Data Driven Computer Animation

HKU COMP 3360

Tutorial 2 - Scripting for Animation

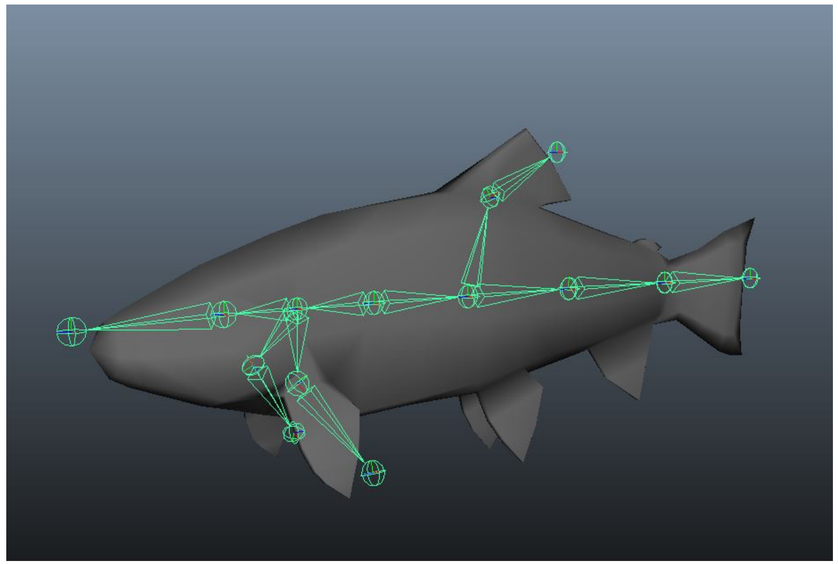
Prof. Taku Komura
TA: Shi Mingyi (myshi@cs.hku.hk)

SECTION 2A, 2021

Some questions by Email

1. Does rigging work on other 3d model like animals?

   Yes. Rigging is a basic technology for helping the manipulation on any kind of mesh with control points. Humanoid model is more common, but for other mesh, the basic idea is same.



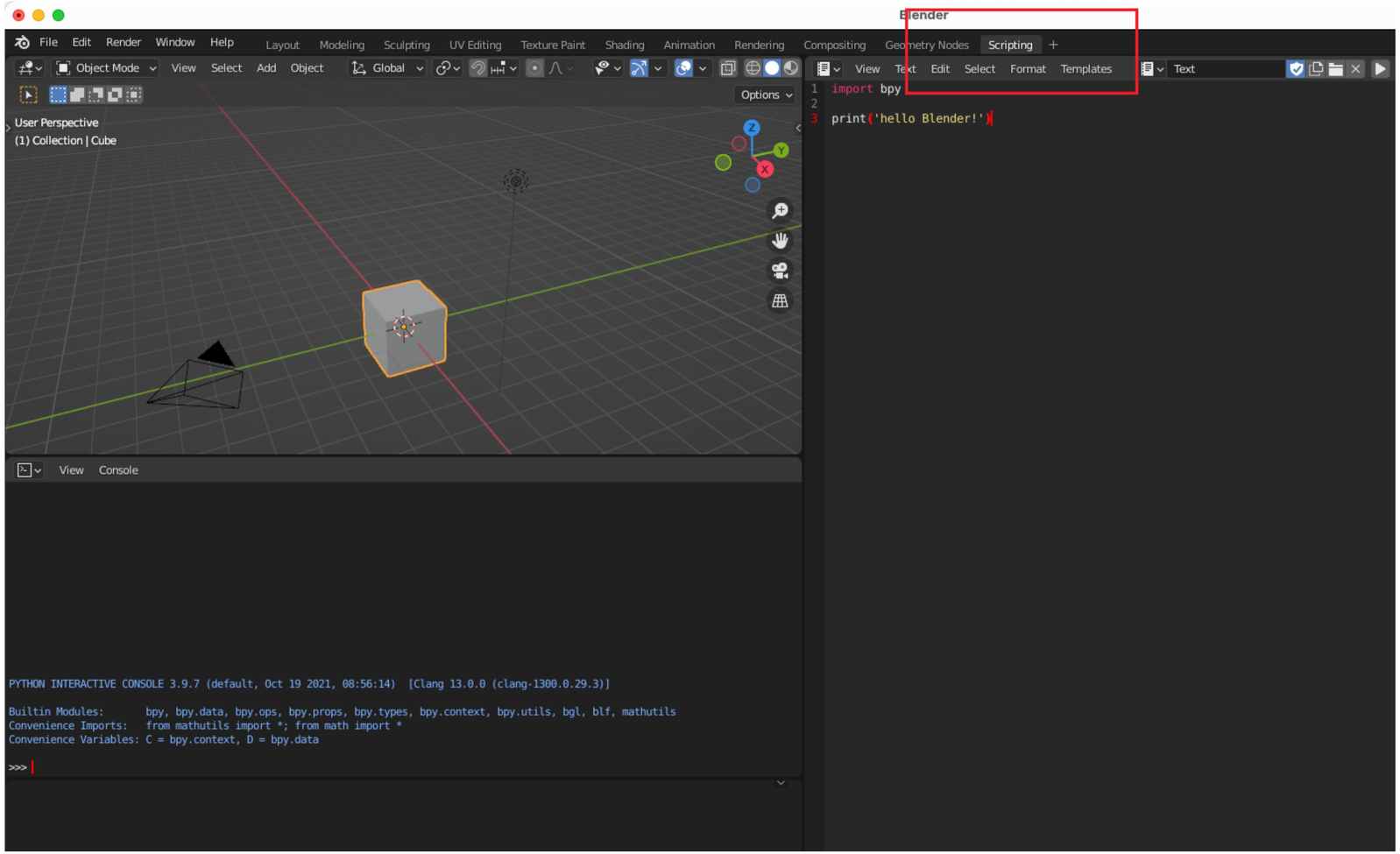
Rigging example on fish model

MoCap on Cat (https://youtu.be/UTB452OV8ak)

Tutorial 2 - Agenda

- Scripting in Animation
- BVH File Structure and Visualization
- Inverse Kinematics

Scripting in Blender



←Activate scripting window

Why we need scripting?

1. Reduce repeating stuff

2. Event simulation

3. More controllability and flexibility *

Operator - Add object

```
def add_cube(location=[0, 0, 0], size=1):
    bpy.ops.mesh.primitive_cube_add(size=size)
    c_obj = bpy.context.object
    # c_obj = bpy.data.objects[bpy.context.active_object.name] # Another selection way
    c_obj.location = location
    c_obj.rotation_euler = [np.radians(0), np.radians(0), np.radians(0)]
    return c_obj
```

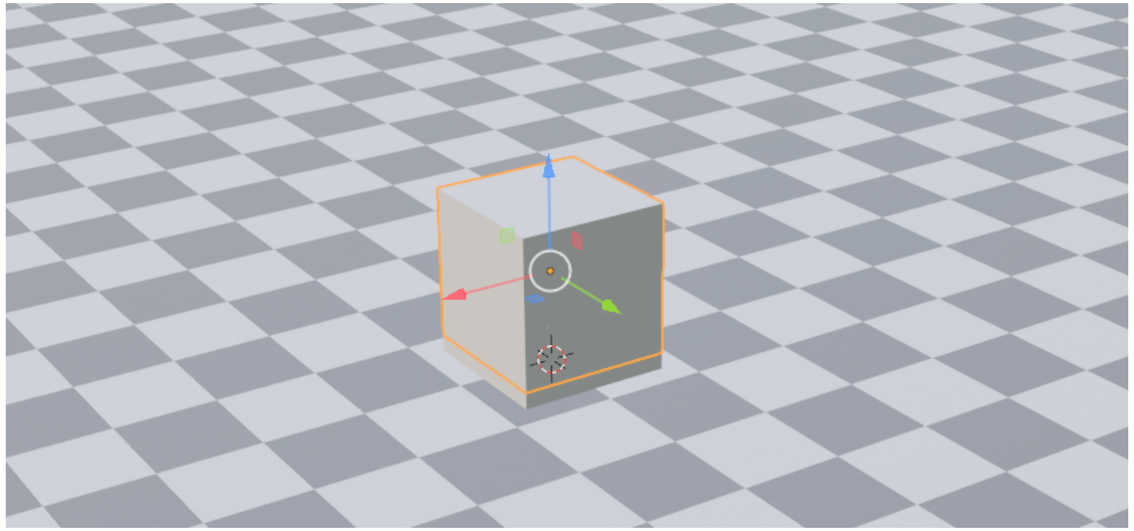
1. Add a new object
2. Make a pointer to this object
3. Custom Properties

Operator - Add materials

```
def add_floor(location=[0, 0, 0]):
    bpy.ops.mesh.primitive_plane_add(size=100, location=location)
    f_obj = bpy.context.object
    f_obj.name = 'floor'

    floor_mat = bpy.data.materials.new(name="floorMaterial")
    floor_mat.use_nodes = True
    bsdf = floor_mat.node_tree.nodes["Principled BSDF"]
    floor_text = floor_mat.node_tree.nodes.new("ShaderNodeTexChecker")
    floor_text.inputs[3].default_value = 150
    floor_mat.node_tree.links.new(bsdf.inputs['Base Color'], floor_text.outputs['Color'])

    f_obj.data.materials.append(floor_mat)
    return f_obj
```

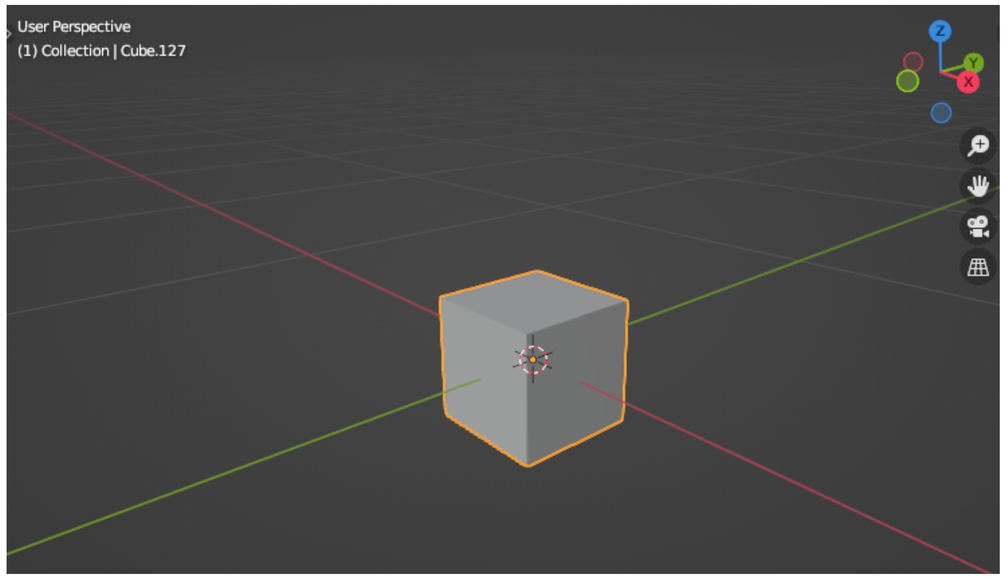


No need in Assignment

Example - Add objects in a loop

```python
def add_cube(location=[0, 0, 0], size=1):
    bpy.ops.mesh.primitive_cube_add(size=size)
    c_obj = bpy.context.object
    # c_obj = bpy.data.objects[bpy.context.active_object.name] # Another selection way
    c_obj.location = location
    c_obj.rotation_euler = [np.radians(0), np.radians(0), np.radians(0)]
    return c_obj
```

