UNITY INTRODUCTION

Corso Realtà Virtuale 2024/20245

susanna.brambilla@unimi.it

WITH UNITY V2022.3.5

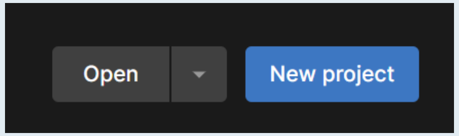
UNITY BASICS

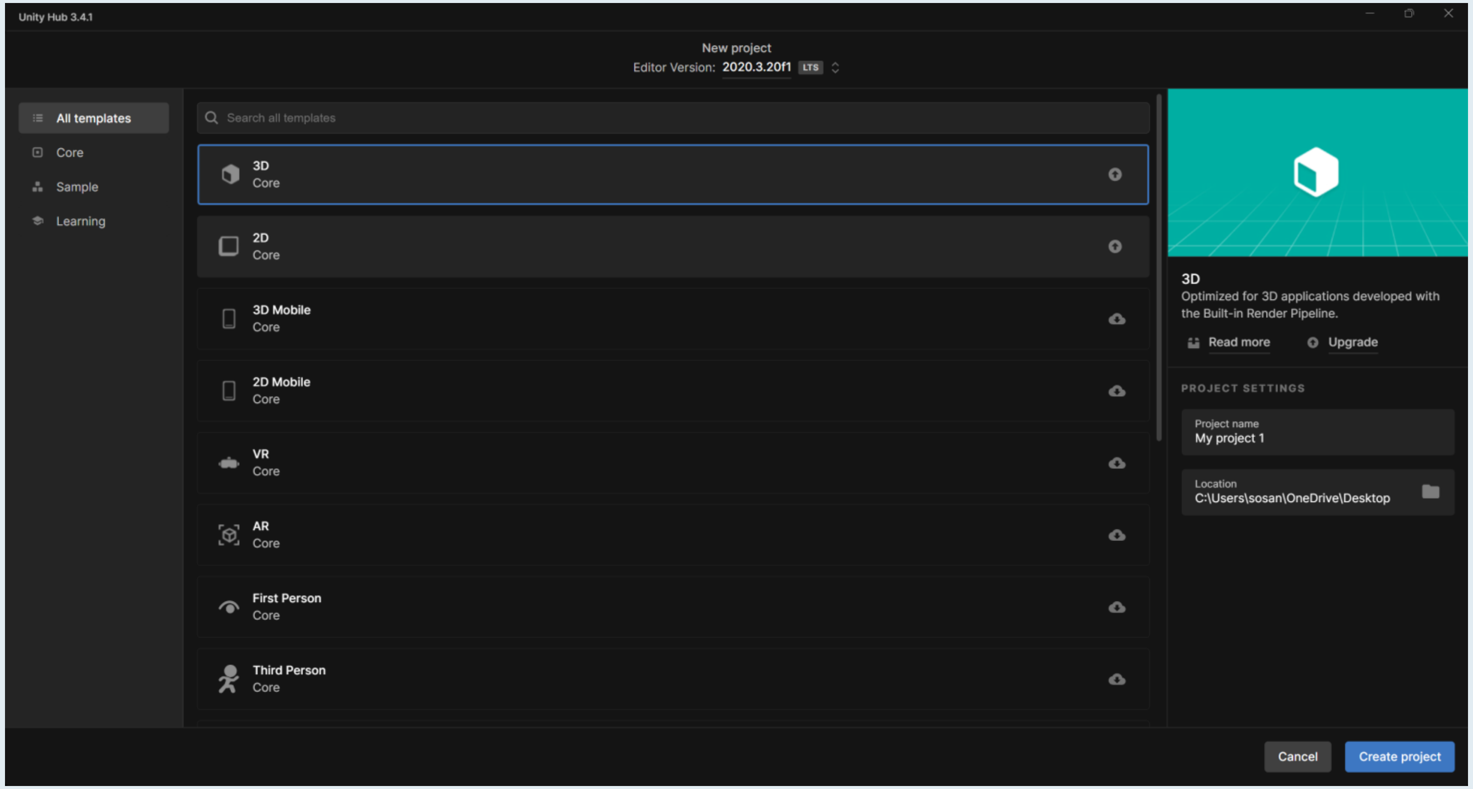
OPEN UNITY

Unity 3D download: https://store.unity.com/

Official Unity 3D tutorials: https://learn.unity.com/

Unity Hub:




Unity Hub 3.4.1

New project

Editor Version: 2020.3.20f1 LTS

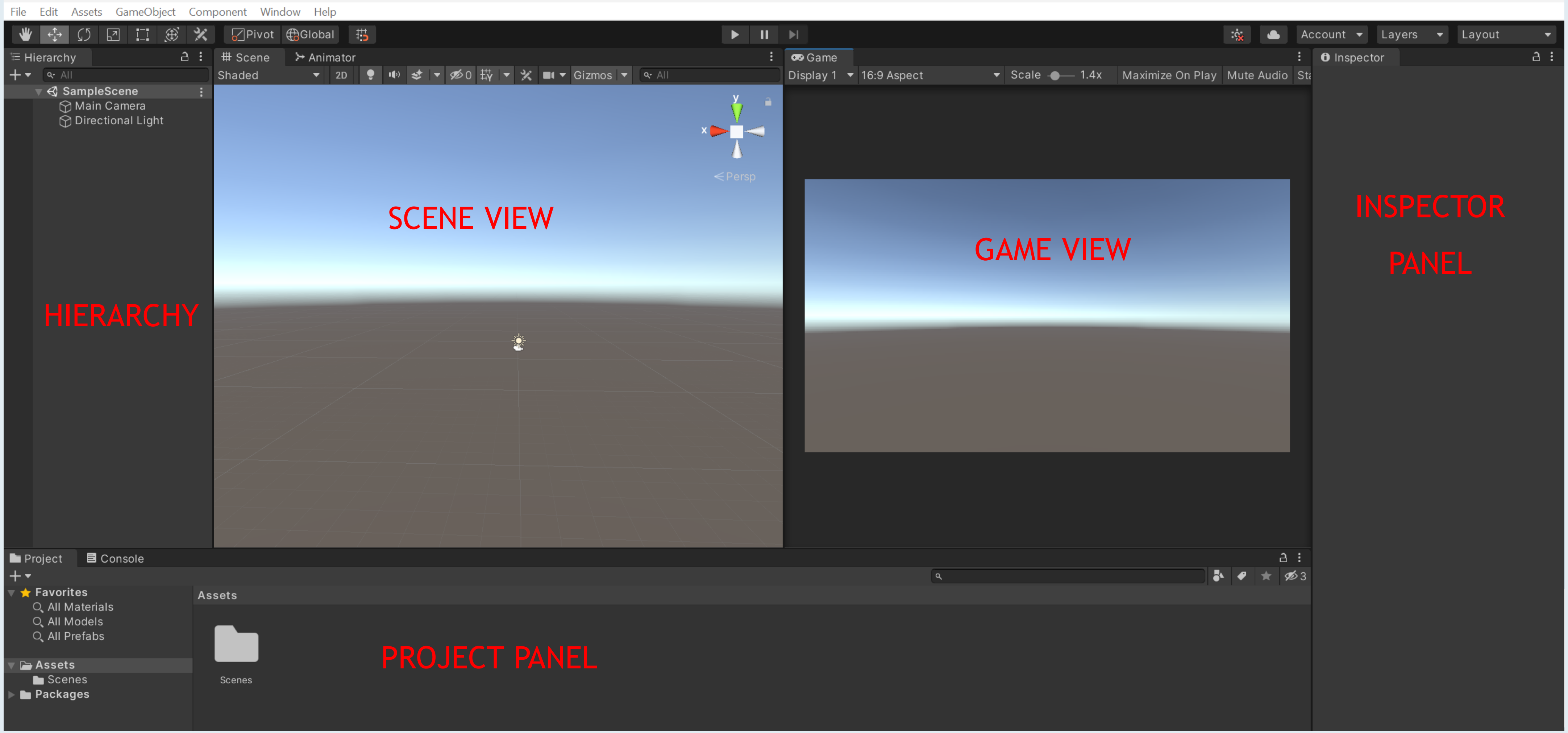
3D
Optimized for 3D applications developed with the Built-in Render Pipeline.

