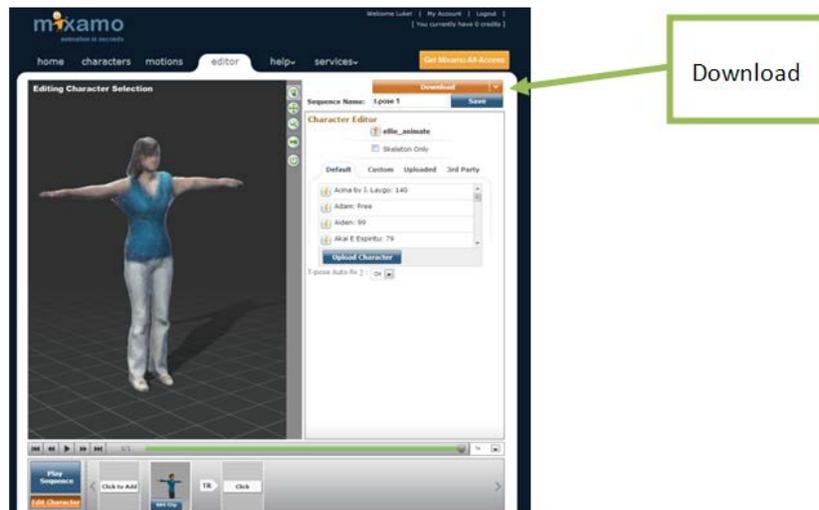
- Setting up a .ZIP file (Section 2.2.1)
- Rigging Skanect model (Section 2.2.2)
- Setting up Unity (Section 2.2.3)

2.2.1. SETTING UP .ZIP FILE

1. Scan a model or load the folder of the scanned model in Skanect and simplify it to 10,000 faces (Mixamo Auto-rigger is free for characters that have fewer than 10,000 faces) and export the model as an .obj. Process the model in Meshlab using the following steps:
 - a. *File* → *Import Mesh* and open the .obj file previously saved.
 - b. *Filters* → *Normals, Curvatures, and Orientation* → *Transform: Scale* → Change the *X, Y, and Z Axis* to 100 and click *Apply*.
 - c. *Filters* → *Texture* → *Parametrization: Trivial Per-Triangle* → *Apply*. If an error appears, make sure the number of faces is 62,000 or fewer.
 - d. *Filters* → *Texture* → *Transfer Vertex Attributes to Texture* (between two meshes). Make sure the *Assign Texture* box is checked. This will save a .png file, which you will use in Unity to give the object color in the same place as the .obj file.
 - e. *File* → *Export Mesh As*. Change the .ply to an .obj extension and save the mesh. This saved file can replace the previous .obj file that you created.
2. Place the .obj, .png, and .obj.mtl files in a new folder.
3. Compress the folder by right clicking, selecting *Send to*, and choosing *Compressed (zipped) folder*.

