



Romel Cabiling ▾



Home

Home > My courses > 121 - MUL101 > 16 [Enter Module Title Here] > Lesson Proper for Week 16

Lesson Proper for Week 16

Introduction to Unity

Unity is an engine for creating games on multiple platforms. Unity was released by Unity Technologies in 2005. The focus of Unity lies in the development of both 3D and 2D games and interactive content. Unity now supports 27 different target platforms for deploying. The most popular platforms are Android, PC, and iOS systems.

- Unity is an integrated platform that is used as a gaming engine and framework.
- Unity allows you to develop once and publish everywhere.
- Although unity is considered to be more appropriate for creating 3D games, it can also be equally used to develop 2D games.
- In Unity, it is possible to develop games with heavy assets without depending on the additional frameworks or engines. It really enhances the experience of users.
- With the help of Unity, our game developers can access a wealth of resources like intuitive tools, ready-made assets, clear documentation, online community, etc. free of cost for creating exciting 3D contents in the games.
- Asset tracking and rendering, scripting are some of the features of Unity game development that we use in reducing time and cost.

Factors that Enhances the Efficiency of Unity

Unity Multiplayer: Multiplayer experience in unity is unparalleled. Unity enables the users to play for the traffic that uses the relay servers and matchmakers.

IDE: Unity provides the text editor for writing the code. To reduce confusion, sometimes, a distinct code editor is also used by our developers. As the IDE (Integrated development editor) of the unity engine supports C# and Unity Script (JavaScript), we use it in our game development process for creating immersive and exciting games.

Platform Support: The Unity engine is highly acceptable due to its ability to support a total of 27 different platforms. It is used for developing and deploying gaming apps

that can be easily shared among personal computers, mobile, and web platforms.

Debugging: Debugging and Tweaking is extremely easier with Unity game development. During the gameplay, the game variables are displayed, and it allows the developers to debug the process at run-time.

Unity Analytics: The built-in analytics of Unity offers indispensable insights regarding your game. It is necessary during the release of a game. Information related to the distribution of the game and feedback of the players can be easily obtained through unity analytics.

Graphics: The unity engine provides high-quality visual effects and audio. The visuals developed by Unity are adaptable on every device and screen without any compromise or distortion with the quality of the images.

Unity is the most preferred gaming engine among today's developers. On Unity, the coding part of the game development process is only about 20%. Hence extensive programming skills are not required.

The free license offered by Unity makes it open for game developers all over the world. The developers can access the resources from the asset store of Unity, which is used to enrich the process of mobile game development.

Installing Unity

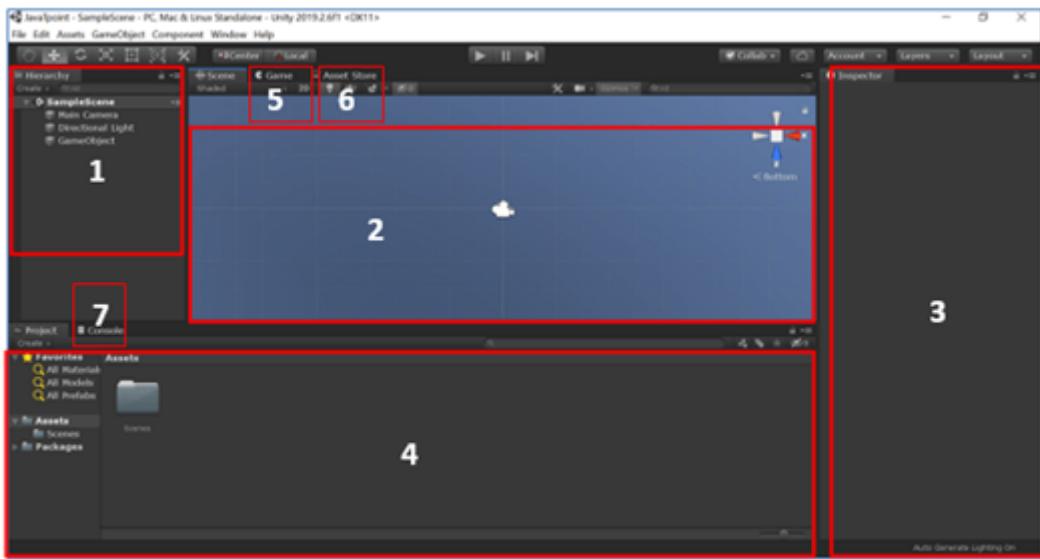
Before we begin to use Unity, we first need to download and install it. Software installation is a very simple and straightforward process these days, and Unity is no exception.

Unity provides three different versions: Unity Personal, Unity Plus, and Unity Pro. Unity Personal is completely free, while the Unity Professional comes at a monthly fee of \$125 and Unity Plus at a monthly fee of \$25. For personal use, the personal edition is completely sufficient.

Unity's interface

The main editor window is made up of tabbed windows that can be rearranged, detached, grouped, and docked. So we can say that the editor looks different from one project to the next, and one developer to the next, depending on personal preference and what type of work you are doing.

Once your new project is created, and Unity opens, the following window appears:



Unity's interface

Hierarchy Window

Scene View

Inspector Window

Project Window

Game Window

Asset Store

Console Window

1. Hierarchy Window

This is the hierarchy window. This is the hierarchical text representation of every object in the scene. It is where all the objects in your recently open scene are listed, along with their parent-child hierarchy.

Each item in the scene has an entry in the hierarchy, so the two windows are linked. The hierarchy defines the structure of how objects are attached to one another.

By default, the Hierarchy window lists GameObjects by order of creation, with the most recently created GameObjects at the bottom. We can reorder the GameObjects by dragging them up or down, or by making the parent or child GameObjects.

2. Scene View

This window is where we will create our scenes. This view allows you to navigate and edit your scene visually.

The scene view can show a 2D or 3D perspective, depending on the type of project you are working on.

We are using the scene view to select and position scenery, cameras, characters, lights, and all other types of GameObject.

Being able to select, manipulate, and modify objects in the scene view are some of the most important skills you must learn to begin working in Unity.

3. Inspector Window

The Inspector window allows you to view and edit all the properties of the currently selected object.

Since different types of objects have different sets of properties, the layout and contents of the inspector window will vary.

In this window, you can customize aspects of each element that is in the scene.

You can select an object in the Hierarchy window or double click on an object in the scene window to show its attributes in the inspector panel.

The inspector window displays detailed information about the currently selected GameObject, including all attached components and their properties, and allows you to modify the functionality of GameObjects in your scene.

4. Project Window

This window displays the files being used for the game. You can create scripts, folders, etc. by clicking create under the project window.

In this view, you can access and manage the assets that belong to your project.

All assets in your project are stored and kept here. All external assets, such as textures, fonts, and sound files, are also kept here before they are used in a scene.

The favorites section is available above the project structure list. Where you can maintain frequently used items for easy access. You can drag items from the list of project structure to the Favorites and also save search queries there.

5. Game Window

This window shows the view that the main camera sees when the game is playing. Means here, you can see a preview window of how the game looks like to the player.

It is representative of your final game. You will have to use one or more cameras to control what the player actually sees when they are playing your game.

6. Asset Store

The Unity asset store is a growing library of free and commercial Assets created both by Unity Technologies and also members of the community.

A wide variety of Assets is available, covering everything from Models, Textures, and animations to whole Project examples, tutorials, and Editor Extensions.

The assets are accessed from a simple interface created into the Unity Editor and are downloaded and imported directly into your project.

7. Console Window

If you are familiar with programming, you will already know that all the output messages, errors, warnings, and debug messages are shown here. It is similar for Unity, except output messages are done a bit differently than you think.

The console window of Unity shows the errors, warnings, and other messages generated by Unity.

You can also show your own messages in the console using the Debug.Log, Debug.LogError, Debug.LogWarning function.

Introduction to Unity 2D

Unity is available for both 2D and 3D games. When you create a new project in Unity, you will have a choice to start in 2D or 3D mode. The choice between starting from 2D or 3D mode determines some settings for the Unity Editor, such as whether images are imported as sprites or textures. You can swap between 2D or 3D mode at any time regardless of the mode you set when you created your project.

Sprites in Unity

Sprites are simple 2D graphic objects that have graphical images (called textures) on them. Unity handles sprites by default when the engine is in 2D mode.

If you are 3D, sprites are essentially just standard textures, but there are special techniques for combining and managing sprite textures for efficiency and convenience during development. When you view the sprite in 3D space, sprites will appear to be paper-thin, because they have no Z-width.

Sprites always face the camera at a right angle unless rotated in 3D space.