× 64

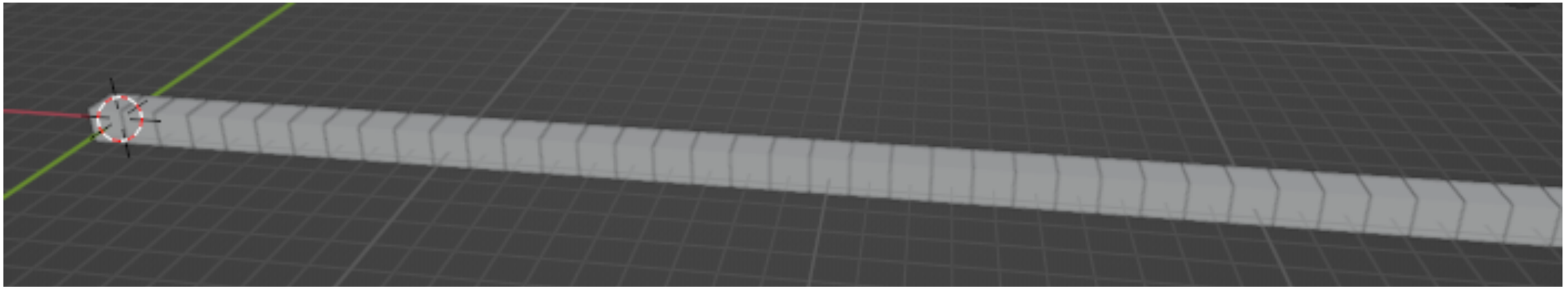


Lots of cube in a same place (looks stupid)

Example - Add objects in a loop

If we apply different location for each cube, like:

```
for i in range(64):
    add_cube(location=[i, 0, 0], size=1)
```

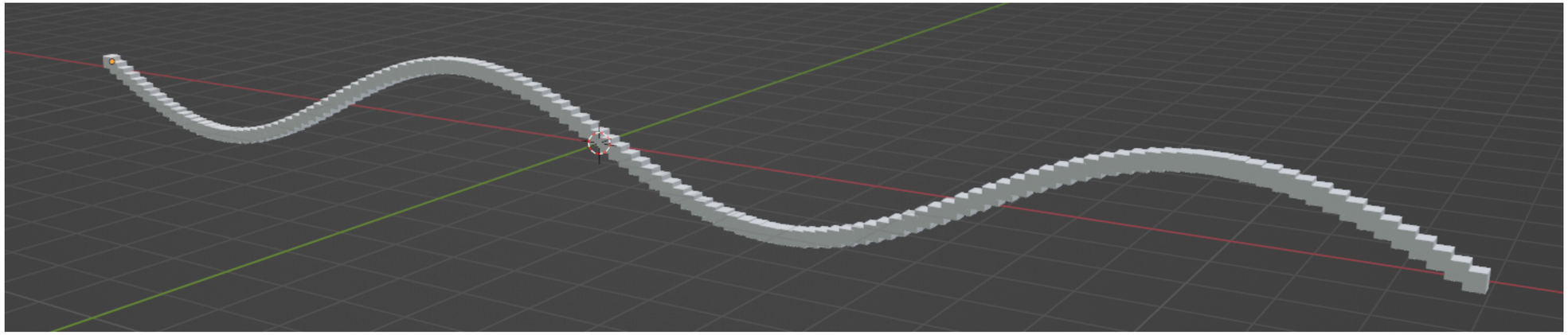


We can get a line of cubes

Example - Add objects in a loop

We can make the location to be more different:

```
for i in range(64):
    z = np.sin(I*0.1)
    add_cube(location=[i, 0, z], size=1)
```



A sin curve (basic point-based animation)

Example - Add objects in a loop with keyframe

We can make the location to be different in frames:

```
def toy_animation():
    N = 128
    frame = 500
    locations = np.zeros((N, frame, 3))
    toy_objects = add_joints([[0, 0, 0]]*N)
    bpy.context.scene.frame_end = frame
    for frame_idx in range(frame):
        for i in range(N):
            toy_objects[i].location = [i, 0, np.sin(i*0.1 + frame_idx*0.05)]
            toy_objects[i].keyframe_insert(data_path='location', frame=frame_idx)
```

Check: example.py

Use this pipeline for human skeleton

They are almost same!

- Lots of key joints
- Joints are always moving
- There are connection between joints

Example - Simple skeleton

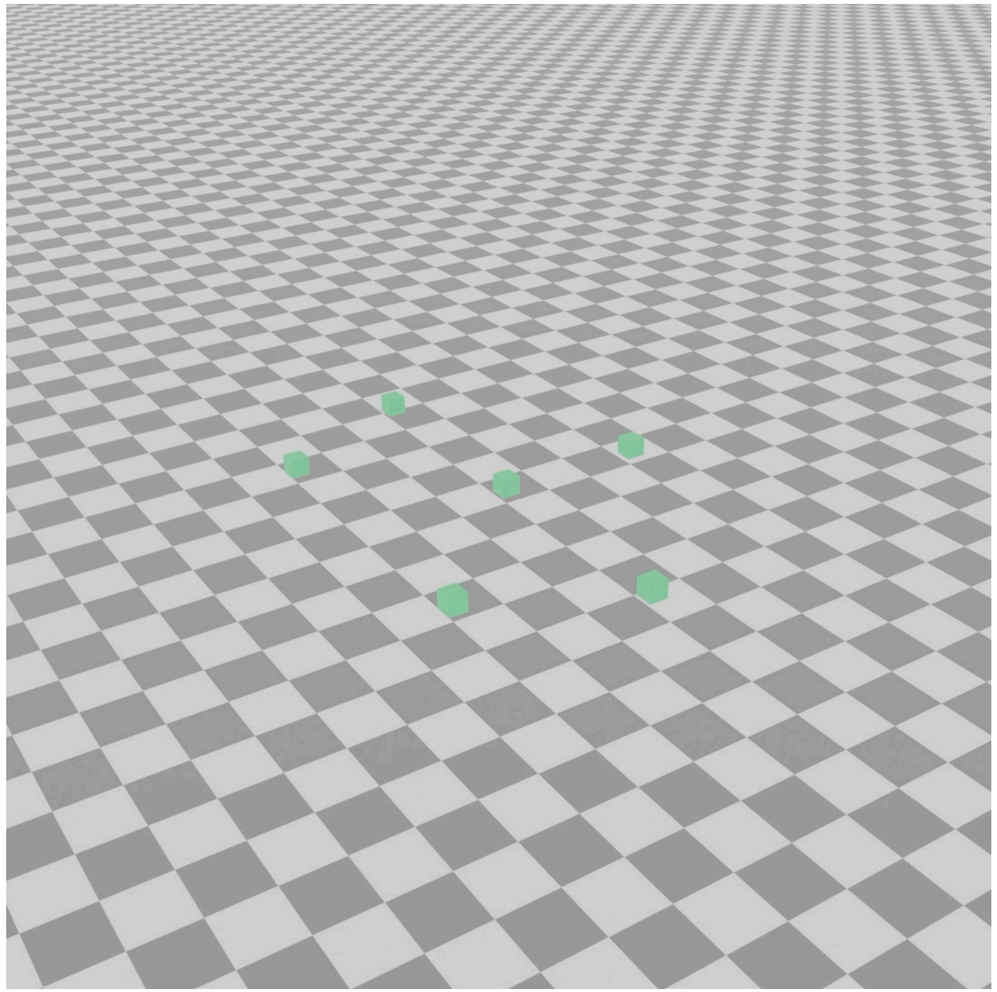
```
skeleton example
     J5           # (0, 4)
      |
J4 - J0 - J1      # (-3, 2), (0, 2), (3, 2)
     / \
   J3   J2        # (-2, 0), (2, 0)

joints = [[0, 2, 0], [3, 2, 0], [2, 0, 0], [-2, 0, 0], [-3, 2, 0], [0, 4, 0]]
parents = [-1, 0, 0, 0, 0, 0] # the index of parent joint, -1 means itself
```

```python
def add_skeleton(joints, parents):
    assert(len(joints) == len(parents))
    bpy.ops.object.armature_add(enter_editmode=False, align='WORLD', location=(0, 0, 0), scale=(1, 1, 1))
    b_obj = bpy.context.object
    bpy.ops.object.mode_set(mode='EDIT', toggle=False)
    edit_bones = b_obj.data.edit_bones
    edit_bones.remove(b_obj.data.edit_bones[0])
    for joint_idx, parent_idx in enumerate(parents):
        if parent_idx == -1:
            continue
        b1 = edit_bones.new('name%s' % joint_idx)
        b1.head = joints[parent_idx]
        b1.tail = joints[joint_idx]
    bpy.ops.object.mode_set(mode='OBJECT')
    print(b_obj.keys())
    return b_obj
```

```python
def set_animation(target, start_positions, end_positions, frames=20):
    assert(len(start_positions) == len(target))
    assert(len(start_positions) == len(end_positions))
    bpy.context.scene.frame_start = 0
    bpy.context.scene.frame_end = frames
    joint_number = len(target)
    start_positions, end_positions = np.array(start_positions, dtype=np.float32), np.array(end_positions, dtype=np.float32),
    base_increment = (end_positions - start_positions)/frames
    for frame_idx in range(frames):
        new_position = start_positions + base_increment*frame_idx
        for joint_idx in range(joint_number):
            target[joint_idx].location = new_position[joint_idx]
            target[joint_idx].keyframe_insert(data_path='location', frame=frame_idx)
```

Check: example.py

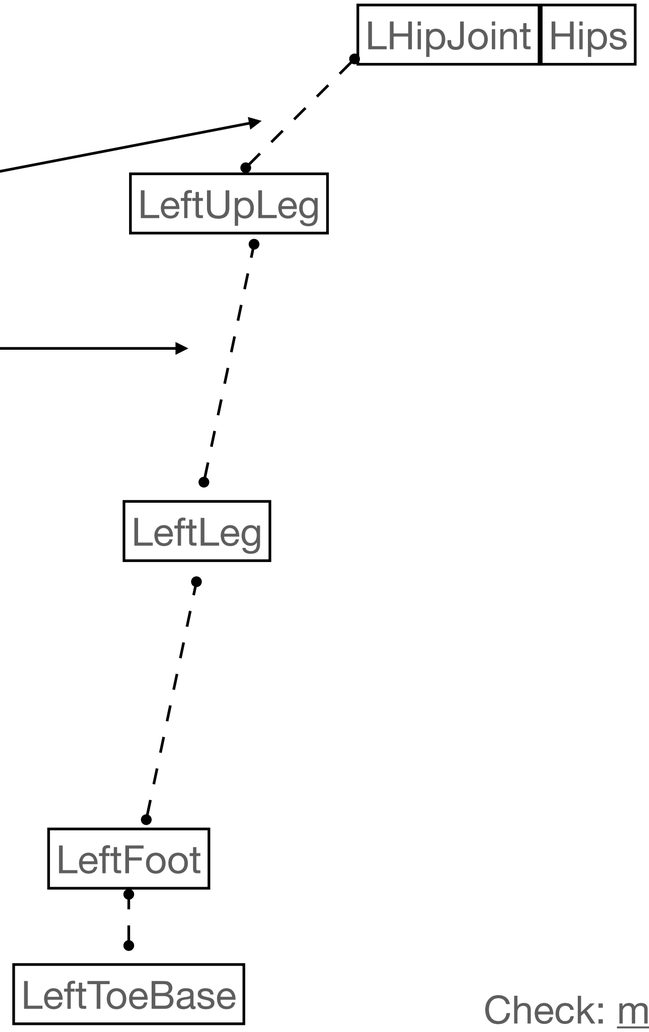


With our simple interpolation, there is no any constrains, so the motion is too bad.