2.2.2. RIGGING SKANECT MODEL

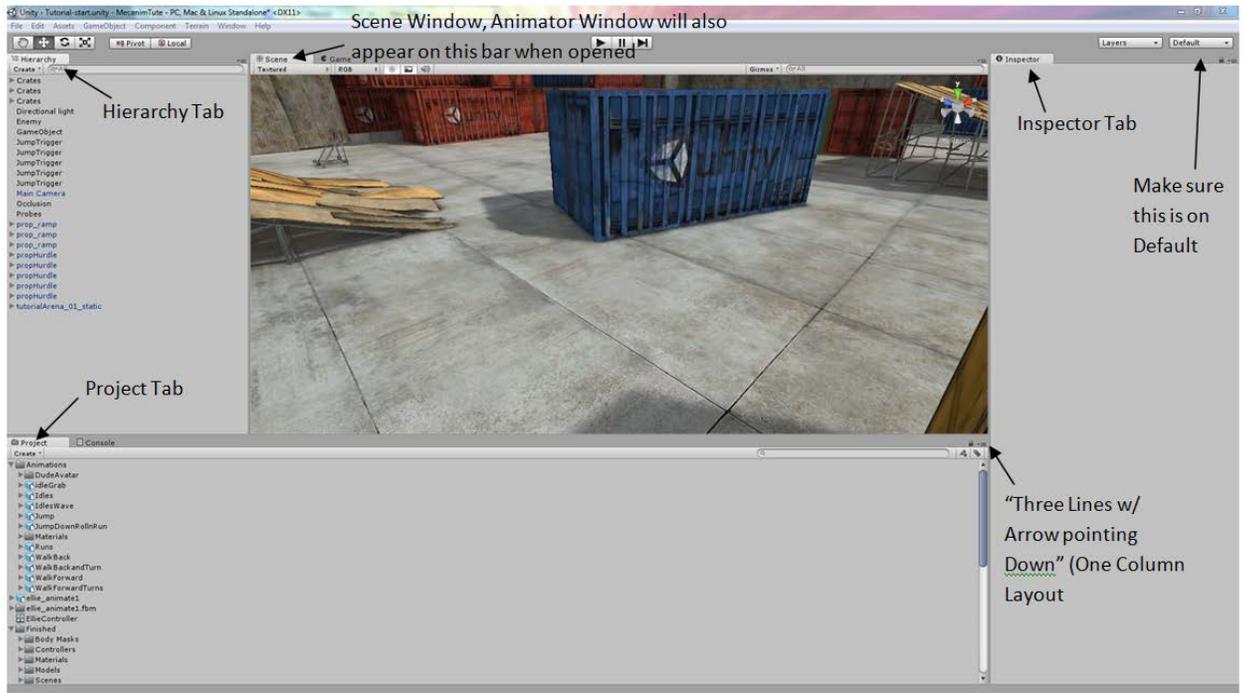
1. After logging into Mixamo Auto-rigger, upload the .zip file by selecting *Upload File* in the *Character* tab. The model should display on the next screen with textures applied. Make sure it is oriented so that the character is facing forward.
2. Click *Rig* and drag the colored markers to their appropriate body parts. Click *Preview*.
3. Click *Finish* → *Apply animation*. At this point, a collection of animation boxes should display on the screen. Scrolling over the boxes will give a preview of the animation
4. Select the *t-pose*. The model should now display in the T-Pose under the *Editor* tab.



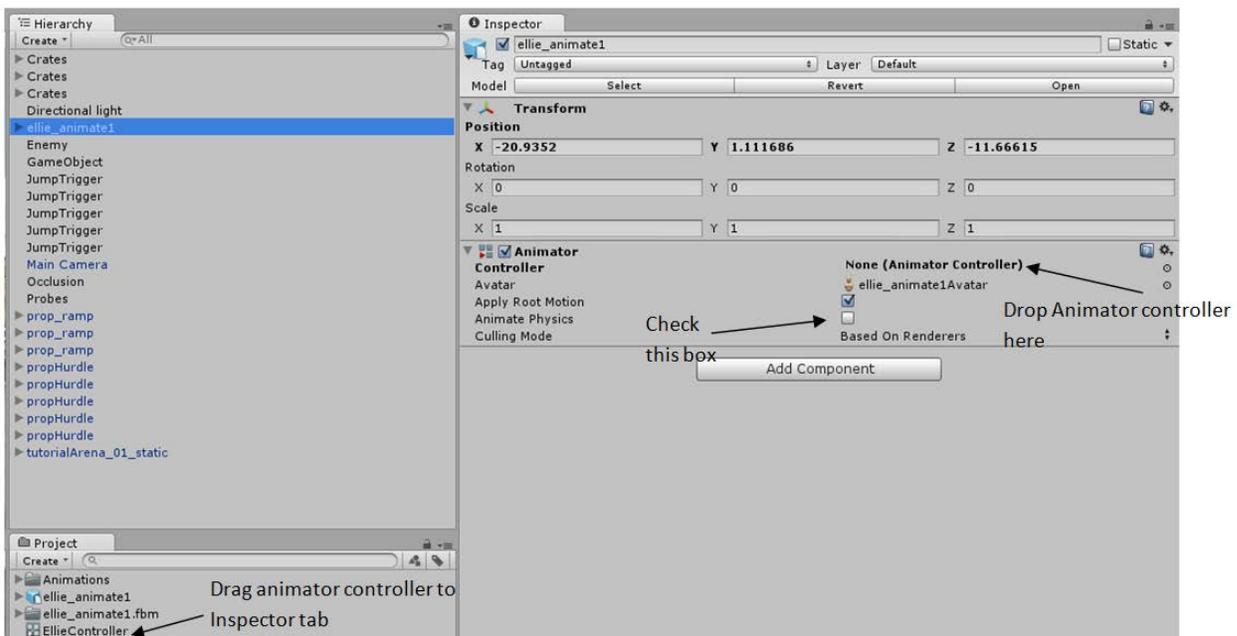
5. Change the name of the model under *Sequence Name* and click on the down arrow in the orange download box. Select *Character Only – Free*.
6. Select *Checkout* on the next window that pops up.
7. A *Downloads* tab will display on the right side of the screen. Click the model to highlight it. In the dropdown menu, change the pose to *T-Pose*.
8. At the bottom of the *Downloads* tab, change the format to FBX for Unity (.fbx) and select *Download*. Place the file in a location where you can easily find it.