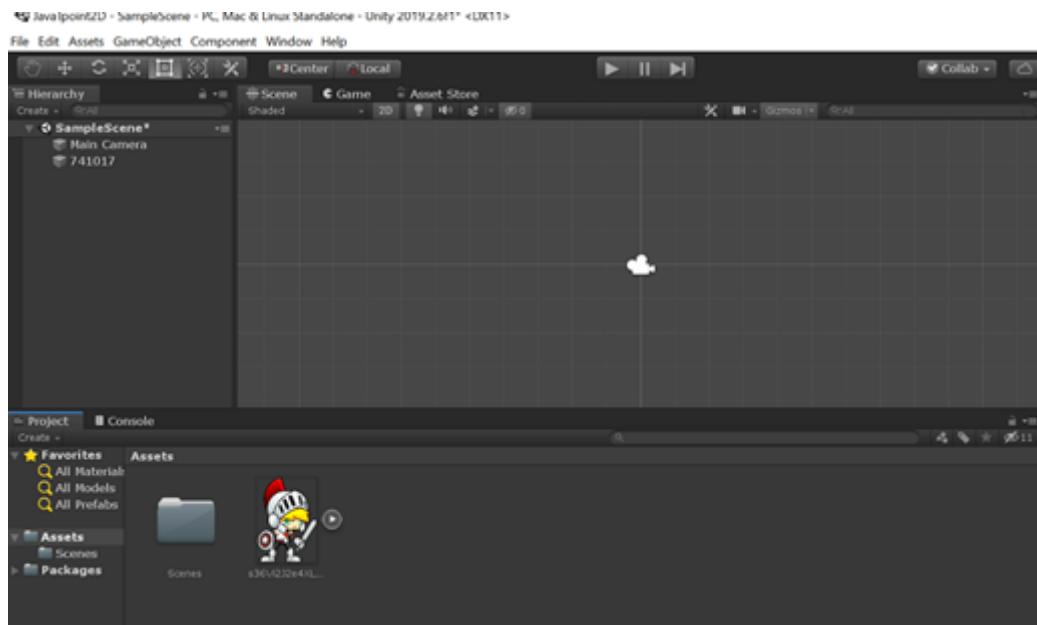
When you create a new sprite, it uses a texture. This texture is then applied on a fresh GameObject, and the Sprite Renderer component is attached to it. This makes our GameObject visible with our texture, as well as its properties related to how it looks on-screen.

Creating Sprites

To create a sprite to your game, you must supply the engine with a texture. Let's create a texture first.

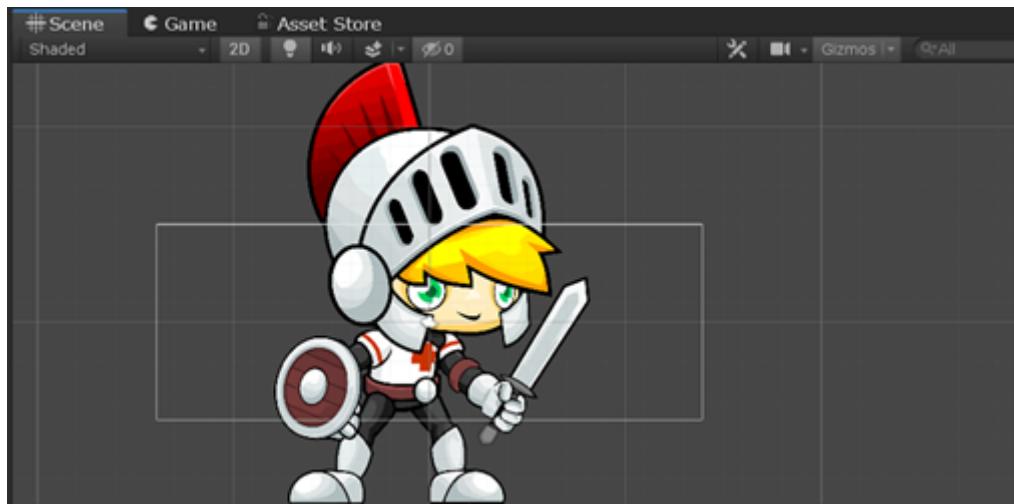
Get an image what you want to add as a sprite in standard image file such as PNG or JPG that you want to use, Save it in your system directory and

Then drag the image into the Assets region of Unity.



Now drag the image from the Assets into the Scene Hierarchy.

You will notice that as soon as you let go of the mouse button, a new GameObject with the name of the texture shows up in the list. You will also get the image now in the middle of the scene in the scene view.



Let us consider the following points while adding a sprite:

By dragging from an external source into Unity, we are putting an asset.

This added asset is an image, so it becomes a texture.

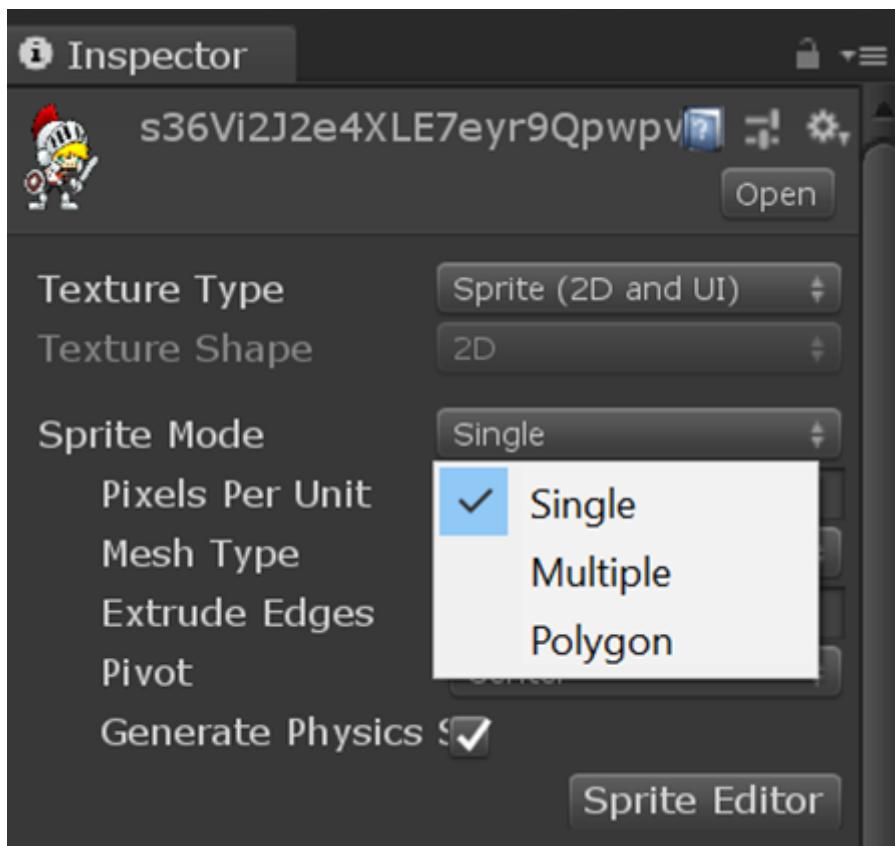
By dragging this texture into the scene hierarchy, we are creating a new GameObject with the same name as our texture, with a sprite renderer attached.

This sprite renderer uses that texture to draw the image in the game.

We have now added a sprite in our scene.

Sprite Modes

This setting is used to specify how the sprite graphic is extracted from the image. To choose the modes, click on a sprite in the Assets/ Sprites folder, in the inspector, there are three different modes in which you can use Sprites:



Single: It is used for a single image sprite.

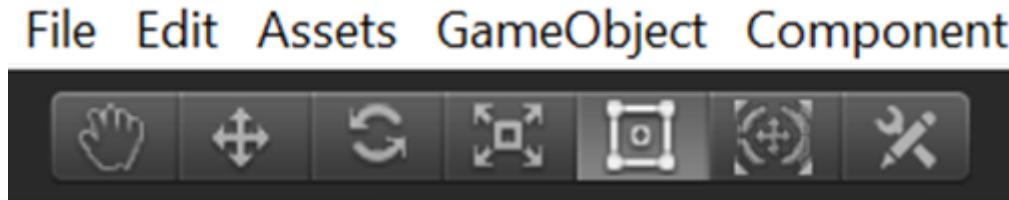
Multiple: It is used for a sprite with multiple elements, such as animations or spritesheets, with different parts for a character.

Polygon: It is used for a custom polygon-shaped sprite that you can create many different types of primitive shapes with, for example, Square, Triangle, Pentagon, Hexagon, etc.

Modifying Sprites

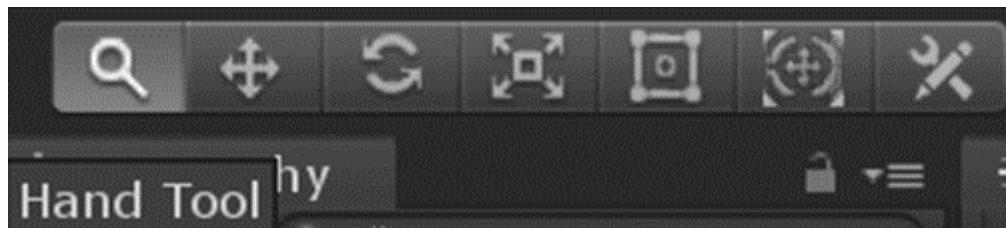
We can manipulate the imported sprites in various ways to change how it looks.