PROJECT SETTINGS

Project name
My project 1

Location
C:\Users\sosan\OneDrive\Desktop

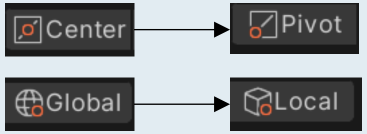
INTERFACE OVERVIEW



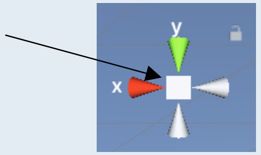
SCENE VIEW

- Pan: left mouse click
- Look around in place: right mouse click
- Orbit around a looking point: [alt]+left mouse click
- Zoom: [alt] + right mouse click

- You can decide to apply transformation using as center:
  - The object center or the object pivot point
  - Using as reference global or local axes

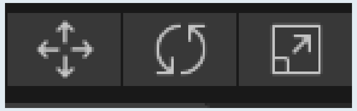


- You can change from perspective 3D to orthogonal view by clicking the center of the axis icon:



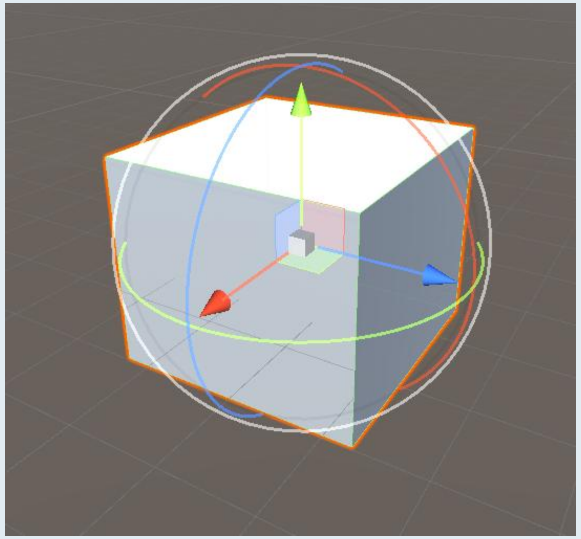
SCENE VIEW – MOVE, ROTATE, SCALE

You can move, scale, or rotate an object in the Scene with the top left tools:



Or:

- MOVE with [W]
- ROTATE with [E]
- SCALE with [R]
- Move, scale and rotate at the same time with [Y]



- To focus on object: double left button click in Hierarchy or select object and [F]

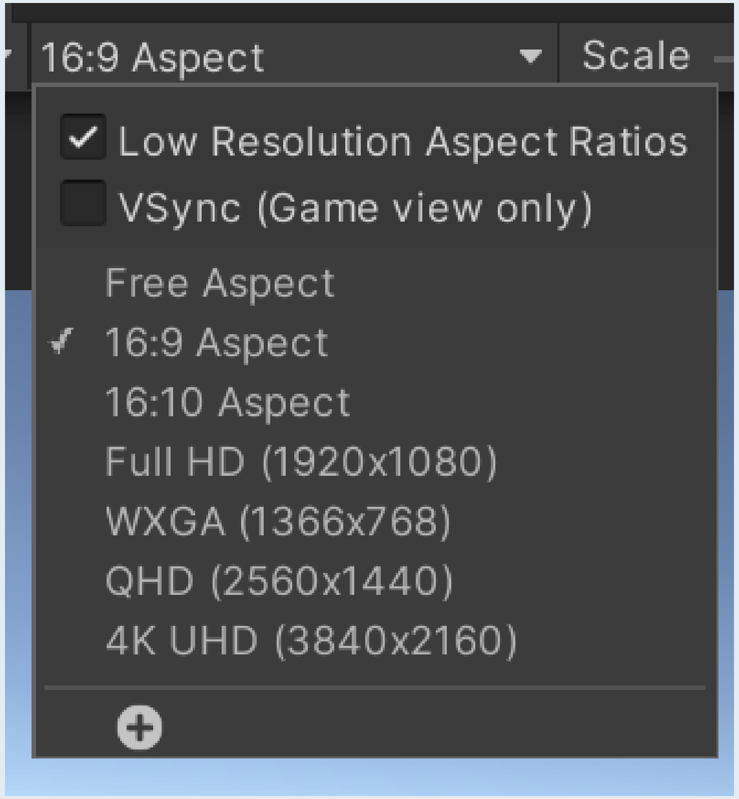
GAME VIEW

It shows what the game will look like from the main camera

The Game view is active when the Play button ▶ is pressed

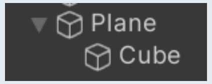
- In play mode every change is not saved
- It is possible to pause or step forward frame by frame ⏸ ⏭

- You can change aspect ratio and resolution (to test deploy platform)

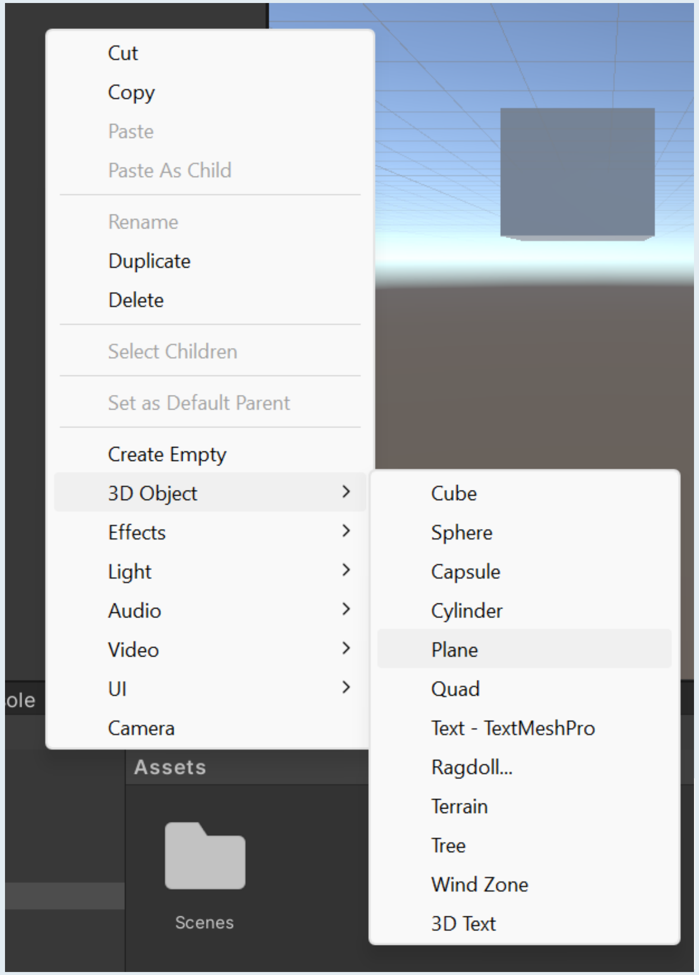


HIERARCHY PANEL

Here, all the objects in the current scene are displayed

- You can create a new 3D Object: click the right mouse button and select the object you want to create
- You can parent an object to another object dragging the child object onto the parent object: 