Motion capture data will be useful to create animation, they are stored by **BVH** file

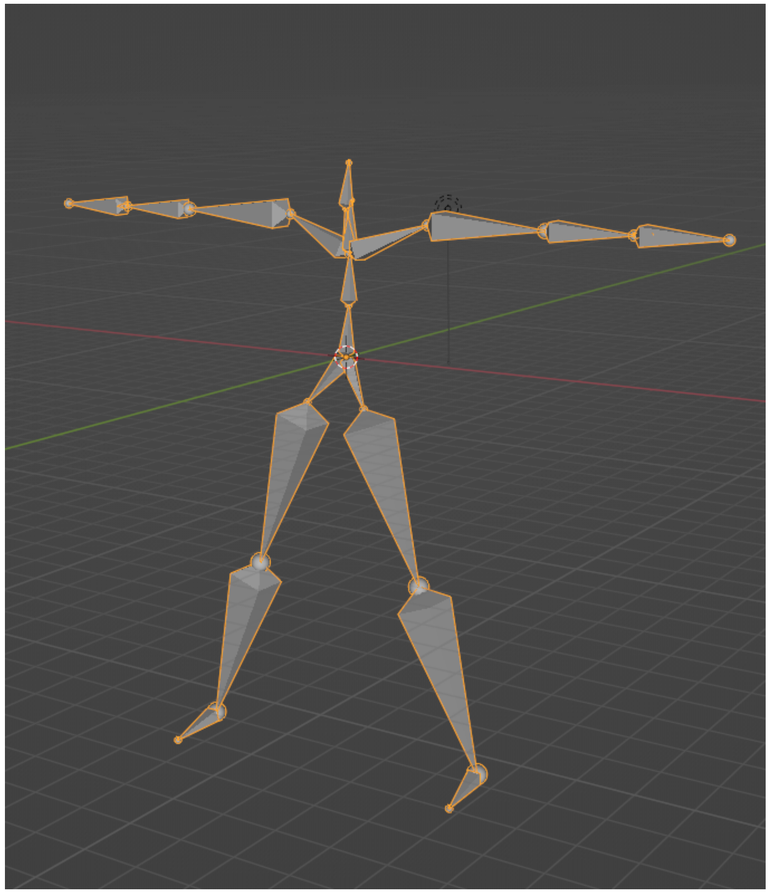
BVH file - example

```
HIERARCHY
ROOT Hips
{
        OFFSET 0.000000 0.000000 0.000000
        CHANNELS 6 Xposition Yposition Zposition Zrotation Yrotation Xrotation
        JOINT LHipJoint
        {
                OFFSET 0.000000 0.000000 0.000000
                CHANNELS 3 Zrotation Yrotation Xrotation
                JOINT LeftUpLeg
                {
                        OFFSET 1.363060 -1.794630 0.839290
                        CHANNELS 3 Zrotation Yrotation Xrotation
                        JOINT LeftLeg
                        {
                                OFFSET 2.448110 -6.726130 0.000000
                                CHANNELS 3 Zrotation Yrotation Xrotation
                                JOINT LeftFoot
                                {
                                        OFFSET 2.562200 -7.039590 0.000000
                                        CHANNELS 3 Zrotation Yrotation Xrotation
                                        JOINT LeftToeBase
                                        {
                                                OFFSET 0.157640 -0.433110 2.322550
                                                CHANNELS 3 Zrotation Yrotation Xrotation
                                                End Site
                                                {
                                                        OFFSET 0.000000 0.000000 0.000000
                                                }
                                        }
                                }
                        }
                }
        }
        JOINT RHipJoint
        {
                OFFSET 0.000000 0.000000 0.000000
                CHANNELS 3 Zrotation Yrotation Xrotation
                JOINT RightUpLeg
                {
                        OFFSET -1.305520 -1.794630 0.839290
```



Check: motion_basket.bvh

BVH file - example



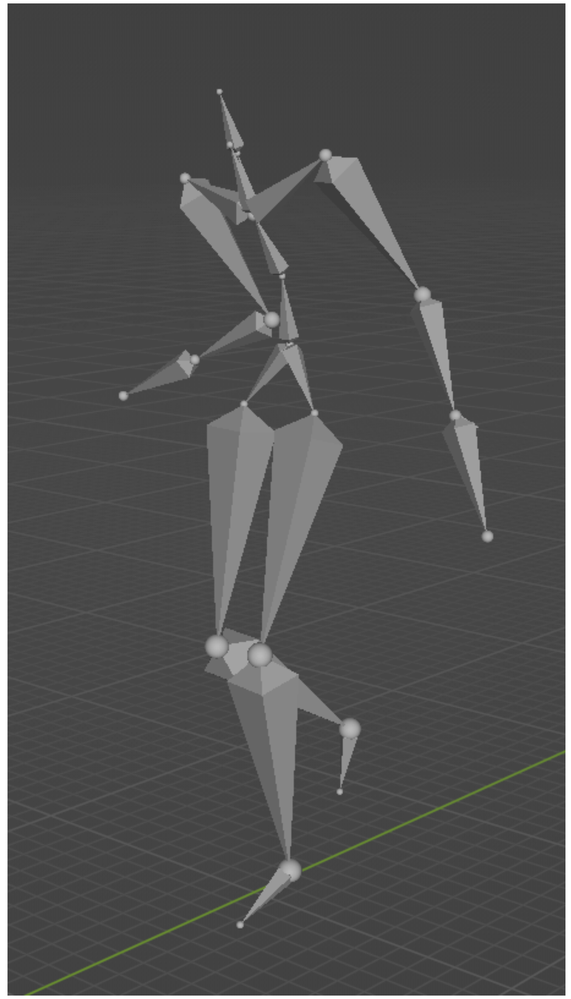
With the OFFSETs, it’s able to create a reset skeleton already

But it cannot move

BVH file - example

```
MOTION
Frames: 1146
Frame Time: 0.008333
-2.063732 18.011760 39.447933 179.127828 2.634550 -177.920153 1.206874 4.754803 4.104683 -23.563419 -11.889129 -18.828557 1.1858
-5.215836 4.517962 18.873920 7.311934 -12.506019 -11.950456 -16.033348 62.352482 2.786593 -18.765751 16.995573 -0.776200 -5.6251
-0.181365 -12.787612 -1.374329 -1.227442 -13.084216 -2.114110 0.138130 17.403246 -1.172200 0.092200 7.219200 -1.832485 1.104533
-0.555222 -6.507470 10.210135 49.935446 7.757400 -7.125000 0.000000 0.000000 -59.405500 74.075000 -44.346900 0.189166 -0.665463
0.207908 -36.148443 -9.663323 -0.878073 0.223500 7.125000 0.000000 0.000000 13.646200 -31.169300 4.732600
-2.070192 18.006791 39.330956 178.872573 1.870162 -178.041505 1.215018 4.778846 4.131116 -23.973632 -12.129823 -18.094388 1.1615
-5.208678 4.507877 18.846497 7.003885 -13.980773 -12.482011 -16.338560 64.272492 2.851874 -18.525429 15.834672 -0.626600 -5.0655
-0.295907 -12.798922 -1.165232 -1.205185 -13.831668 -2.205391 0.165837 17.659073 -1.229600 0.092900 7.421300 -1.823523 1.105351
-0.553590 -6.773480 10.030447 50.815154 7.713000 -7.125000 0.000000 0.000000 -62.397000 74.371400 -47.269400 0.208751 -0.650728
0.221464 -34.827075 -9.766024 -0.469250 0.160600 7.125000 0.000000 0.000000 13.528600 -30.787700 4.817200
-2.111436 18.010667 39.146001 178.712021 2.518295 -178.525697 1.201978 4.838431 4.197904 -24.494111 -11.742339 -16.851865 1.1540
-5.210053 4.512771 18.537530 7.616456 -14.618746 -12.903965 -16.568441 65.835281 2.761408 -17.782671 15.080130 -0.470500 -4.3993
-0.260308 -12.585790 -1.197374 -1.072545 -13.684112 -2.457030 0.417679 17.679626 -1.277200 0.197600 7.415200 -1.820126 1.086665
-0.557626 -7.064862 10.429341 51.289808 8.087700 -7.125000 0.000000 0.000000 -63.799200 74.705000 -48.872200 0.216774 -0.651851
0.213348 -33.597487 -10.203720 -0.316924 0.140900 7.125000 0.000000 0.000000 13.191500 -30.645100 4.737500
-2.130220 18.004704 39.013819 178.581251 2.288734 -178.515897 1.220191 4.844444 4.203782 -24.824199 -11.667838 -16.281330 1.1383
-5.198383 4.501691 18.479783 7.736503 -16.089027 -13.356942 -16.822103 67.453385 2.829419 -17.577012 13.901420 -0.380600 -3.9618
-0.331326 -12.970865 -1.353835 -1.044161 -13.069214 -2.736947 0.562816 17.683820 -1.314800 0.258500 7.312600 -1.840209 1.094920
-0.562172 -6.813896 10.627283 51.151272 8.232100 -7.125000 0.000000 0.000000 -63.221000 74.731300 -48.386300 0.228027 -0.655531
0.213939 -33.157573 -10.275641 -0.231756 0.127300 7.125000 0.000000 0.000000 13.131900 -30.564500 4.741100
-2.154570 18.002219 38.867524 178.482929 2.267110 -178.654585 1.221228 4.877126 4.248485 -25.126794 -11.487232 -15.552914 1.1294
-5.191192 4.508421 18.443492 7.935466 -17.303557 -13.742063 -17.022801 68.822310 2.979118 -17.597506 12.427723 -0.357400 -3.8410
-0.409895 -12.616380 -1.437290 -1.040795 -12.860014 -2.796393 0.562540 17.460794 -1.322500 0.248600 7.222200 -1.836681 1.054439
-0.560976 -6.422287 9.959104 51.050415 7.687100 -7.125000 0.000000 0.000000 -63.237300 74.438300 -48.084000 0.254412 -0.652732 0
0.222695 -33.150211 -10.278689 -0.151111 0.115200 7.125000 0.000000 0.000000 13.121100 -30.496000 4.759100
-2.159837 17.992579 38.760585 178.343237 1.459571 -178.682082 1.239828 4.889306 4.257075 -25.381015 -11.524257 -15.043042 1.1257
-5.184132 4.502501 18.532326 7.684861 -18.844429 -14.108236 -17.223905 70.114736 3.161199 -17.777201 10.866564 -0.369200 -3.9029
-0.565171 -12.855348 -1.297146 -0.974048 -12.543931 -3.087013 0.678504 17.376072 -1.415200 0.298600 7.147100 -1.842484 1.049521
-0.555024 -6.248881 11.878535 50.144795 9.109900 -7.125000 0.000000 0.000000 -58.981700 74.804400 -44.745900 0.281839 -0.645240
0.232628 -33.625174 -10.257540 -0.035880 0.097400 7.125000 0.000000 0.000000 13.119500 -30.395800 4.790800
-2.180807 17.985820 38.628801 178.190322 1.303193 -178.710378 1.250646 4.907288 4.273854 -25.716849 -11.321019 -14.525437 1.1326
-5.173345 4.507122 18.426008 7.899075 -20.199854 -14.382289 -17.390261 71.141430 3.280793 -17.683031 9.243297 -0.481700 -4.45090
-12.857105 -1.051684 -0.831465 -12.824773 -3.383156 0.943970 17.725193 -1.515600 0.418400 7.307000 -1.852672 1.045659 -0.977942
```