If you look at the top left corner of the unity interface, you will get a toolbar, as shown below:

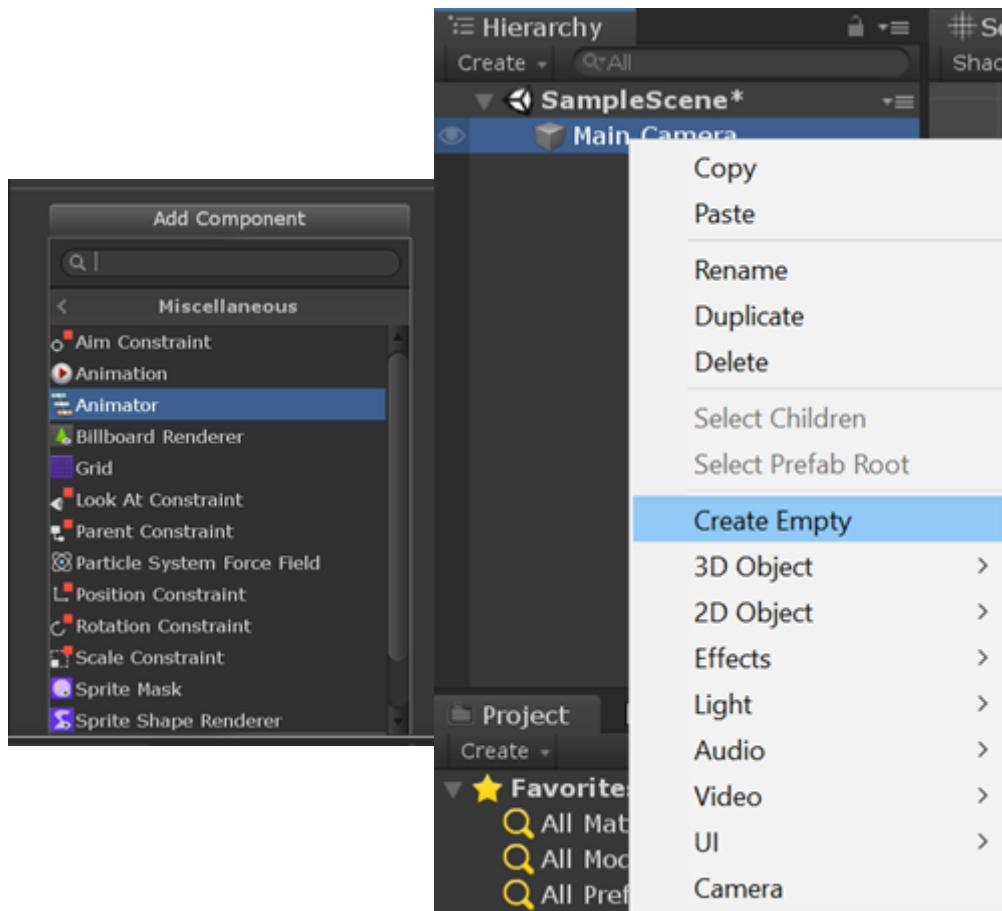


Let's see the functions of these buttons:

A **first-Hand** tool is used to move around the scene without affecting any objects.

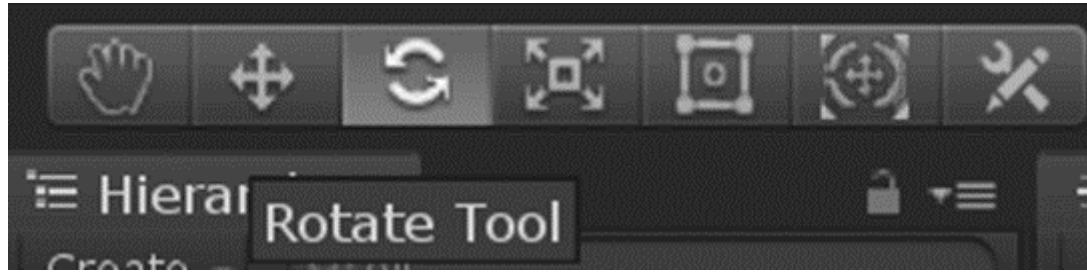


The next tool is the **Move** tool. This is used to move the objects in the game world around.

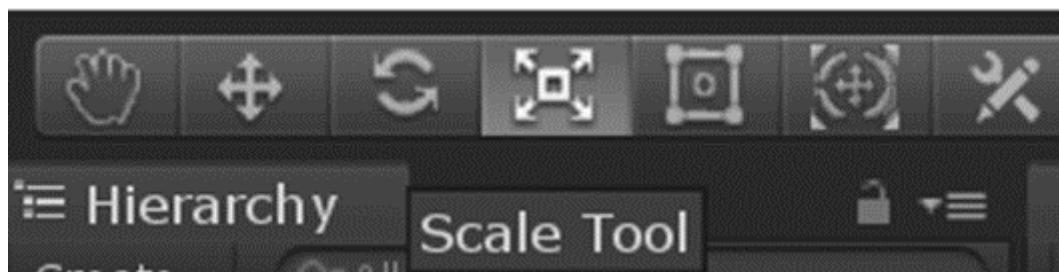




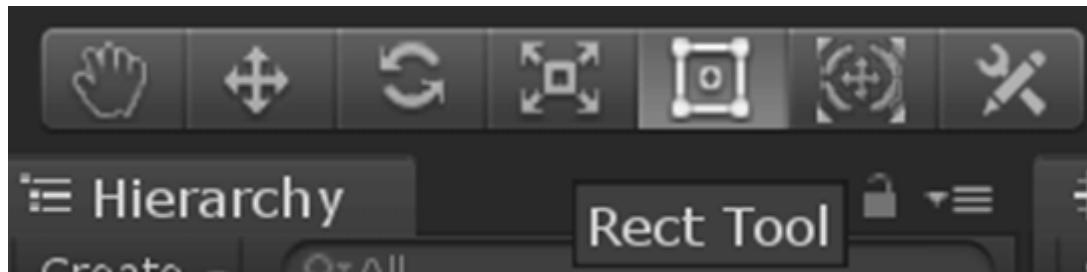
The next tool is the **Rotate** tool, which is used to rotate objects along the Z-axis of the game world or parent object



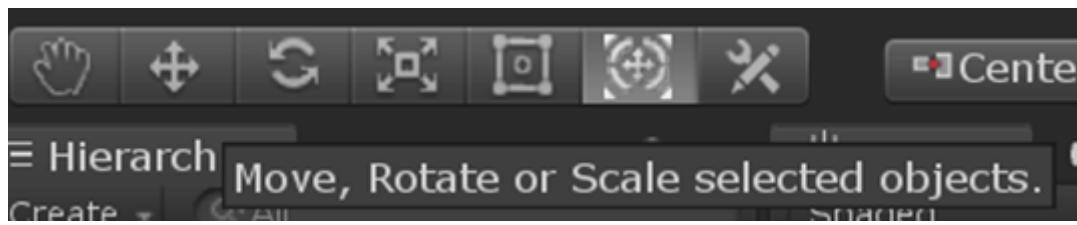
The centered tool is the **Scale** tool. This tool allows you to modify the size (scale) of the objects along certain axes.



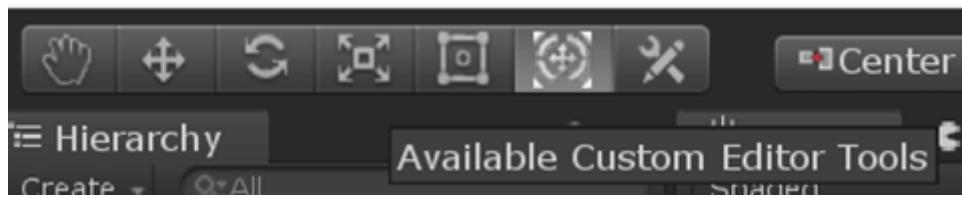
The next tool is the **Rect** tool. This tool behaves like a combination of the Move and the Scaling tool but is prone to loss of accuracy. It is more useful in arranging the UI elements.



The next tool is the **Move**, **Rotate**, and a **Scale** tool. It is used to move, rotate, and scale the selected object.



And finally, the last tool is the **Custom Editor** tool.



These tools are very useful and worthy as the complexity of the project increases.

2D Sprite Sheet

In a nutshell, a sprite sheet is a way of packing images together as one image, which is then used to create animations and sprite graphics as it will use low memory and increase the performance of games.

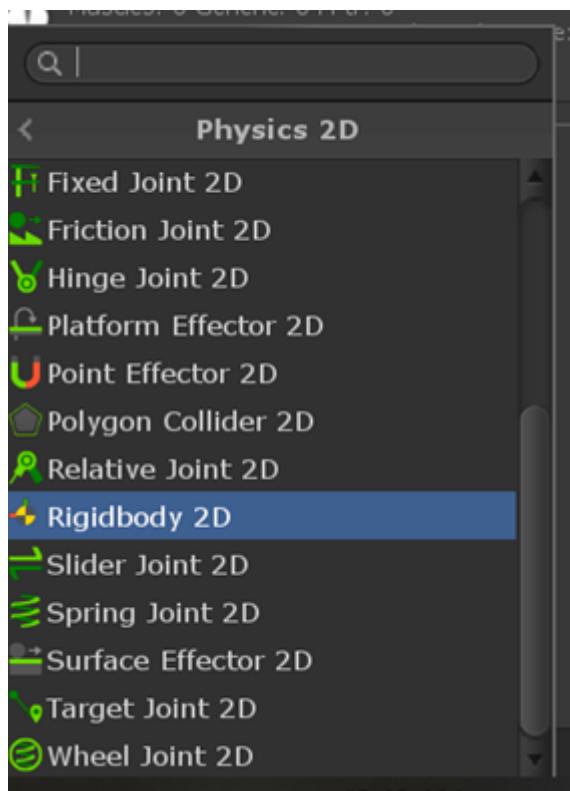
Creating GameObject and Adding Components

First of all, we need a GameObject in our scene. For this Right Click on the Hierarchy tab and select Create Empty.