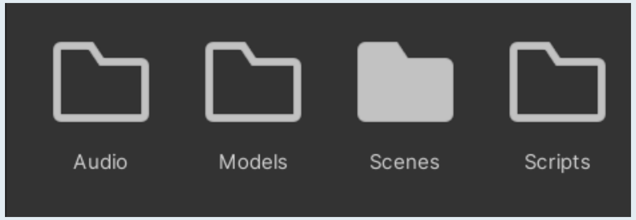
- You can change also the objects’ order by dragging them
- You can search an object using the search bar; search could be filtered by type: t:Light, t:AudioSource …



PROJECT PANEL

It shows all the contents in Asset folder

- Used to organize all the assets
  - You can create new folder with right mouse button Create > Folder



- New object could be imported:
  - into the project folder
  - by drag and drop
- New object could be created by clicking right mouse button
- You can search for objects using the search bar

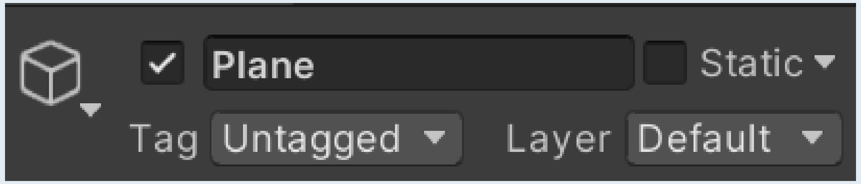
INSPECTOR

The Inspector panel shows the characteristics of the selected objects

Here, you can find all components attached to the object and edit object’s properties

You can set up tag and layer for each object:

- **Tags** are markers that you can use to identify object in your Project and find them by code
- **Layers** are used to create groups of objects that share particular characteristics

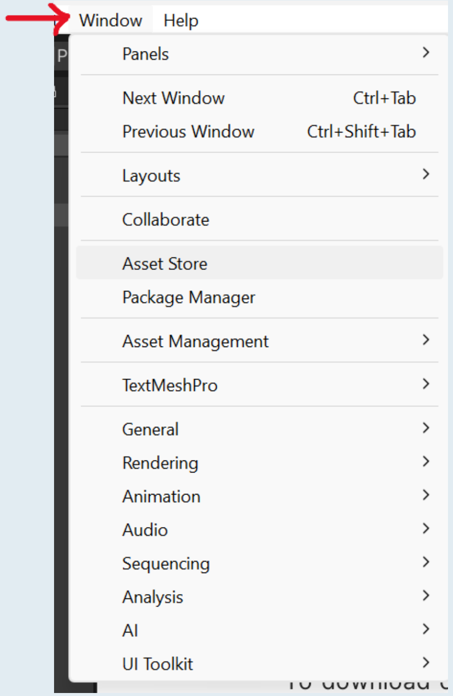


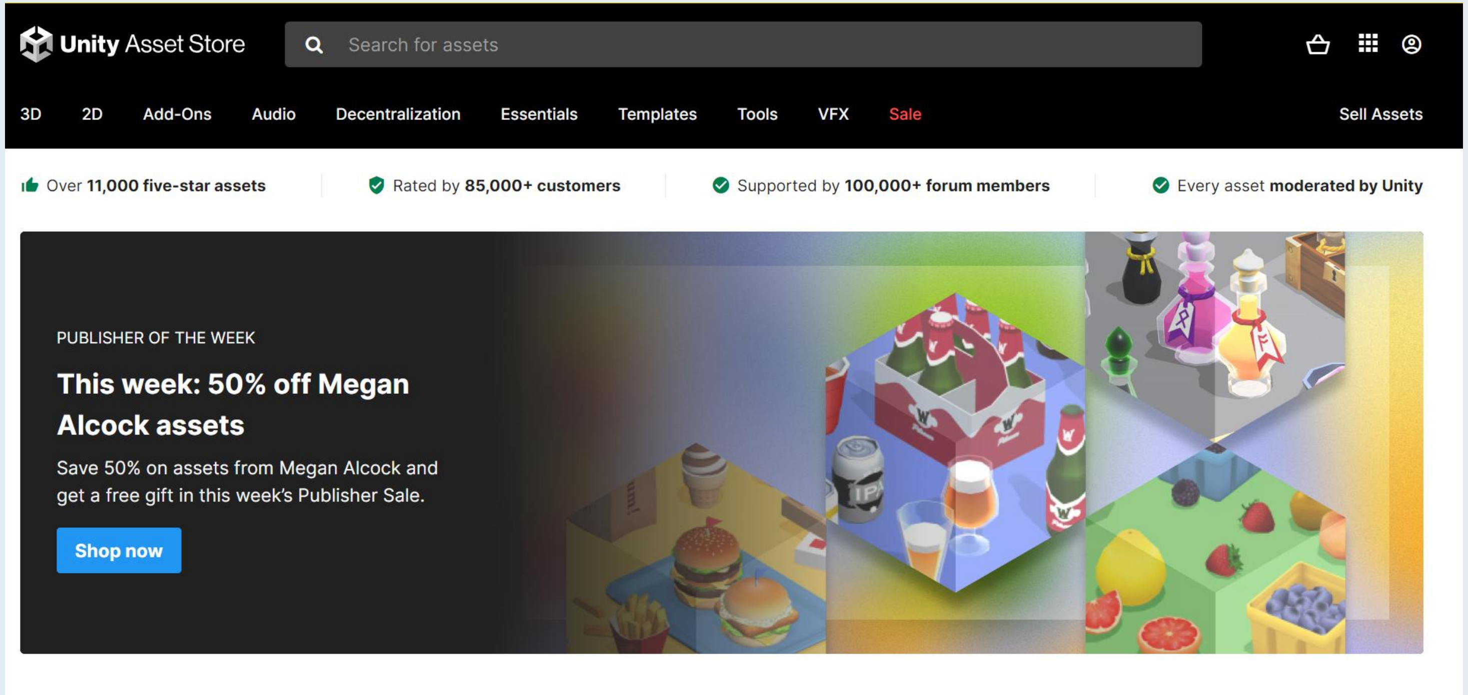
☑ Plane □ Static

Tag Untagged Layer Default

ASSET STORE

In the Unity Asset Store, you can find free and commercial Assets to use in your project:





Over **11,000 five-star assets**

Rated by **85,000+ customers**

Supported by **100,000+ forum members**

Every asset **moderated by Unity**

PUBLISHER OF THE WEEK

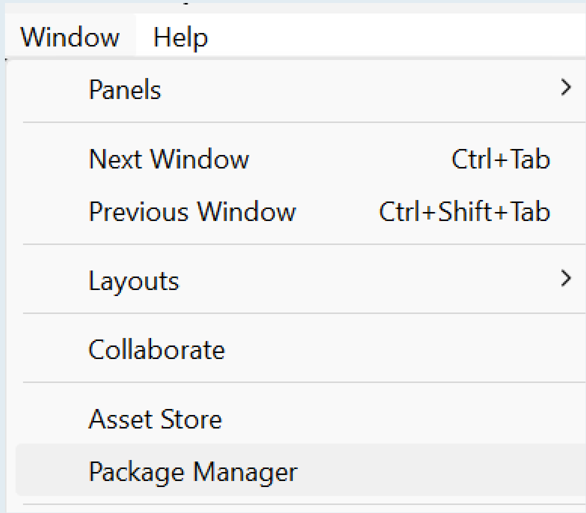
This week: 50% off Megan Alcock assets

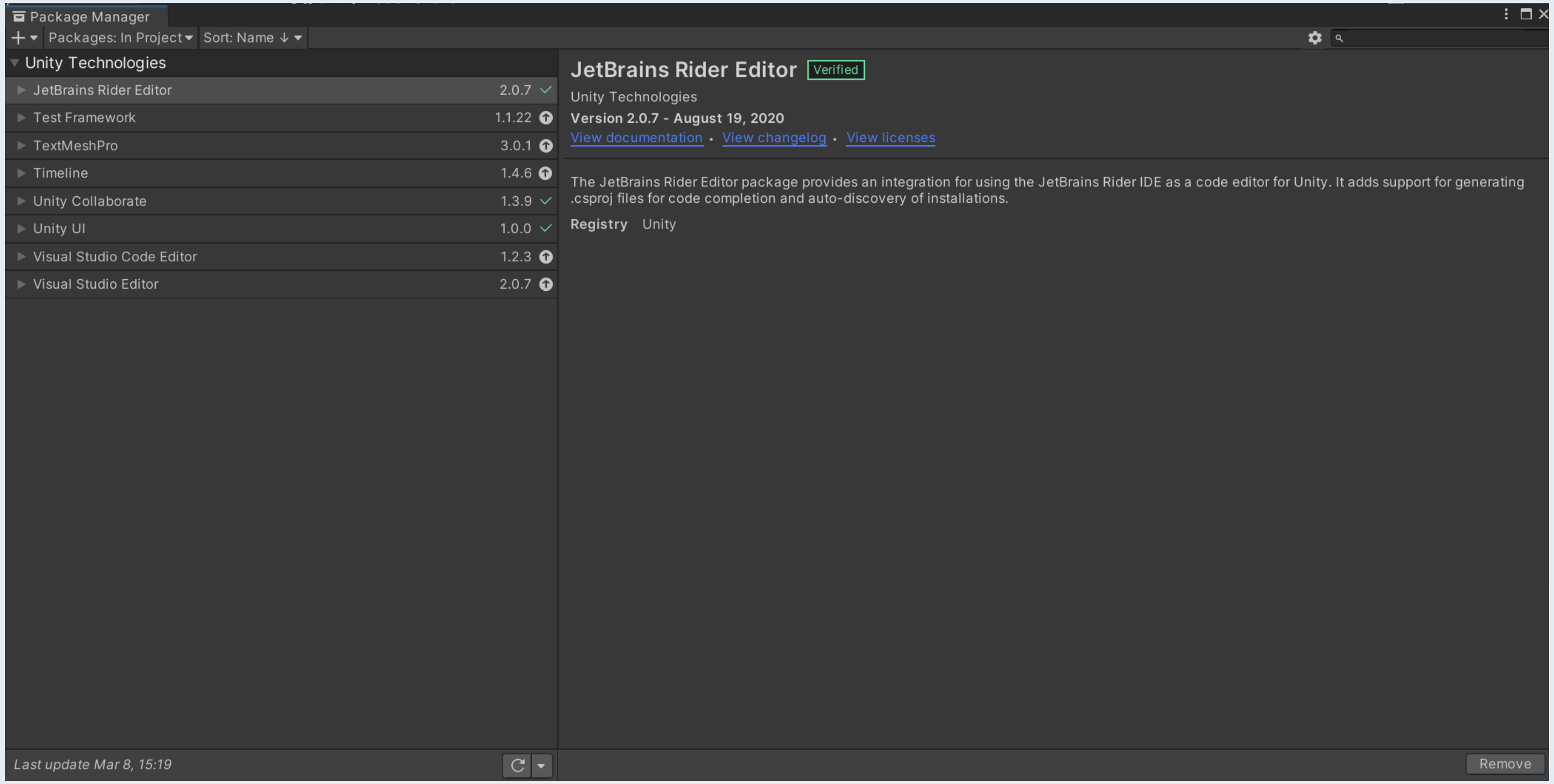
Save 50% on assets from Megan Alcock and get a free gift in this week's Publisher Sale.

https://assetstore.unity.com/

UNITY PACKAGE

Use the Unity Package Manager (in Unity’s top menu: Window > Package Manager) to view which packages are installed or can be installed in your project





JetBrains Rider Editor	2.0.7 ✓
Test Framework	1.1.22 ↑
TextMeshPro	3.0.1 ↑
Timeline	1.4.6 ↑
Unity Collaborate	1.3.9 ✓
Unity UI	1.0.0 ✓
Visual Studio Code Editor	1.2.3 ↑
Visual Studio Editor	2.0.7 ↑

Last update Mar 8, 15:19

JetBrains Rider Editor Verified

Unity Technologies

Version 2.0.7 - August 19, 2020

View documentation · View changelog · View licenses

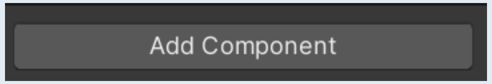
The JetBrains Rider Editor package provides an integration for using the JetBrains Rider IDE as a code editor for Unity. It adds support for generating .csproj files for code completion and auto-discovery of installations.

Registry Unity

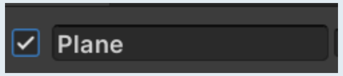
GAME OBJECT

GOs are the object which populate the Unity Scene

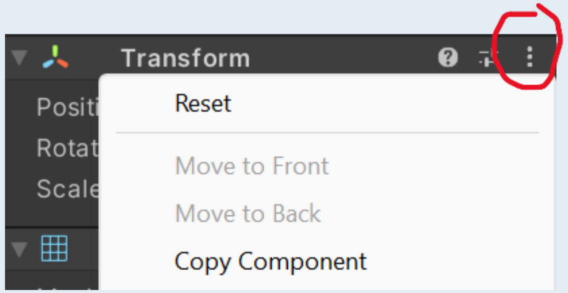
- GOs are made up by several components that could be added through Add Component button in the Inspector:



- Every GO has a transform component
- The GO could be activated or deactivated by checking or unchecking the box near the object’s name:



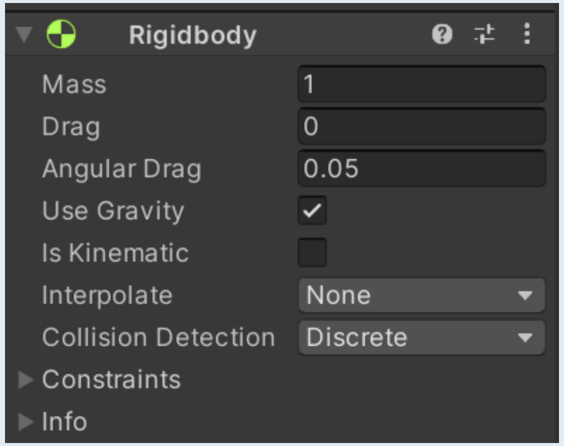
- Components could be copied and pasted between GOs:



PHYSICS

Unity helps you to simulate physics

- **Rigidbody** component enables physics-based behavior (movement, gravity, and collision)



Rigidbody

Mass 1

Drag 0

Angular Drag 0.05

Use Gravity ☑

Is Kinematic □

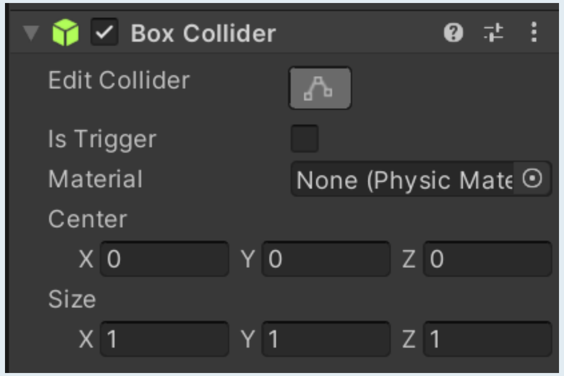
Interpolate None

Collision Detection Discrete

▶ Constraints

▶ Info

- To manage collision between GOs, you need a **Collider** component
  - You can change the shape of Collider for the purposes of physical collisions
  - use Colliders to manage collision events



☑ Box Collider

Edit Collider

Is Trigger □

Material None (Physic Mate

Center

X 0 Y 0 Z 0

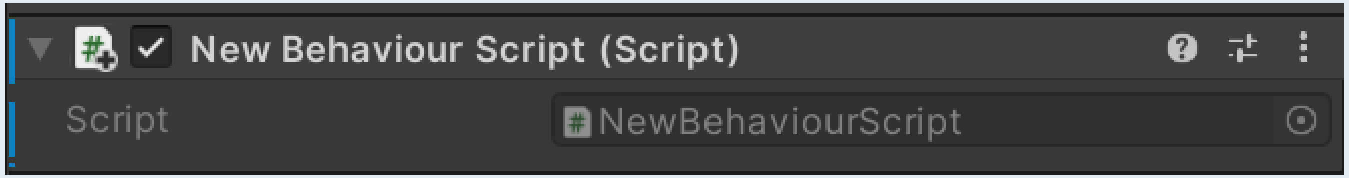
Size

X 1 Y 1 Z 1

SCRIPTING

The scripting language used by Unity is C#

- Scripts are mainly used to implement game mechanics, but can be used to customize the editor
- A script could be created in Project panel: click the right mouse button > Create > C# Script
- A script is a component, and it could be attached to a GO dragging it over the inspector or using the Component Add button



▼ ☑ New Behaviour Script (Script)

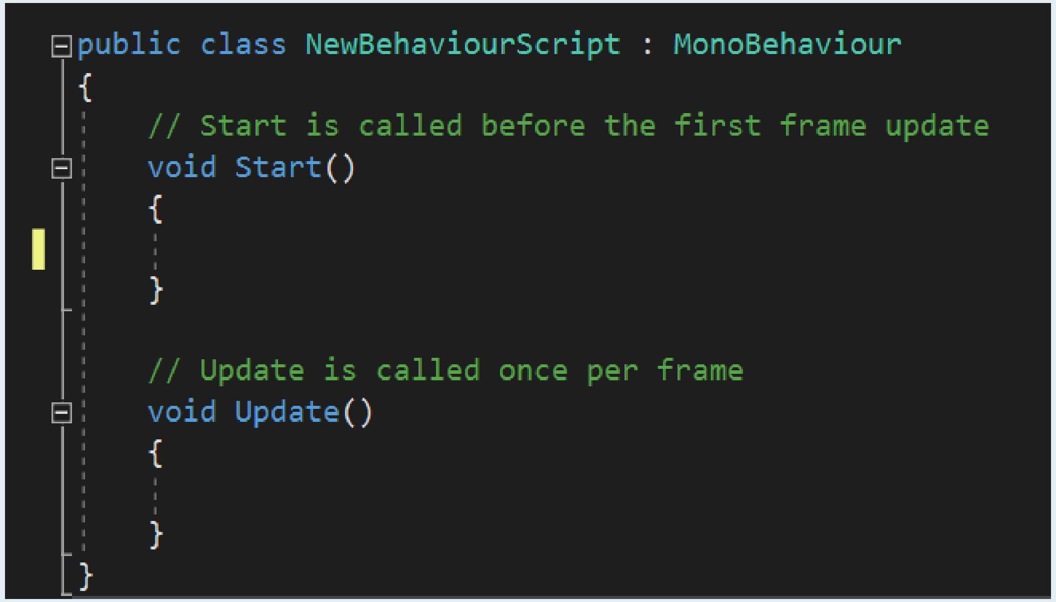
Script: NewBehaviourScript

- The script will start working when you press Play and run the game

SCRIPTING – START, UPDATE

Two important functions:

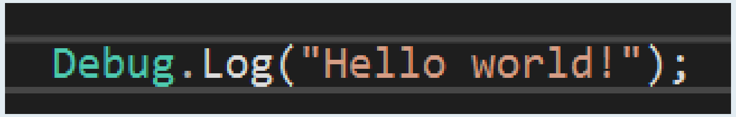
- Start(): called by Unity before gameplay begins, useful for any initialization
- Update(): called each frame, useful for handle everything which needs to be handled over time during gameplay



```
public class NewBehaviourScript : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

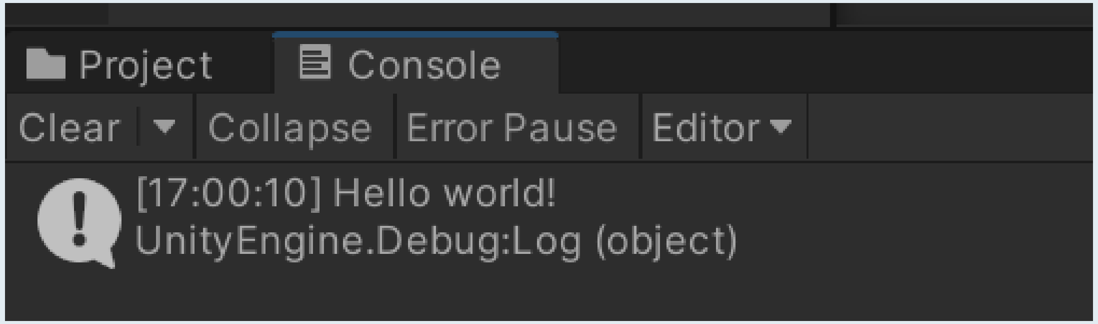
SCRIPTING – DEBUG.LOG

You can use Debug.Log command to print a message to Unity’s console output:



```
Debug.Log("Hello world!");
```

When you press Play:

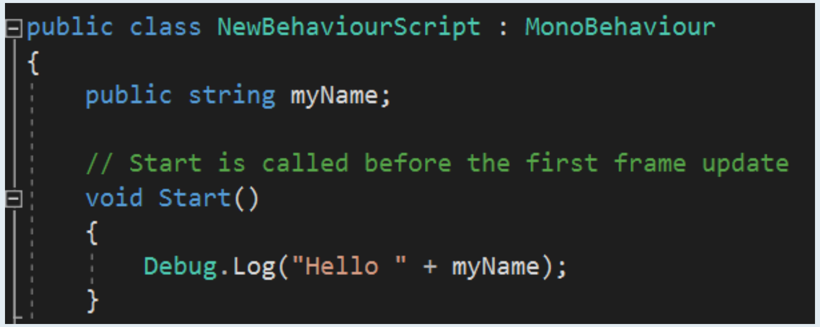


