Casual Girl

Casual Girl is a 3d character model created for your game project. The character model is optimised for use across all platforms, mobile, console and desktop and the average polygon count for each iteration / prefab is **4000** triangles. The asset comes with a number of pre-set prefabs containing different variations that can be used as a starting point within your project. The characters color information is material based and fully adjustable, leaving unlimited options when creating new variations. The character model is fully rigged and will work with humanoid animations from both **Mixamo.com** and the **Unity Asset Store**.

Animations are not included.

Due to the **Mixamo.com EULA**, animations created on the platform cannot be distributed within a **Unity Asset Pack**. To create and download animations from **Mixamo.com** and use them with this character pack, please follow the next section.

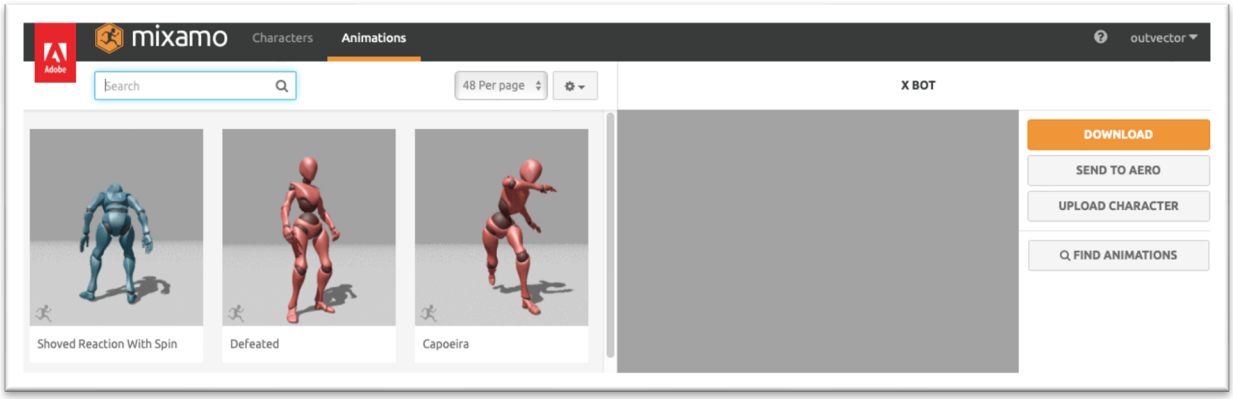
Animation Tutorial

Stage 1: Open the character / character pack within **Unity**.

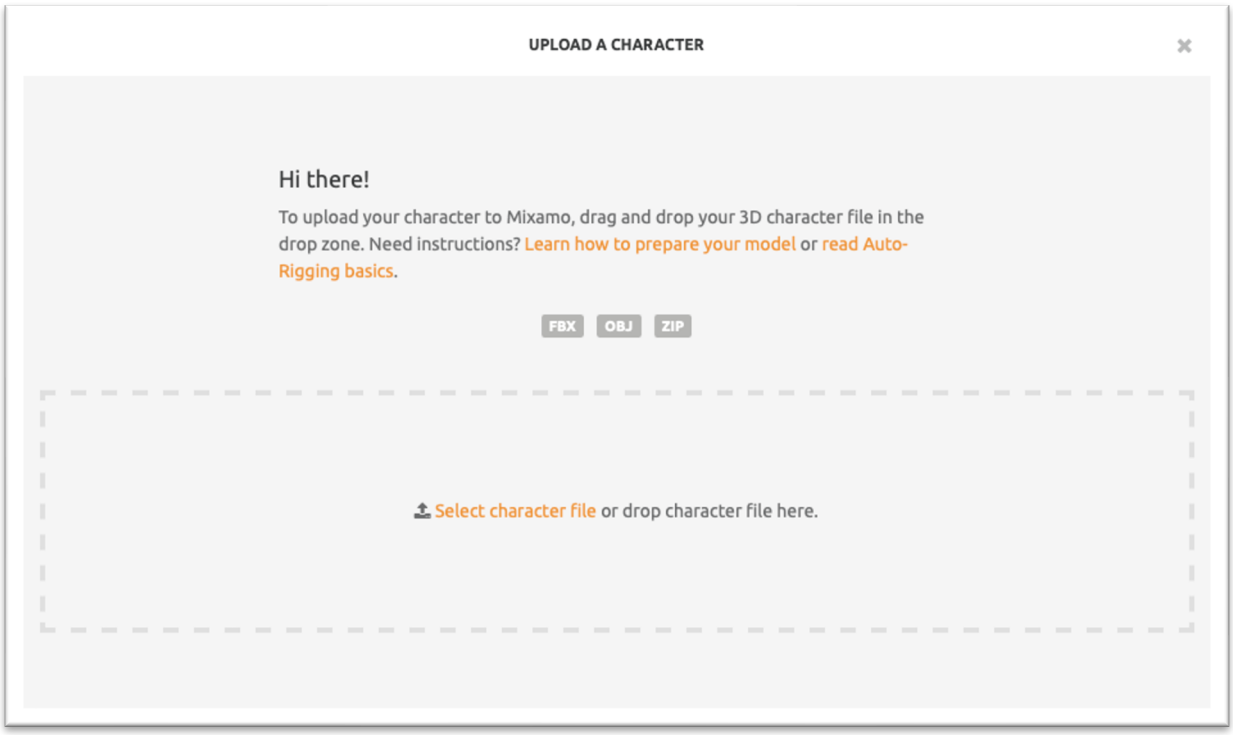
Stage 2: In the project view, under the asset folder, find the **Meshes** folder. Within this folder, you will find your **character** model in **fbx** format. The model can be used within **Mixamo** to visualise the animation and how it will appear on your character.

Stage 3: Drag and drop the model from this folder to somewhere easily accessible, for instance, your **Desktop**.

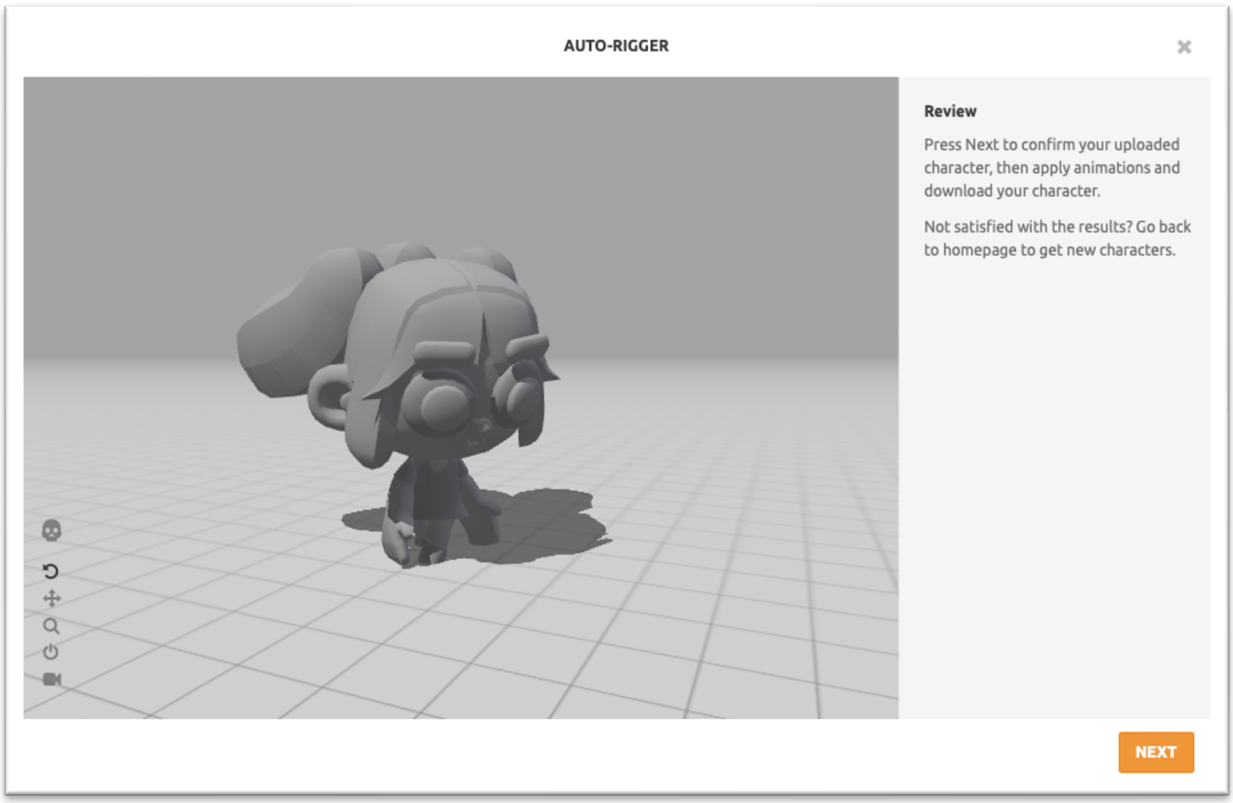
Stage 4: Open a browser windows, and go the URL **mixamo.com**, log in to your account, or if you do not have an account with **Mixamo**, then register. Once you have logged into your account, select **Upload Character** from the menu on the right side.



