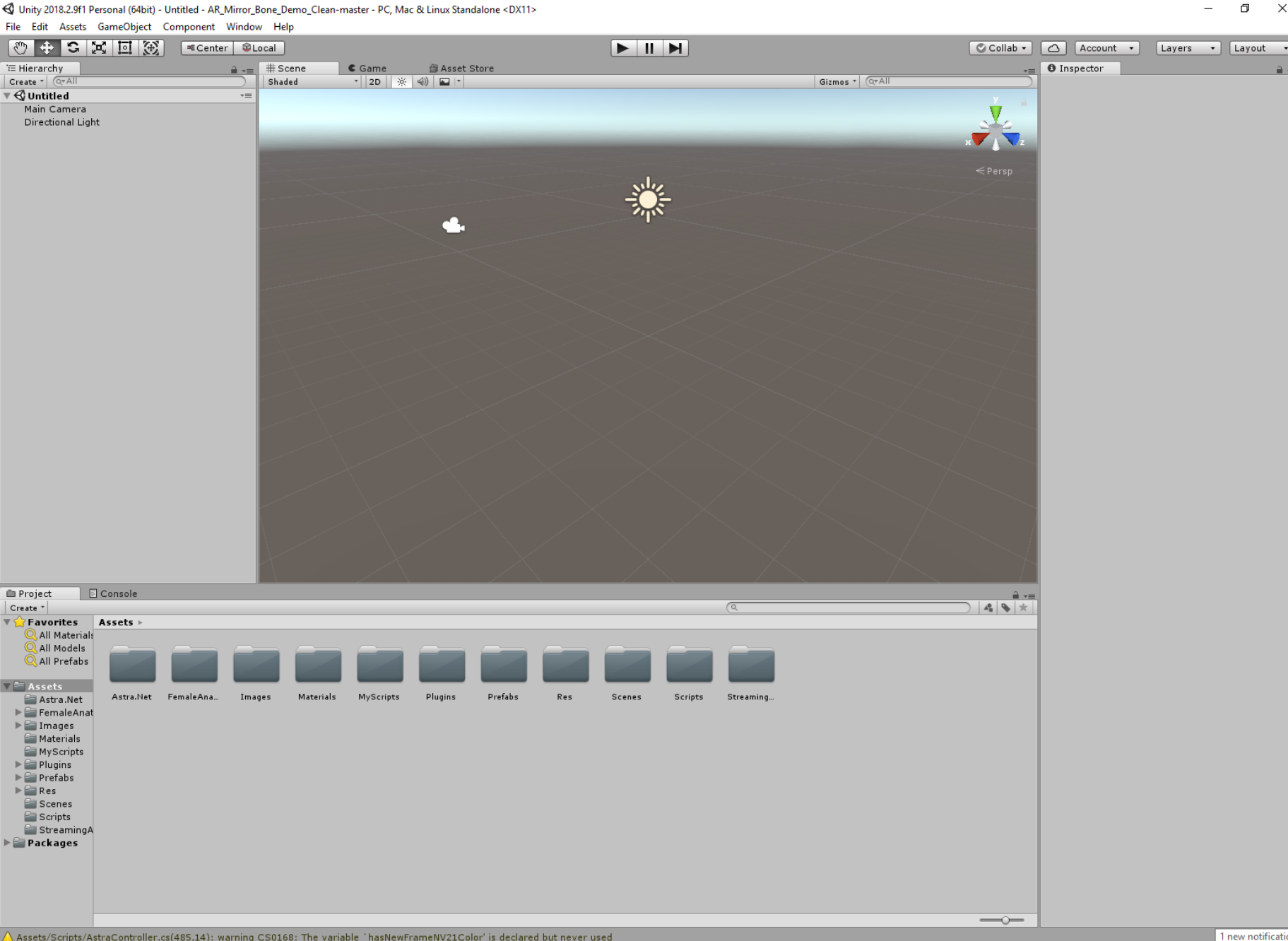


Augmented Reality Mirror (ar_mirror) Demo Tutorial and Lesson Plan Guide



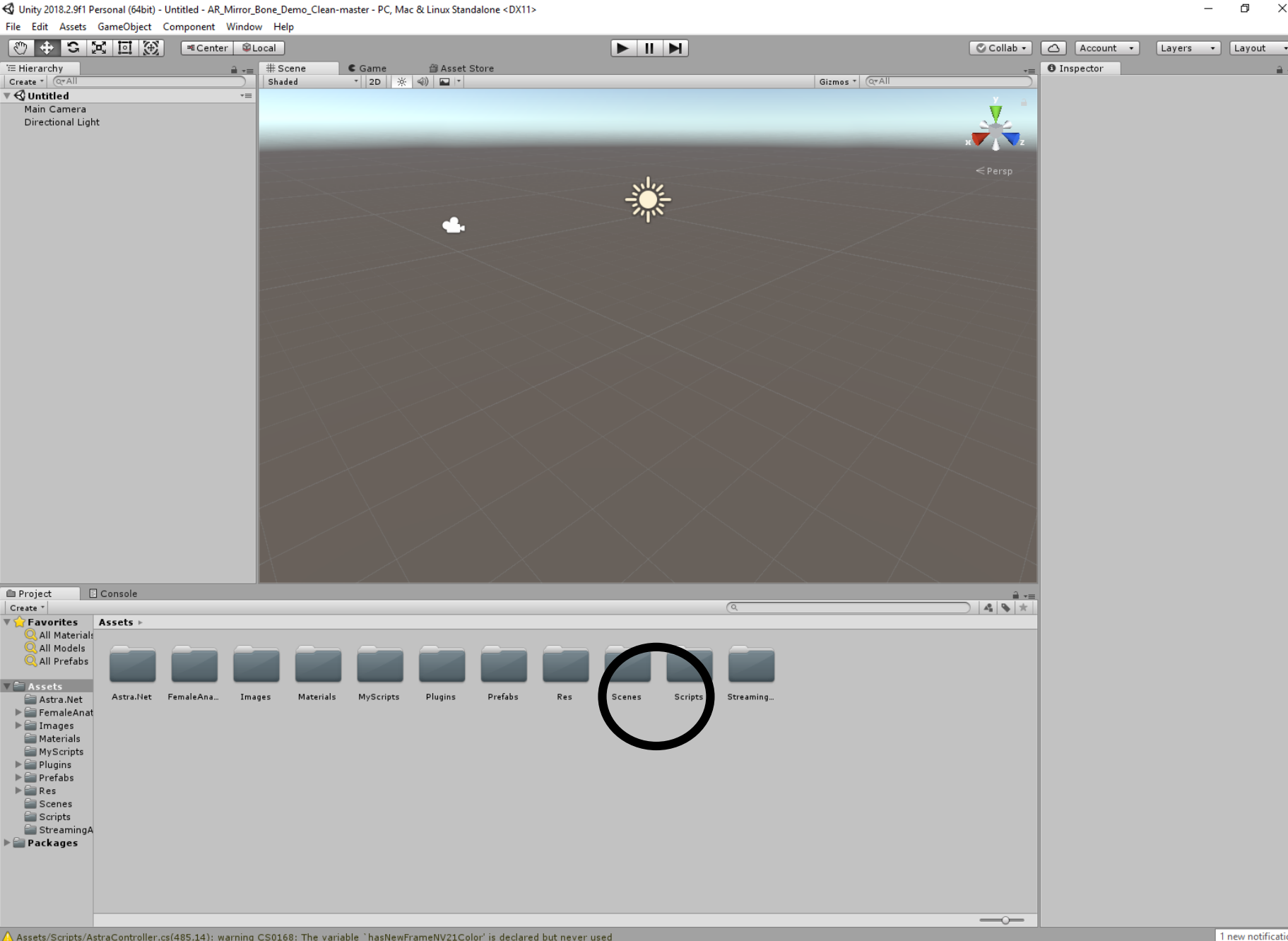
Guide Objectives

- **Provide teachers a general guide how to navigate the ar_mirror physiology demos and to provide students with project ideas.**



