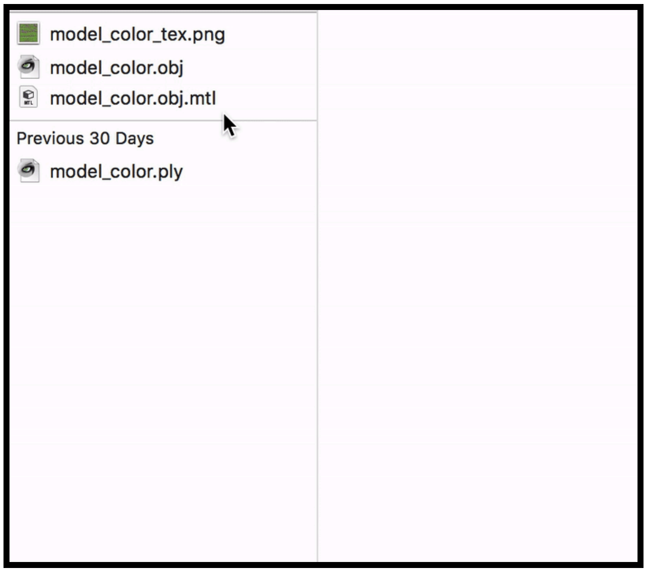
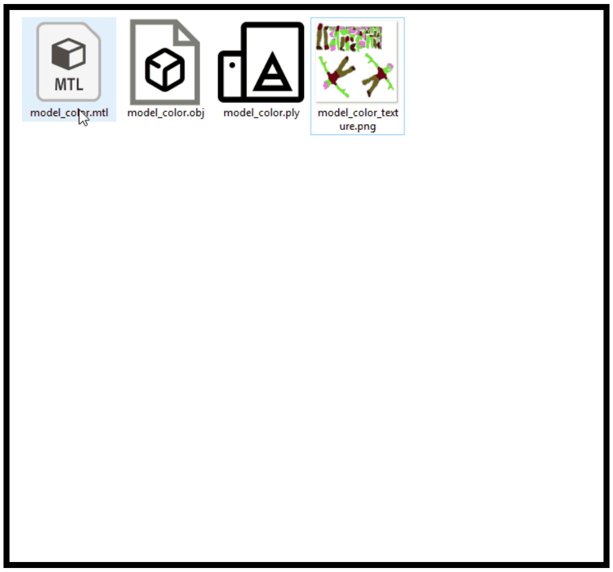
Importing Your Character into Unity

By now, you've probably fallen hopelessly in love with running around in your virtual world with our little demo person. But this love, like all things, must pass, and, anyway, now you've got *your own awesome* character to put into your world. Let's look into how to do that.

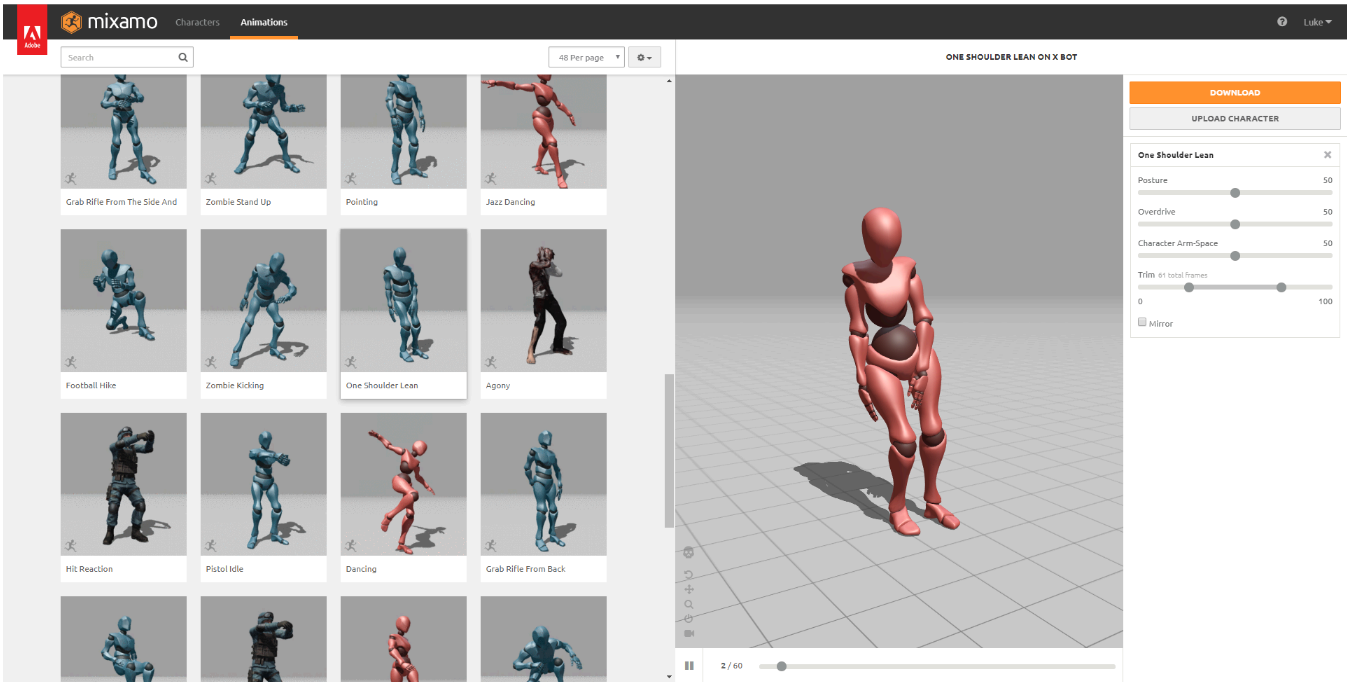
To start, you should have your **.ZIP** file with the .OBJ model .MTL material, and (usually) .PNG texture inside, which we made to animate your character in Mixamo. To refresh, the way you ZIP files together is:

1. Select the files you want to ZIP together (on Windows, hold CTRL while clicking, on Mac hold Command)
2. On **Mac**, **control-click > Compress 3 Items**. (Left image below)
3. Or on **Windows**, **right-click> Send to> Compressed(Zipped) Folder**. (Right image below)
4. Rename the zipped folder as something memorable ("player.zip")