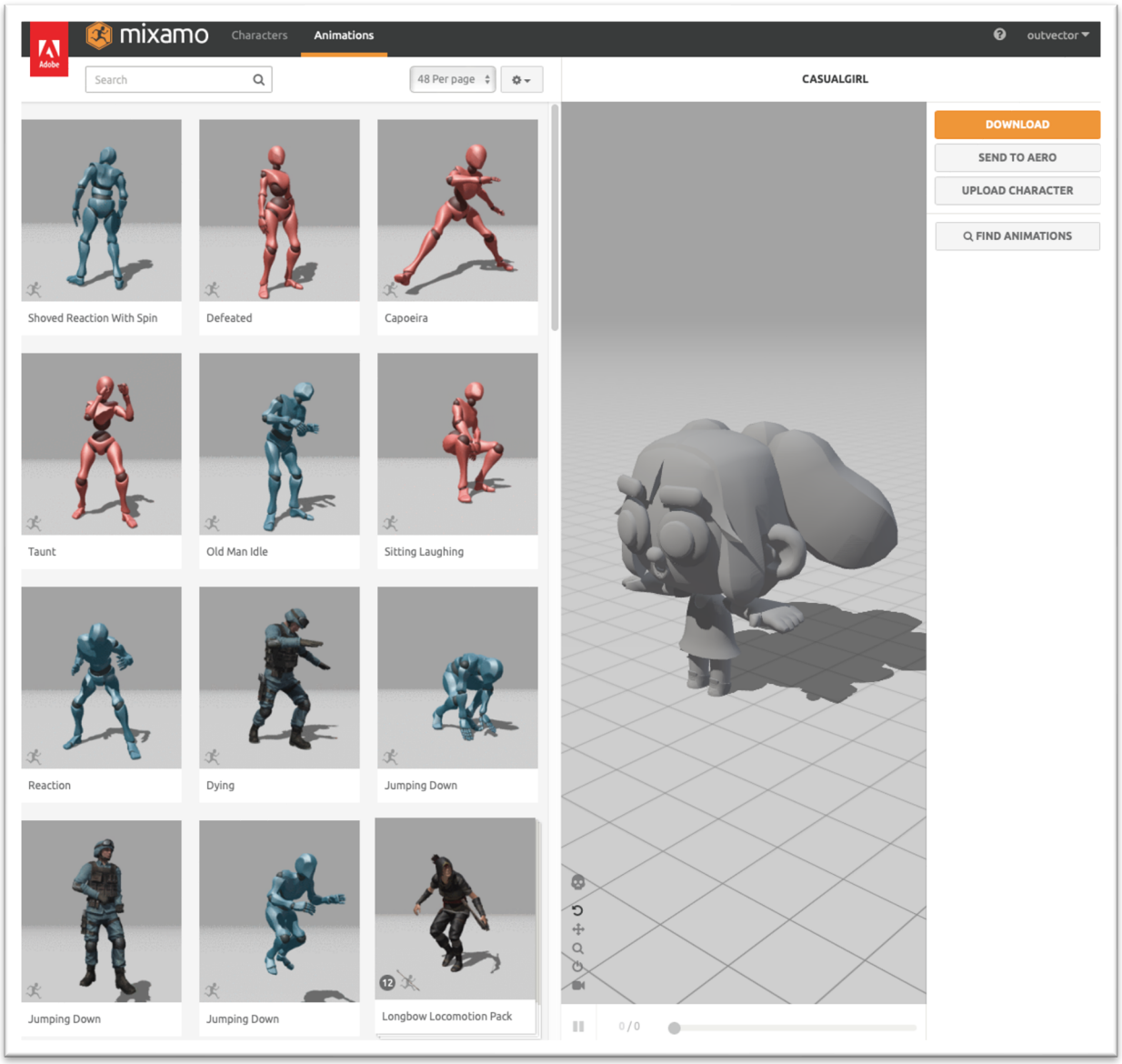
Mixamo will then show the following window, prompting you to drag and drop your 3d character model, select the file from your **desktop**, then **drag** and **drop** it into this window.



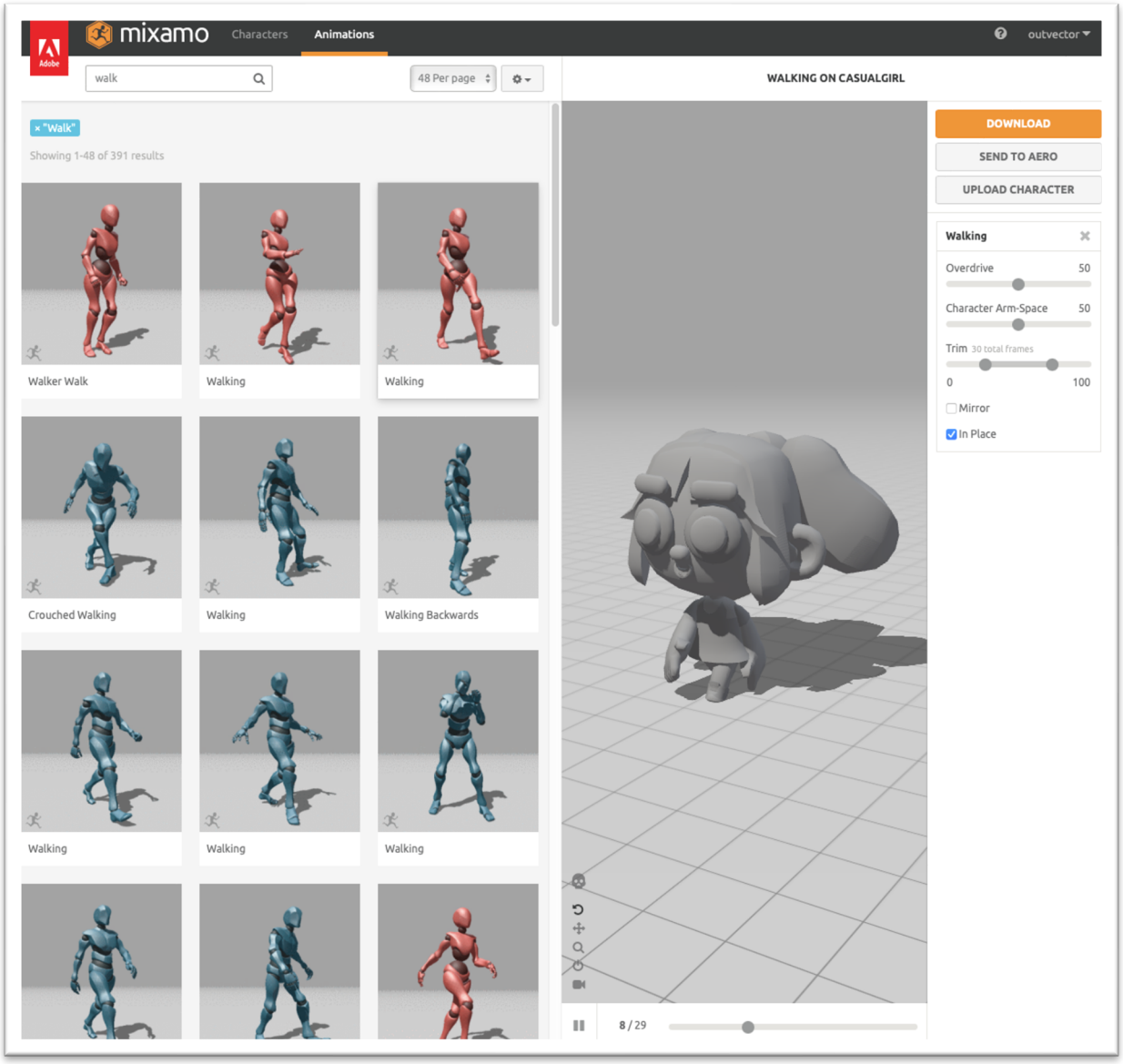
Mixamo will upload your character, the character is already rigged, so **Mixamo** will automatically prepare your character model for animation, you will then be presented with a review window which will contain your character and a basic idle animation will be playing, at this stage, select **NEXT**.