The second part, represents the values which are defined in first part



Check: motion_basket.bvh

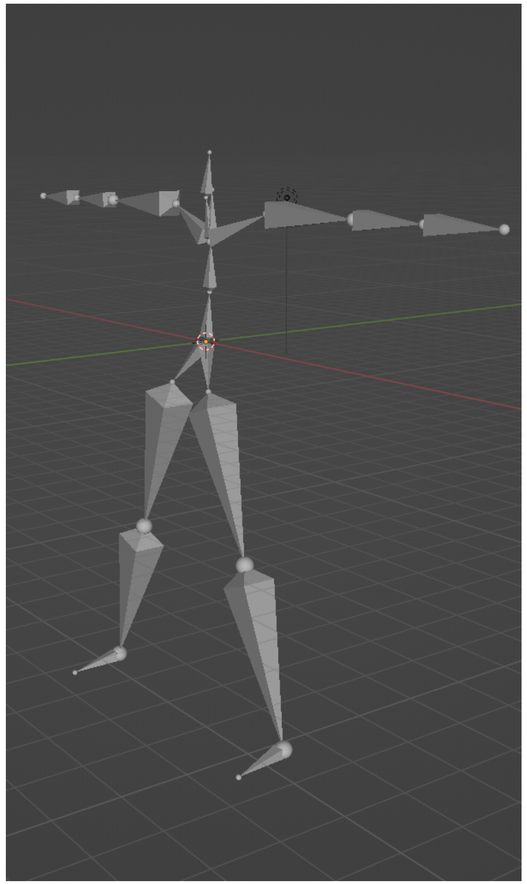
BVH file - example

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-2.063732 18.01176 39.447933 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-2.063732 18.01176 39.447933 179.127828 2.63455 -177.920153 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-2.063732 18.01176 39.447933 179.127828 2.63455 -177.920153 1.206874 4.754803 4.104683 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-2.063732 18.01176 39.447933 179.127828 2.63455 -177.920153 1.206874 4.754803 4.104683 -23.563419 -11.889129 -18.828557 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-2.063732 18.01176 39.447933 179.127828 2.63455 -177.920153 1.206874 4.754803 4.104683 -23.563419 -11.889129 -18.828557 1.185826 7.951782 27.500826 0 0 0 0
-2.063732 18.01176 39.447933 179.127828 2.63455 -177.920153 1.206874 4.754803 4.104683 -23.563419 -11.889129 -18.828557 1.185826 7.951782 27.500826 -0.6341
-2.063732 18.01176 39.447933 179.127828 2.63455 -177.920153 1.206874 4.754803 4.104683 -23.563419 -11.889129 -18.828557 1.185826 7.951782 27.500826 -0.6341
-2.063732 18.01176 39.447933 179.127828 2.63455 -177.920153 1.206874 4.754803 4.104683 -23.563419 -11.889129 -18.828557 1.185826 7.951782 27.500826 -0.6341
-2.063732 18.01176 39.447933 179.127828 2.63455 -177.920153 1.206874 4.754803 4.104683 -23.563419 -11.889129 -18.828557 1.185826 7.951782 27.500826 -0.6341
-2.063732 18.01176 39.447933 179.127828 2.63455 -177.920153 1.206874 4.754803 4.104683 -23.563419 -11.889129 -18.828557 1.185826 7.951782 27.500826 -0.6341
-2.063732 18.01176 39.447933 179.127828 2.63455 -177.920153 1.206874 4.754803 4.104683 -23.563419 -11.889129 -18.828557 1.185826 7.951782 27.500826 -0.6341
-2.063732 18.01176 39.447933 179.127828 2.63455 -177.920153 1.206874 4.754803 4.104683 -23.563419 -11.889129 -18.828557 1.185826 7.951782 27.500826 -0.6341
-2.063732 18.01176 39.447933 179.127828 2.63455 -177.920153 1.206874 4.754803 4.104683 -23.563419 -11.889129 -18.828557 1.185826 7.951782 27.500826 -0.6341