```
[17:00:10] Hello world!
UnityEngine.Debug:Log (object)
```

SCRIPTING - VARIABLES

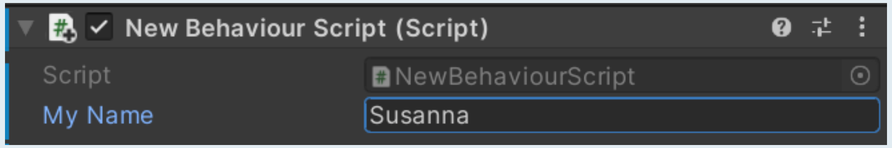
When you attach a script to a GO as component, you can edit its properties in the inspector

- You can define public variables in the script and edit their values in the inspector:



```
public class NewBehaviourScript : MonoBehaviour
{
    public string myName;

    // Start is called before the first frame update
    void Start()
    {
        Debug.Log("Hello " + myName);
    }
```

Script

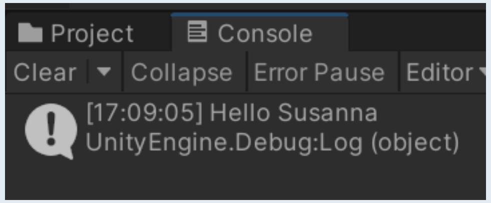


New Behaviour Script (Script)

Script NewBehaviourScript

My Name Susanna

Inspector



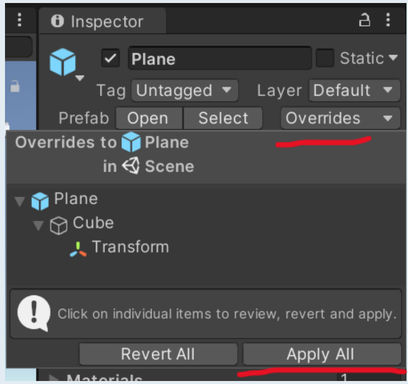
```
[17:09:05] Hello Susanna
UnityEngine.Debug:Log (object)
```

Play mode

PREFAB

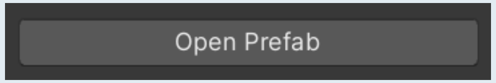
- To store a GO and its components as an asset we can create a Prefab
- To create a prefab simply drag the GO from the Hierarchy to the Asset folder (or prefab folder)
- The prefab could be then instantiated runtime or added to the scene

- If you modify a prefab in the scene, you can apply changes in the original prefab by clicking on Overrides > Apply All (or revert them with Revert All)



Click on individual items to review, revert and apply.

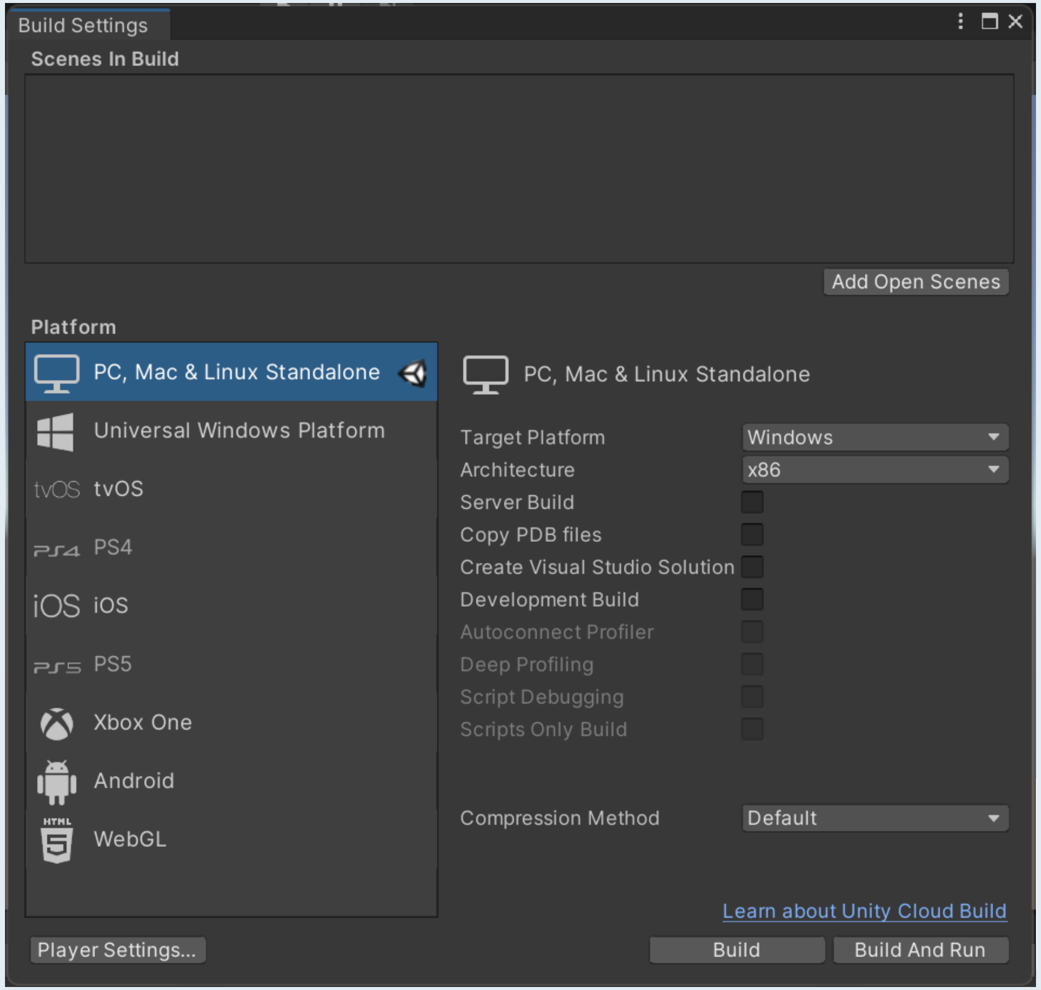
- You can also modify the original prefab:
  - Select the prefab in the folder
  - Click on Open Prefab



DEPLOY

You can build your project with File > Build Settings

- Add scenes to build dragging in “Scenes in Build” panel
- Select platform
- Modify player settings with Player Settings
- Click on Build or Build and Run to also run the application



Target Platform: Windows
Architecture: x86
Server Build □
Copy PDB files □
Create Visual Studio Solution □