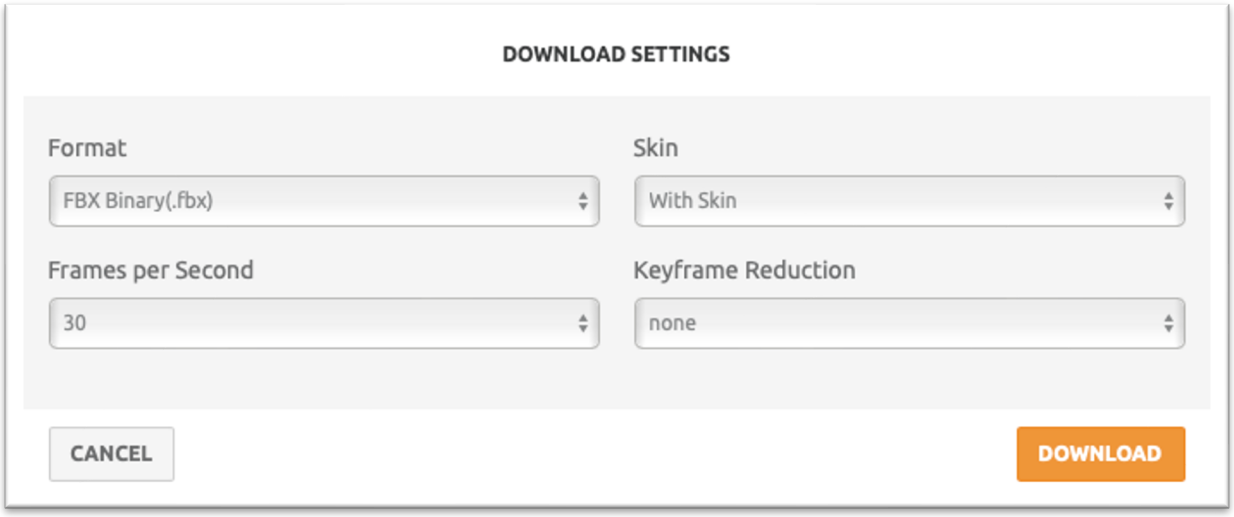
Stage 5: After clicking **Next**, your character will appear in the main **Mixamo** preview window in a static pose like the following screenshot, from here, we can apply animations. Please note, do not worry at this stage that the character model does not have color data applied, we are simply using the model as a **reference** for the **animation**, and we will **export** the animation data only, which can be applied directly within **Unity** on the main character (***this will be documented later on in the tutorial***).



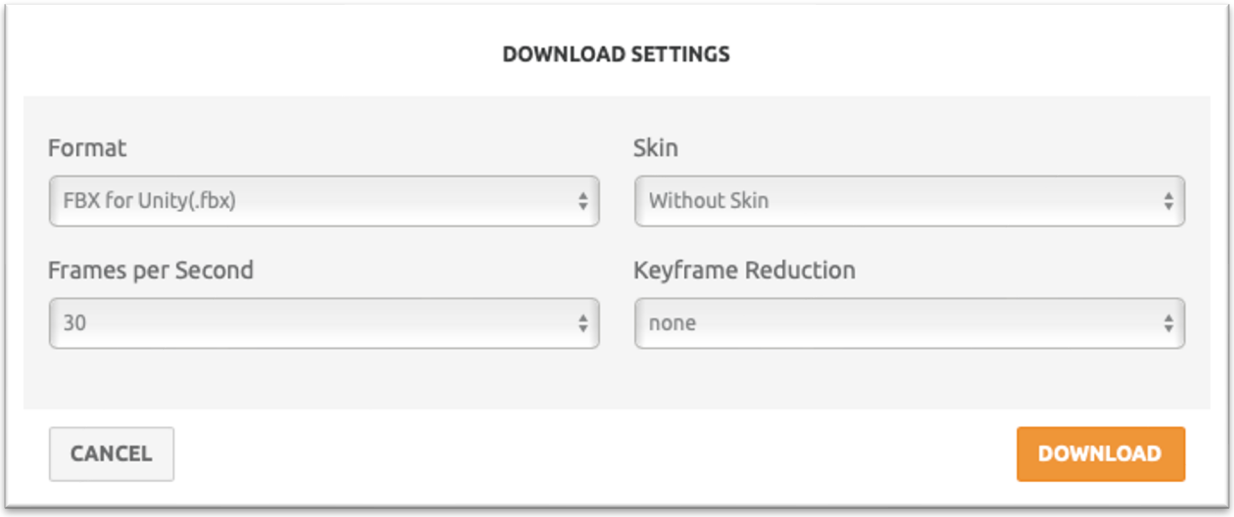
Stage 6: Using the search function on **Mixamo**, find an animation that you would like to use on your character model, for this tutorial instance, we will look for and apply a **walk cycle**. In the search bar, enter '**walk'** and press the **Enter** key, all walk animations will then be presented to you. From the list presented, select an animation that appeals to you, it will automatically be applied to your character, from the menu to the right of your preview window select '**In Place**', in place showcases the animation as if applied like a walk cycle / looping animation, typically how we would set up animation for cyclic data in **Unity**, such a running, walking, etc.



Stage 7: To export and use this animation within **Unity**, we can now download it, click the **DOWNLOAD** button from the right-hand side menu. The following window showcasing the download settings will appear: -



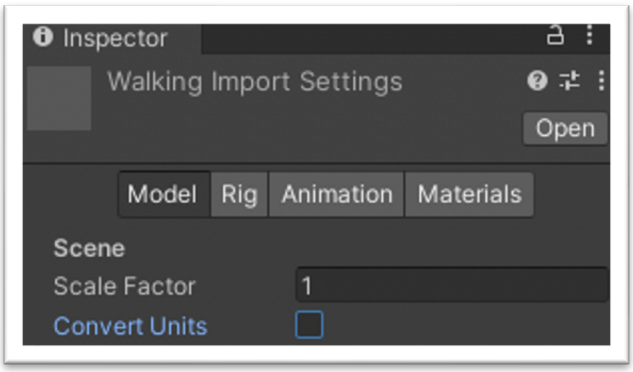
We need to change 2 settings, under **Format**, select **FBX for Unity(.fbx)**, under the **Skin** drop down, select **Without-Skin**.