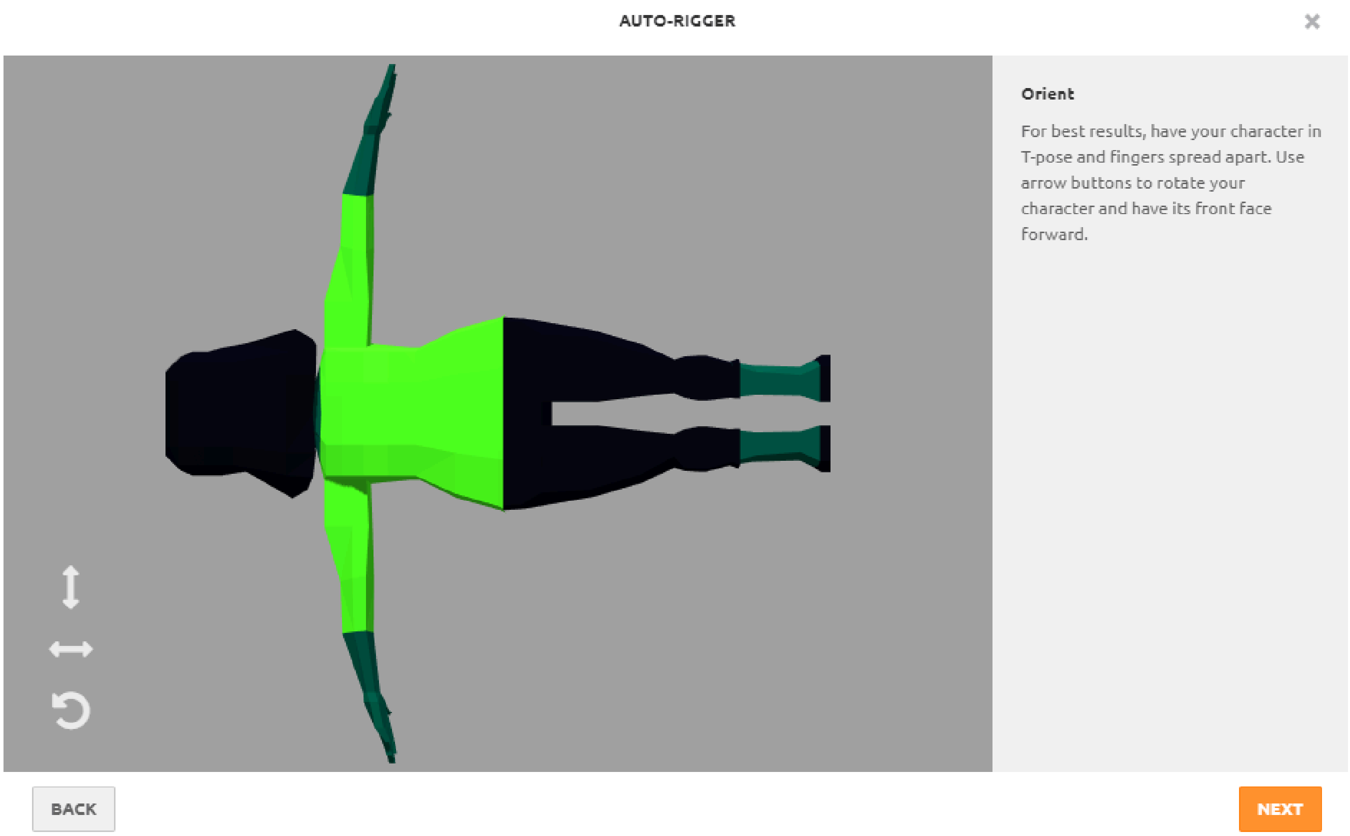
Tutorial



1. We're going to use **Mixamo** to **rig** our models. Rigging is the process of giving your model a "skeleton" with digital joints and bones that allow it to be animated. Animators often build custom rigs for their characters, but Mixamo has a nice basic rig that can work for any model with two legs and two arms.

   (Unity calls this a 'humanoid' rig - but not all humans have two arms and two legs, and not all things with two arm and two legs are human, so it's not the best term, really...)

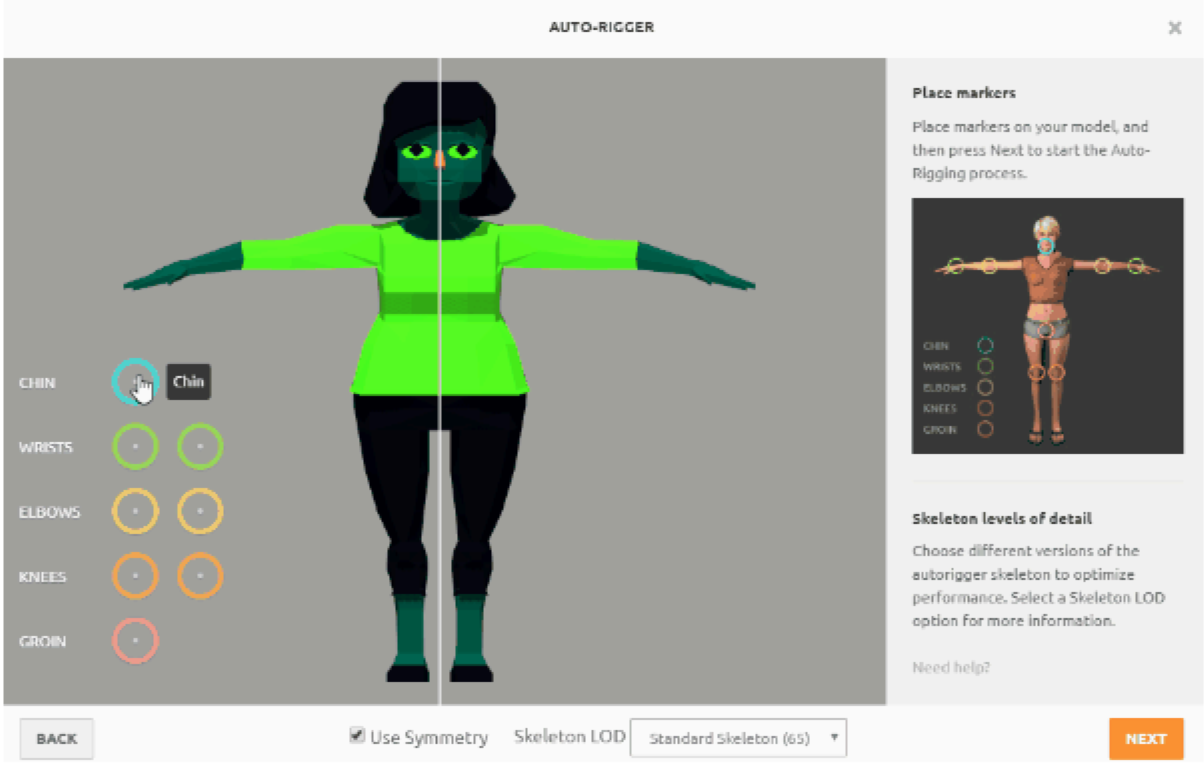
Click on the “Upload Character” (UPLOAD CHARACTER) button, and either select or drop your .ZIP file in. It will process your model, and you’ll see a window like this:



Orient

For best results, have your character in T-pose and fingers spread apart. Use arrow buttons to rotate your character and have its front face forward.

2. If your model is not facing you, use the rotation buttons in the lower left to make your character stand **upright, facing toward you**. Then click “next” (NEXT). ↕ ↔ ↺
3. Drag the joints onto the appropriate parts of your model. Then click “next” (NEXT).



Place markers

Place markers on your model, and then press Next to start the Auto-Rigging process.

Skeleton levels of detail

Choose different versions of the autorigger skeleton to optimize performance. Select a Skeleton LOD option for more information.