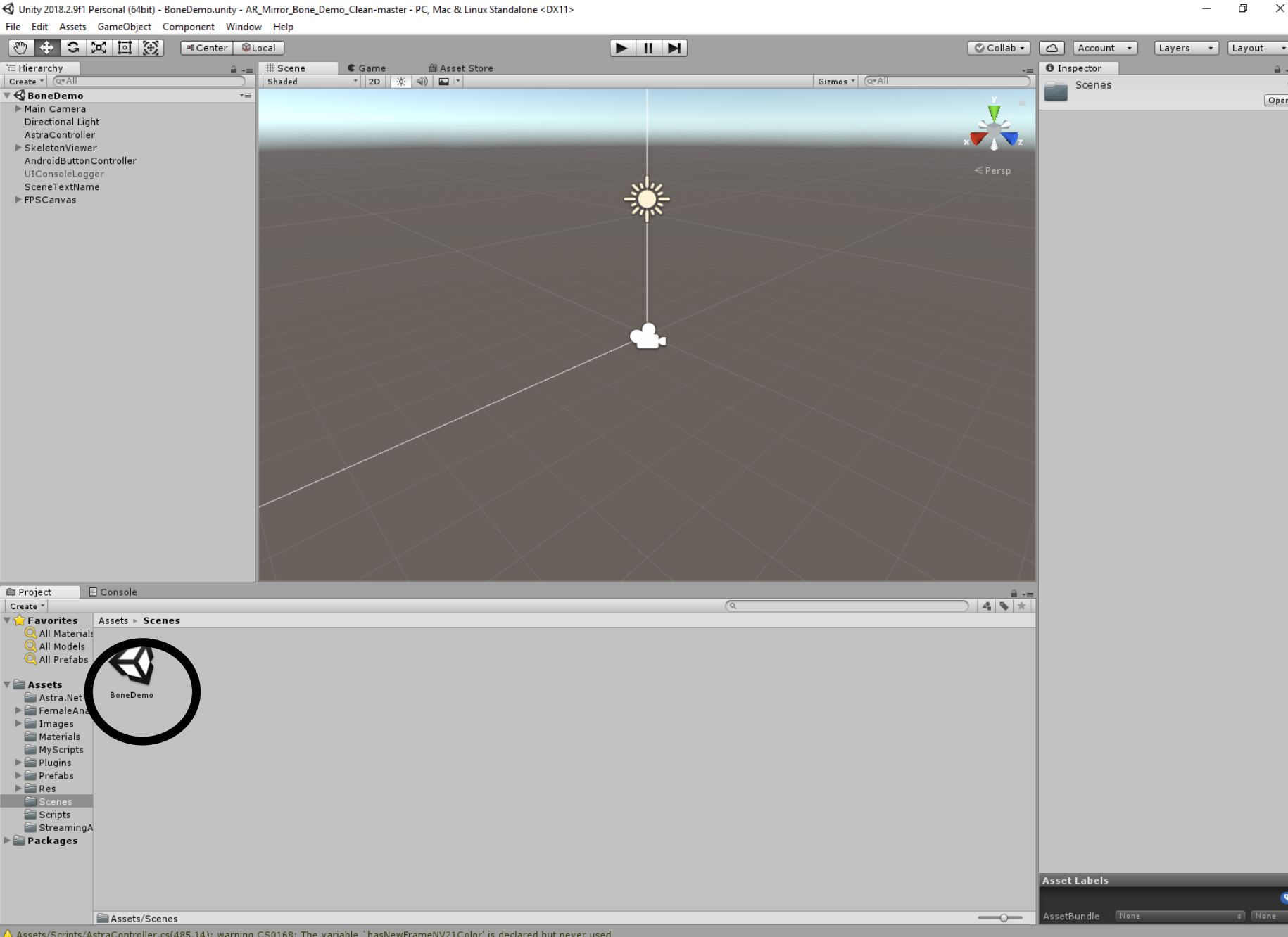
**Open
AR_Mirror_Bone_
Demo in Unity**

**The main screen
should look
something like
this...**



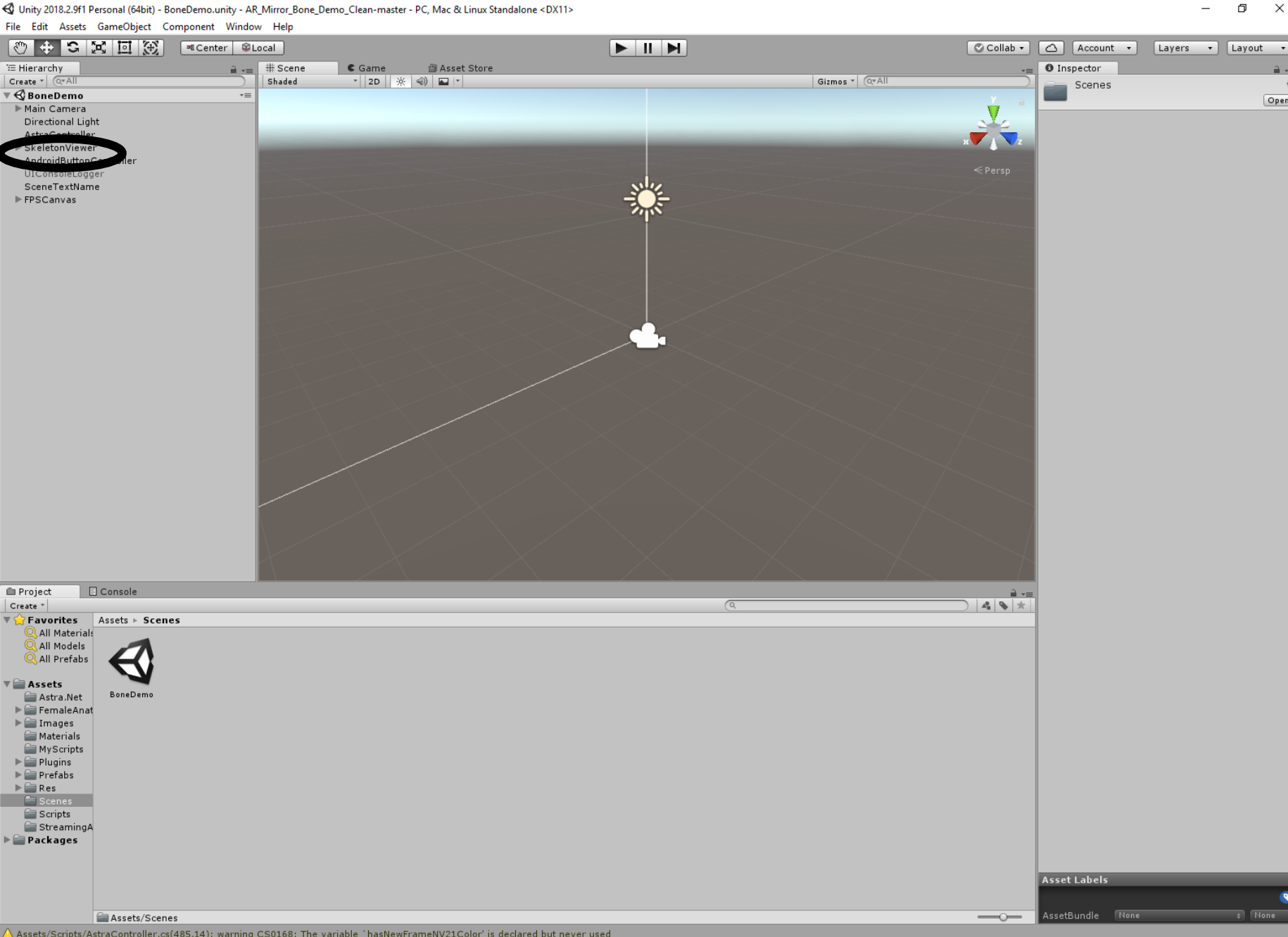
You'll notice many assets folders available. Unity allows for multiple scenes to be created.

The demo's only have one scene, so we will open it and check it out.