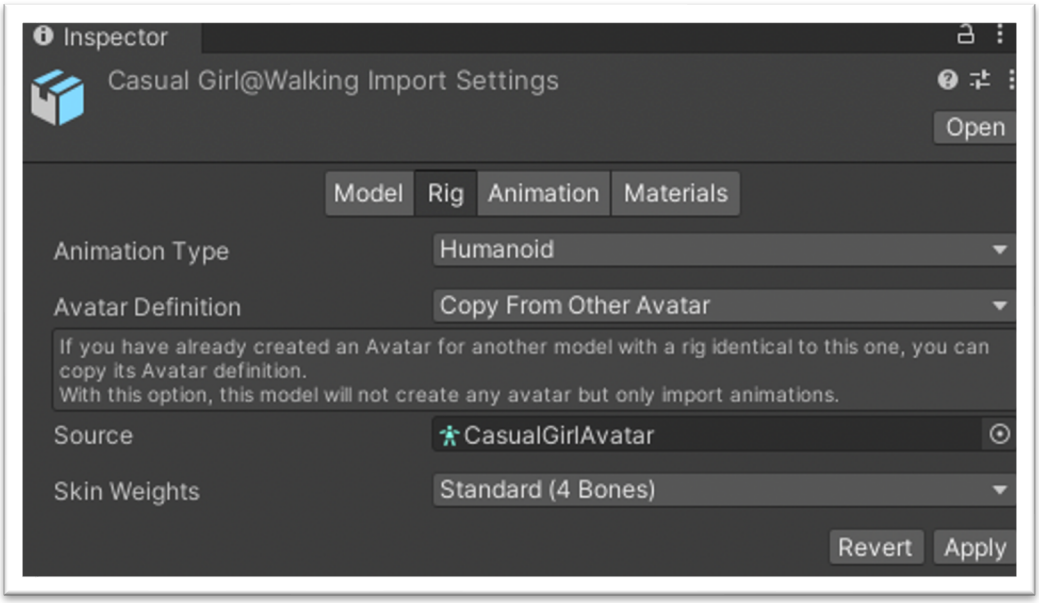
We chose '**Without-Skin**' as we already have our **skin** (our character model) within **Unity**. Finally, click **DOWNLOAD** and **Save** your file.

Stage 8: Go back to **Unity**, create an **Animations** folder, and then drag and drop your **downloaded file** into this folder, if you exported the walking animation, the downloaded file will take the format '**ModelName@Walking.fbx**'. **Select** the file you just imported into the **Animations** folder, in the **Inspector**, you will be presented with the **import settings**, change the following options under each **TAB** respectively: -

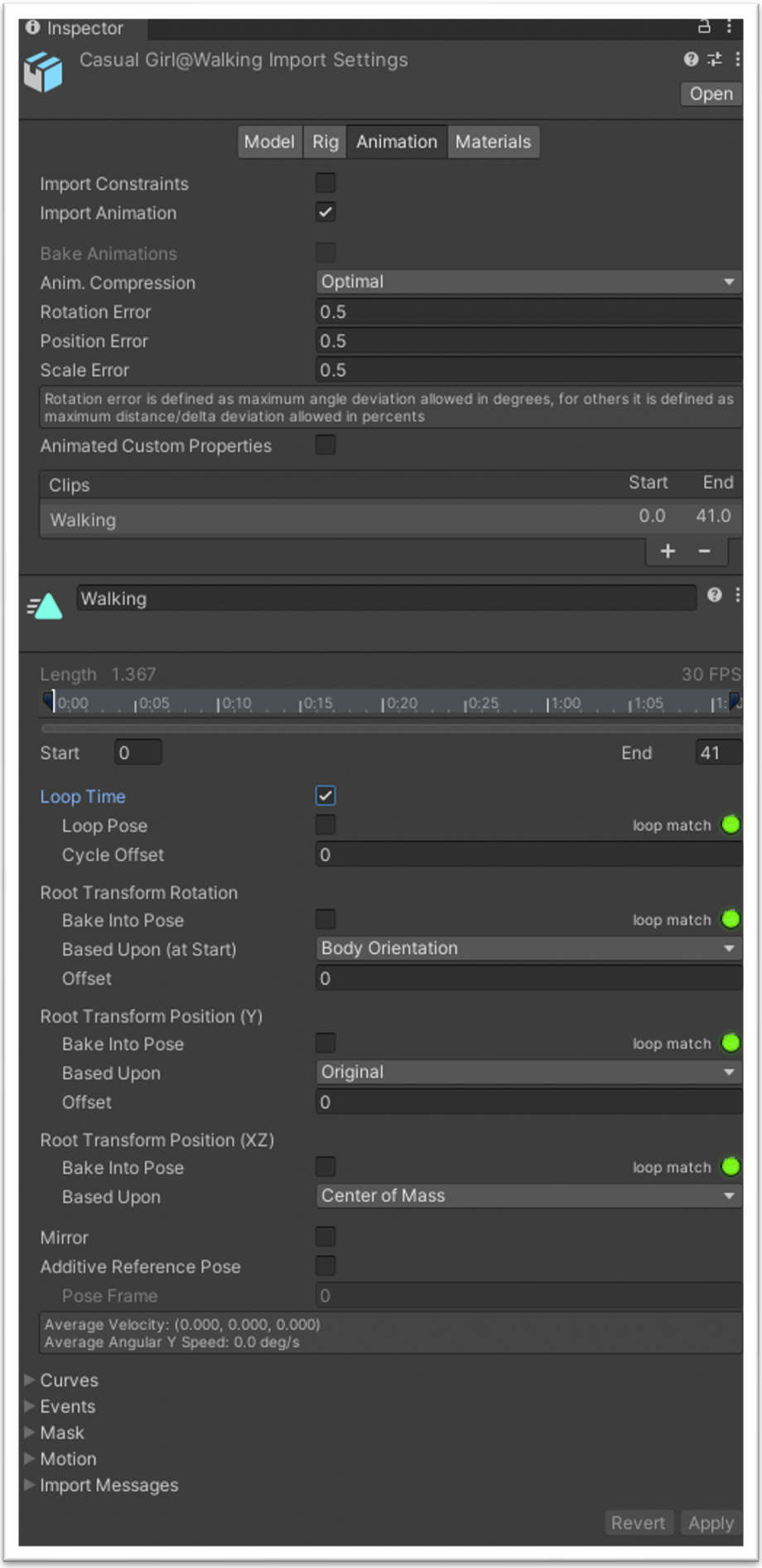
Under the **Model** TAB, uncheck '**Convert Units**',



Under the **Rig** TAB, set **Animation Type** to '**Humanoid**', and under '**Avatar Definition**' choose '**Copy From Other Avatar**', a **Source** prompt will appear, choose your **main** characters **Avatar**: -