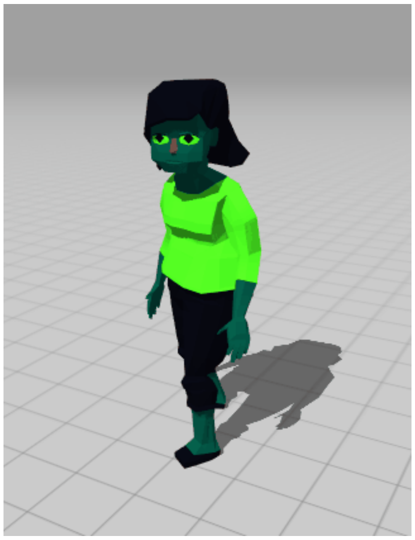
Need help?

☑ Use Symmetry

Skeleton LOD: Standard Skeleton (65)

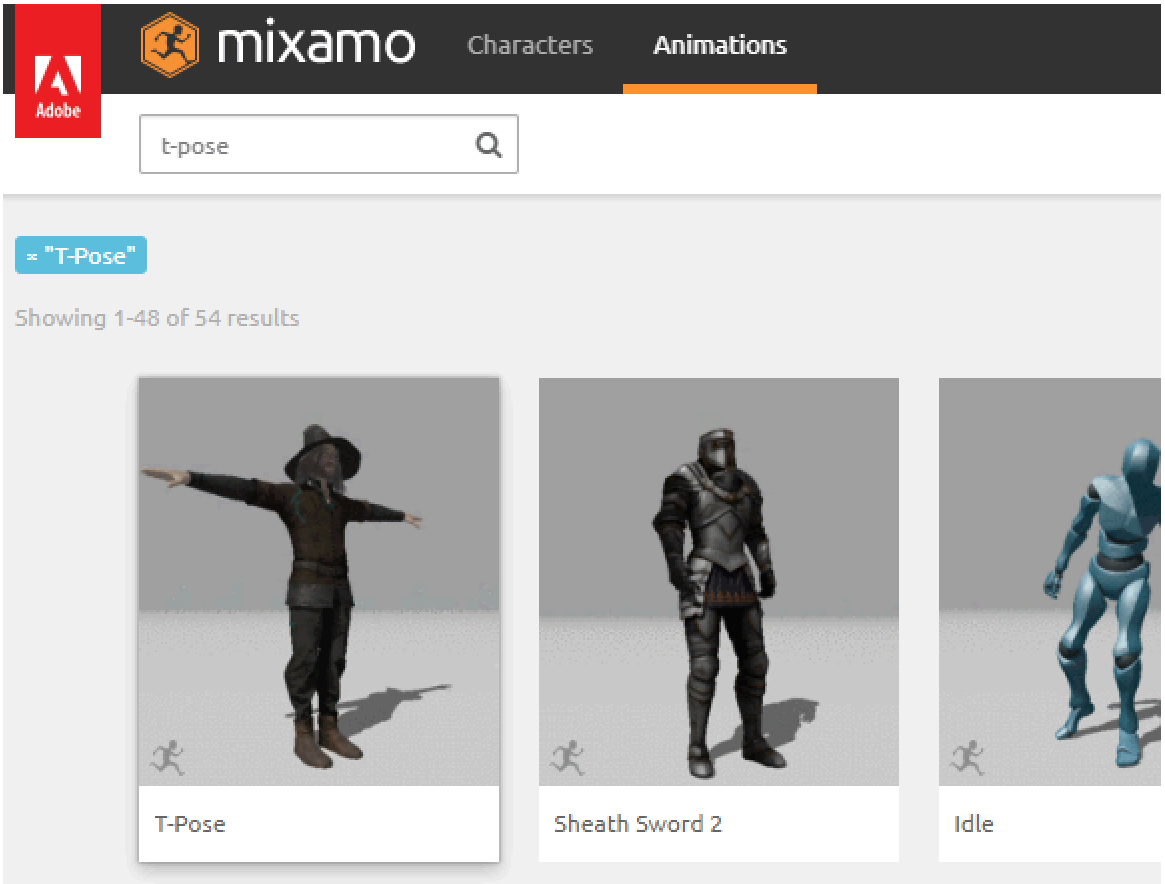
It will work for a few moments applying the rig to your character. Be patient!
When it's done, you'll see your character looking around.

Click "next" (NEXT).



4. It's tempting now to want to make your character do a bunch of silly dances (and take a moment to indulge yourself, if you want!), but right now we're on Mixamo for one thing: **a T-pose**. This will provide a basic rigged model for Unity to use in our game.

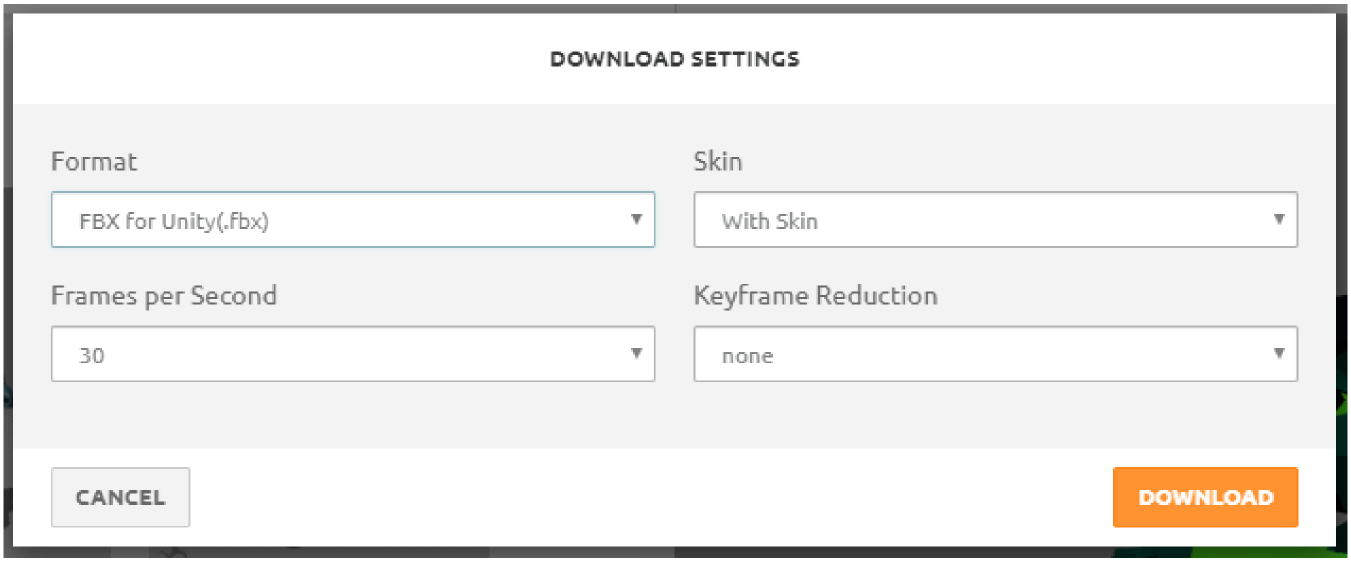
   In the upper-left-hand corner, search for "t-pose," and then click on the first result (a wizard-looking person doing a T-pose).



Search: t-pose

You'll see your character do the same.