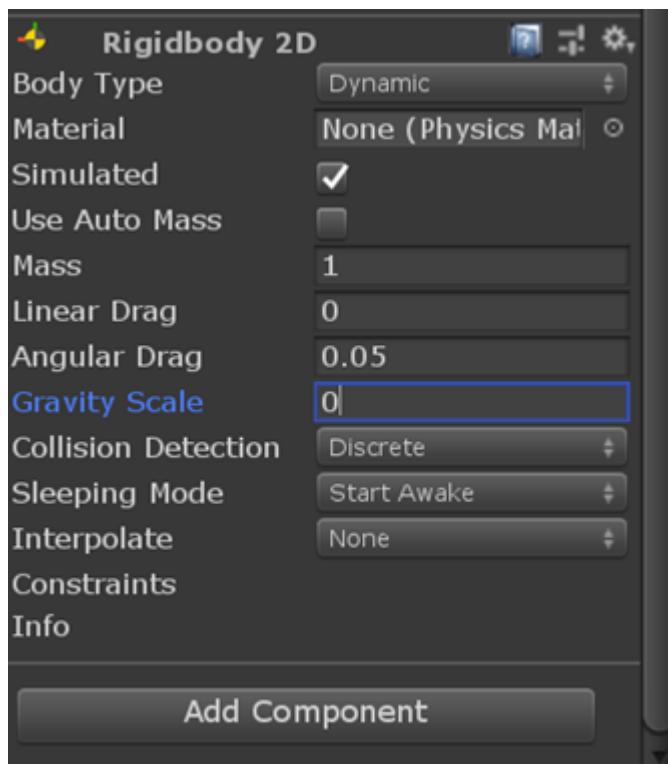
Rename the new GameObject. Here, we renamed it to Player.

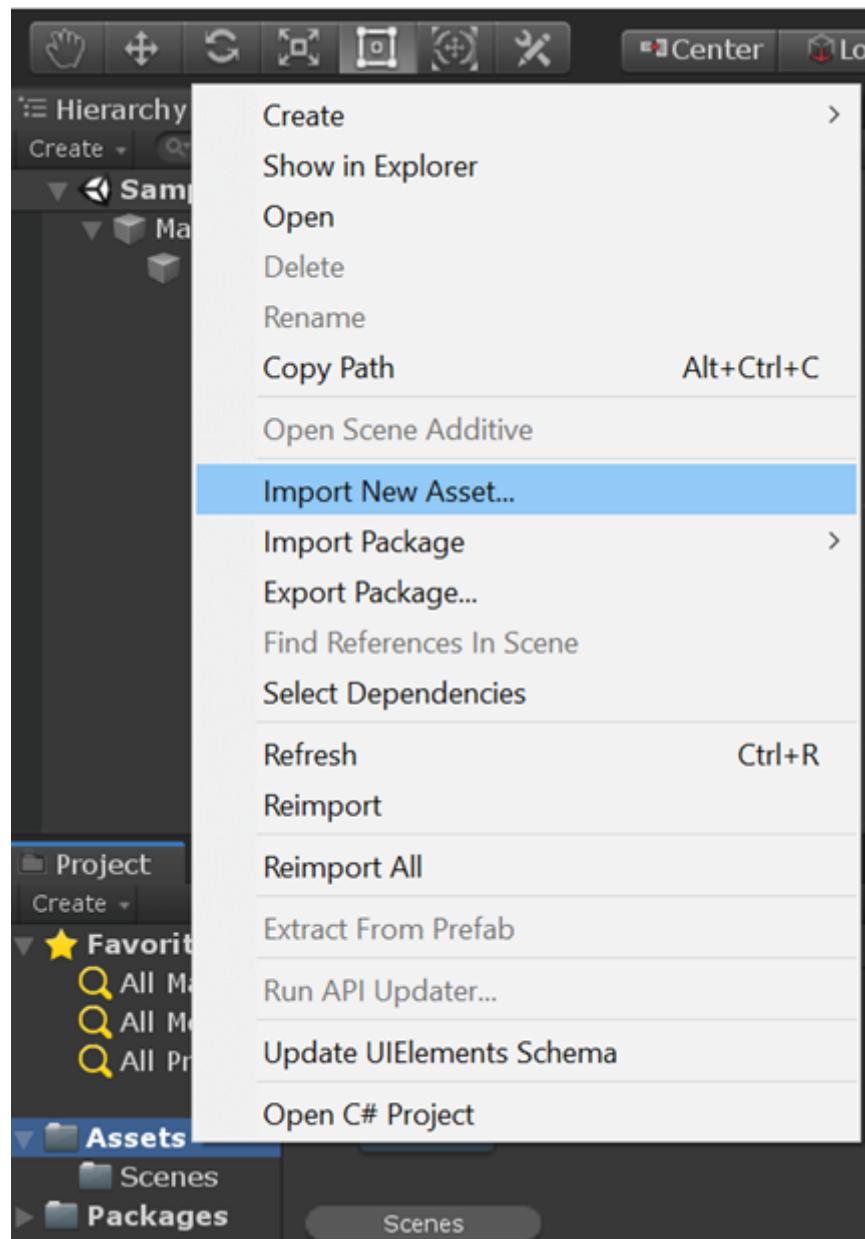
Select the Player GameObject and Go to the Inspector tab. In the Inspector tab, click the Add Component button. Choose miscellaneous -> Animator. Make sure to choose Animator nor Animation.

Add another component, Rigidbody2D. For this, click on the Add Component button and choose Physics 2D -> Rigidbody 2D.

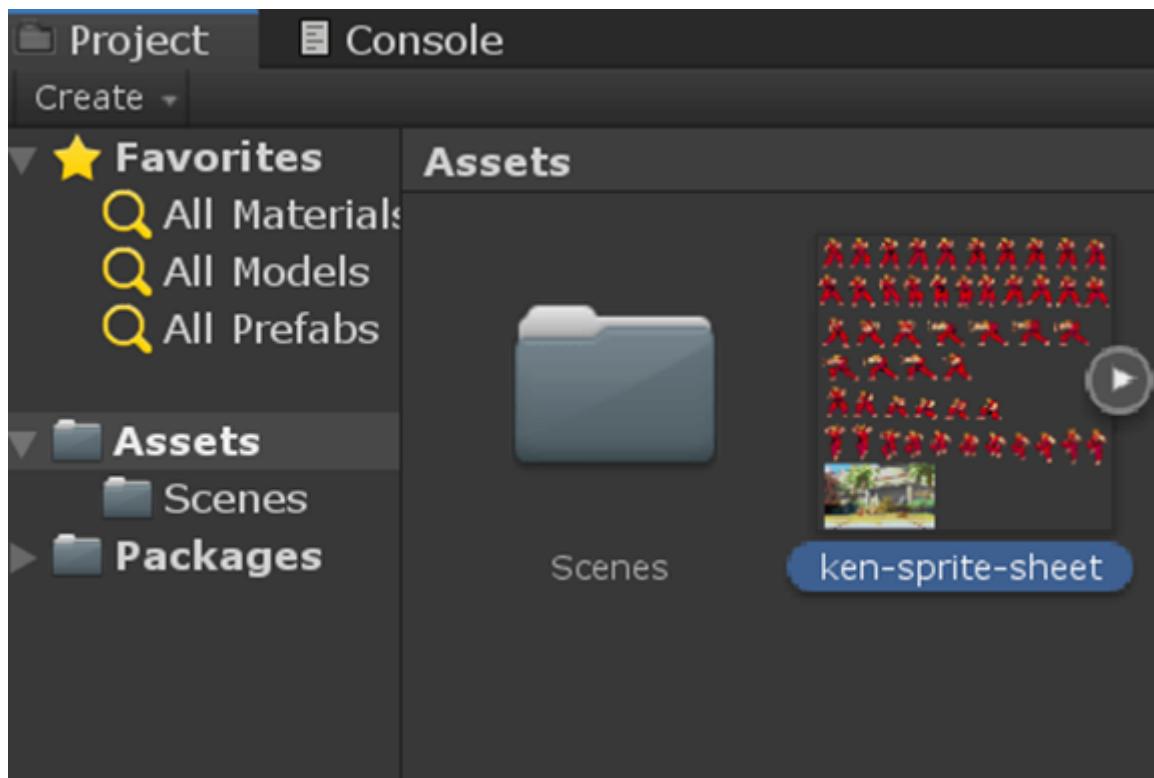


Now, in the Rigidbody 2D component, set the Gravity Scale to zero.