2.2.3. SETTING UP UNITY

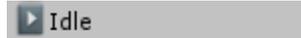
1. Download Mecanim Tutorial at <http://video.unity3d.com/video/7362044/unity-40-mecanim-animation-tutorial>.
2. Unzip and open the MecanimTute folder: MecanimTute → Assets → Tutorial → Tutorial-start Unity Scene File
3. Once Unity is open, go to Assets → Import New Asset... then find the Mixamo .FBX file and click Import. The FBX file will display in the Project Tab.
4. In the upper right hand corner of Unity there should be two drop down menus, one labeled Layers and the other labeled with either 2 by 3, 4 Split, Default, etc. Make sure that two menus are Layers and Default.
5. Under the Project tab, look to the right of the word Project and there will be a lock icon and three lines with an arrow pointing down. Click the three lines with the arrow and select One Column Layout. This will make it easier to see the layout of the Project tab.



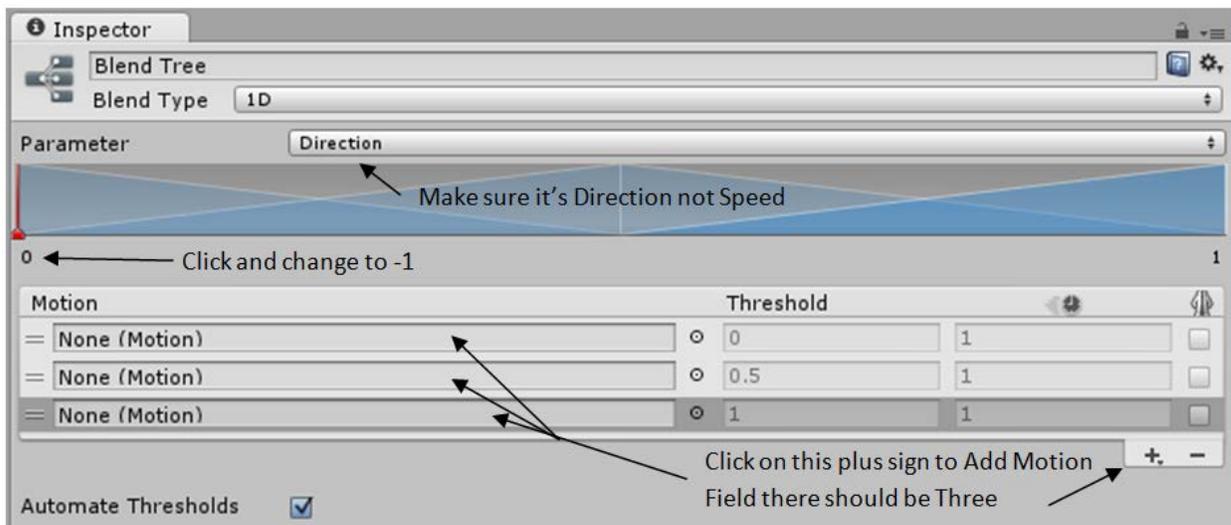
6. Click the FBX File. In the *Inspector* tab go to *Rig*, click on *Generic* and change it to *Humanoid*, click *Apply*, then click and drag it from the *Project* tab into the *Scene* window.
7. In the *Hierarchy* tab click the FBX file and change the *Position* to *X = -21, Y = 1, and Z = -10*.
8. In the *Project* tab, right click on the *Assets* folder and *Create* → *Animator Controller*. It will appear in the *Project* tab as *NewAnimatorController*. Rename it to something recognizable.
9. In the *Hierarchy* tab, click the character. From the *Project* tab, click and drag the *Animator Controller* to the *Inspector* tab and drop it under *Animator* → *Controller* → *None* (Animator Controller). Also under *Animator* check *Animate Physics*.