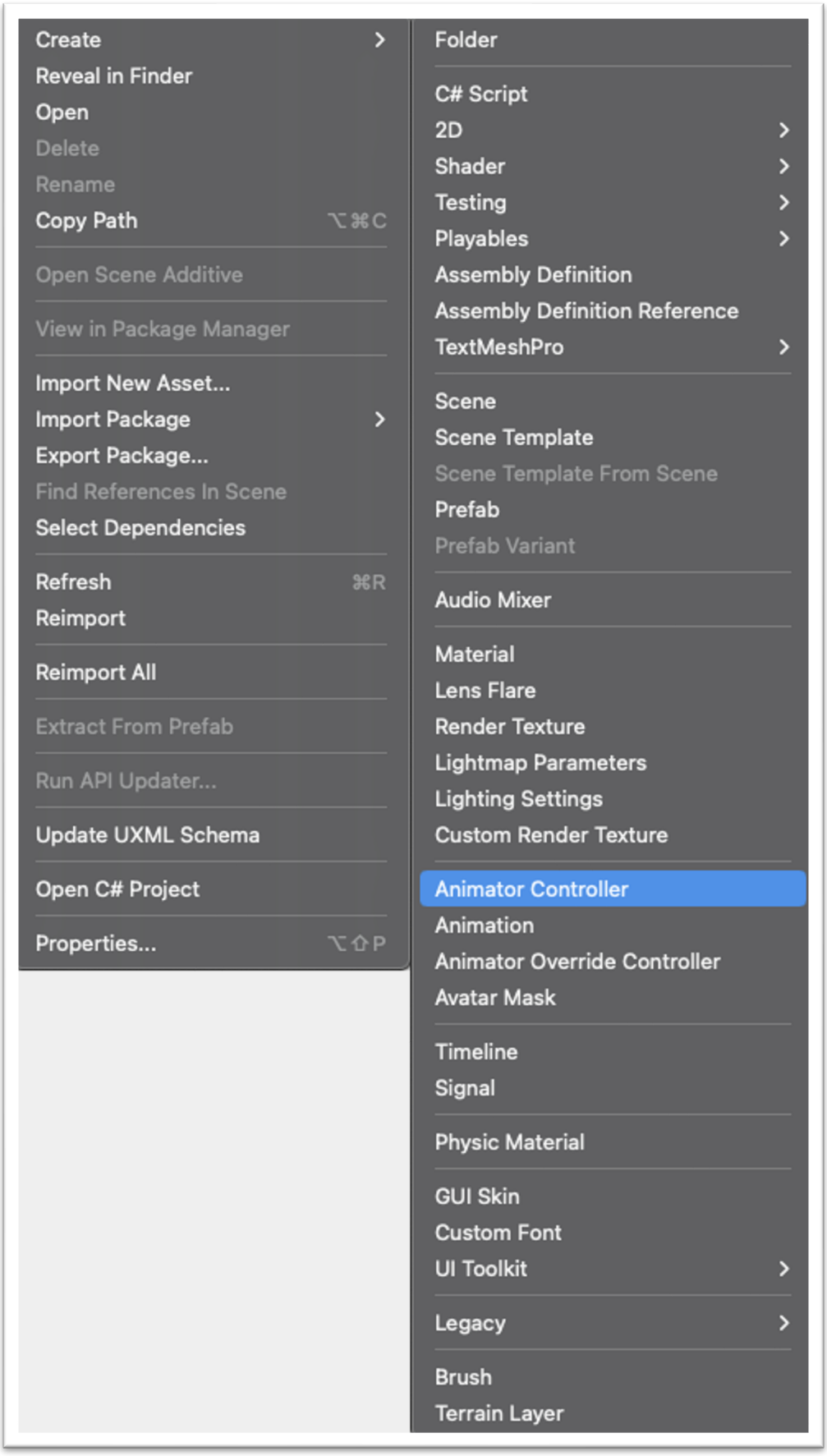


Under the **Animation** TAB, **tick check** the **Loop Time** box, as we want our animation to loop. **Finally**, under the **Materials** TAB, select **None**. Click **Apply**.

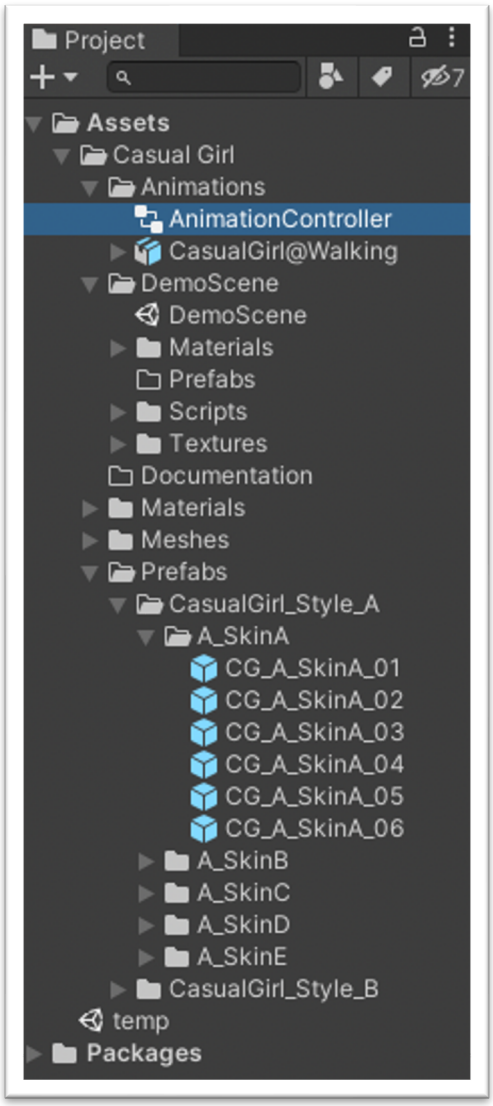

Inspector
Casual Girl@Walking Import Settings
Open
Model
Rig
Animation
Materials
Import Constraints
Import Animation
Bake Animations
Anim. Compression
Optimal
Rotation Error
0.5
Position Error
0.5
Scale Error
0.5
Rotation error is defined as maximum angle deviation allowed in degrees, for others it is defined as maximum distance/delta deviation allowed in percents
Animated Custom Properties
Clips
Start
End
Walking
0.0
41.0
Walking
Length 1.367
30 FPS
0:00
0:05
0:10
0:15
0:20
0:25
1:00
1:05
Start
0
End
41
Loop Time
Loop Pose
loop match
Cycle Offset
0
Root Transform Rotation
Bake Into Pose
loop match
Based Upon (at Start)
Body Orientation
Offset
0
Root Transform Position (Y)
Bake Into Pose
loop match
Based Upon
Original
Offset
0
Root Transform Position (XZ)
Bake Into Pose
loop match
Based Upon
Center of Mass
Mirror
Additive Reference Pose
Pose Frame
0
Average Velocity: (0.000, 0.000, 0.000)
Average Angular Y Speed: 0.0 deg/s
Curves
Events
Mask
Motion
Import Messages
Revert
Apply

Step 9: To **Apply** the **Animation** to one of the character model **Prefabs**, first drag and drop a model **Prefab** into the **Scene Hierarchy**. Expand the **Prefab** group, and click on the **child** of the main group, you will see find the **Animator** component for the character.

We need to add an **Animator Controller**. In the **Project** view, **right click**, select **Create > Animator Controller**.