Development Build □
Autoconnect Profiler □
Deep Profiling □
Script Debugging □
Scripts Only Build □
Compression Method: Default

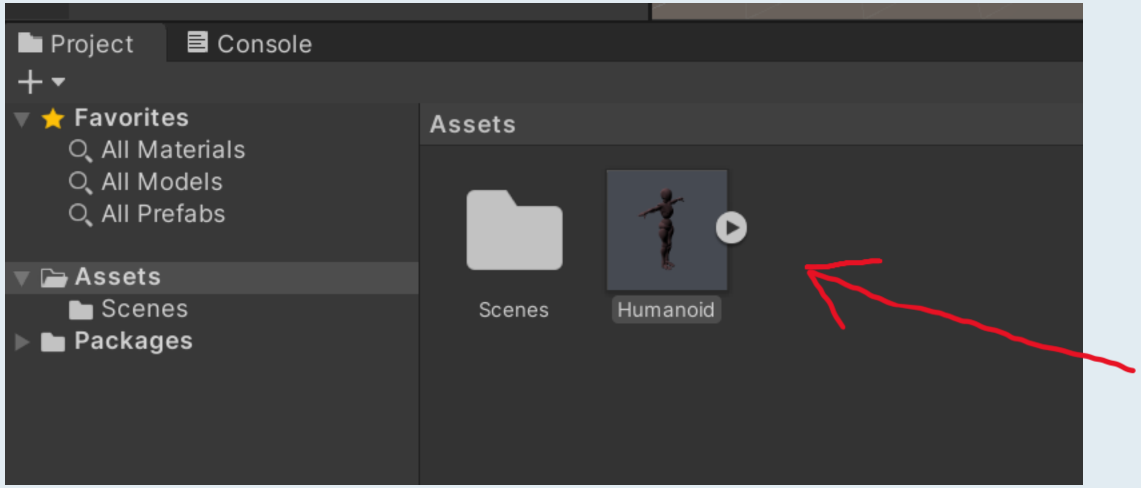
IMPORT CHARACTERS AND ANIMATIONS IN UNITY

CHARACTER SETUP: IMPORT

Open the Humanoid.blend file of the past lesson (you can find it also in Ex02-01 folder on Github):

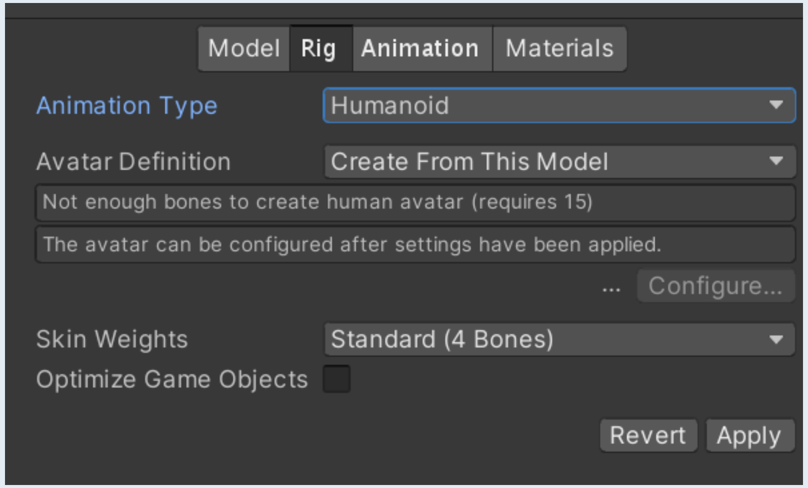
You can select the avatar, the beta_joints and the metarig export it as an FBX (with the option “selection only” enabled)

- Import the .fbx object in Unity:
  1. Drag the .fbx into the Project Window > Asset folder of the Unity Project
  2. Select it



CHARACTER SETUP: RIG TAB

1. In the Inspector panel, click on Rig
2. Change the Animation Type from *Legacy* to *Humanoid*
3. Change Avatar Definition to *Create From this Model*
4. Apply
5. Now an ‘Avatar’ appears in the Project Window



Animation Type: Humanoid

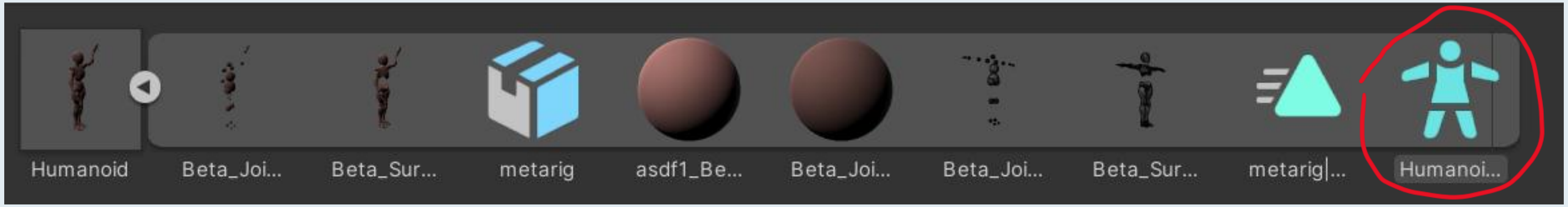
Avatar Definition: Create From This Model

Not enough bones to create human avatar (requires 15)

The avatar can be configured after settings have been applied.

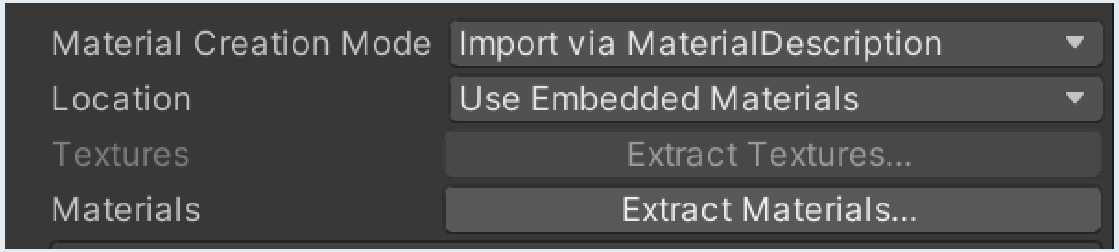
Skin Weights: Standard (4 Bones)

Optimize Game Objects: □



CHARACTER SETUP: MATERIALS TAB

1. In the Inspector Panel, select Materials Tab
2. Check Materials to open the settings for importing Materials from your .fbx
3. You can:
   - use embedded materials
   - extract materials and textures from the model



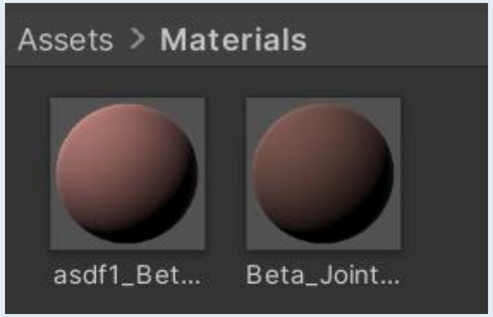
Material Creation Mode: Import via MaterialDescription

Location: Use Embedded Materials

CHARACTER SETUP: EXTRACT MATERIALS

1. Select *Use Embedded Materials* as Location and click on:
    - Extract Materials button to extract Materials that are embedded in your imported Asset