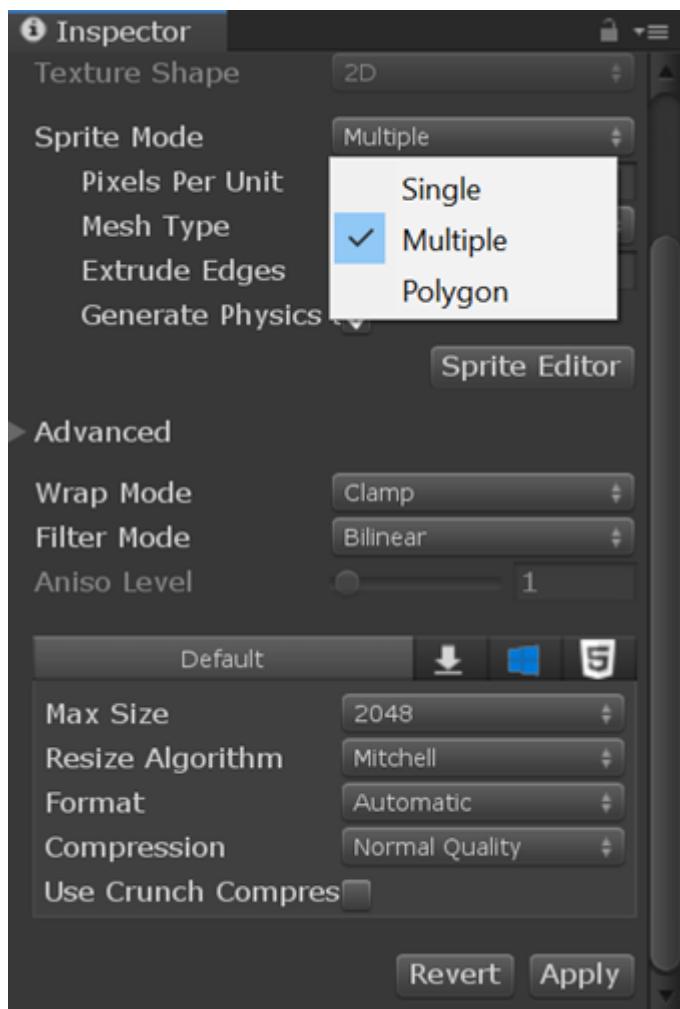




Browse to the sprite sheet image you downloaded and click the import button



Select this imported Asset, and in the Inspector window, change the Sprite mode option from Single to Multiple. Click the Apply button in the Inspector tab.

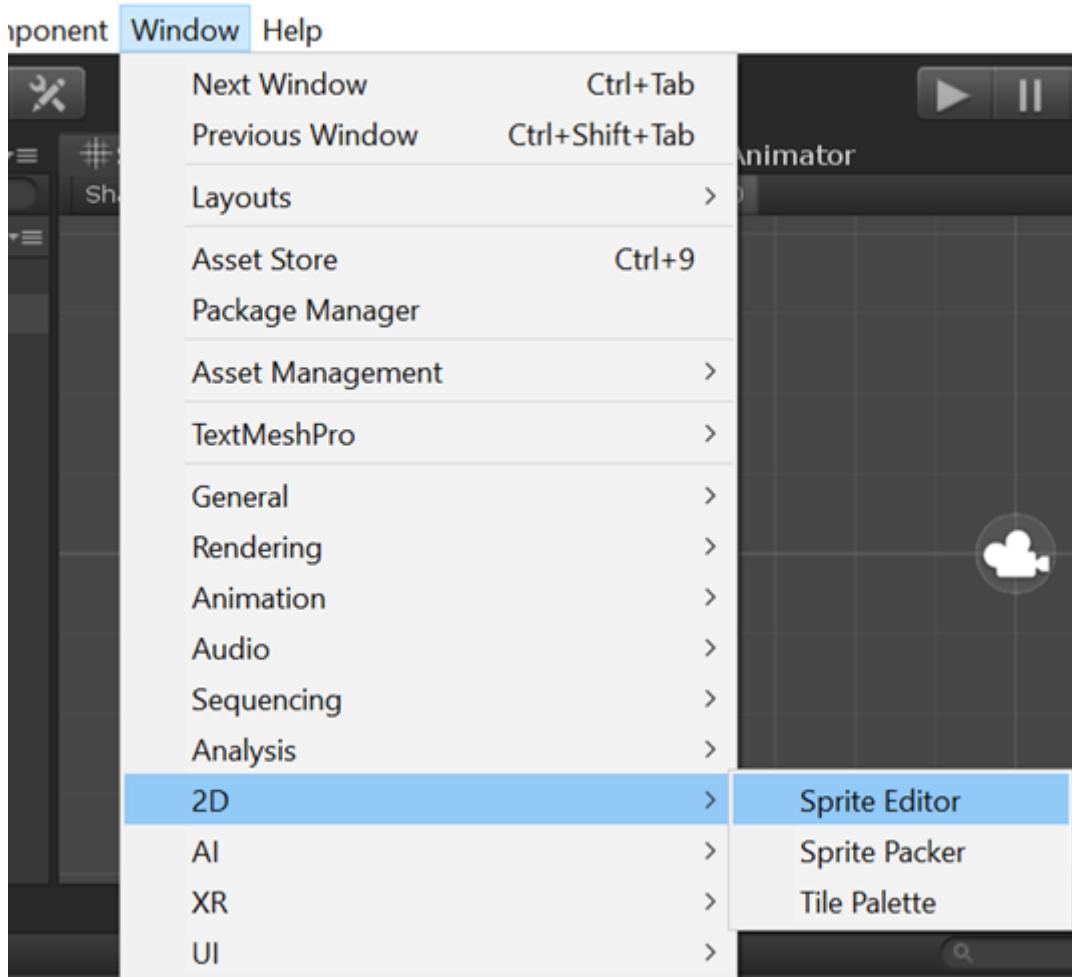


Now Unity will treat ken-sprite-sheet.png as a sprite sheet with multiple frames of animation.

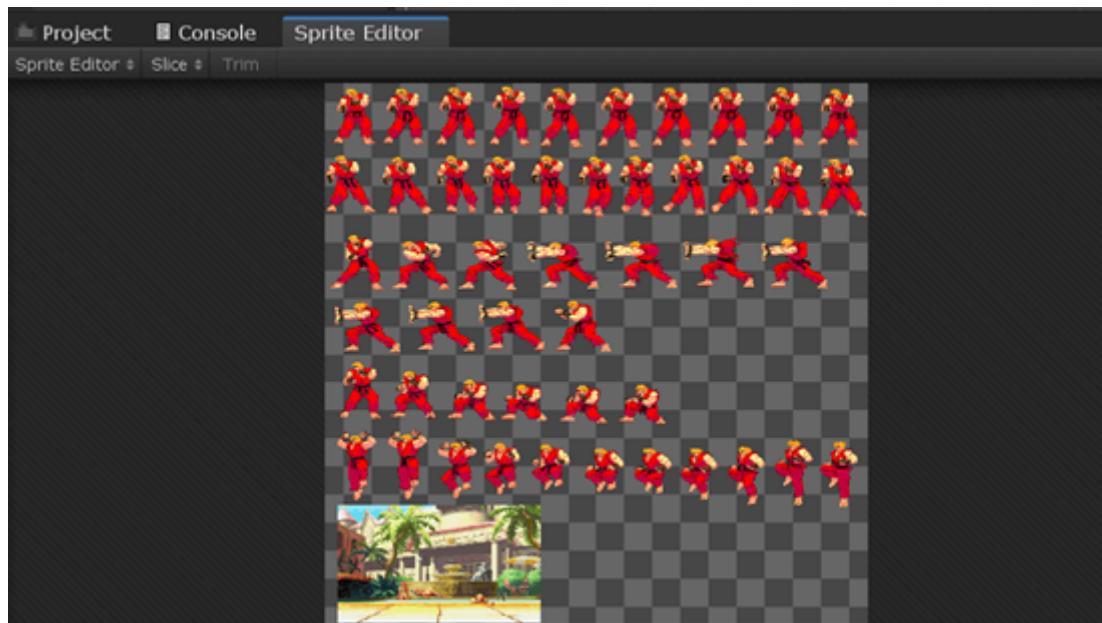
Slicing Sprite Sheet

Go to the Windows menu and select 2D -> Sprite Editor.