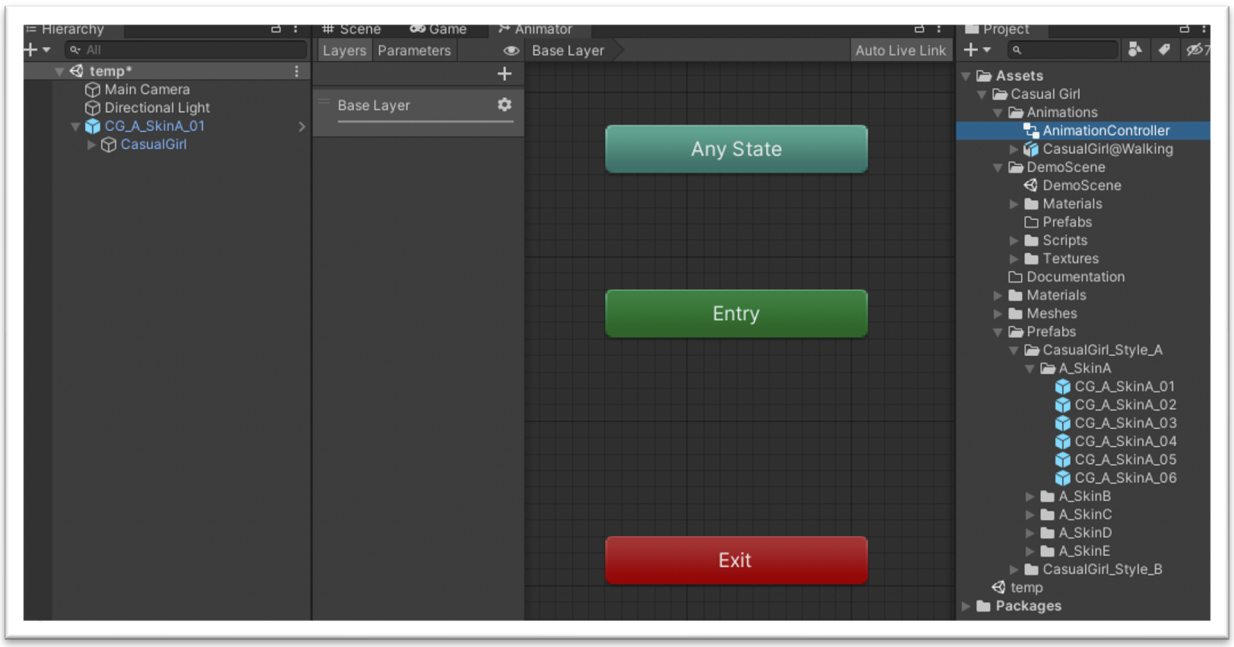

Create
Reveal in Finder
Open
Delete
Rename
Copy Path ⌥⌘C
Open Scene Additive
View in Package Manager
Import New Asset...
Import Package
Export Package...
Find References In Scene
Select Dependencies
Refresh ⌘R
Reimport
Reimport All
Extract From Prefab
Run API Updater...
Update UXML Schema
Open C# Project
Properties... ⌥⇧P
Folder
C# Script
2D
Shader
Testing
Playables
Assembly Definition
Assembly Definition Reference
TextMeshPro
Scene
Scene Template
Scene Template From Scene
Prefab
Prefab Variant
Audio Mixer
Material
Lens Flare
Render Texture
Lightmap Parameters
Lighting Settings
Custom Render Texture
Animator Controller
Animation
Animator Override Controller
Avatar Mask
Timeline
Signal
Physic Material
GUI Skin
Custom Font
UI Toolkit
Legacy
Brush
Terrain Layer

Rename the **New Animation Controller** if required, for the purpose of this tutorial, we will rename it to **AnimationController**. To keep all our animation related content together, we will move the **AnimationController** to the **Animations** folder.

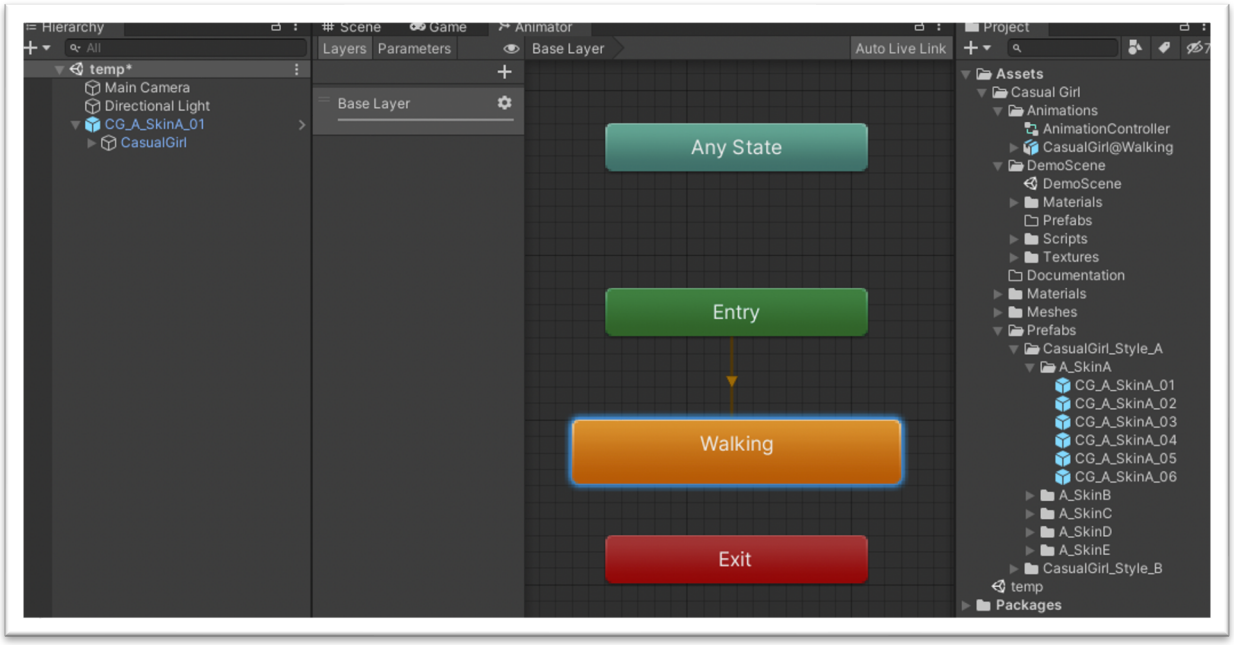


Add the **AnimationController** to the **Character Prefab**, within the **Animator** component.

Step 10: Select **Window** > **Animation** > **Animator** from the main **Unity Menu**. In the **Project View**, click on the **AnimationController** to see it within the **Animator**.



Drag and drop your '**walking**' animation into the **Animator** view, it will automatically connect.



Switch back to your main **Scene**, the press **Play** to see your walking animation applied to your character model prefab. When your press **Play**, the **Game** view will activate, you can switch back the **Scene** view whilst the **Game** view is active to view your animation without any camera restrictions. Simply press the **Play** button again to deactivate.

Customisation

The character model can be customised to create new variations quite easily.

On a **mesh** level, the Prefabs contain extra meshes (hidden) that can be switched on and off as required, depending on the visual aesthetic you require.

On a **material** level, new color combinations can be easily achieved to truly make this character your own. There are some common materials with classic colors already bundled with the package, but there are no limits to what you can achieve with your own materials.

Customisation – Mesh

To create your own mesh variant Prefab, make sure you have at least one Prefab in your Scene, in the **Hierarchy**, right click on it, and select **Prefab > Unpack Completely**. This will give us a base to work with. Open the Prefabs parent child relationship, and expand the contents of the **visualMesh**.

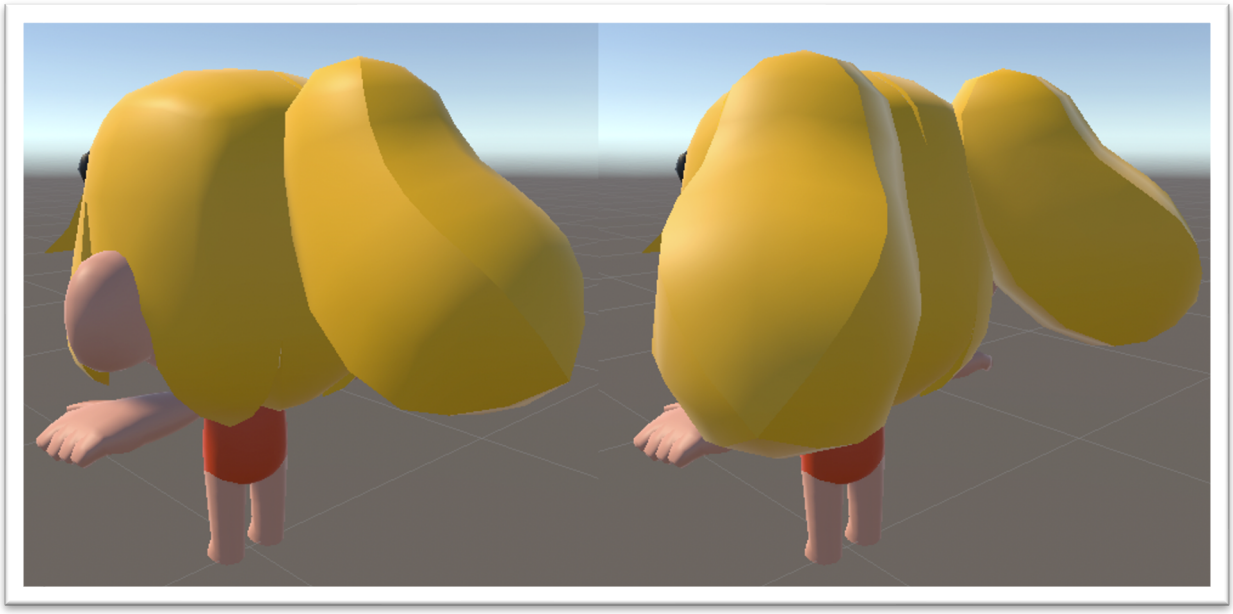
In the expanded view, you will see some objects are greyed out, these are hidden from view, and can be switched on or off as required for your customised mesh. Let's look at some of the options available.