5. Now, click the "download" button (DOWNLOAD). You'll see this window:



DOWNLOAD SETTINGS

Format: FBX for Unity(.fbx)

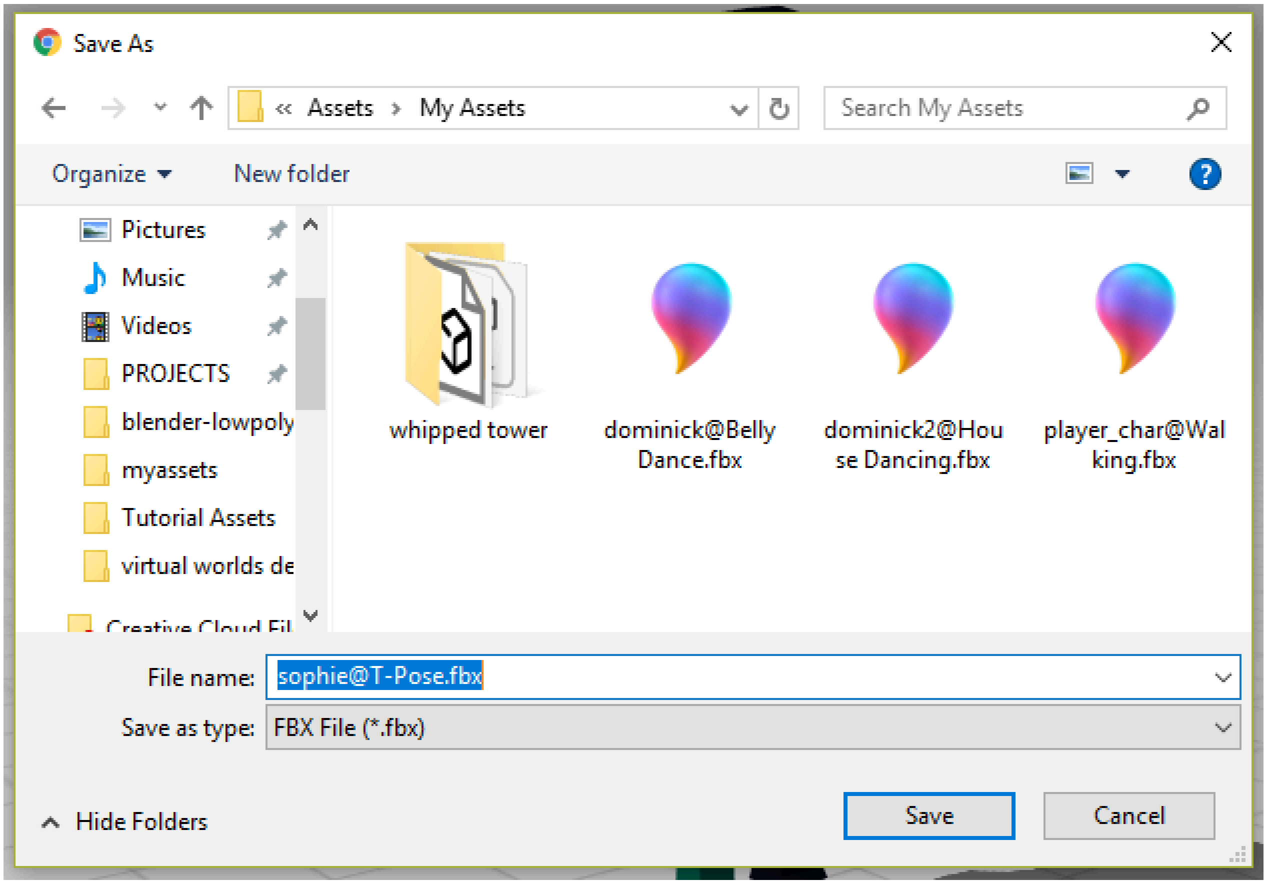
Skin: With Skin

Frames per Second: 30

Keyframe Reduction: none

6. **Make sure the "Format" box says "FBX for Unity(.fbx),"** and then click "Download." Save your .FBX file right into your Assets folder of your unity project (I like to make my

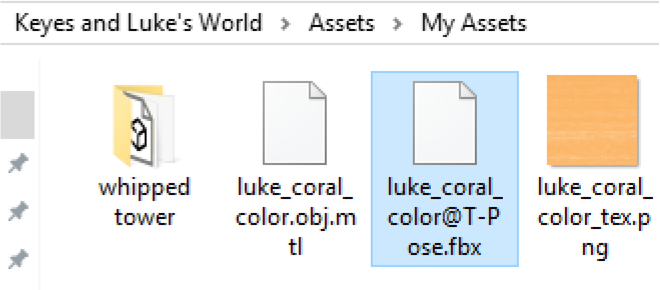
own folder called “luke_assets” to separate the things I made from the tutorial models and assets).