-2.063732 18.01176 39.447933 179.127828 2.63455 -177.920153 1.206874 4.754803 4.104683 -23.563419 -11.889129 -18.828557 1.185826 7.951782 27.500826 -0.6341
-2.063732 18.01176 39.447933 179.127828 2.63455 -177.920153 1.206874 4.754803 4.104683 -23.563419 -11.889129 -18.828557 1.185826 7.951782 27.500826 -0.6341
```

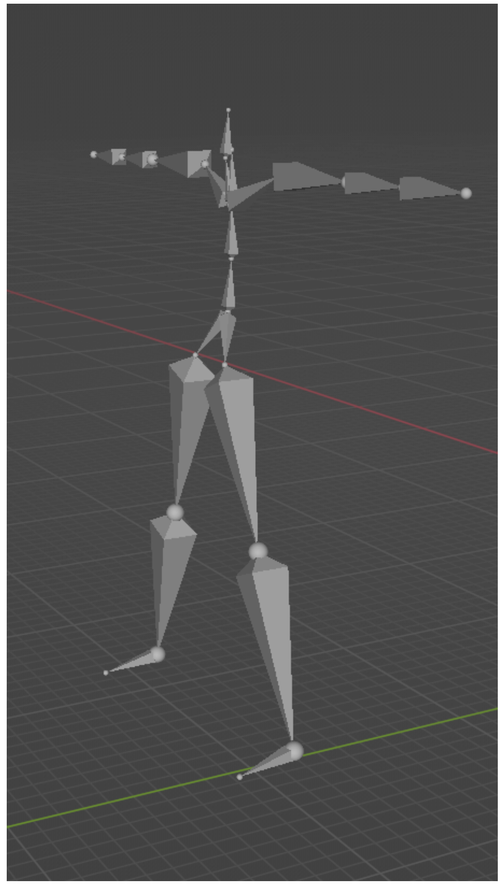
Let's check how three numbers impact the skeleton

Check: motion_toy.bvh

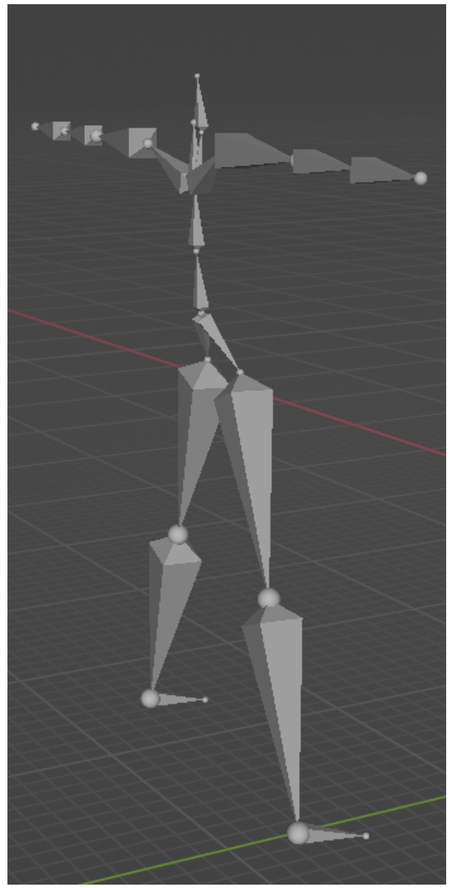
BVH file - example



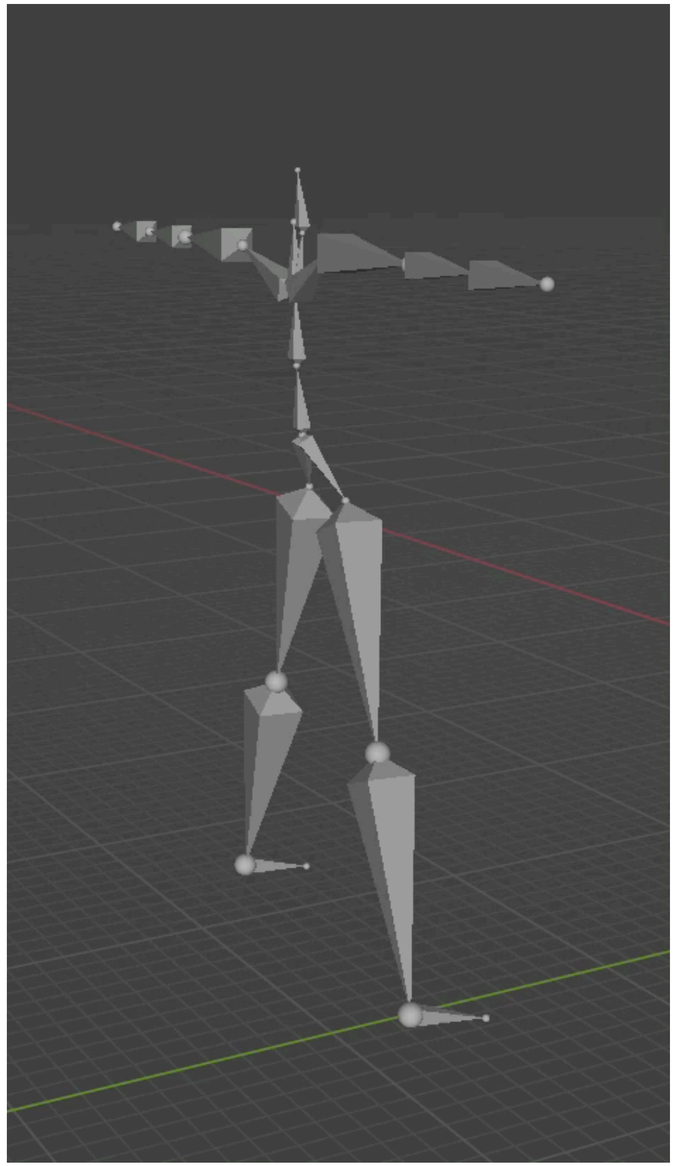
All zeros



Three value -
Global position



Next three value -
Global rotation



Then

Check: motion_toy.bvh

Assignment 1 Part 1 - BVH visualizer

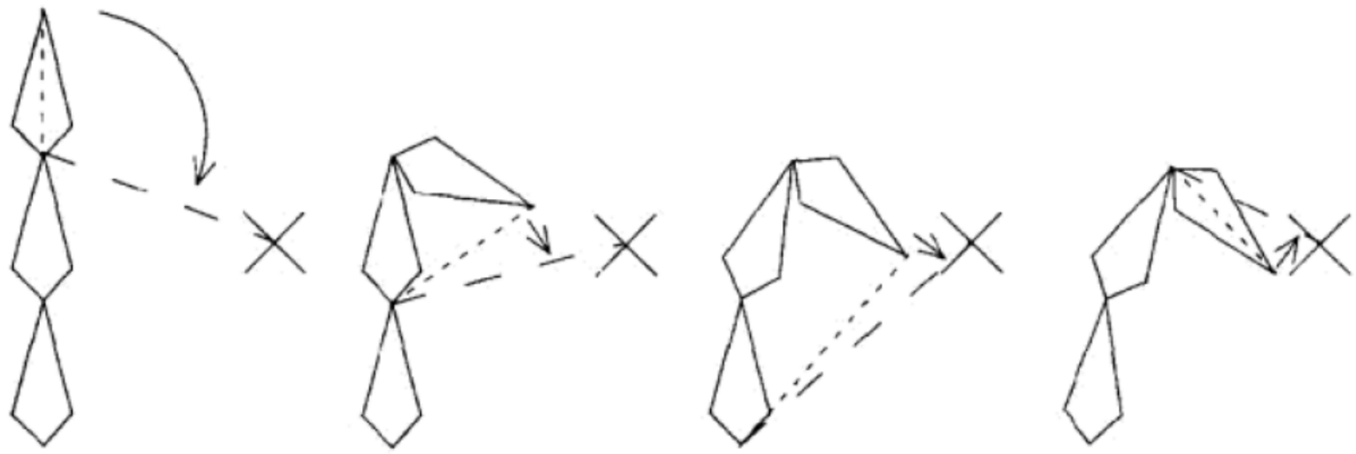
To-Do: Write a script in Blender with Python to visualize BVH file

- The text parsing has been done
- We prepared a version already, you can refer to that
- You are required to calculate the joint position in 3d space, and put a cube on the location of joint.
- Points: 1. Build rest skeleton with OFFSETs 2. Calculate joint position by given rotation 3. Key framing insertion

Next Part - Inverse Kinematics Solver

Environment: Unity
Language: C-sharp (c#)

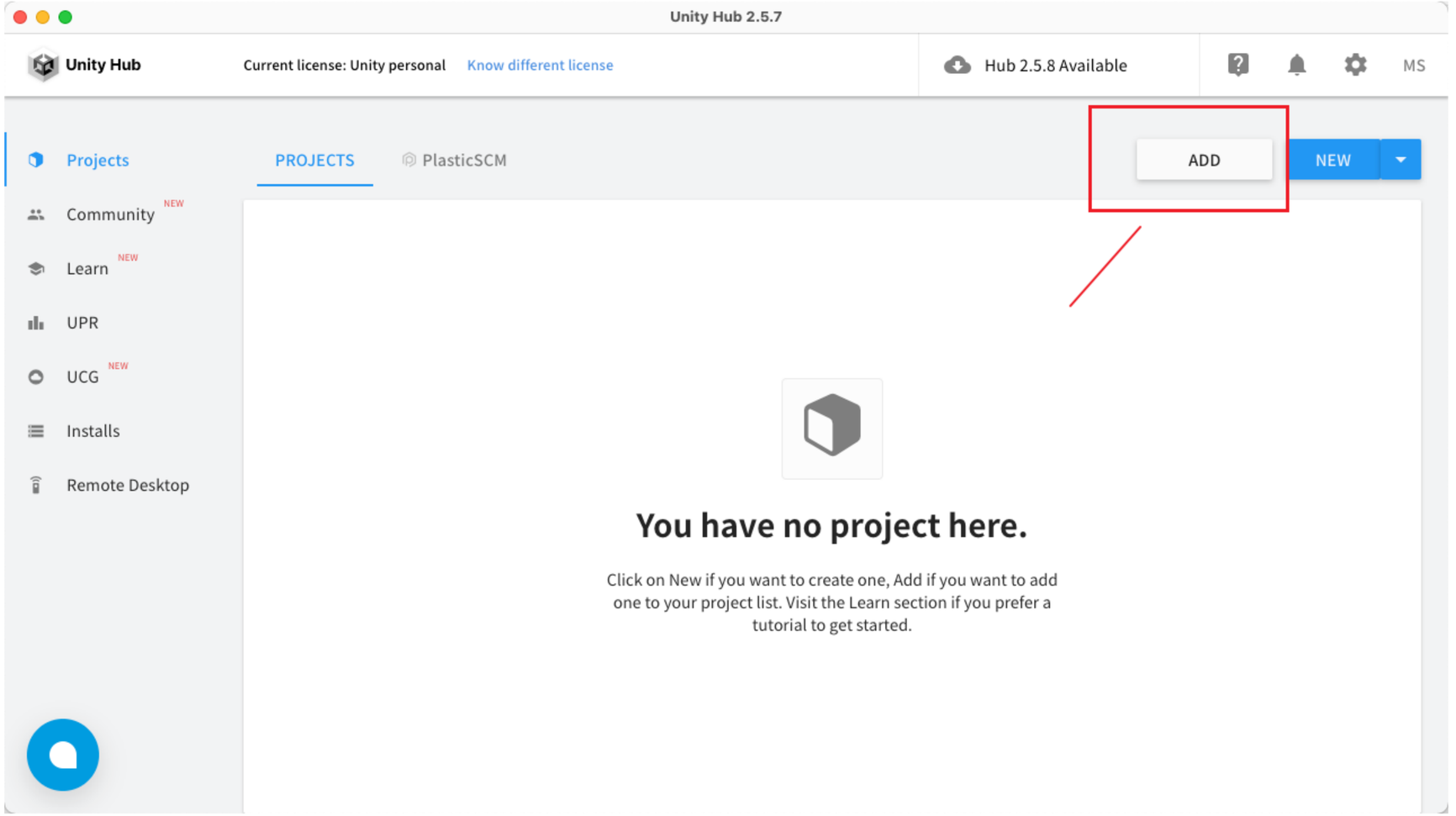
CCD IK (Position)



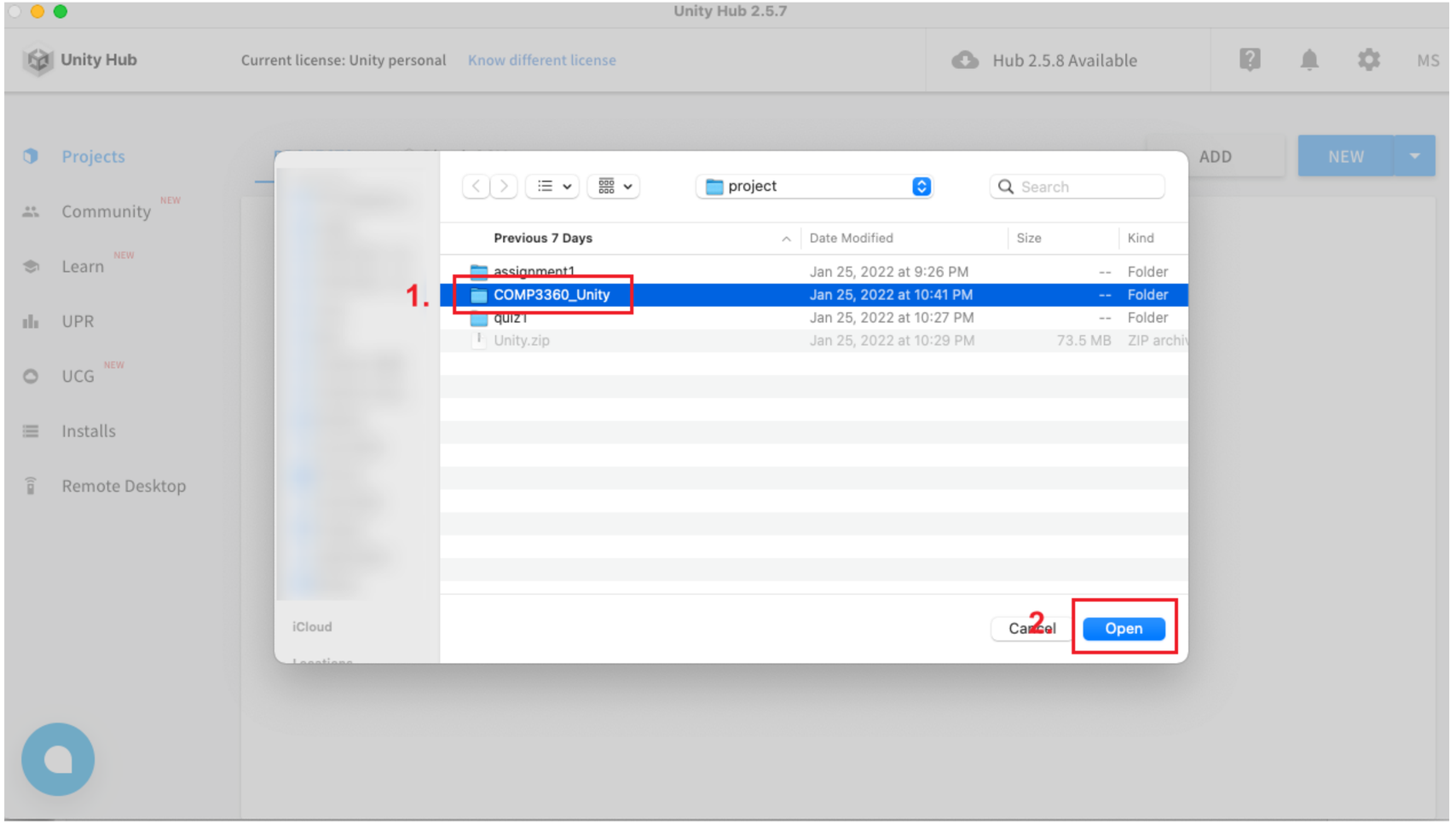
Loop N times:

1. Start from the deepest child bone - rotate the bone segment respect to its (origin) joint towards the target.
2. Rotate the upper-level parent bone respect to its (origin) joint so that the line between the joint and the end-effector can point to the target.
3. Repeat the same operation for the higher-level bones until the root bone is rotated

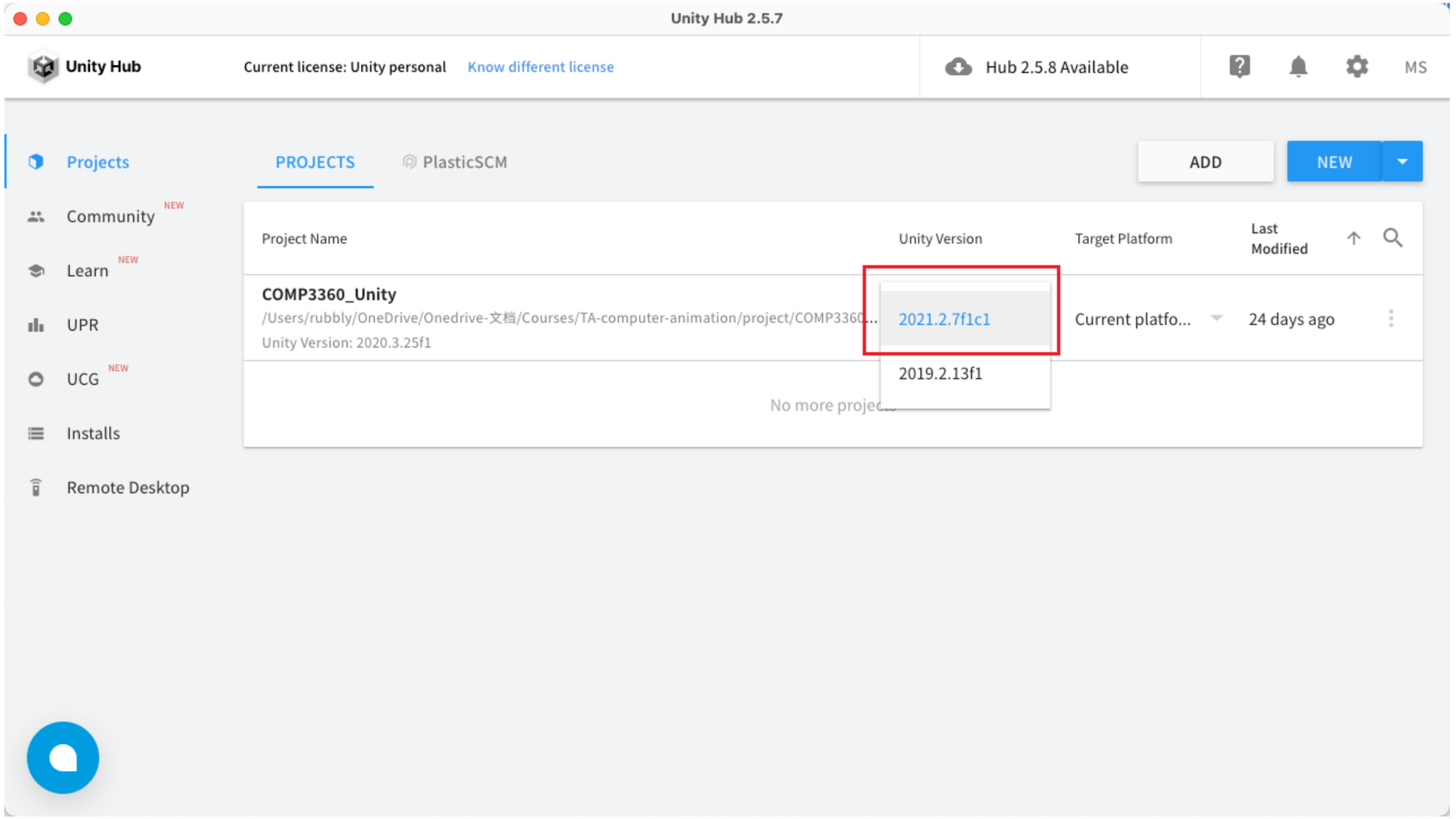
Unity Project Setting Up



Unity Project Setting Up

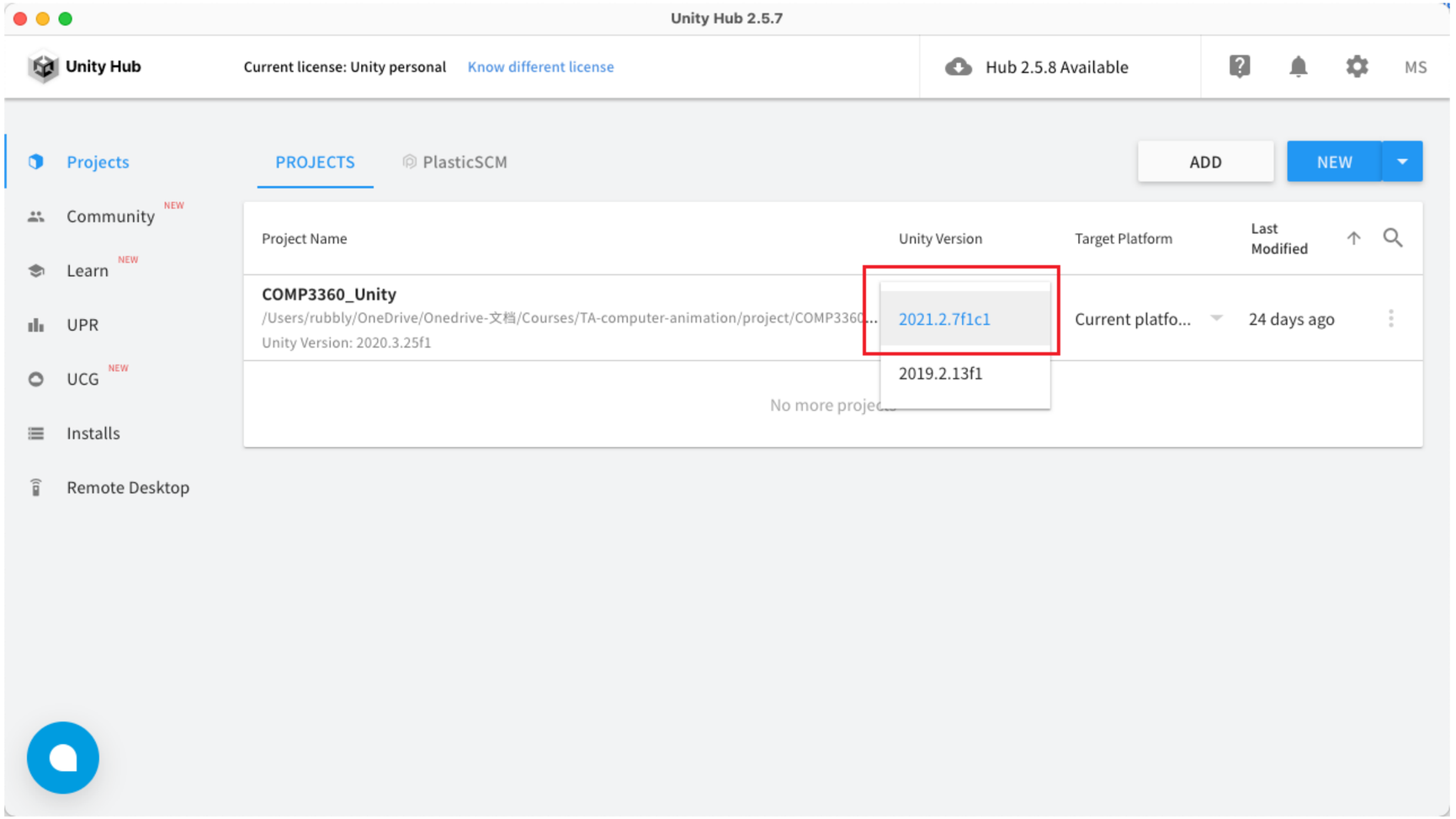


Unity Project Setting Up



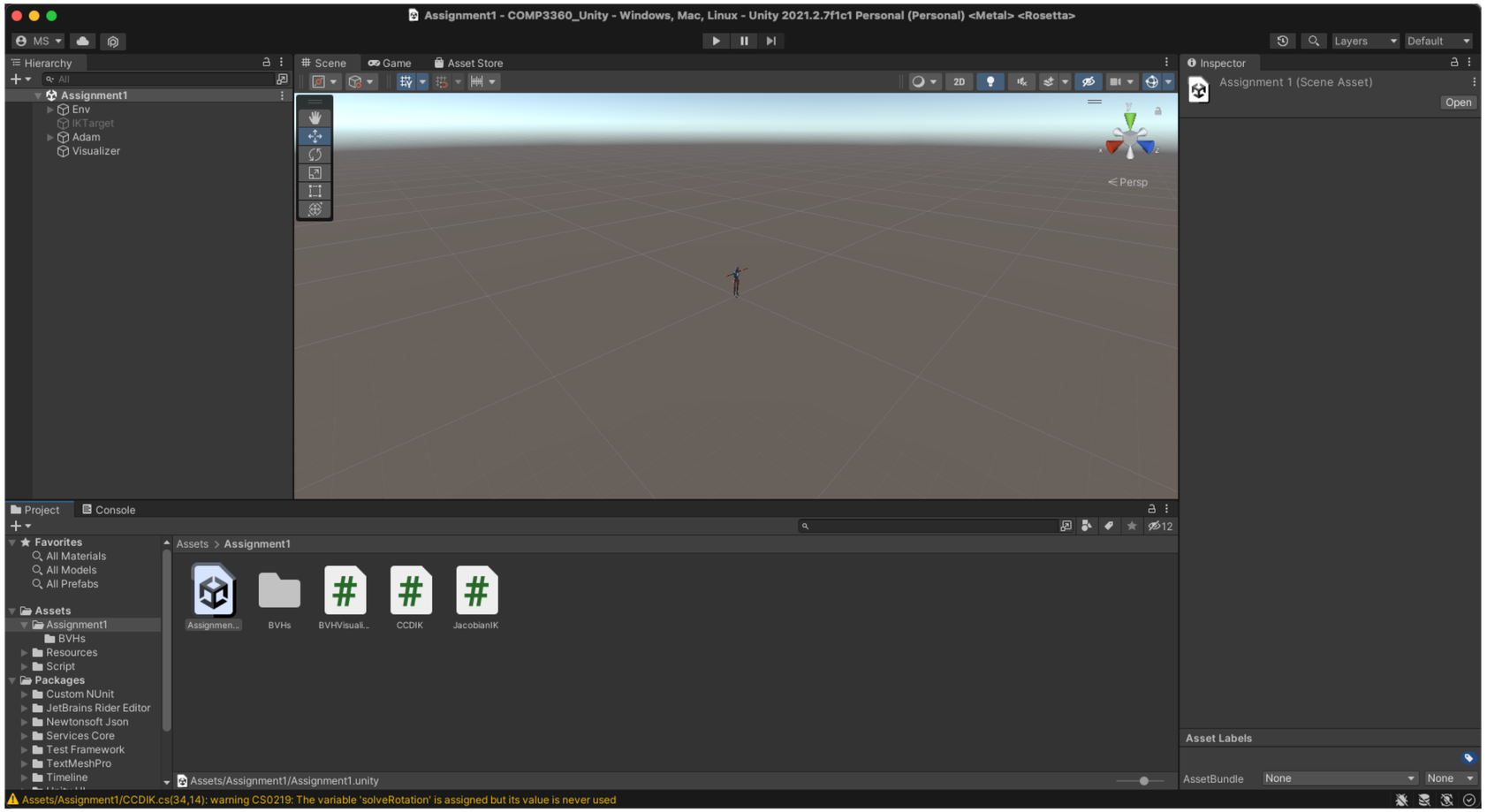
Select a Unity in your machine, original version - 2020.3.25f1, the the new version also works well

Unity Project Setting Up

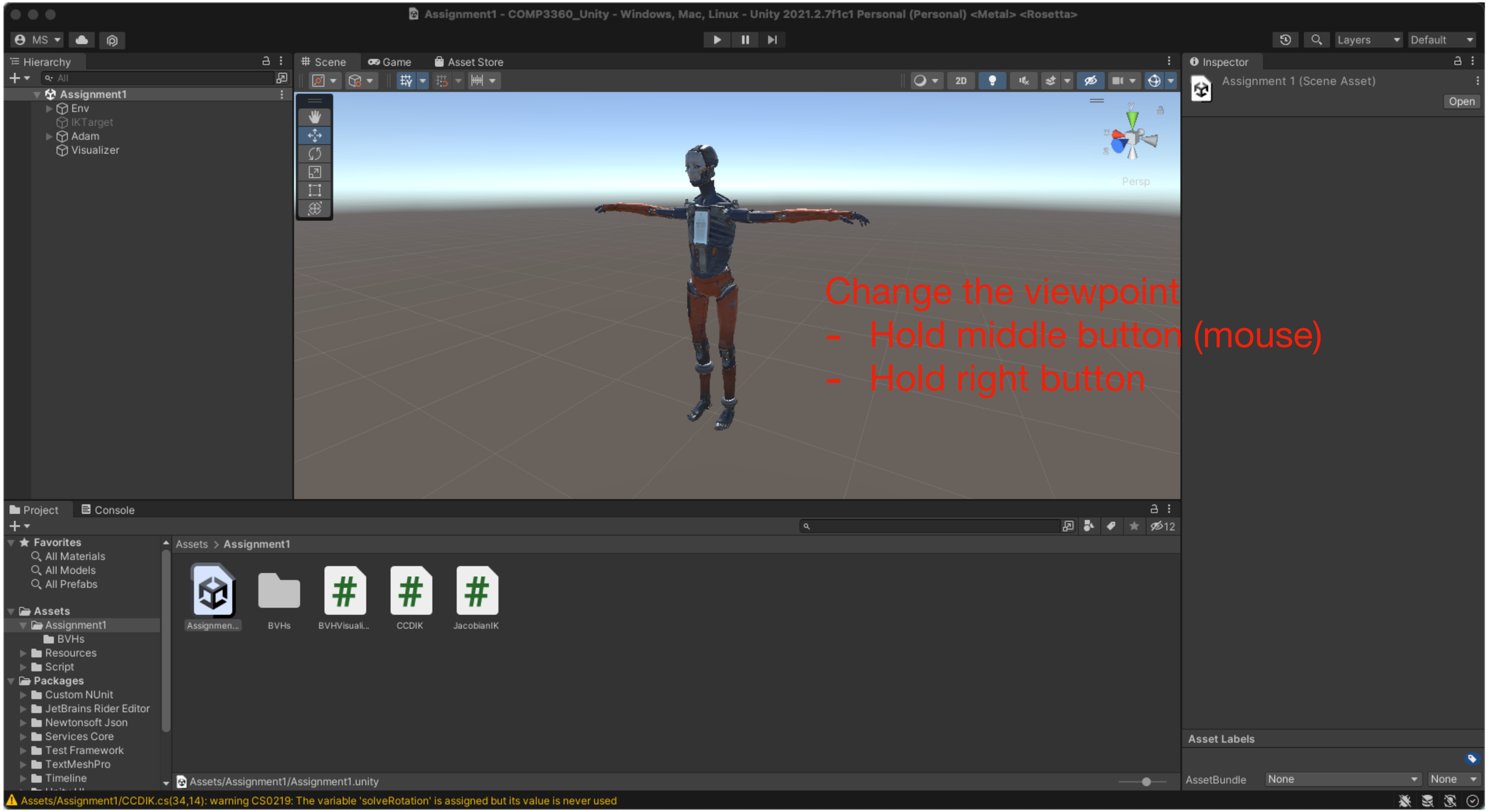


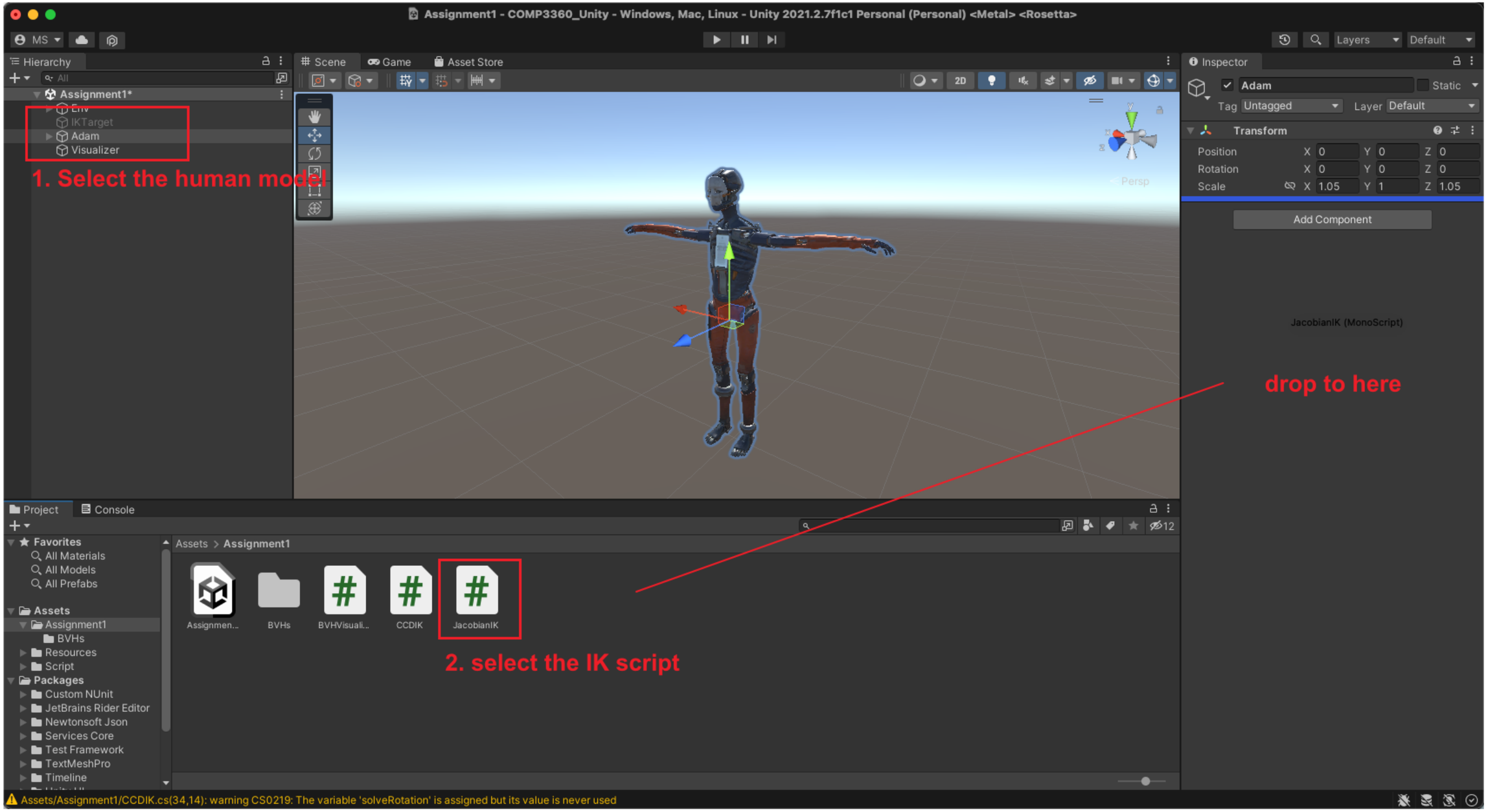