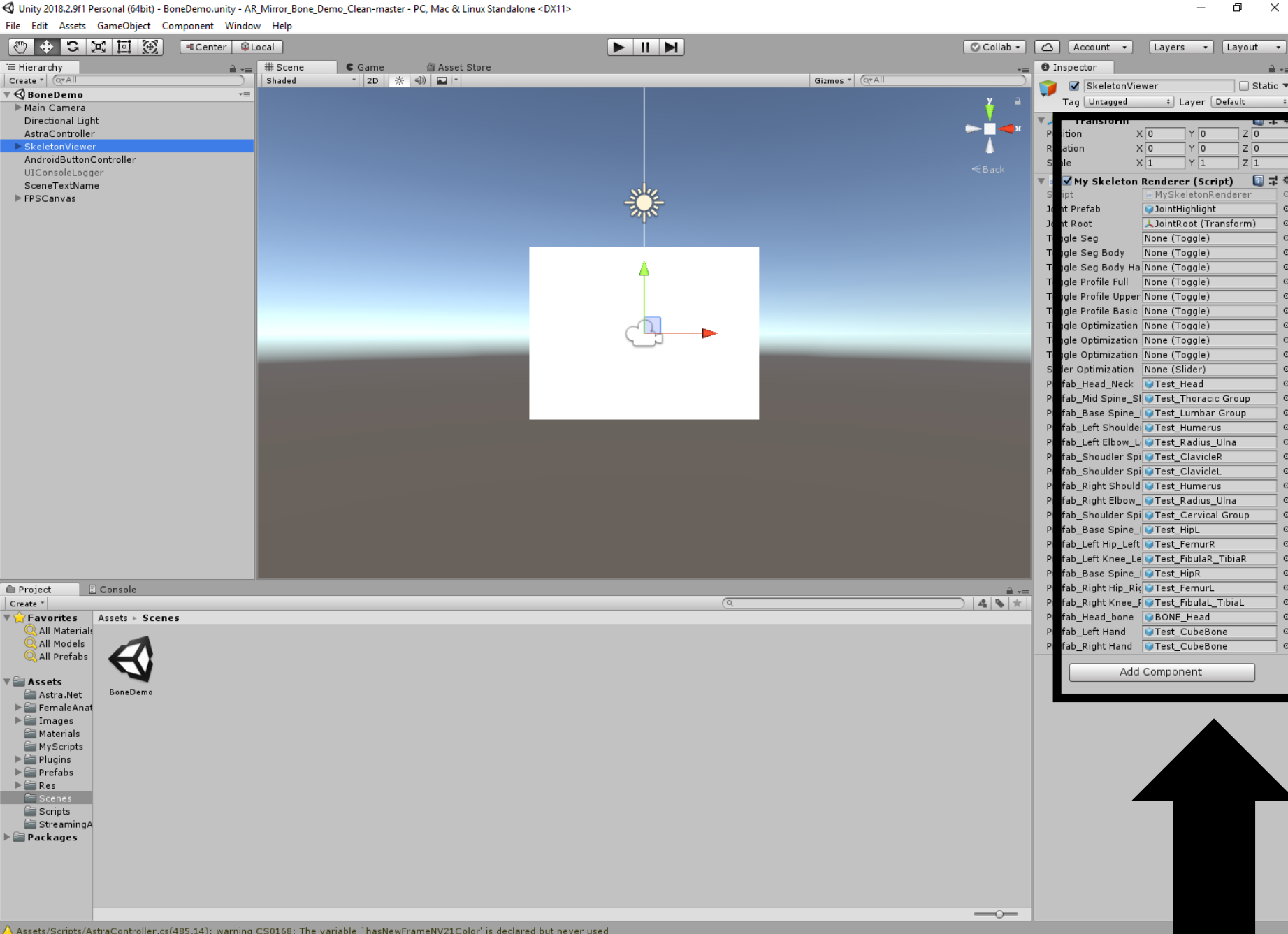


**Open the
BoneDemo scene**

Changing Bone Overlays



**Open the
SkeletonViewer**

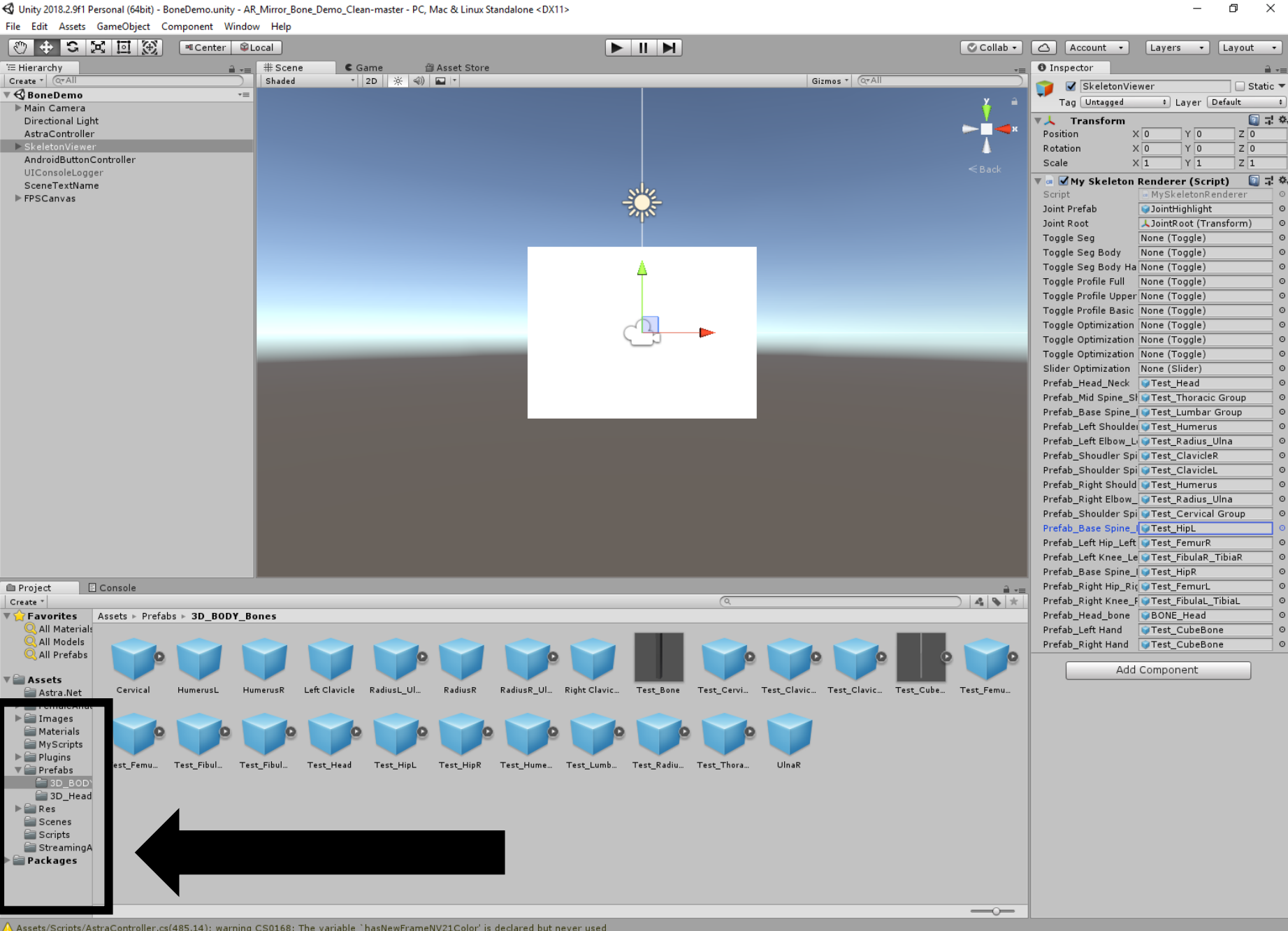


This provides you access to where the specific bone images are linked to each respective joint. You can change these to whatever images you have.