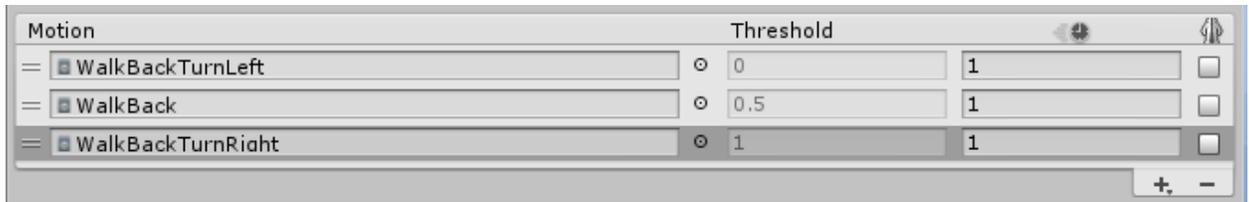
10. Go to *Window* → *Animator*, and another tab labeled *Animator* should display.
11. In the *Project* tab, click the *Animator Controller*. A gray area with a green box labeled *Any State* should display.
12. In the *Project* tab, expand the *Animations* folder and then expand the *Idle* box. Find the *Idle* animation.



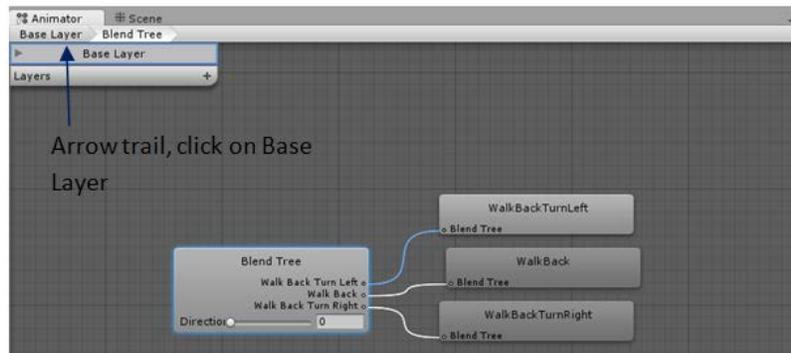
13. Click and drag the *Idle* animation into the *Animator* tab. An orange box labeled *Idle* will display in the *Animator* tab.
14. Right click in the *Animator* tab and *Create State* → *From New Blend Tree*. A gray box labeled *Blend Tree* will display; click it. At the top of the *Inspector* tab, rename it *Walk Back*.
15. Double click on *Walk Back* and a new screen should display with a single gray box labeled *Blend Tree*. In the bottom left hand corner there should be a box labeled *Parameters*. Click the plus sign → *Float*, and label this *Float Speed*.
16. Create another *Parameter* → *Float*; label it *Direction*.
17. Click on *Blend Tree* and, in the *Inspector* tab under *List is Empty*, click the plus sign → *Add Motion Field*. Do this two more times.
18. Above *Motion* is *Parameter* and it should be currently set to *Speed*. Click on *Speed* and change it to *Direction*.
19. Under *Parameter* there should be a box with a red line in it. Under the red line should be a 0. Click on the 0 and change it to -1.



20. In the *Project* tab, expand *WalkBack*. Find the *WalkBack* animation and drag it to the middle *None (Motion)* in the *Inspector* tab.
21. In the *Project* tab, expand *WalkBackandTurn*, find *WalkBackTurnLeft* and drag it to the top *None (Motion)* and then click and *WalkBackTurnRight* and drag it to the bottom *None (Motion)*.



22. Click Base Layer (below the labeled Animator tab is an arrow trail).
23. In the *Animator* tab, right click on *Idle* → *Make Transition*. An arrow will display. Drag the arrow to the *Walk Back* box and select it.