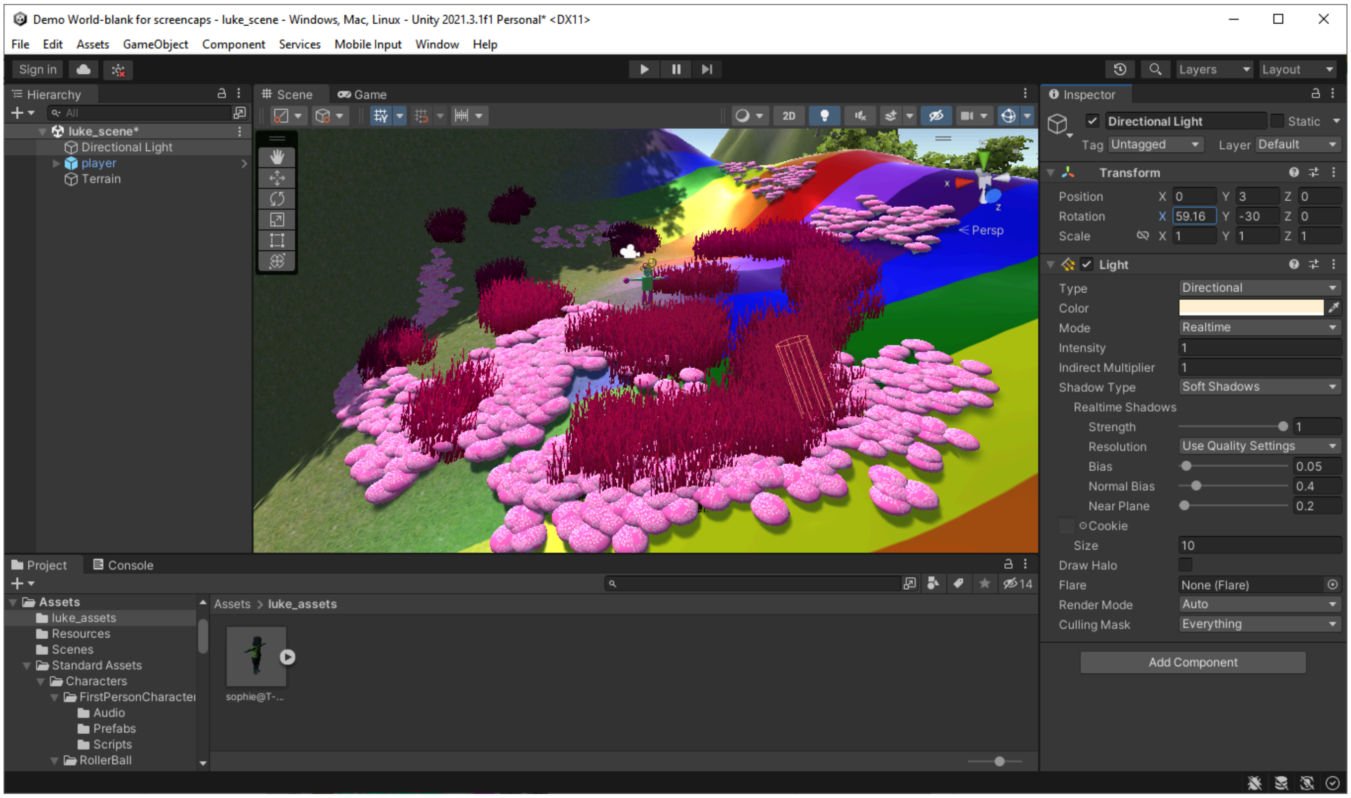
File name: sophie@T-Pose.fbx

Save as type: FBX File (*.fbx)

7. If you made your character in SculptGL, and used Blender to apply the texture, your model’s **.MTL** and **.PNG** texture files should also be copied into your “[name]_assets” folder next to your **.FBX**. (My Sophie model wasn’t made in SculptFAB, so the following screenshot is of a different model named “Coral”).


Keyes and Luke's World › Assets › My Assets
whipped tower
luke_coral_color.obj.mtl
luke_coral_color@T-Pose.fbx
luke_coral_color_tex.png

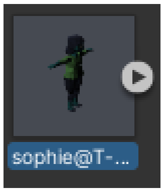
8. Now, open up your project in Unity, and go to your scene that has our demo character standing in it. In the Assets pane, go to the folder where you just saved your new rigged model.



Directional Light
Static
Tag Untagged
Layer Default

Transform
Position X 0 Y 3 Z 0
Rotation X 59.16 Y -30 Z 0
Scale X 1 Y 1 Z 1

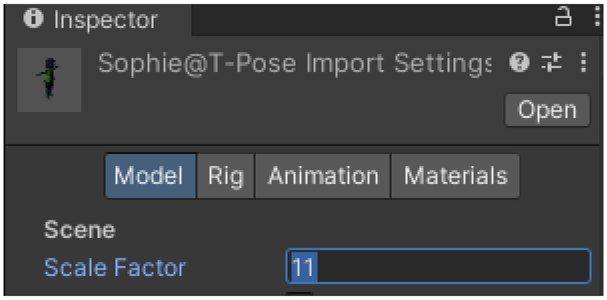
Light
Type Directional
Color
Mode Realtime
Intensity 1
Indirect Multiplier 1
Shadow Type Soft Shadows
Realtime Shadows
Strength 1
Resolution Use Quality Settings
Bias 0.05
Normal Bias 0.4
Near Plane 0.2
Cookie
Size 10
Draw Halo
Flare None (Flare)
Render Mode Auto
Culling Mask Everything

9. Click on the model to select it (

), and the Inspector pane on the right will change to be about your model. There are a few things we need to tweak here.

   First, set the **scale factor** to **30 for TinkerCAD models or 1.5 for SculptGL models**. Otherwise, our model will be too small (you can change this value later if the size is still off).

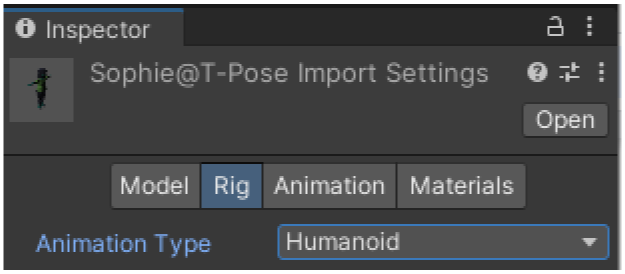
10. Then, click on the **'Rig' button** above, and in that window, change the **Animation Type** from 'Generic' to '**Humanoid**.' Click **Apply**.