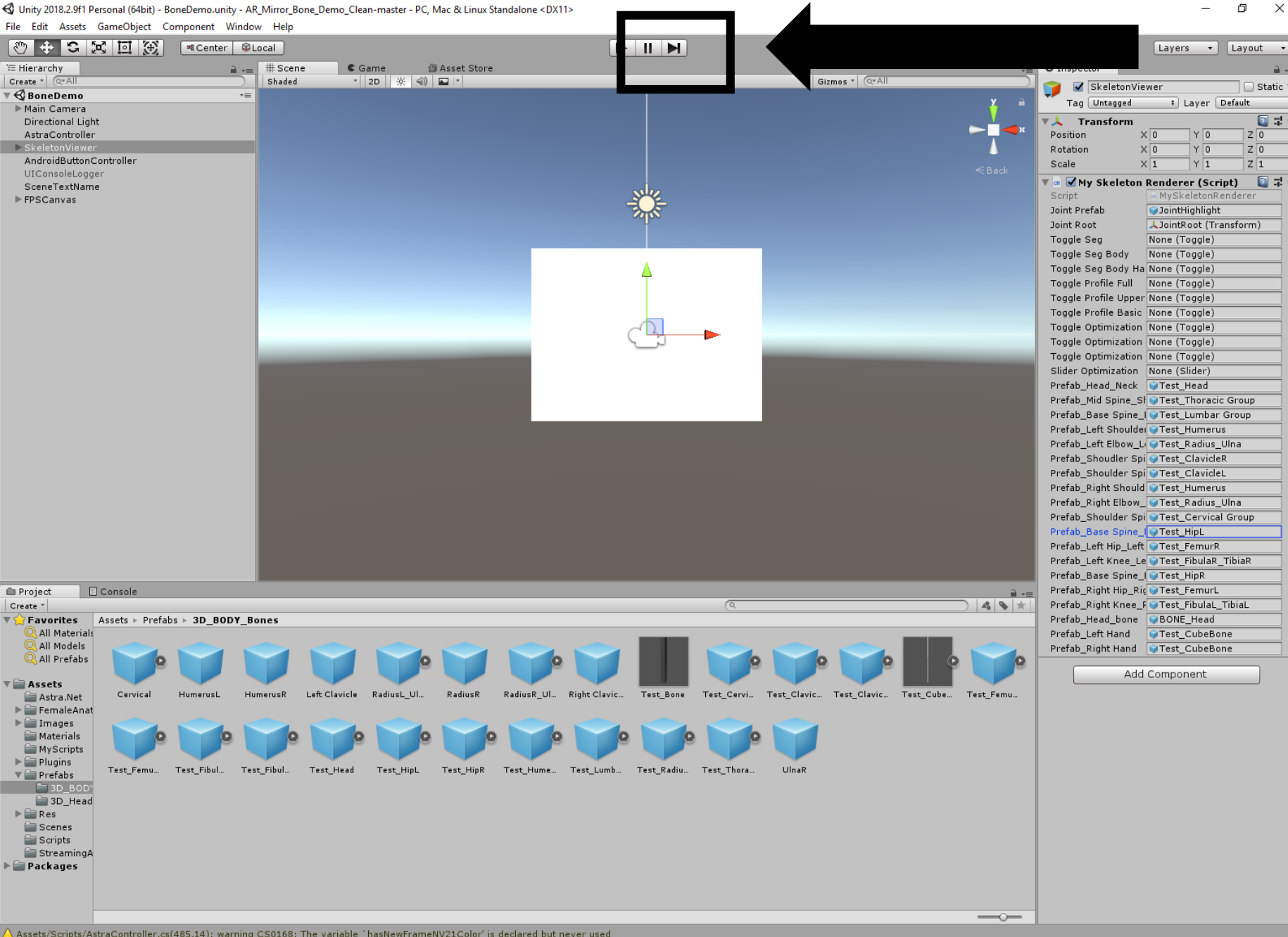


Note: The Orbbec camera recognizes each of the labelled joints. Technically, you can associate a 3D image between any of these respective joints.

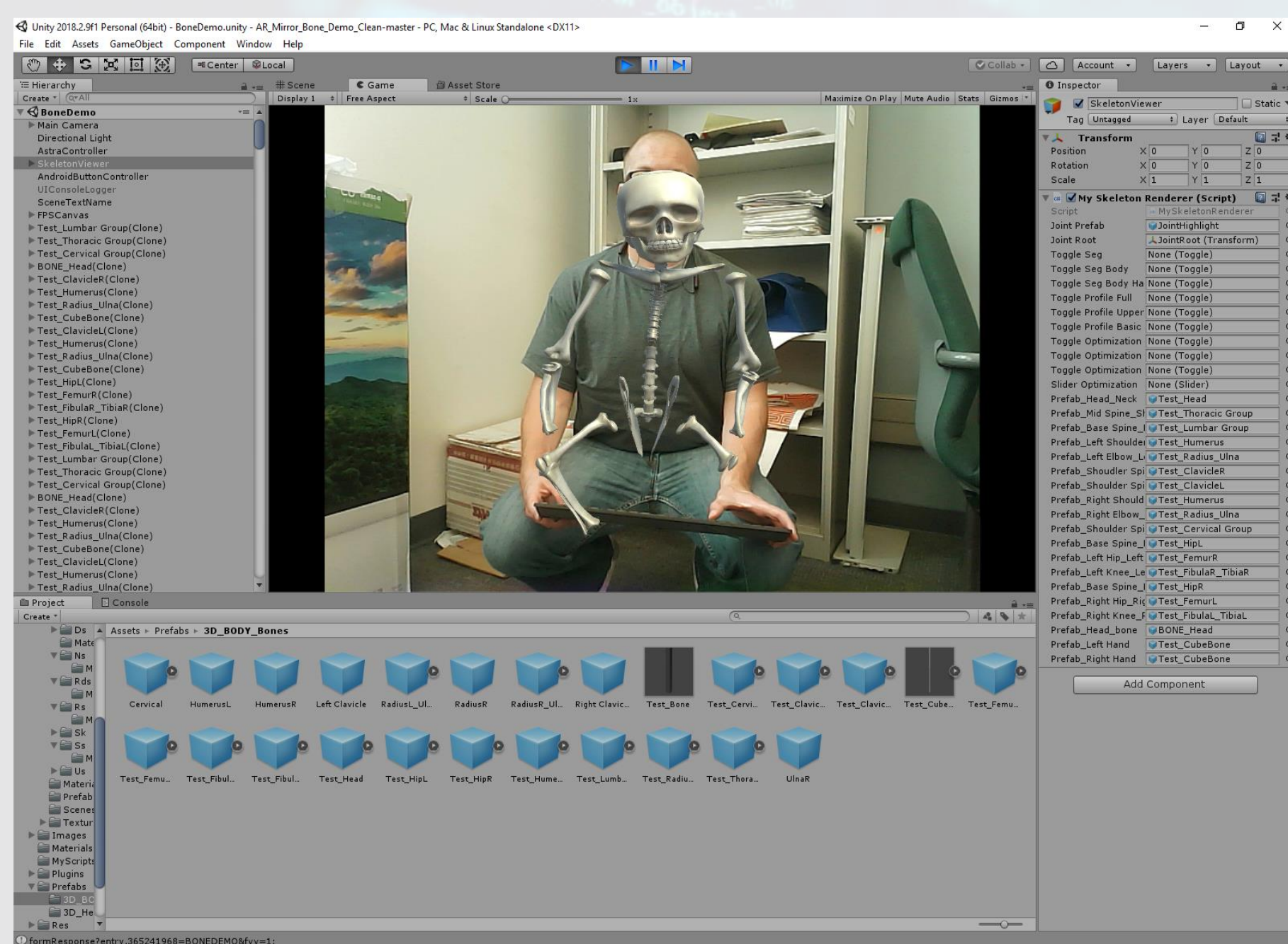
You could have the students overlay images in any combination.



The images are saved in the “Prefabs” folder, and you can add more images here to allow you to plug and play.