& Linux Standalone - Unity 2019.2.6f1* <DX11>



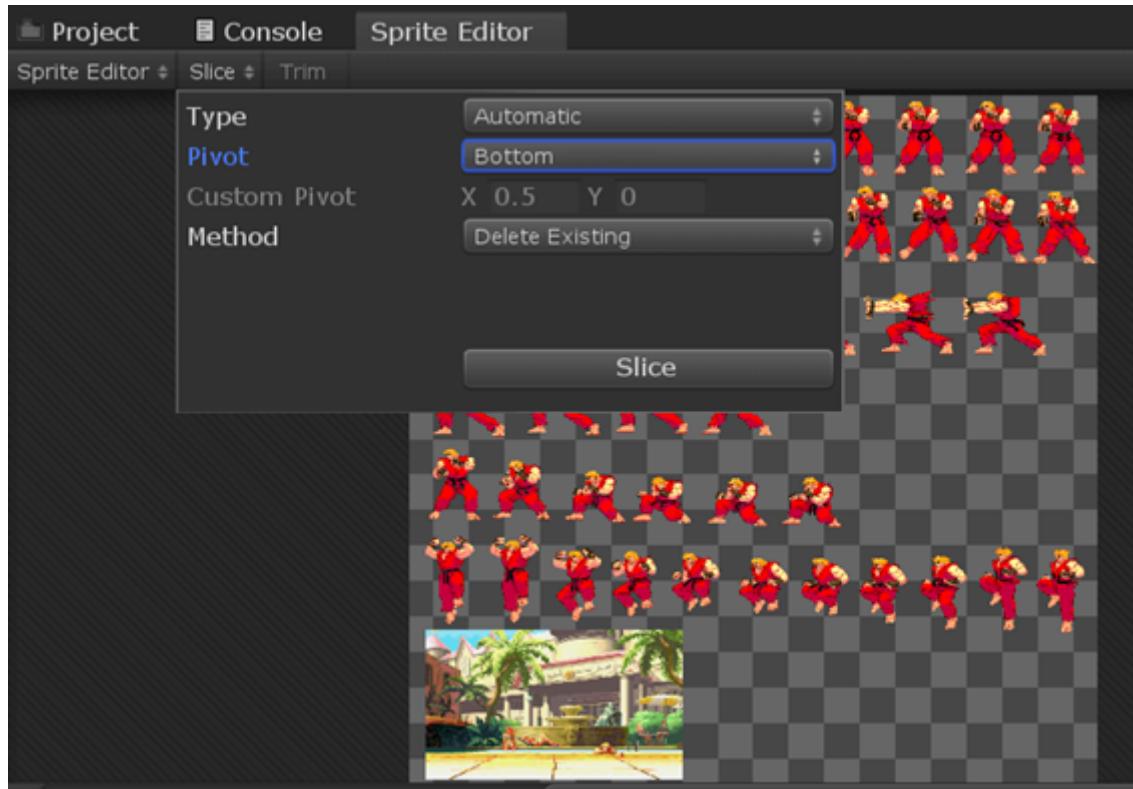
Drag the Sprite Editor window and dock it. I docked mine alongside the Console tab.



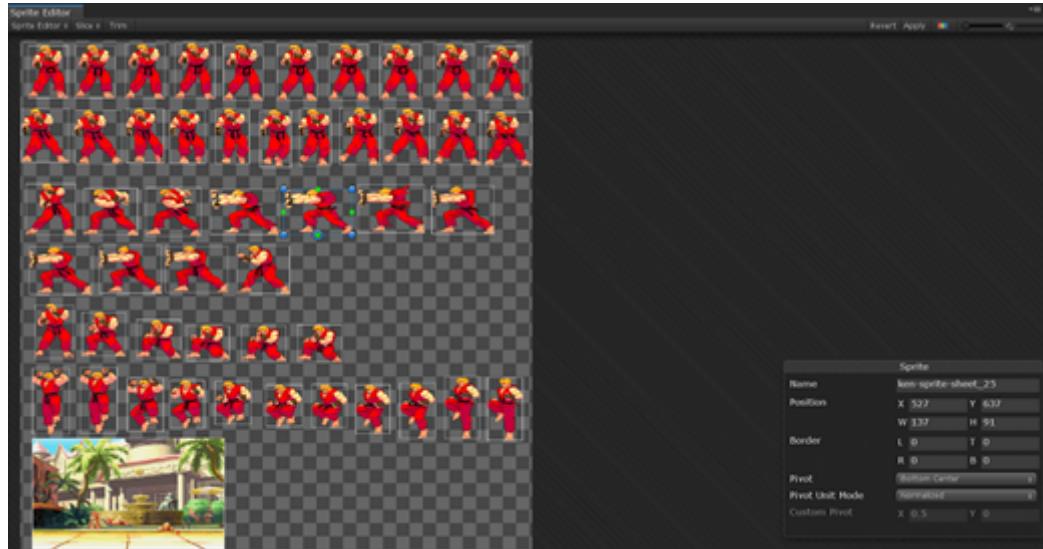
Click on the Slice drop-down, and here we can see that the options Automatic, Centre, and Delete Existing are default chosen. Some sprite-sheets have their images ordered in different ways. That is why there are a number of

options to choose from.

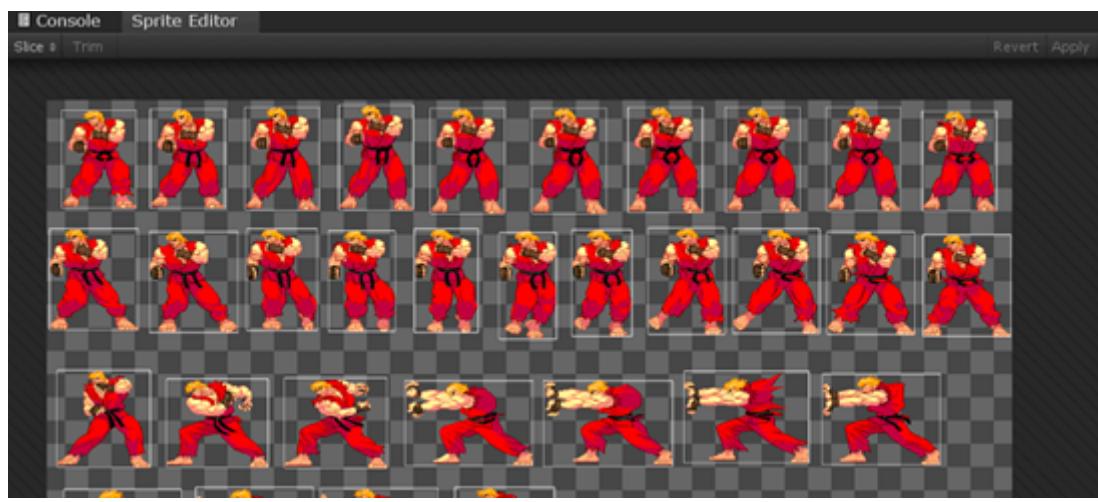
Here I changed the Pivot value from Centre to Bottom. When you change the Pivot to Bottom means it sets the pivot point to the center bottom of the sprite, and slicing Type in this particular case should be set to be automatic.



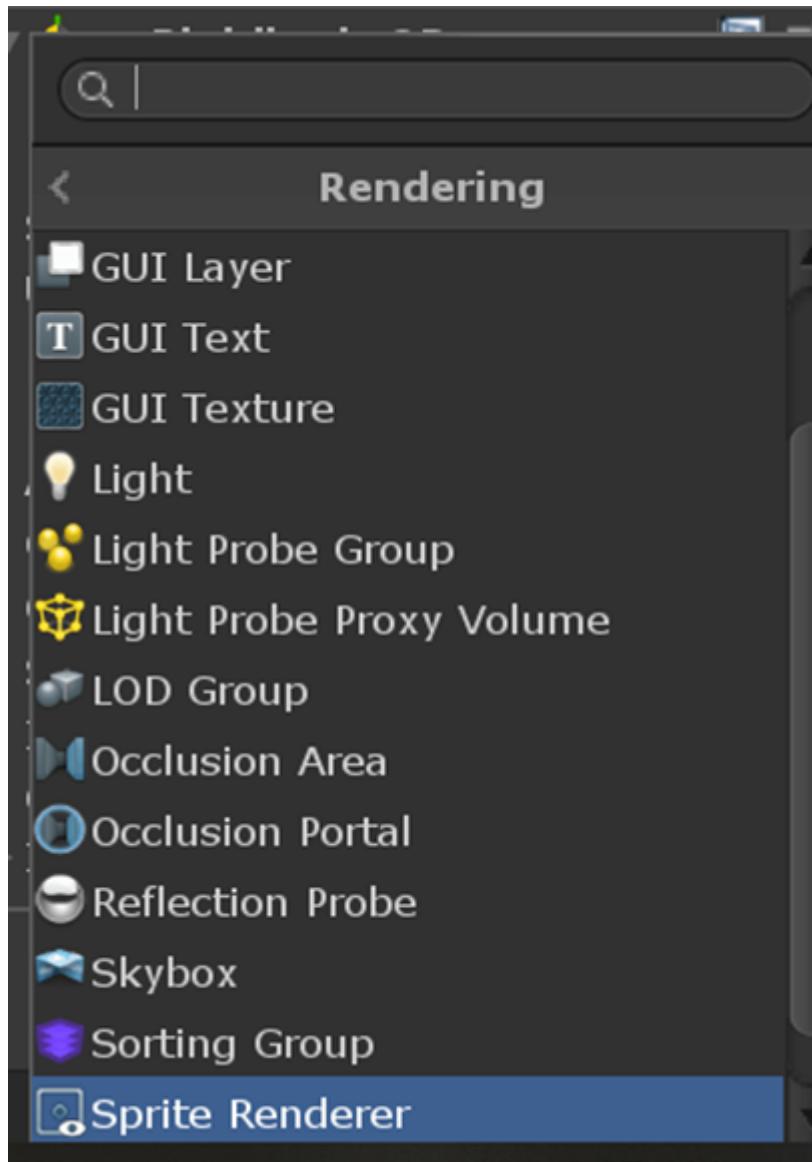
Then click on the Slice button. Unity has now separated all the sprites. Each sprite should have its own bounding box, clicking them provide information on each sprite and allow fine-tuning of sizes and pivot points.