Select a Unity in your machine, original version - 2020.3.25f1, the the new version also works well so you can upgrade the project

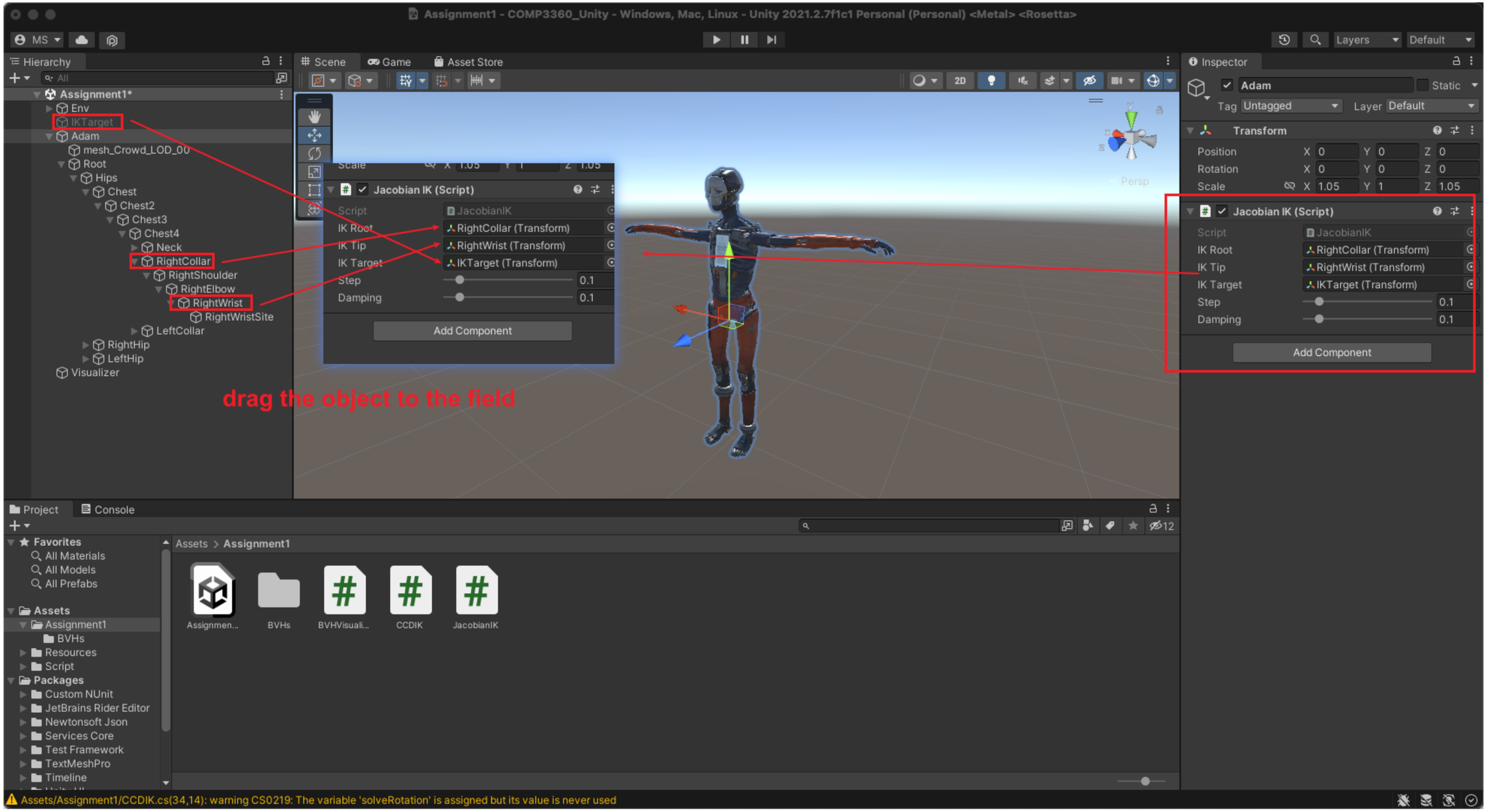
Unity Project Setting Up

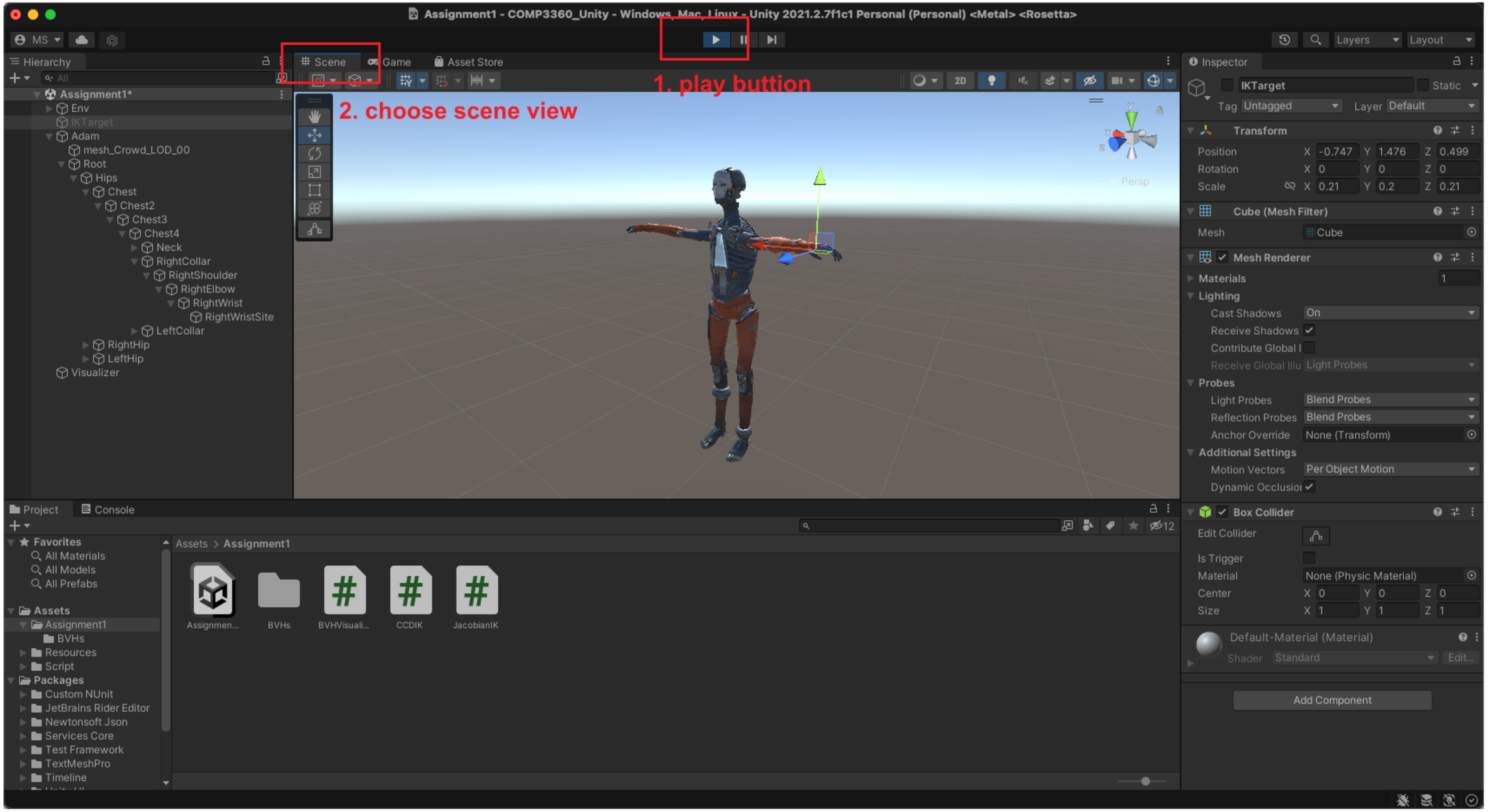


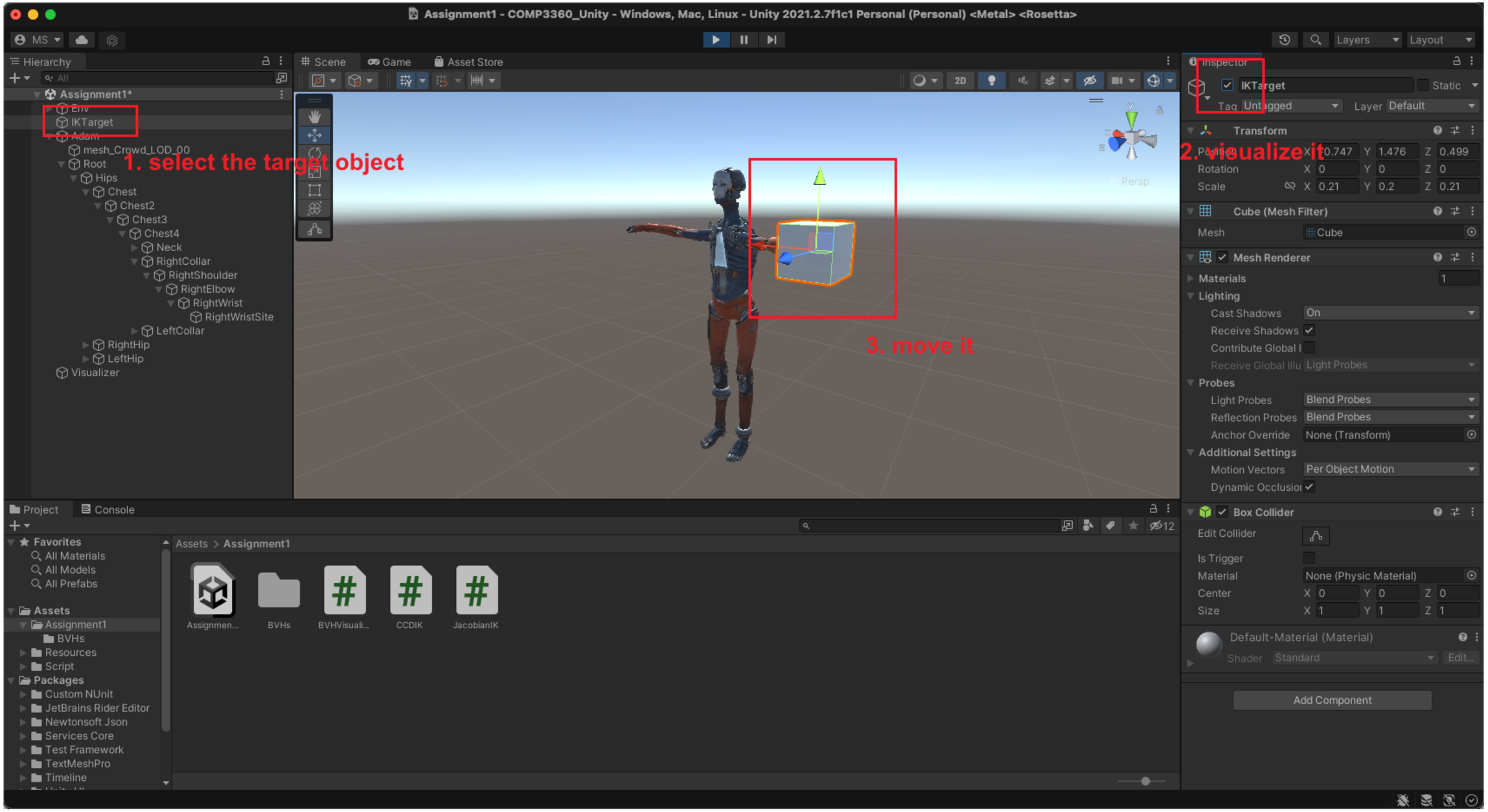
Select the scene file in Assets/Assignment1/Assignment1.unity


Assignment1 - COMP3360_Unity - Windows, Mac, Linux - Unity 2021.2.7f1c1 Personal (Personal) <Metal> <Rosetta>
Hierarchy
Assignment1
Env
IKTarget
Adam
Visualizer
Scene
Game
Asset Store
Persp
Change the viewpoint
- Hold middle button (mouse)
- Hold right button
Inspector
Assignment 1 (Scene Asset)
Open
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Assignment1
BVHs
Resources
Script
Packages
Custom NUnit
JetBrains Rider Editor
Newtonsoft Json
Services Core
Test Framework
TextMeshPro
Timeline
Assets > Assignment1
Assignmen...
BVHs
BVHVisuali...
CCDIK
JacobianIK
Assets/Assignment1/Assignment1.unity
Asset Labels
AssetBundle
None
None
Assets/Assignment1/CCDIK.cs(34,14): warning CS0219: The variable 'solveRotation' is assigned but its value is never used

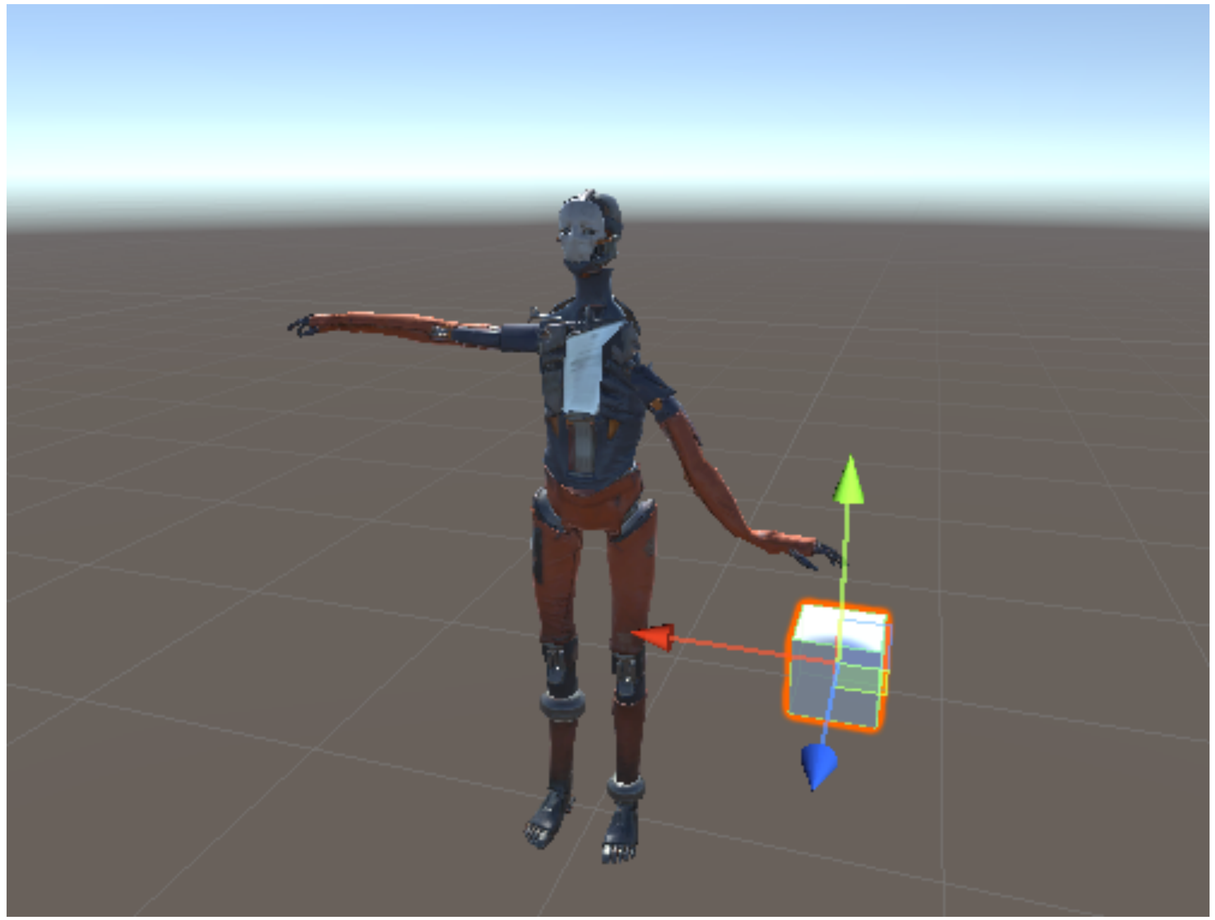

Assignment1 - COMP3360_Unity - Windows, Mac, Linux - Unity 2021.2.7f1c1 Personal (Personal) <Metal> <Rosetta>
Hierarchy
Assignment1*
IKTarget
Adam
Visualizer
1. Select the human model
Scene
Game
Asset Store
Persp
Inspector
Adam
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1.05 Y 1 Z 1.05
Add Component
JacobianIK (MonoScript)
drop to here
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Assignment1
BVHs
Resources
Script
Packages
Custom NUnit
JetBrains Rider Editor
Newtonsoft Json
Services Core
Test Framework
TextMeshPro
Timeline
Assets > Assignment1
Assignmen...
BVHs
BVHVisuali...
CCDIK
JacobianIK
2. select the IK script
Assets/Assignment1/CCDIK.cs(34,14): warning CS0219: The variable 'solveRotation' is assigned but its value is never used


Assignment1 - COMP3360_Unity - Windows, Mac, Linux - Unity 2021.2.7f1c1 Personal (Personal) <Metal> <Rosetta>
Hierarchy
Assignment1*
Env
IKTarget
Adam
mesh_Crowd_LOD_00
Root
Hips
Chest
Chest2
Chest3
Chest4
Neck
RightCollar
RightShoulder
RightElbow
RightWrist
RightWristSite