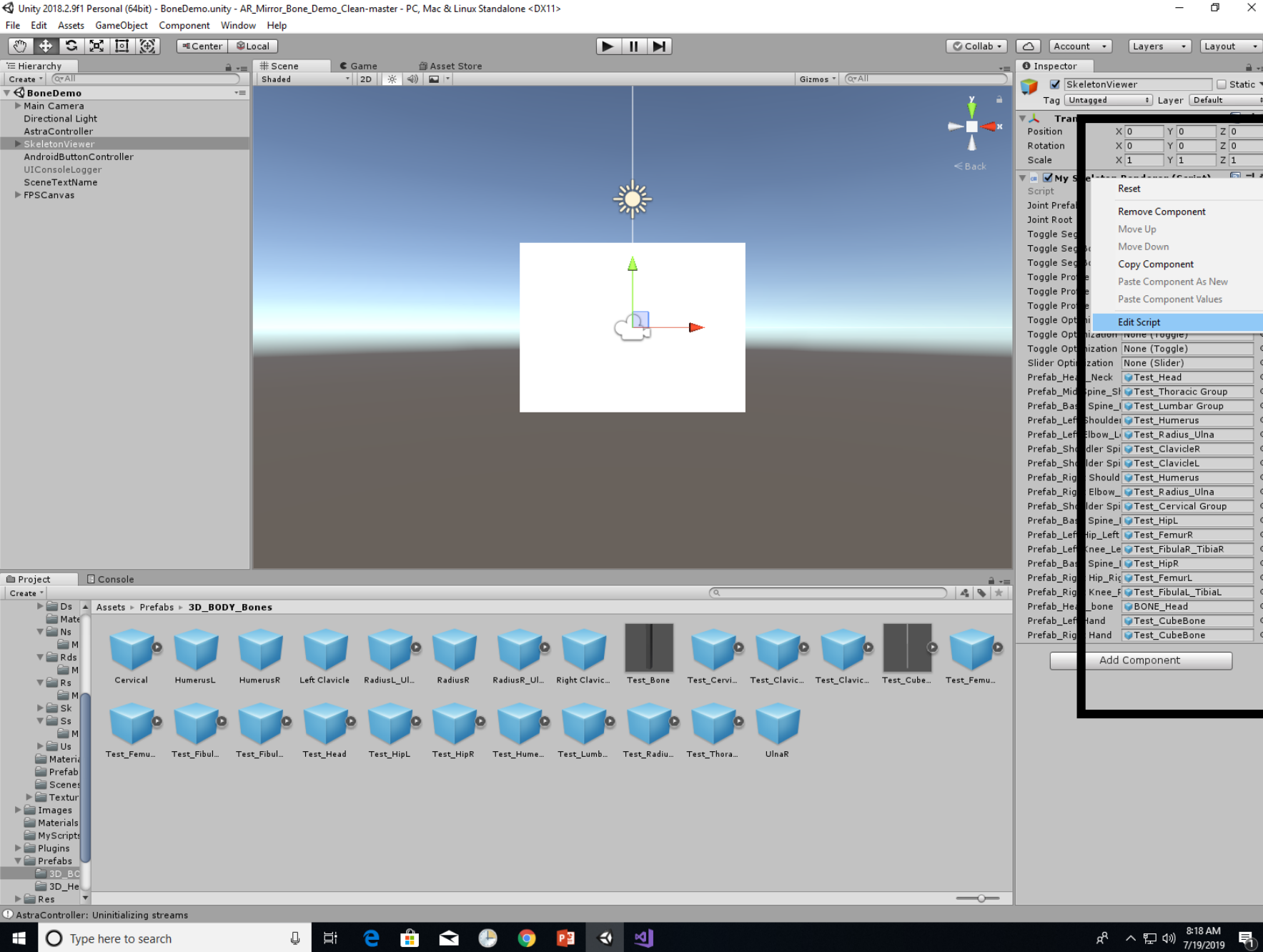


If you press the play button, you can see how your changes affect the program.



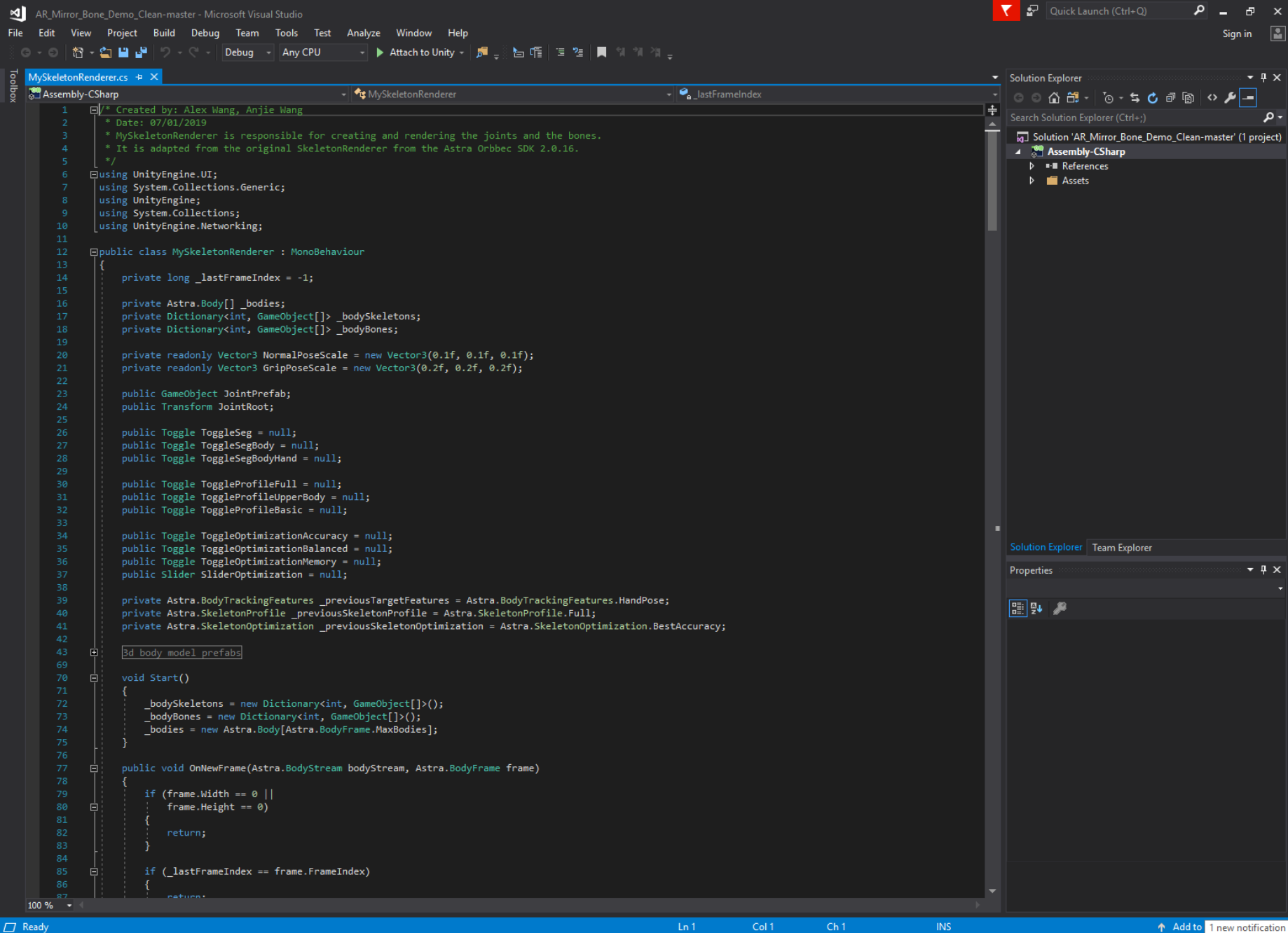
What you see in the main screen is what you'll see when your demo is running.

When you're finished checking, press the play button again to stop the demo



**You can also
manipulate the
code by right
clicking in box
and selecting
“Edit Script”**

**This opens Visual
Studio to view
your C# script.**



This allows you to check out the script responsible for rendering the bone overlay in the program.

Some suggestions are to have the you annotate the code or see what happens if they modify the code.

Example: Adding a bone between the hand joints

AR_Mirror_Bone_Demo_Clean-master (Running) - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Process: [N/A] Lifecycle Events Thread: Stack Frame:

MySkeletonRenderer.cs

```
37 public Slider SliderOptimization = null;
38
39 private Astra.BodyTrackingFeatures _previousTargetFeatures = Astra.BodyTrackingFeatures.HandPose;
40 private Astra.SkeletonProfile _previousSkeletonProfile = Astra.SkeletonProfile.Full;
41 private Astra.SkeletonOptimization _previousSkeletonOptimization = Astra.SkeletonOptimization.BestAccuracy;
42
43 #region 3d body model prefabs
44 //Bone Prefabs
45 public GameObject Prefab_Head_Neck;
46 public GameObject Prefab_MidSpine_ShoulderSpine;
47 public GameObject Prefab_BaseSpine_MidSpine;
48 public GameObject Prefab_LeftShoulder_LeftElbow;
49 public GameObject Prefab_LeftElbow_LeftWrist;
50 public GameObject Prefab_ShoulderSpine_LeftShoulder;
51 public GameObject Prefab_ShoulderSpine_RightShoulder;
52 public GameObject Prefab_RightShoulder_RightElbow;
53 public GameObject Prefab_RightElbow_RightWrist;
54 public GameObject Prefab_ShoulderSpine_Neck;
55 public GameObject Prefab_BaseSpine_LeftHip;
56 public GameObject Prefab_LeftHip_LeftKnee;
57 public GameObject Prefab_LeftKnee_LeftFoot;
58 public GameObject Prefab_BaseSpine_RightHip;
59 public GameObject Prefab_RightHip_RightKnee;
60 public GameObject Prefab_RightKnee_RightFoot;
61 public GameObject Prefab_Head_bone;
62 public GameObject Prefab_LeftHand;
63 public GameObject Prefab_RightHand;
64 public GameObject Prefab_LeftHand_RightHand;
65
```

Step 1.