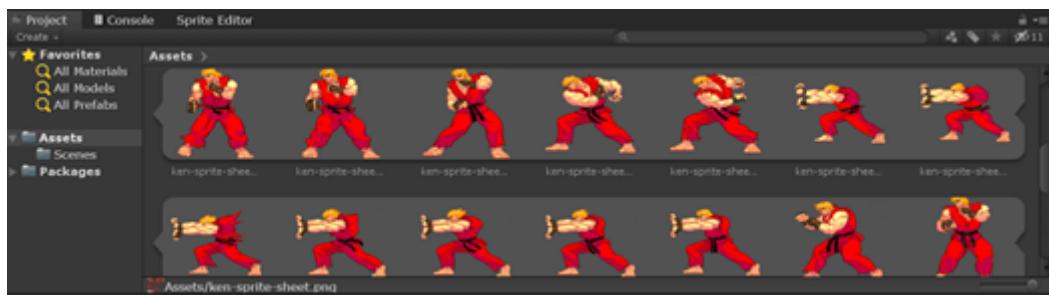
Now, click on Apply in the top right corner of the Sprite Editor window.



Select the Player object in the Hierarchy tab and then in the Inspector tab click on Add Component button. Then select Rendering -> Sprite Renderer.

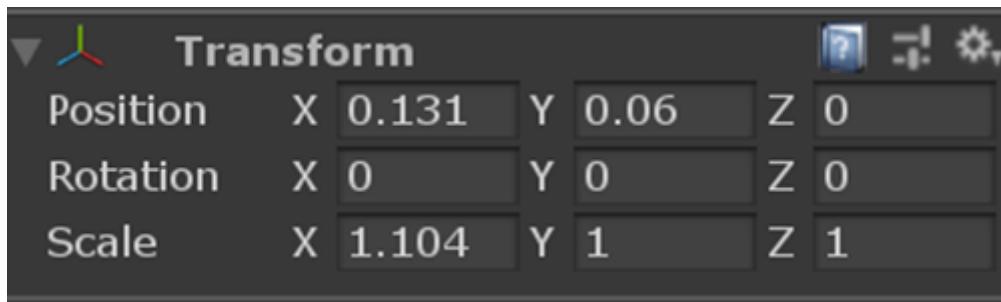


Finally, you will be able to see all of your Sprites as an individual object in the Project -> Assets Folder.



Transforms and Object Parenting

In Unity, the Transform component has three visible properties - the position, rotation, and scale. Each of these properties has three values for the three axes. Means, Transform is used to determine the Position, Rotation, and Scale of each object in the scene. Every GameObject has a Transform.



Properties

Position: This is the position of the transform in X, Y, and Z coordinates. 2D games generally do not focus on the Z-axis when it comes to positioning. The most frequent use of the Z-axis in 2D games is in the creation of parallax.

Rotation: This property defines the amount of rotation (measured in degree) an object is rotated about that axis with respect to the game world or the parent object.

Scale: The scale of the object defines how large it is when compared to its original or native size. For example, let us take a square of 2x2 dimensions. If the square is scaled against the X-axis by 3 and the Y-axis by 2 we will get a square of size 6x4.

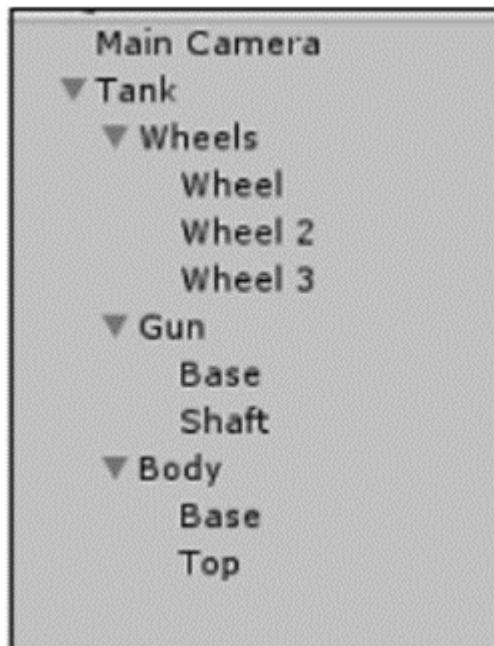
These properties are measured relative to the transform's parent. If the transform has no parent, the properties are calculated in world space.

Object Parenting

In Unity, GameObjects follow a Hierarchy system. Using this hierarchy system, GameObjects can become parents of other GameObjects. When a GameObject has a parent, it will perform all its transform changes with respect to another GameObject instead of the game world.

A parent object causes all children objects to move and rotate the same way the parent object does, although moving children objects does not have any effect on the parent. Children themselves can be parents; e.g., your hand is the child of your arm, and fingers are children of your hand.

Parenting GameObjects has a number of uses. For example, all the different parts of a tank could be separate GameObjects, parented under a single GameObject named "tank." Hence, when this "tank" parent GameObject moves, all the parts move along with it because their positioning is updated constantly according to their parent.



Internal Assets

Along with the external assets that you import from other programs such as image, audio files, 3D models, etc., Unity also offers the creation of internal assets. These assets are formed within Unity itself, and as such, do not need any external program to create or modify.

Let's see some examples of internal assets:

Scenes: These act as levels.

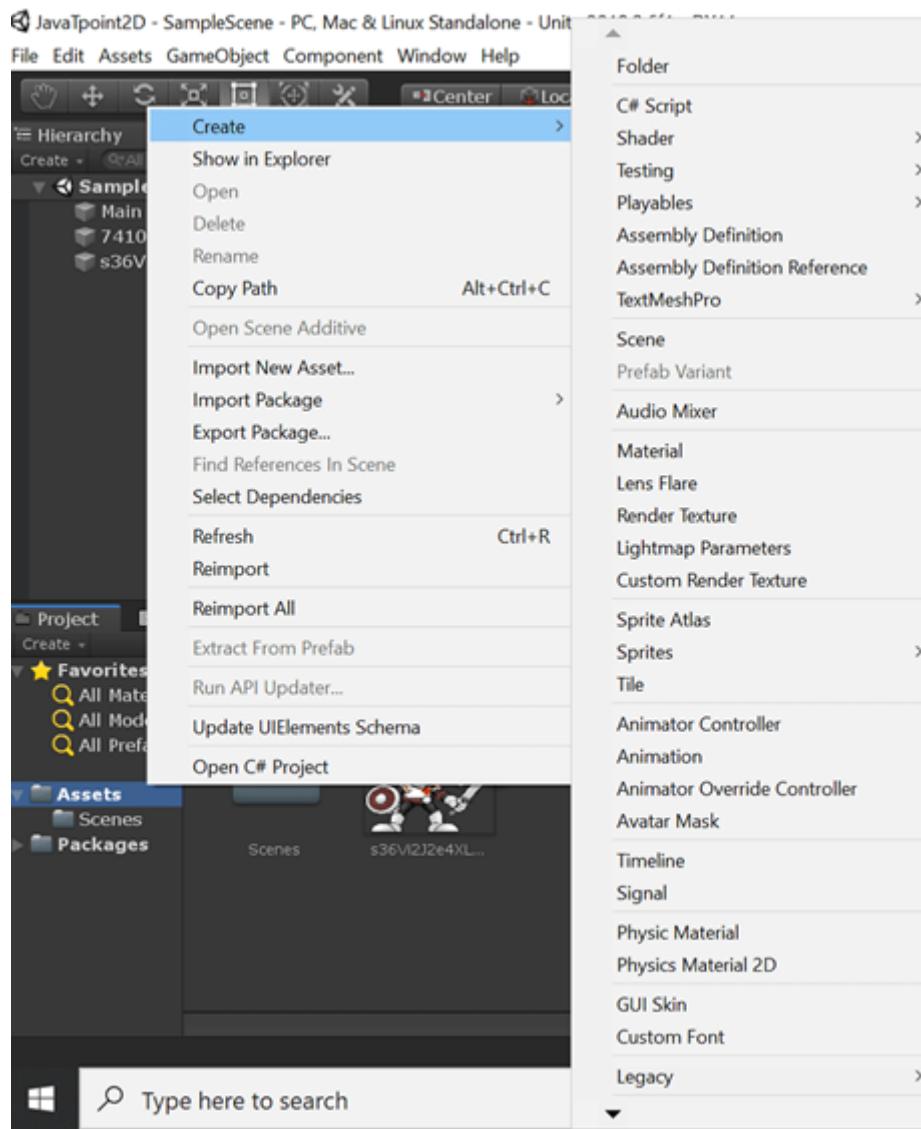
Animations: These contain data for the animation of GameObject.

Materials: These are used to define how lighting affects the appearance of an object.

Scripts: The code which will be written for the GameObjects.