Sophie@T-Pose Import Settings

Scene

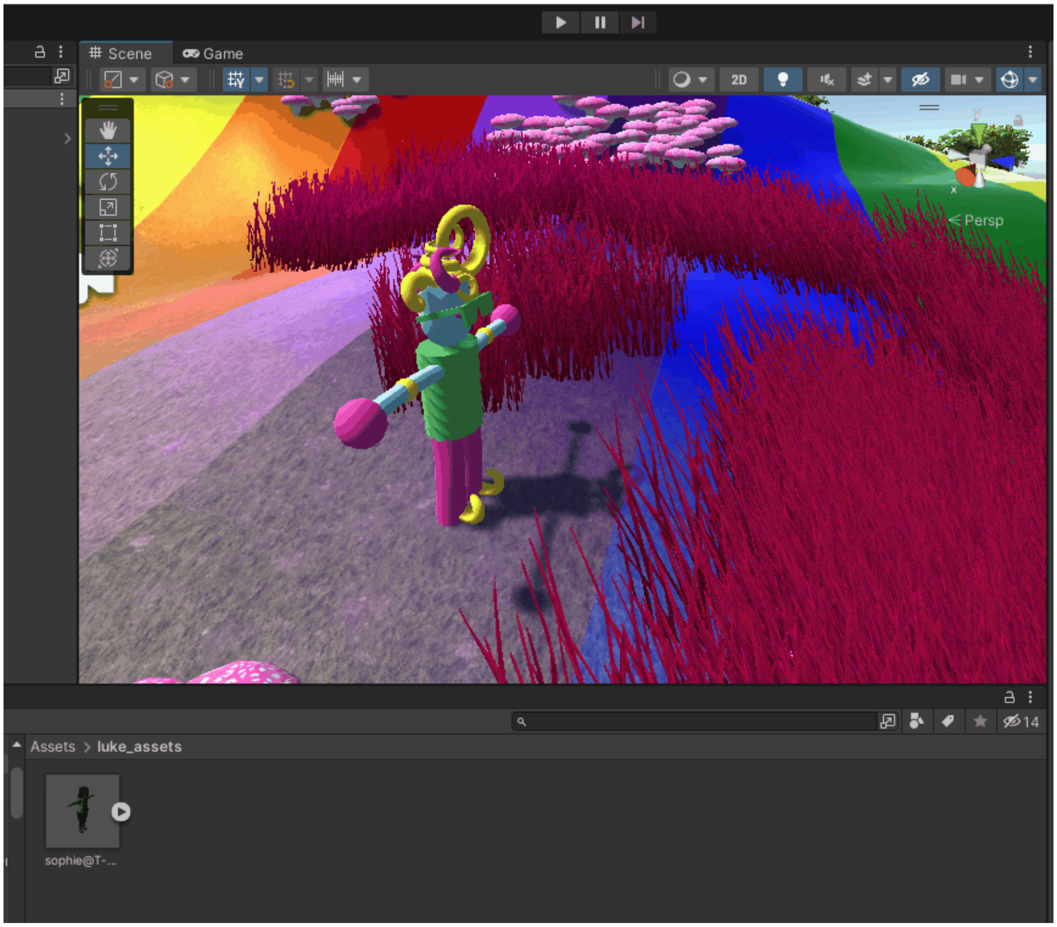
Scale Factor 11



Sophie@T-Pose Import Settings

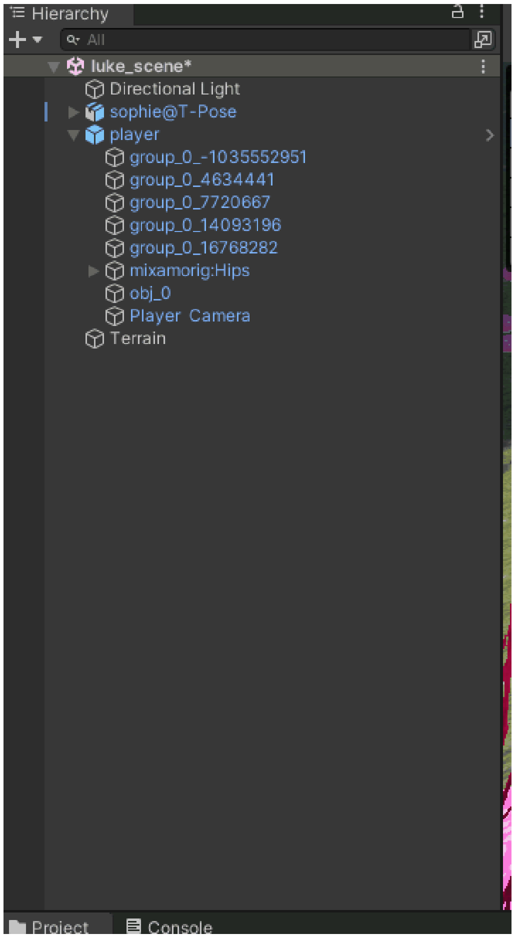
Animation Type Humanoid

11. Now, drag your model into the scene:



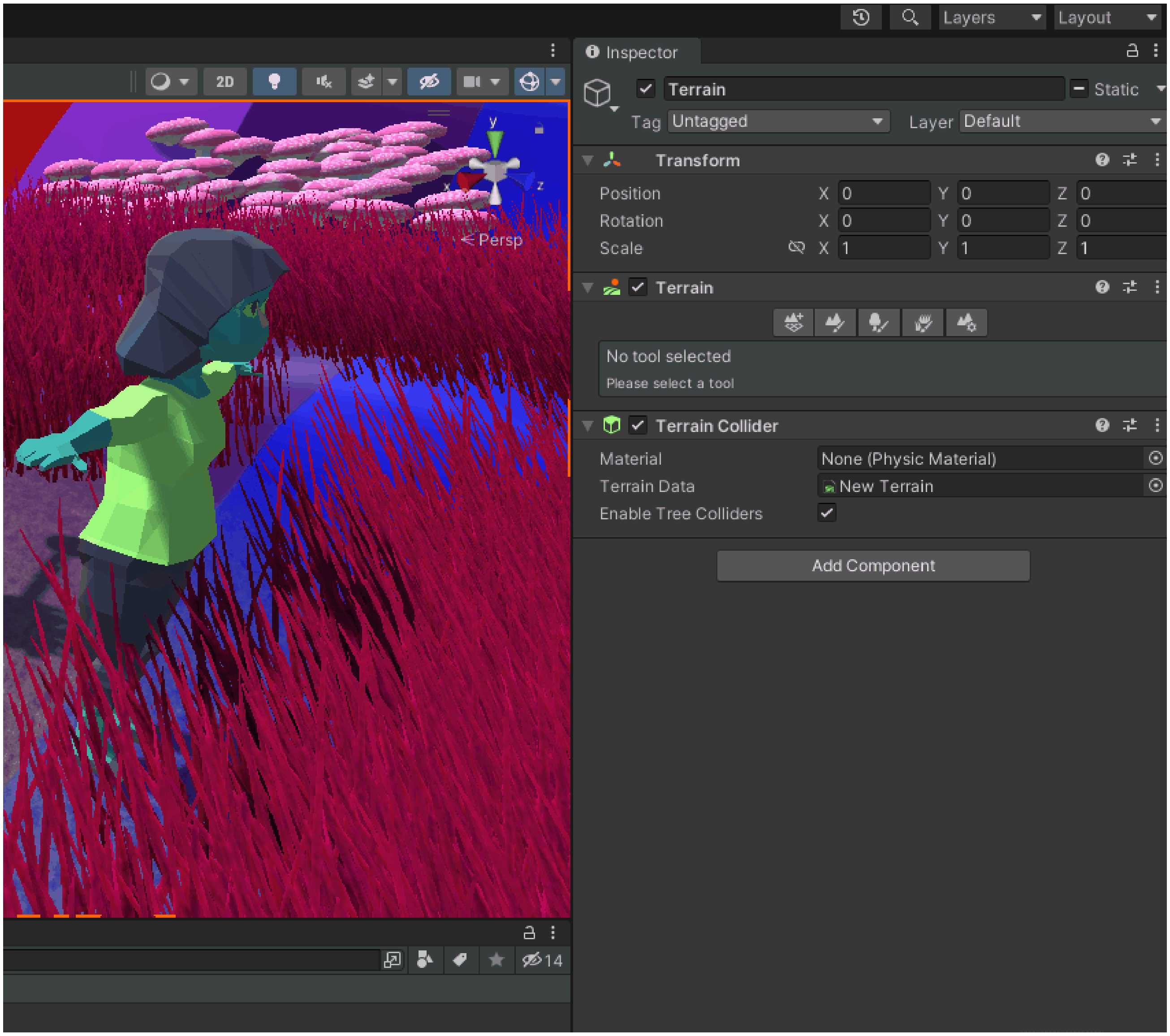
Now, we can give our model the same properties that make our player character tick.

12. First, we'll give our new character their own camera to follow them. **Right-click on your character's name** in the hierarchy, click "**Camera**," and name the new camera "Player Camera."



luke_scene*
- Directional Light
- sophie@T-Pose
- player
  - group_0_-1035552951
  - group_0_4634441
  - group_0_7720667
  - group_0_14093196
  - group_0_16768282
  - mixamorig:Hips
  - obj_0
  - Player Camera
- Terrain

13. Click on your model in the scene, so that it is selected. In the Inspector pane, on the right, in the **Animator** pane, click on the little circle next to "Controller," and choose the **"CharacterController"**. This is the animation controller that tells the player character when to play different animations, like running, jumping, falling...or FLYING.