**Define a new
GameObject.**

AR_Mirror_Bone_Demo_Clean-master (Running) - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Process: [N/A] Lifecycle Events Thread: Stack Frame:

MySkeletonRenderer.cs

```
Assembly-CSharp MySkeletonRenderer.Bone
127 _bodySkeletons.Add(body.Id, joints);
128
129 //Instantiate bone gameobjects
130 bones = new GameObject[Bones.Length];
131 for (int i = 0; i < bones.Length; i++)
132 {
133     bones[0] = (GameObject)Instantiate(Prefab_BaseSpine_MidSpine, Vector3.zero, Quaternion.identity);
134     bones[1] = (GameObject)Instantiate(Prefab_MidSpine_ShoulderSpine, Vector3.zero, Quaternion.identity);
135     bones[2] = (GameObject)Instantiate(Prefab_ShoulderSpine_Neck, Vector3.zero, Quaternion.identity);
136     bones[3] = (GameObject)Instantiate(Prefab_Head_bone, Vector3.zero, Quaternion.identity);
137     bones[4] = (GameObject)Instantiate(Prefab_ShoulderSpine_LeftShoulder, Vector3.zero, Quaternion.identity);
138     bones[5] = (GameObject)Instantiate(Prefab_LeftShoulder_LeftElbow, Vector3.zero, Quaternion.identity);
139     bones[6] = (GameObject)Instantiate(Prefab_LeftElbow_LeftWrist, Vector3.zero, Quaternion.identity);
140     bones[7] = (GameObject)Instantiate(Prefab_LeftHand, Vector3.zero, Quaternion.identity);
141     bones[8] = (GameObject)Instantiate(Prefab_ShoulderSpine_RightShoulder, Vector3.zero, Quaternion.identity);
142     bones[9] = (GameObject)Instantiate(Prefab_RightShoulder_RightElbow, Vector3.zero, Quaternion.identity);
143     bones[10] = (GameObject)Instantiate(Prefab_RightElbow_RightWrist, Vector3.zero, Quaternion.identity);
144     bones[11] = (GameObject)Instantiate(Prefab_RightHand, Vector3.zero, Quaternion.identity);
145     bones[12] = (GameObject)Instantiate(Prefab_BaseSpine_LeftHip, Vector3.zero, Quaternion.identity);
146     bones[13] = (GameObject)Instantiate(Prefab_LeftHip_LeftKnee, Vector3.zero, Quaternion.identity);
147     bones[14] = (GameObject)Instantiate(Prefab_LeftKnee_LeftFoot, Vector3.zero, Quaternion.identity);
148     bones[15] = (GameObject)Instantiate(Prefab_BaseSpine_RightHip, Vector3.zero, Quaternion.identity);
149     bones[16] = (GameObject)Instantiate(Prefab_RightHip_RightKnee, Vector3.zero, Quaternion.identity);
150     bones[17] = (GameObject)Instantiate(Prefab_RightKnee_RightFoot, Vector3.zero, Quaternion.identity);
151     bones[18] = (GameObject)Instantiate(Prefab_LeftHand_RightHand, Vector3.zero, Quaternion.identity);
152
```

Step 2.

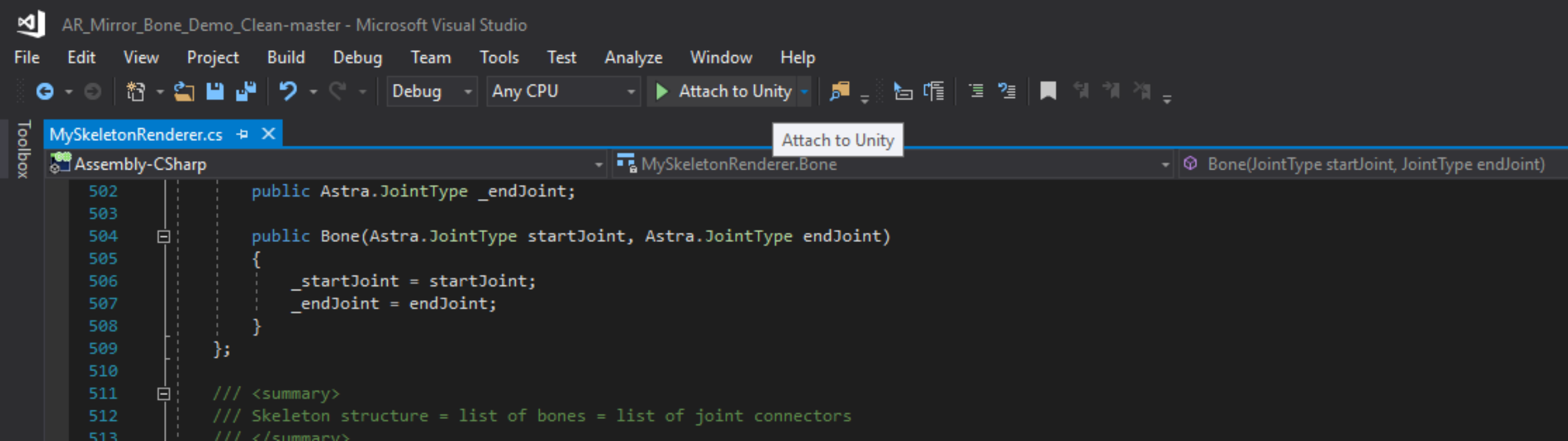
Define a new bone. This allows you to select an image for this in Unity

```
AR_Mirror_Bone_Demo_Clean-master (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Process: [N/A] Lifecycle Events Thread: Stack Fr
MySkeletonRender.cs
Assembly-CSharp MySkeletonRender.Bone
511 /// <summary>
512 /// Skeleton structure = list of bones = list of joint connectors
513 /// </summary>
514 private Bone[] Bones = new Bone[]
515 {
516     // spine, neck, and head
517     new Bone(Astra.JointType.BaseSpine, Astra.JointType.MidSpine),
518     new Bone(Astra.JointType.MidSpine, Astra.JointType.ShoulderSpine),
519     new Bone(Astra.JointType.ShoulderSpine, Astra.JointType.Neck),
520     new Bone(Astra.JointType.Neck, Astra.JointType.Head),
521     // left arm
522     new Bone(Astra.JointType.ShoulderSpine, Astra.JointType.LeftShoulder),
523     new Bone(Astra.JointType.LeftShoulder, Astra.JointType.LeftElbow),
524     new Bone(Astra.JointType.LeftElbow, Astra.JointType.LeftWrist),
525     new Bone(Astra.JointType.LeftWrist, Astra.JointType.LeftHand),
526     // right arm
527     new Bone(Astra.JointType.ShoulderSpine, Astra.JointType.RightShoulder),
528     new Bone(Astra.JointType.RightShoulder, Astra.JointType.RightElbow),
529     new Bone(Astra.JointType.RightElbow, Astra.JointType.RightWrist),
530     new Bone(Astra.JointType.RightWrist, Astra.JointType.RightHand),
531     // left leg
532     new Bone(Astra.JointType.BaseSpine, Astra.JointType.LeftHip),
533     new Bone(Astra.JointType.LeftHip, Astra.JointType.LeftKnee),
534     new Bone(Astra.JointType.LeftKnee, Astra.JointType.LeftFoot),
535     // right leg
536     new Bone(Astra.JointType.BaseSpine, Astra.JointType.RightHip),
537     new Bone(Astra.JointType.RightHip, Astra.JointType.RightKnee),
538     new Bone(Astra.JointType.RightKnee, Astra.JointType.RightFoot),
539     // between hands
540     new Bone(Astra.JointType.LeftHand, Astra.JointType.RightHand),
541 };
542 #endregion
```

Step 3.