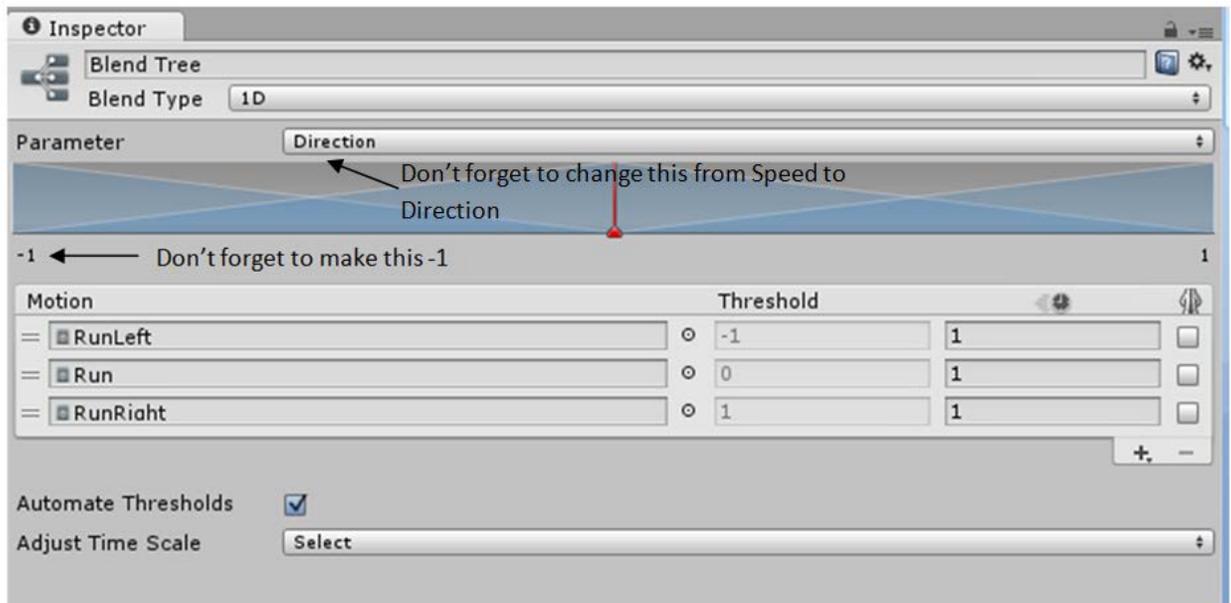


24. Right click on *Walk Back* → *Make Transition*. Drag the arrow to *Idle* and select it.
25. Click the arrow going from *Idle* to *Walk Back*. In the *Inspector* tab under *Conditions* click *Exit Time* and change it to *Speed*. Change *Greater* to *Less*, and change 0 to -0.1.
26. Click the arrow going from *Walk Back* to *Idle*. In the *Inspector* tab under *Conditions* click *Exit Time* and change it to *Speed*. Leave it on *Greater* and change 0 to -0.1.
27. Click the character in the *Hierarchy* tab. Go to *Component* → *Scripts* → *Bot Control Script*.
28. In the *Inspector* tab, under *Rigidbody*, change *Mass* to 50, expand *Constraints*, and under *Freeze Rotation* check *X*, *Y*, and *Z*. Under *Capsule Collider*, change *Center Y* = 1, *Radius* to .3, and *Height* to 2.
29. In the *Hierarchy* tab, click *Main Camera*. Go to *Component* → *Scripts* → *Third Person Camera*.
30. In the *Project* tab, expand the *Prefabs* folder, click and drag *CamPos*, and drop it on the *FBX* file in the *Hierarchy* tab

Note: If the camera jumps to a blue box, save and restart Unity and to fix the problem.

31. Press the *Play* button and use the down, right, and left arrow keys or A,S, and D keys to make sure that the character can walk backwards and turn. Exit *Play* mode.
32. Click the *Animator Controller* in the *Project* tab and select the *Animator* window. Right click in the *Animator* window and select *Create State* → *From New Blend Tree*.
33. Click the new blend tree and name it *Locomotion*.

34. Double click *Locomotion* and follow the same steps for the *Walk Back*. However, this time use the *Runs* animations from the *Project* tab.



35. Go back to the *Base Layer*, right click on *Locomotion* → *Make Transition* and select *Idle*.
36. Right click on *Idle* → *Make Transition* and select *Locomotion*.
37. Click the arrow going from *Idle* to *Locomotion*, change *Exit Time* to *Speed*, keep *Greater*, and change 0 to 0.1
38. Click the arrow going from *Locomotion* to *Idle*, change *Exit Time* to *Speed*, change *Greater* to *Less*, and change 0 to 0.1
39. Click *Play* to make sure that everything is working. Use all the arrows keys or A,S,D,W keys to move the character. Exit *Play* mode.

Note: If the camera jumps to a blue box, save and restart Unity to fix the problem.

40. Click on the *Animator* window, go to the *Project* tab in the *Animations* folder, and expand *Jump*. Click and drag the animation into the *Animator* window.
41. Right click on *Locomotion* → *Make Transition* and select *Jump*.
42. Right click on *Jump* → *Make Transition* and select *Locomotion*.
43. Click on the plus sign on *Parameters* (in the lower left hand corner) and select *Bool* and name it *Jump*
44. Click on the arrow going from *Locomotion* to *Jump* and change *Exit Time* to *Jump*, keep it true.
45. Press *Play* and with the arrow keys and space bar. The character should be able to run forward, walk backwards, and jump.