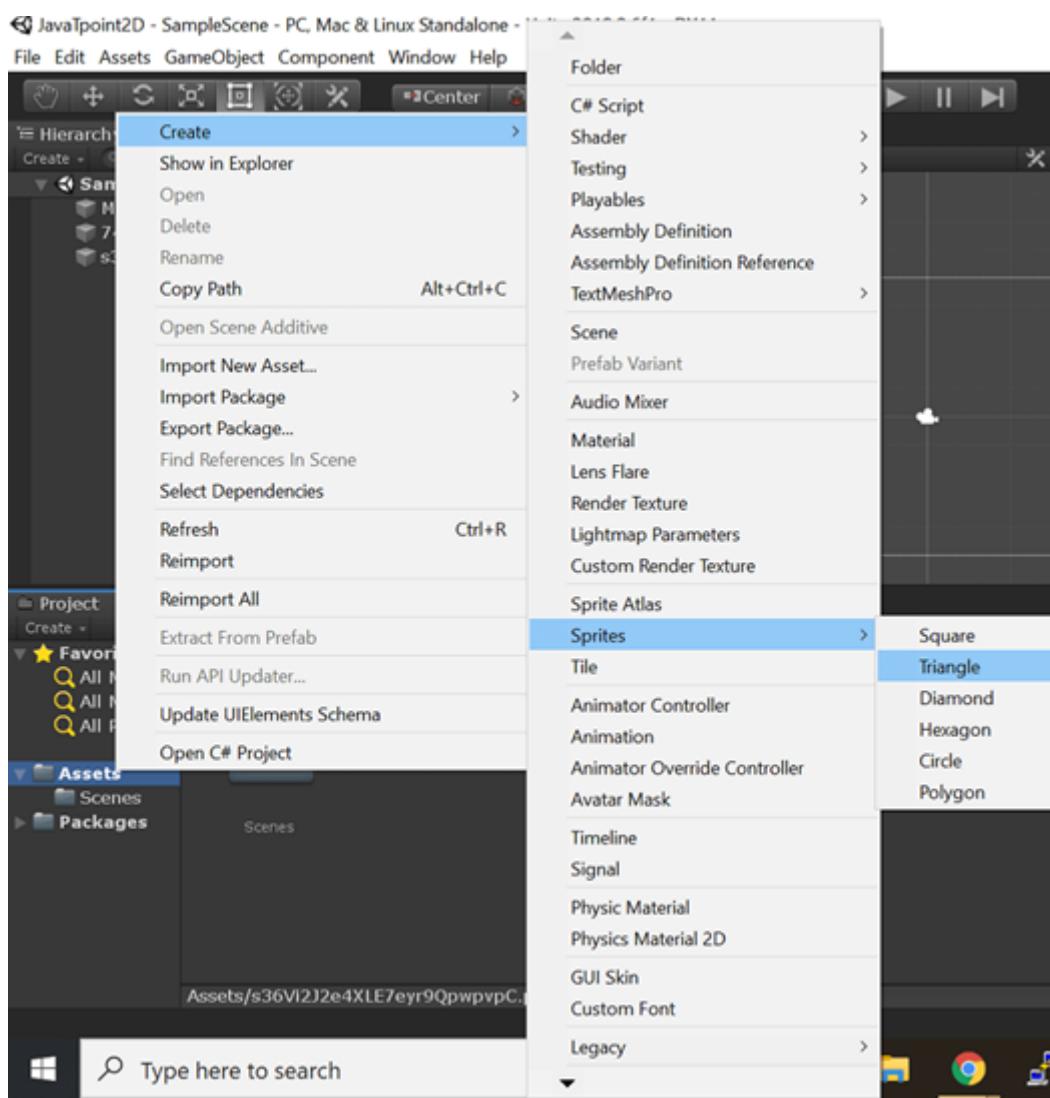
Prefabs: These act as a blueprint for GameObjects so they can be generated at runtime.

To create an internal asset, go to the Assets folder and right-click on that folder and select **Create**.

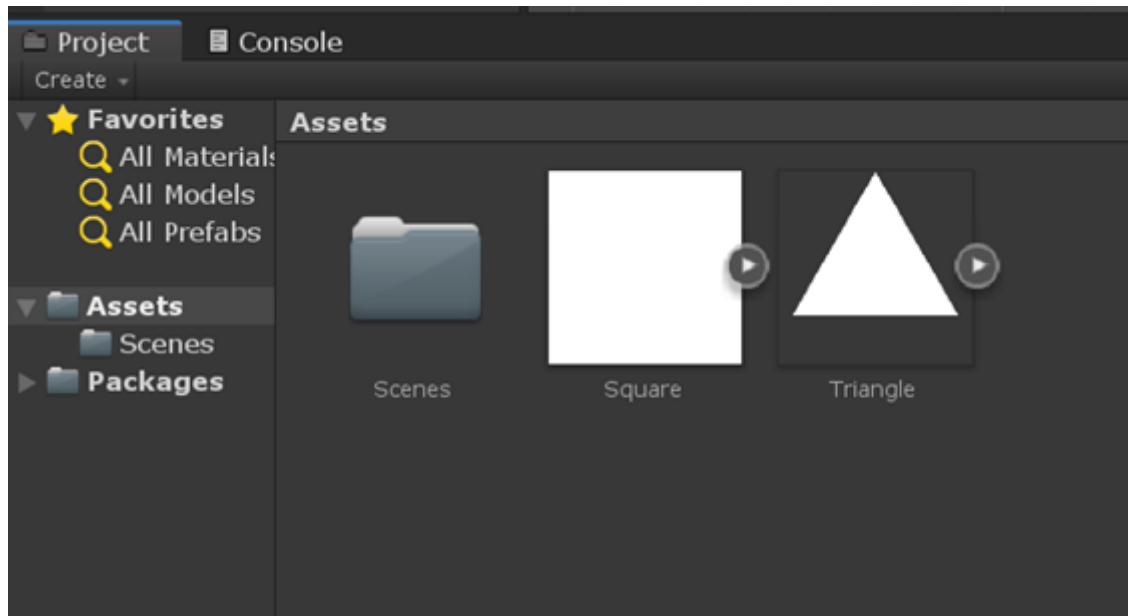


In this example, we will create a Triangle and a Square.

To create a Triangle asset, right click on the asset -> Create -> Sprites -> Triangle



Repeat the same process for Square, and now you should have two new graphic assets.



As we move along, we will explore more of these internal assets, since they are very crucial to build a proper game.

Analysis, Application, and Exploration for Week 16 ►

Navigation

[Home](#) [Dashboard](#)[Site pages](#)[My courses](#)[121 - CC106](#)[121 - BPM101 / DM103](#)[121 - OAELEC2](#)[121 - ITE3](#)[121 - MUL101](#)[Participants](#) [Grades](#)[General](#)[01 \[Enter Module Title Here\]](#)[02 \[Enter Module Title Here\]](#)[03 \[Enter Module Title Here\]](#)[04 \[Enter Module Title Here\]](#)[05 \[Enter Module Title Here\]](#)[06 - Preliminary Examination](#)[07 \[Enter Module Title Here\]](#)[08 \[Enter Module Title Here\]](#)[09 \[Enter Module Title Here\]](#)[10 \[Enter Module Title Here\]](#)[11 \[Enter Module Title Here\]](#)[12 - Midterm Examination](#)[13 \[Enter Module Title Here\]](#)[14 \[Enter Module Title Here\]](#)[15 \[Enter Module Title Here\]](#)[16 \[Enter Module Title Here\]](#) [Preliminary Activity for Week 16](#) [**Lesson Proper for Week 16**](#) [Analysis, Application, and Exploration for Week 16](#) [Generalization for Week 16](#) [Evaluation for Week 16](#) [Assignment for Week 16](#)[17 \[Enter Module Title Here\]](#)[121 - ITSP2B](#)[121 - WEB101 / CCS3218](#)

Fair Warning

NOTICE: Please be reminded that it has come to the attention of the Publishing Team of eLearning Commons that learning materials published and intended for ***free use only by students and faculty members within the eLearning Commons network were UNLAWFULLY uploaded in other sites without due and proper permission.***

PROSECUTION: Under Philippine law (Republic Act No. 8293), copyright infringement is punishable by the following: Imprisonment of between 1 to 3 years and a fine of between 50,000 to 150,000 pesos for the first offense. Imprisonment of 3 years and 1 day to six years plus a fine of between 150,000 to 500,000 pesos for the second offense.

COURSE OF ACTION: Whoever has maliciously uploaded these concerned materials are hereby given an ultimatum to take it down within 24-hours. Beyond the 24-hour grace period, our Legal Department shall initiate the proceedings in coordination with the National Bureau of Investigation for IP Address tracking, account owner identification, and filing of cases for prosecution.

2nd Semester Enrollment



The advertisement features a blue-toned background image of a modern building with a glass facade and a tall antenna tower against a cloudy sky. In the upper right corner, the letters "BCP" are displayed in large green block letters. Below them, the text "visit www.bcp.edu.ph" is written in blue. The main text in the center reads "Enrollment registration is now Ongoing" in large red letters, followed by "For 2nd Semester SY 2021 - 2022" in blue, and "We are accepting new students, returnees and transferees." in blue. At the bottom left, there is an envelope icon followed by the email address "bcp-inquire@bcp.edu.ph". At the bottom right, there is a phone receiver icon followed by the phone numbers "(8)442-8601 | (8)518-8050". A small circular logo for "BCP" is located in the bottom right corner of the advertisement area. The entire advertisement is framed by a thin blue border.

Activities

-  Assignments
 -  Forums
 -  Quizzes
 -  Resources
-

Bestlink College of the Philippines
College Department

Powered by [eLearning Commons](#)