LeftCollar
RightHip
LeftHip
Visualizer
Scene
Game
Asset Store
Jacobian IK (Script)
Script
JacobianIK
IK Root
RightCollar (Transform)
IK Tip
RightWrist (Transform)
IK Target
IKTarget (Transform)
Step
0.1
Damping
0.1
Add Component
drag the object to the field
Inspector
Adam
Static
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
1.05
Persp
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Assignment1
BVHs
Resources
Script
Packages
Custom NUnit
JetBrains Rider Editor
Newtonsoft Json
Services Core
Test Framework
TextMeshPro
Timeline
Assignmen...
BVHs
BVHVisuali...
CCDIK
JacobianIK
Assets/Assignment1/CCDIK.cs(34,14): warning CS0219: The variable 'solveRotation' is assigned but its value is never used


Assignment1 - COMP3360_Unity - Windows, Mac, Linux - Unity 2021.2.7f1c1 Personal (Personal) <Metal> <Rosetta>
1. play buttion
2. choose scene view
Hierarchy
Scene
Game
Asset Store
Inspector
Assignment1*
Env
IKTarget
Adam
mesh_Crowd_LOD_00
Root
Hips
Chest
Chest2
Chest3
Chest4
Neck
RightCollar
RightShoulder
RightElbow
RightWrist
RightWristSite
LeftCollar
RightHip
LeftHip
Visualizer
Persp
Transform
Position
Rotation
Scale
Cube (Mesh Filter)
Mesh
Cube
Mesh Renderer
Materials
Lighting
Cast Shadows
Receive Shadows
Probes
Light Probes
Reflection Probes
Anchor Override
Additional Settings
Motion Vectors
Box Collider
Edit Collider
Is Trigger
Material
Center
Size
Default-Material (Material)
Shader
Standard
Add Component
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Assignment1
BVHs
Resources
Script
Packages
Custom NUnit
JetBrains Rider Editor
Newtonsoft Json
Services Core
Test Framework
TextMeshPro
Timeline
Assignmen...
BVHs
BVHVisuali...
CCDIK
JacobianIK


Assignment1 - COMP3360_Unity - Windows, Mac, Linux - Unity 2021.2.7f1c1 Personal (Personal) <Metal> <Rosetta>
Hierarchy
Assignment1*
Env
IKTarget
Adam
mesh_Crowd_LOD_00
Root
Hips
Chest
Chest2
Chest3
Chest4
Neck
RightCollar
RightShoulder
RightElbow
RightWrist
RightWristSite
LeftCollar
RightHip
LeftHip
Visualizer
1. select the target object
Scene
Game
Asset Store
2D
Persp
3. move it
2.visualize it
Inspector
IKTarget
Static
Tag Untagged