Define the connecting points.

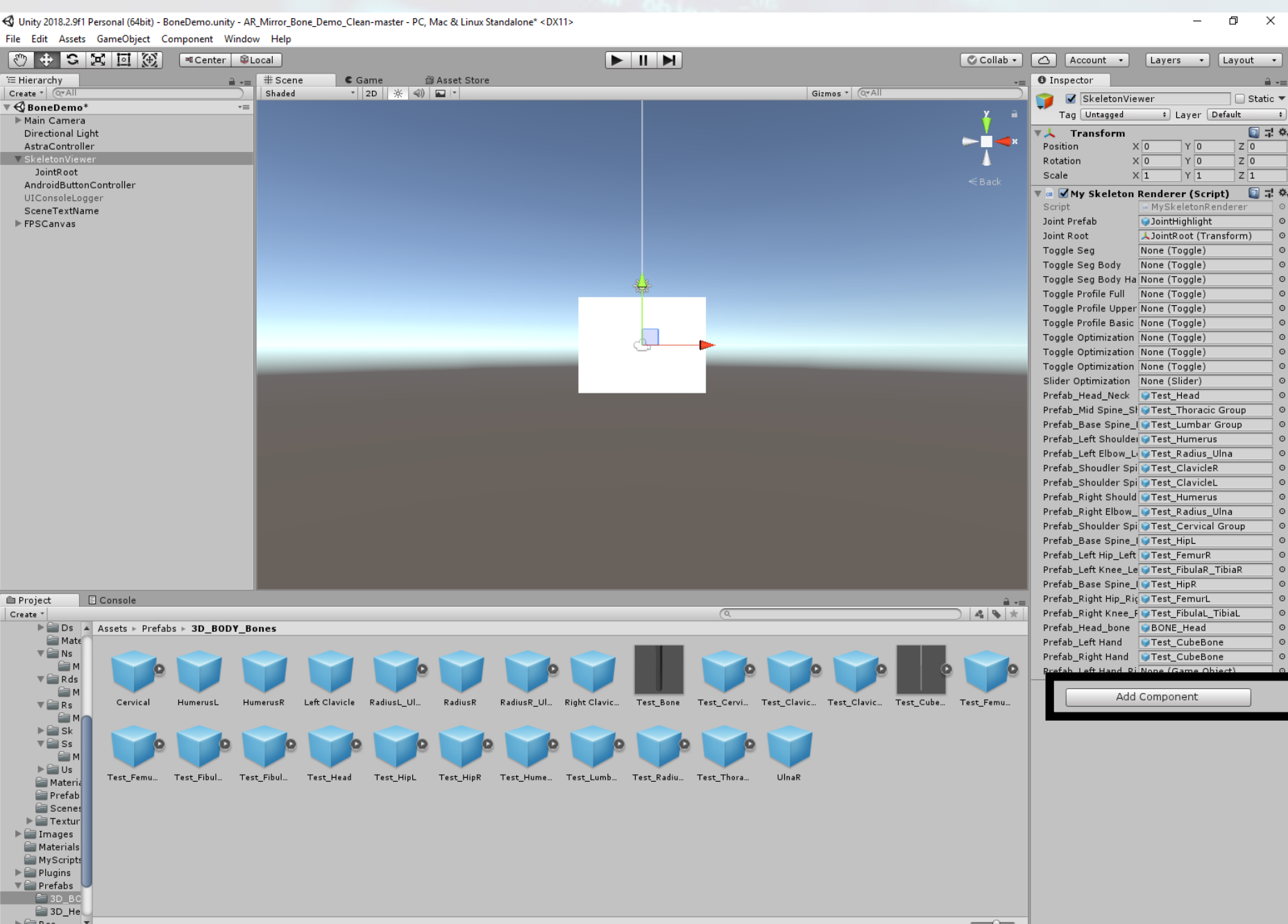
In this case, the bone you defined previously will now be connected between these two joints.



Step 4.

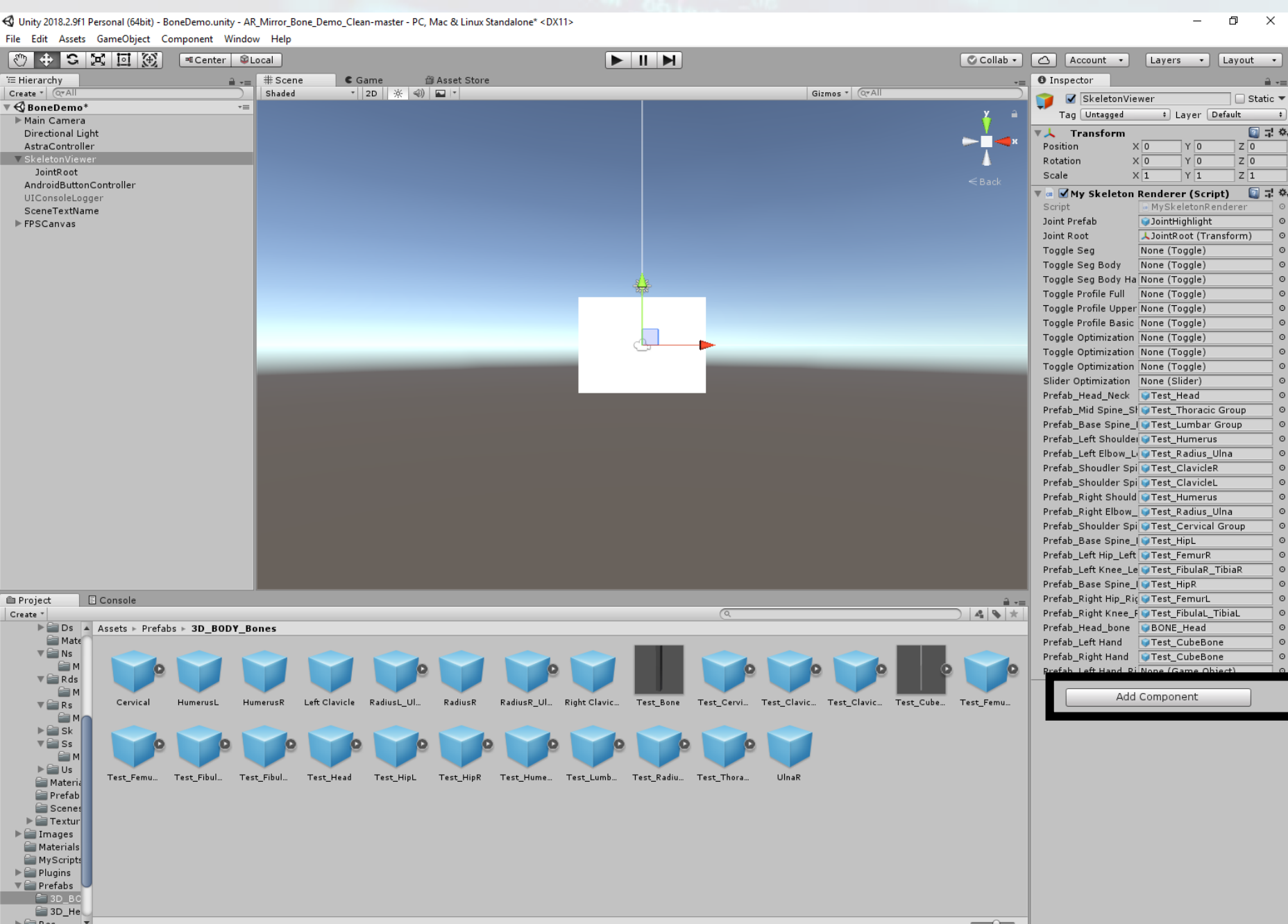
Attach your changes to Unity. This will allow you to now update the images to attach, including the option to attach an image between the hand joints.

Note: If any syntax errors occurred, Visual Studio will flag it and prompt you to make any necessary corrections/changes.