Inspector

☑ Terrain Static

Tag Untagged Layer Default

Transform

Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

☑ Terrain

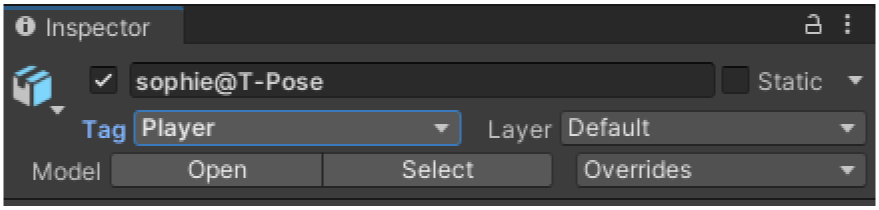
No tool selected
Please select a tool

☑ Terrain Collider

Material None (Physic Material)
Terrain Data New Terrain
Enable Tree Colliders ☑

Add Component

14. At the very top right, set the **Tag** to **"Player."** This will let other code in our game know that this is now the player character!



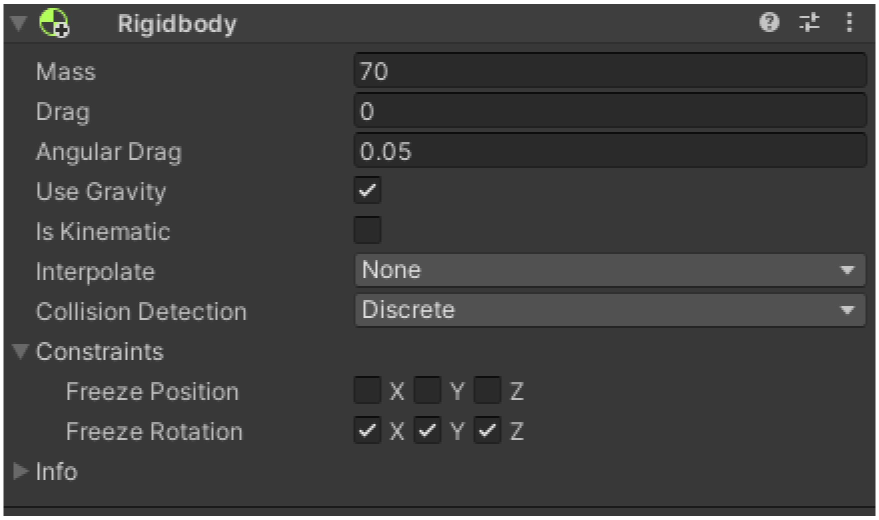
Inspector

☑ sophie@T-Pose □ Static

Tag Player Layer Default

Model Open Select Overrides

15. Now we need to add some components to our model to make it into an actual controllable character. In the Inspector pane to the right, click “Add Component” (Add Component). Type ‘rigid’ in the search bar, and select **“Rigidbody”** (Rigidbody) when it comes up. You’ll get a new little panel like this:



Rigidbody

Mass 70

Drag 0

Angular Drag 0.05

Use Gravity ☑

Is Kinematic □

Interpolate None

Collision Detection Discrete

Constraints

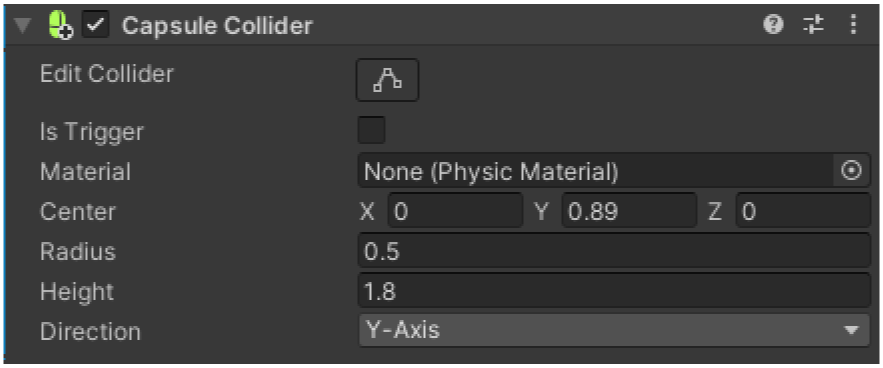
Freeze Position □ X □ Y □ Z

Freeze Rotation ☑ X ☑ Y ☑ Z

Info

Rigidbodies are what give objects in the game weight and heft. In science terms, they control the physics of the objects. Set the Rigidbody properties like they are above: **Mass: 70**, and **Freeze Rotation for X, Y, Z**. That second part is important - otherwise our character will just fall over!

16. In the Inspector pane to the right, click “Add Component” (Add Component). Type ‘Capsule’ in the search bar, and select **“Capsule Collider”** (Capsule Collider) when it comes up. You’ll get a new little panel like this:



☑ Capsule Collider

Edit Collider

Is Trigger □

Material None (Physic Material)

Center X 0 Y 0.89 Z 0

Radius 0.5

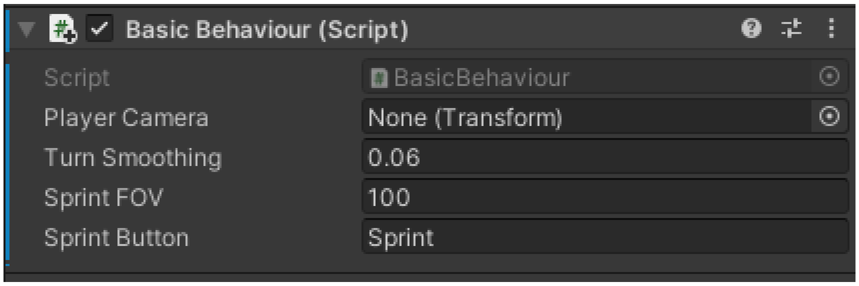
Height 1.8

Direction Y-Axis

Set each of the settings as they are in the image. **Material: Character**, **Center: X: 0, Y: 0.89, Z: 0**, **Radius: 0.5**, and **Height: 1.8**. You should see a green ‘capsule’ appear around your player character. This will be what”collides” with your world, and keeps the character from falling through the floor or running through solid objects.

Later, you can tweak those numbers to get a better fit for your model, if you have issues. But these will work, in general, for most character models.

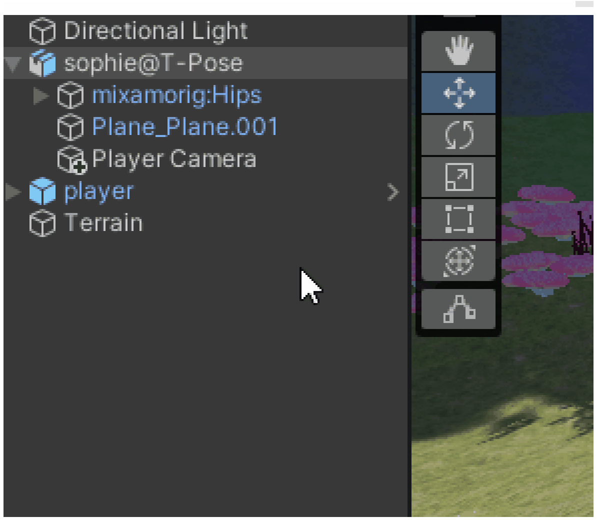
17. "Add Component" (Add Component), again. This time, we're going to add a **script**, or a pre-made chunk of code that does a job. Type "basic" and select the **"Basic Behavior"** script (# Basic Behaviour). You'll get a new pane that looks like this:



Basic Behaviour (Script)

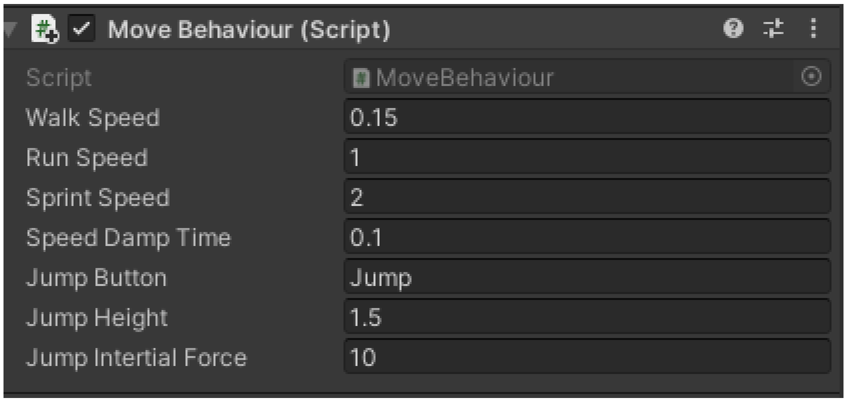
Script	BasicBehaviour
Player Camera	None (Transform)
Turn Smoothing	0.06
Sprint FOV	100
Sprint Button	Sprint

18. The only thing we need to change here is the **Player Camera**. Drag your camera from the Hierarchy on the left, all the way into that box:



Directional Light
sophie@T-Pose
mixamorig:Hips
Plane_Plane.001
Player Camera
player
Terrain

19. **"Add Component"** again. We're going to add another **script**. Type "move" and select the **"Move Behavior"** script (# Move Behaviour). Leave it as-is. (Later you can play with these numbers to change how your character moves.)

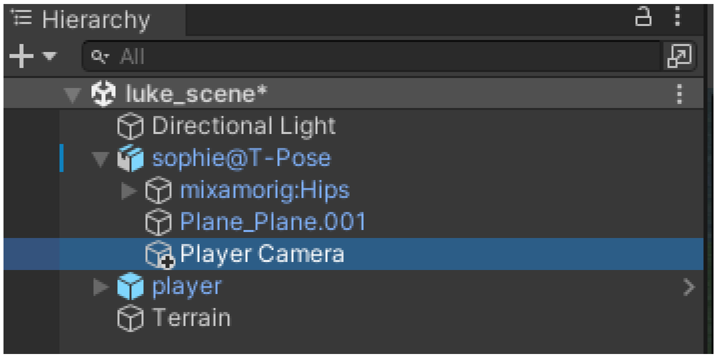


Move Behaviour (Script)

Script	MoveBehaviour
Walk Speed	0.15
Run Speed	1
Sprint Speed	2
Speed Damp Time	0.1
Jump Button	Jump
Jump Height	1.5
Jump Intertial Force	10

20. Our player now has all the scripts it needs to run around, but we're going to add one more script to give us a little more freedom: FLIGHT! **"Add Component"** again. Type "fly" and select the **"Fly Behavior"** script (# Fly Behaviour). Leave it as-is.

21. Our character is rigged and scripted! There are just two quick things we need to do. First, we need to tell the camera to follow our new player character. Click on your character's Player Camera in the Hierarchy on the left:

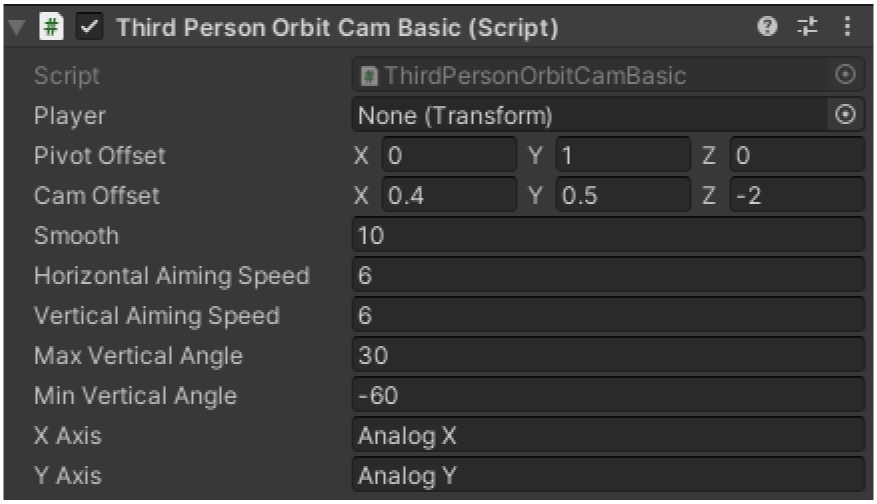


Hierarchy

All

- luke_scene*
  - Directional Light
  - sophie@T-Pose
    - mixamorig:Hips
    - Plane_Plane.001
    - Player Camera
  - player
  - Terrain

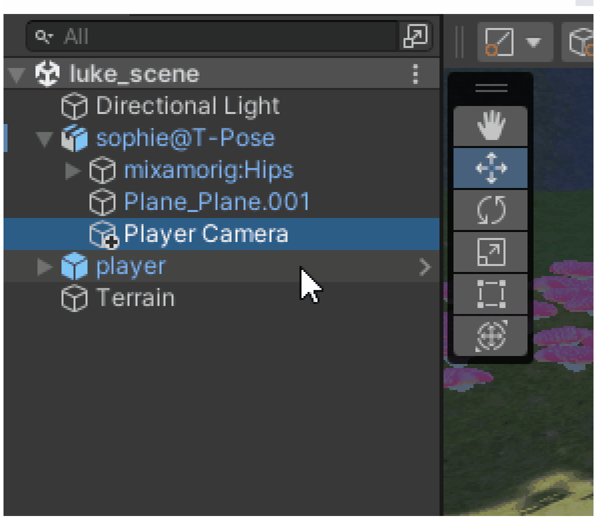
22. In the Inspector on the right, click "**Add Component"** and type "orbit." Click to add the script "**Third Person Orbit Cam Basic**" (# Third Person Orbit Cam Basic).



Third Person Orbit Cam Basic (Script)

Script	ThirdPersonOrbitCamBasic					
Player	None (Transform)					
Pivot Offset	X	0	Y	1	Z	0
Cam Offset	X	0.4	Y	0.5	Z	-2
Smooth	10					
Horizontal Aiming Speed	6					
Vertical Aiming Speed	6					
Max Vertical Angle	30					
Min Vertical Angle	-60					
X Axis	Analog X					
Y Axis	Analog Y					

23. Then drag the name of your new character all the way from the hierarchy into the “Player” slot in this new script.



24. Last, go into the hierarchy and **delete the old player** by clicking on it, and pressing “delete” on your keyboard.

25. Press play, and watch your little so-and-so scoot around your world!

    W, A, S, and D move your character.

    Mouse moves the camera.

    ‘F’ turns flying mode on and off.