    - Extract Textures button to extract Materials that are embedded in your imported Asset

In both cases, decide a Folder where to place the materials/texture (generally Materials/Texture folder)

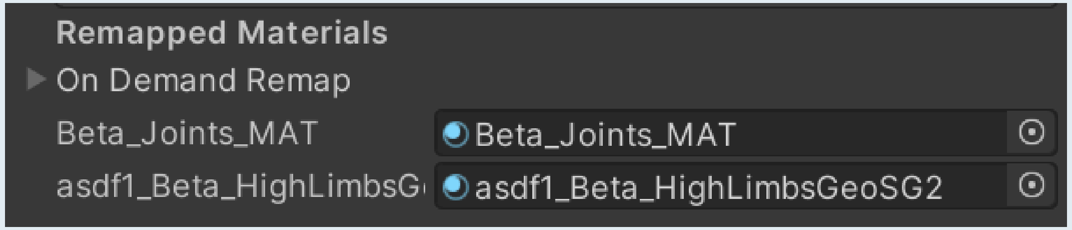


CHARACTER SETUP: REMAP MATERIALS

The materials will be automatically assigned

You can remap the materials searching for a Material in *On Demand Remap*

Click ‘Apply’ to apply all changes



Remapped Materials

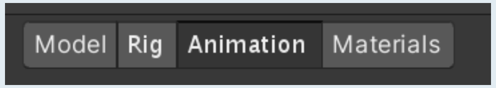
▶ On Demand Remap

Beta_Joints_MAT ● Beta_Joints_MAT ⊙

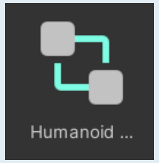
asdf1_Beta_HighLimbsG ● asdf1_Beta_HighLimbsGeoSG2 ⊙

CHARACTER SETUP: ANIMATION TAB

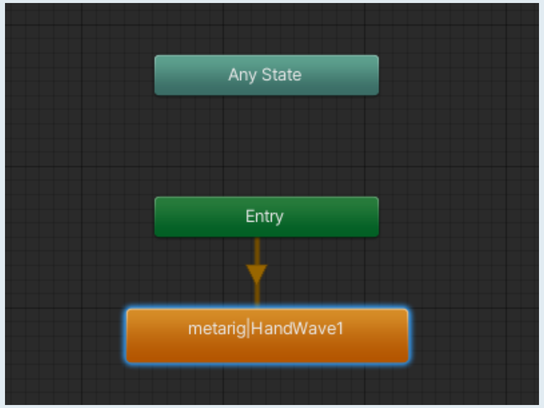
In the Animation Tab you can see and control the character animations



1. In the Project Window right click to add a new Animator Controller



2. Double click on the icon to open it

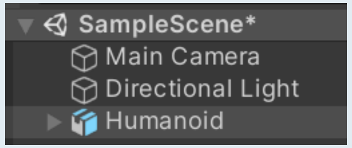


3. Drag the animation of the FBX inside the window, it will be automatically bind as a state in the Animator Finite State Machine

CHARACTER SETUP: ANIMATION

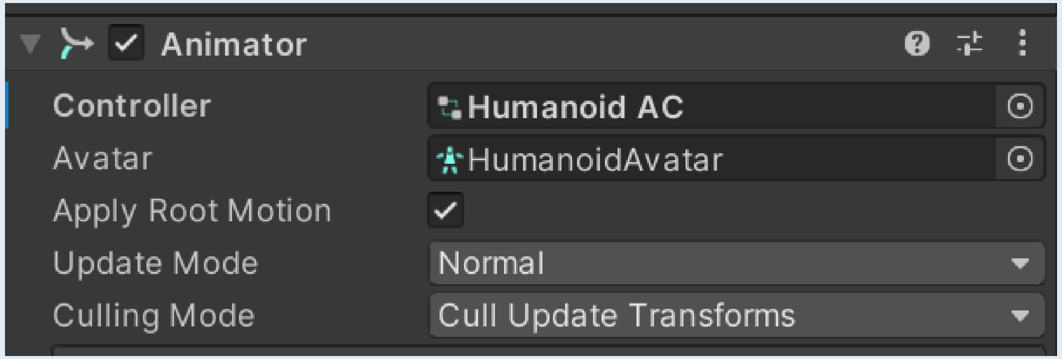
4. Drag the FBX in the Hierarchy or Scene, to transform it into a GameObject

5. Select the GO in the Hierarchy



SampleScene*
- Main Camera
- Directional Light
- Humanoid

6. Add the animator to the avatar GO dragging it into the GO inspector: Animator > Controller

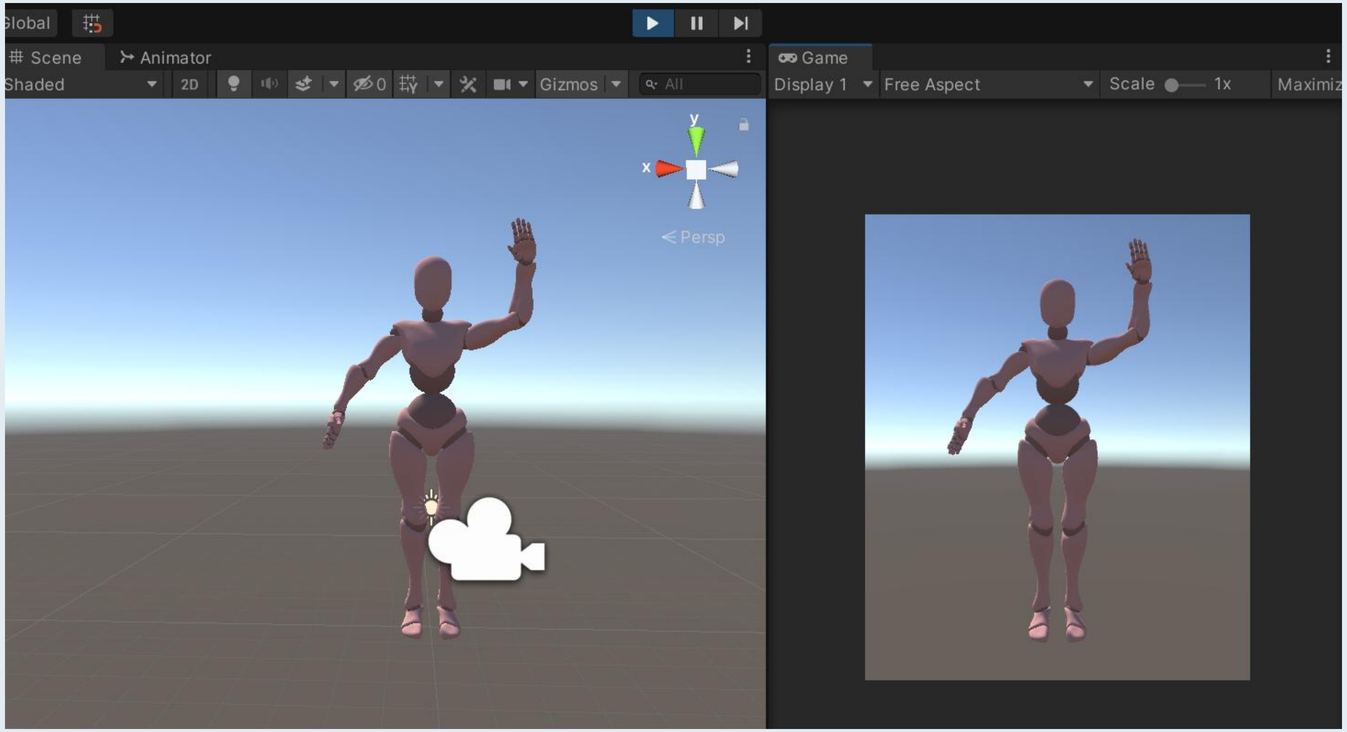


☑ Animator

Controller	Humanoid AC
Avatar	HumanoidAvatar
Apply Root Motion	☑
Update Mode	Normal
Culling Mode	Cull Update Transforms

PLAY IN THE EDITOR

You can play and check the results!


Global
Scene
Animator
Shaded
2D
0
Gizmos
All
y
x
< Persp
Game
Display 1
Free Aspect
Scale
1x
Maximiz