Layer Default
Transform
Position X 0.747 Y 1.476 Z 0.499
Rotation X 0 Y 0 Z 0
Scale X 0.21 Y 0.2 Z 0.21
Cube (Mesh Filter)
Mesh Cube
Mesh Renderer
Materials 1
Lighting
Cast Shadows On
Receive Shadows
Contribute Global I
Receive Global Illu Light Probes
Probes
Light Probes Blend Probes
Reflection Probes Blend Probes
Anchor Override None (Transform)
Additional Settings
Motion Vectors Per Object Motion
Dynamic Occlusio
Box Collider
Edit Collider
Is Trigger
Material None (Physic Material)
Center X 0 Y 0 Z 0
Size X 1 Y 1 Z 1
Default-Material (Material)
Shader Standard
Edit...
Add Component
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Assignment1
BVHs
Resources
Script
Packages
Custom NUnit
JetBrains Rider Editor
Newtonsoft Json
Services Core
Test Framework
TextMeshPro
Timeline
Assets > Assignment1
Assignmen...
BVHs
BVHVisuali...
CCDIK
JacobianIK

Unity Project - IK solver



- Try the different IK root and IK Tip setting

- Read the code of Jacobian IK

- Write you own CCD IK

Assignment 1 Part 2 - CCD IK Solver

To-Do: Fill the function in *CCDIK.cs*

- Only the positional solution is required
- Points: 1. Solve Tip Position 2. Heuristic IK Weight Computation
- The reference source should be mentioned if you use any open-source code

Assignment 1 Part 2 - CCD IK Solver

Tips: Unity Functions：

- Get tip/end effector position/rotation :

  tip.position/tip.rotation
- Get position/rotation of the jth joint on the hierarchical chain :

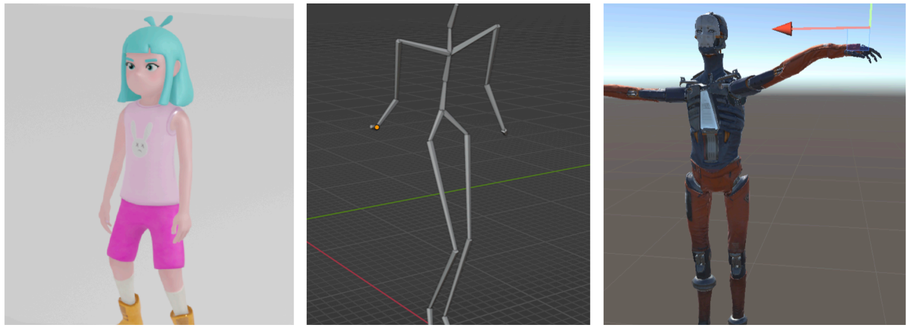
  chain[j].position/chain[j].rotation
- Get target position/rotation

  target.position/target.rotation
- Creates a rotation(Quaternion) which rotates from vectorA to vectorB doc

  Quaternion.FromToRotation(Vector3 vectorA, Vector3 vectorB);
- Rotate the jth joint on the chain by a Rotation R(in Quaternion)

  *chain[j].rotation = R * chain[j].rotation rotations*

Assignment 1 - Summary



1. Key-framing Animation and Rendered video (15%)
2. BVH visualizer code implement, ***bvh_visualizer.py*** (45%)
3. Positional CCD IK implement, ***CCDIK.cs*** (30%)
4. Report file, ***uid_name_1.pdf*** (10%)

Due to: 13th. Feb, by Moodle