BodyGroup

If we turn **OFF** **ArmL_Sleeve**, **ArmR_Sleeve** and **BodyA**, and then turn **ON** **ArmL**, **ArmR** and **BodyB**, we get a character ready for the beach!

HairGroup

We can add a ponytail to the character by displaying the **HairPonyTail** by turning it on in the **Inspector** if required, or we can give the character hair bunches by displaying the **HairBunches** object instead. We can even just hide the whole **HairGroup** to create a bald character.



HeadGroup

We can create another whole visual look for our character by simply turning **OFF** **EyeMainL** and **EyeMainR**, and turning **ON** **EyeL** and **EyeR**.

Combine these **Mesh** changes with **Material** changes and you can really make this character your own with a unique look ready for your game or prototype.

MouthGroup

Under the **MouthGroup**, we have a number of mouth shapes from neutral to extended smile, for negative mouth shapes, we can simple take any of the given mouth shapes and rotate **180 degrees** in the Inspector, turn that smile upside down!

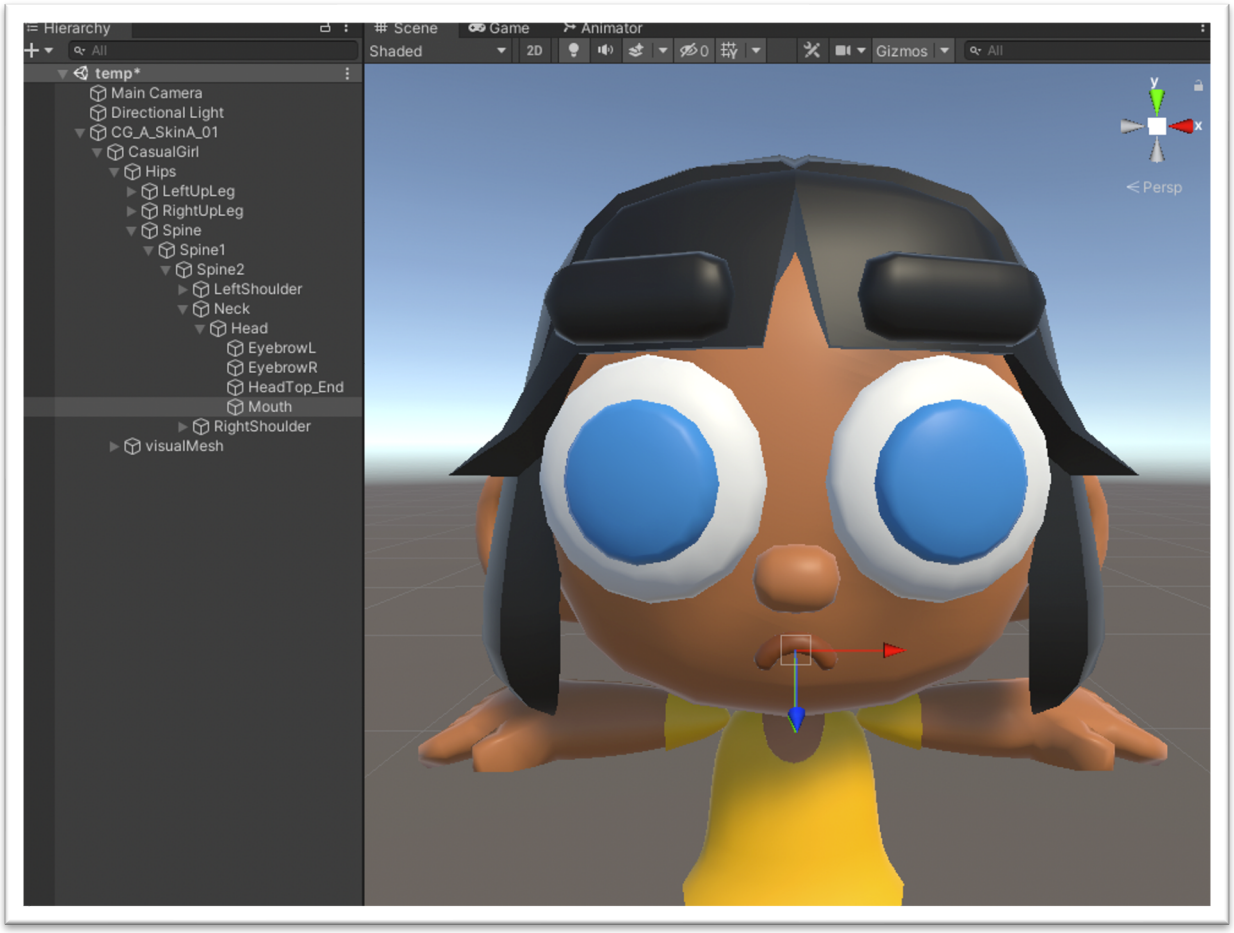
Customisation - Materials

By combining **Mesh** changes with **Material** assignments, we can customise this character with many possible combinations, if you expand the **Materials** folder, and then expand the **Character** and **Colors** folder therein, you will see there is some common materials to get you started.