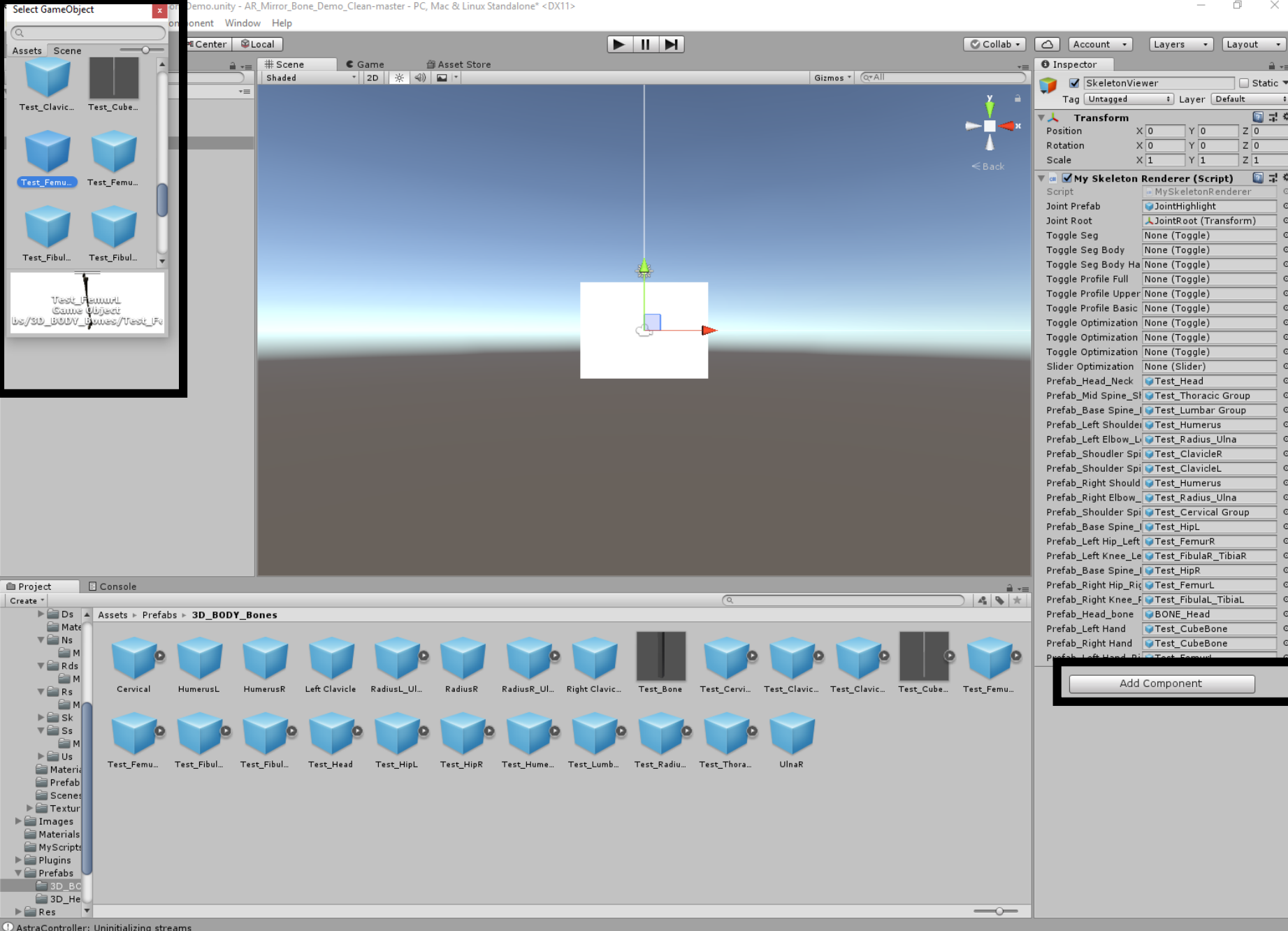
Back in Unity, the renderer has been updated, and a new option to add an image is available.

When you click on it, it opens the image folder and allows you to select an image.



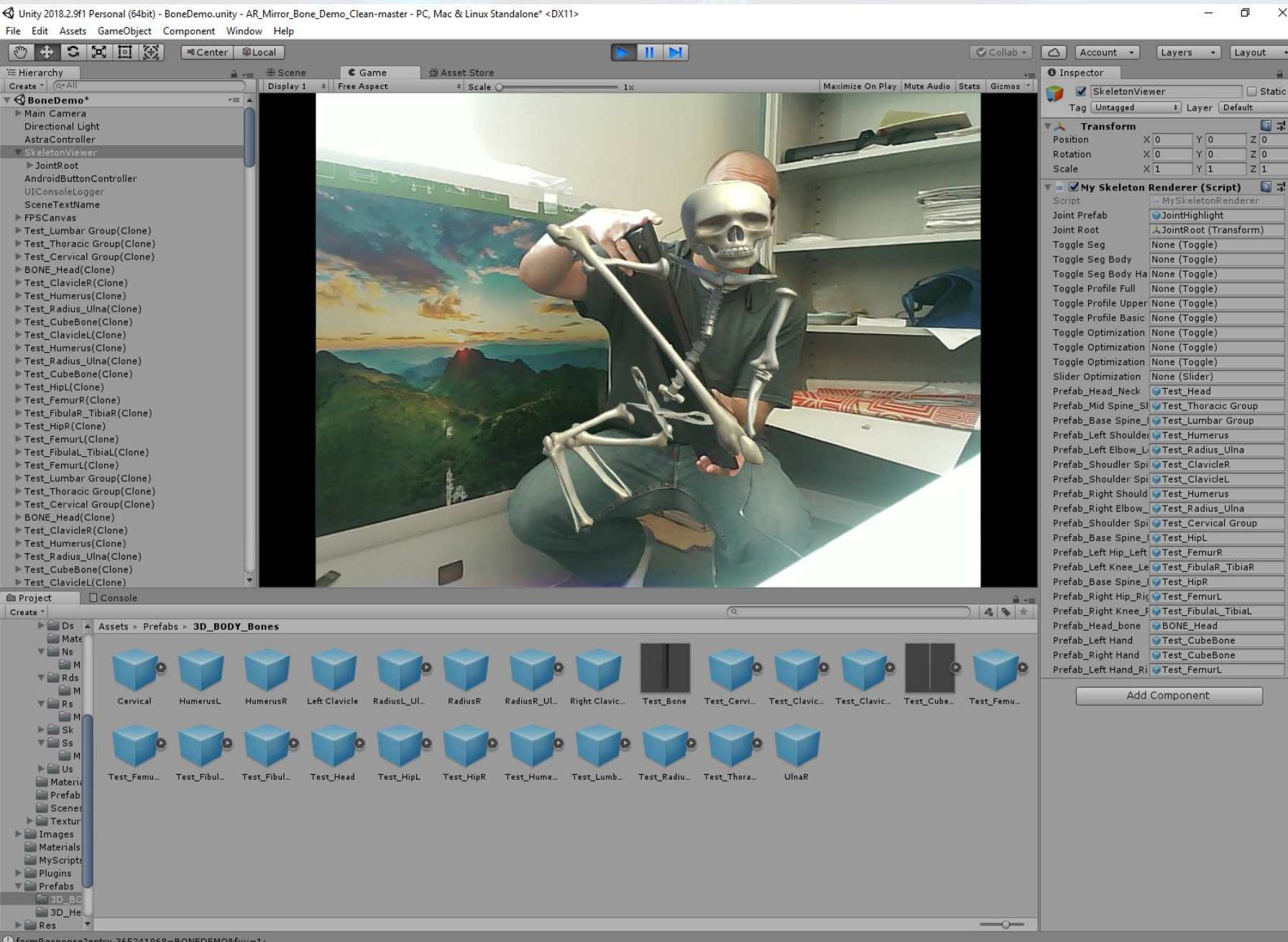
Back in Unity, the renderer has been updated, and a new option to add an image is available.

When you click on it, it opens the image folder and allows you to select an image.



Back in Unity, the renderer has been updated, and a new option to add an image is available.

When you click on it, it opens the image folder and allows you to select an image.



Femur is attached!

Note: Since the model is not upright, the rendering can be a bit off. It works best when the user is upright.