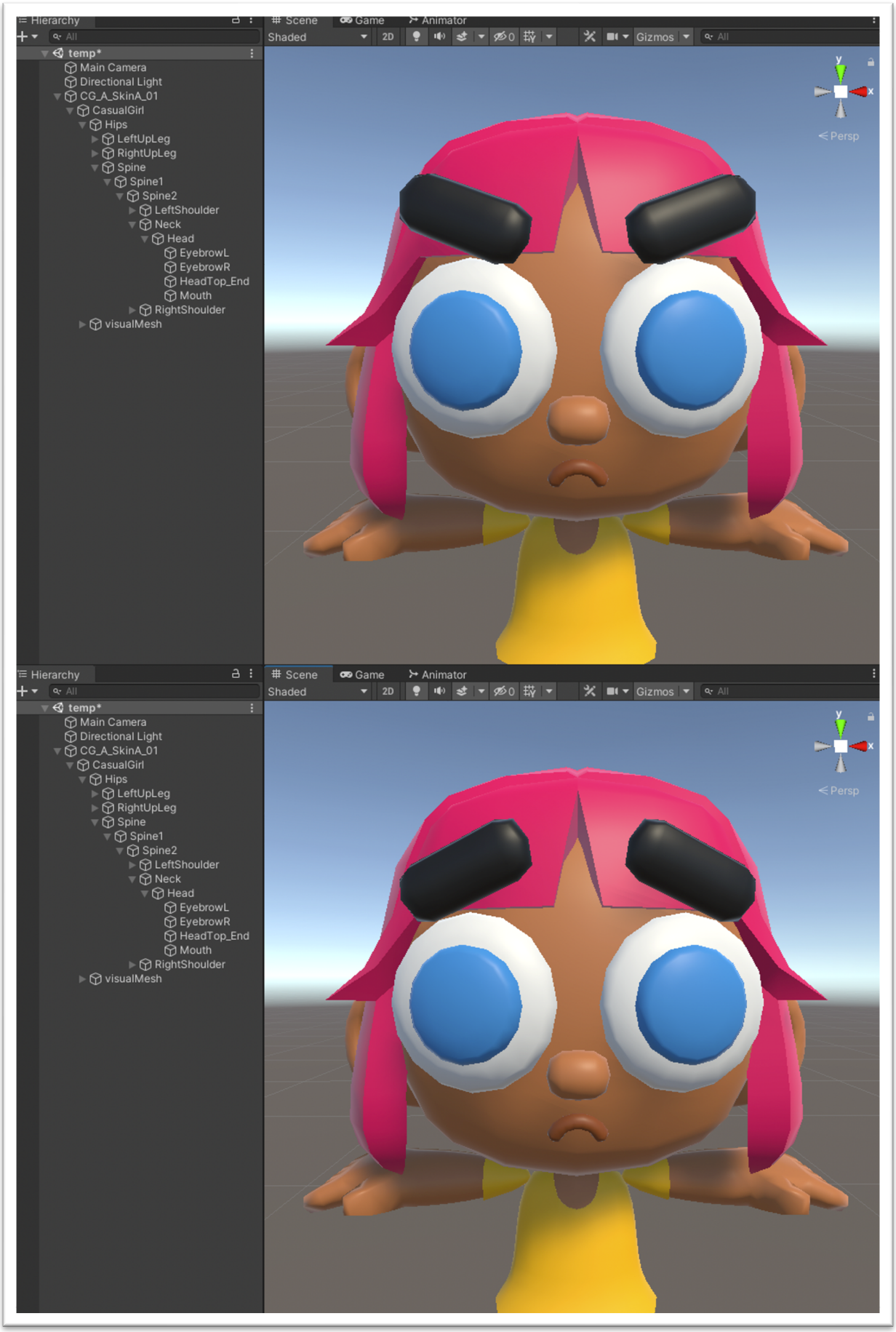
Under the **Character** folder we have some specific **Materials** for **Character Personalisation**, **Materials** to change eye attributes, eyebrow color and skin color (with a component for skin lip color also). In this instance we have modified our characters skin color, eyes, hair and clothes using the default material selection.

Further Customisation

Further Customisation can be achieved by modifying the character rig itself, as previously mentioned, we can create an unhappy expression by rotating the mouth, to do this, we expand our rig in the **Hierarchy**. Then expand the **Spine** child object until you find the **Mouth** bone, to create an un-happy expression, we would rotate this bone **180 degrees** on the **Z Axis**.



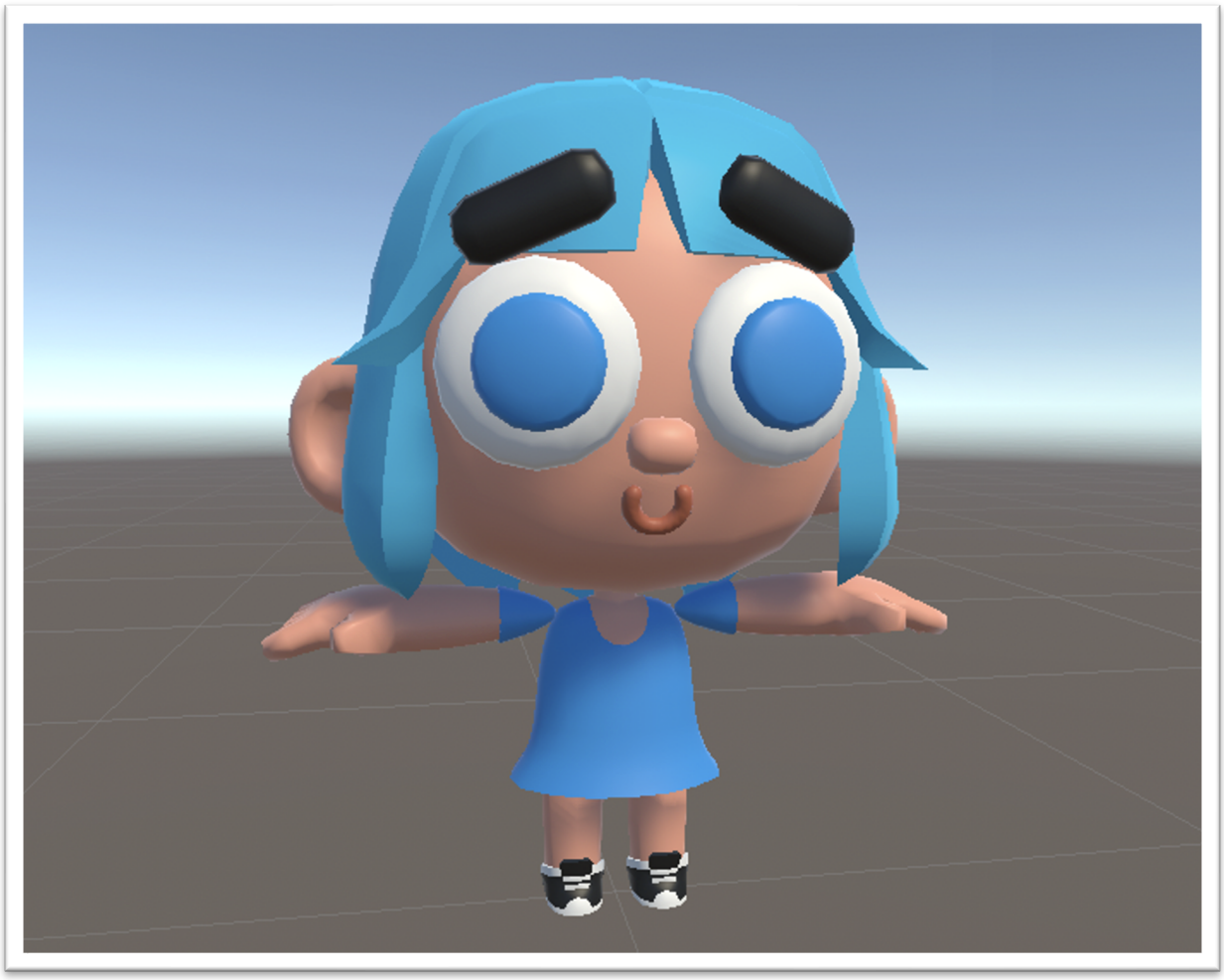
Likewise, we can manipulate the Eyebrows also to create an angry or despondent look.


Hierarchy
Scene
Game
Animator
Shaded
2D
Gizmos
All
temp*
Main Camera
Directional Light
CG_A_SkinA_01
CasualGirl
Hips
LeftUpLeg
RightUpLeg
Spine
Spine1
Spine2
LeftShoulder
Neck
Head
EyebrowL
EyebrowR
HeadTop_End
Mouth
RightShoulder
visualMesh
Persp
Hierarchy
Scene
Game
Animator
Shaded
2D
Gizmos
All
temp*
Main Camera
Directional Light
CG_A_SkinA_01
CasualGirl
Hips
LeftUpLeg
RightUpLeg
Spine
Spine1
Spine2
LeftShoulder
Neck
Head
EyebrowL
EyebrowR
HeadTop_End
Mouth
RightShoulder
visualMesh
Persp

The **Mouth** and **Eyebrow Bones** are '**free bones**', this means they can be manipulated without any worry that they will be affected if you import an animation in from **Mixamo** or another source, it is **not** recommended to change the orientation of other bones in the character hierarchy, as these are labelled to work within a **Humanoid** rig, and would be more than likely to be called upon should you use external animations designed for the Humanoid Rig.

New Prefabs

Once you have customised your own character model to your liking, simply select the root level of the Prefab we initially unpacked. To avoid conflicts, I'd recommend renaming it to something different from any other pre-existing Prefabs, then, all we need to do is drag and drop the customised model into the **Prefabs** directory to create your new character Prefab.



Rate

If you like this Unity Asset, please Rate it!
Thank you!