



Diving Into Unity

Today's Agenda

- Understand the essential views used in Unity Editor.
- Learn how to move, rotate, and scale objects within Unity.
- Modify component values and create new Game Objects.
- Familiarize yourself with the core views provided by the Unity Engine.

Video

Alongside the slide, you can watch the accompanying video, which covers the same topics in a less detailed manner.

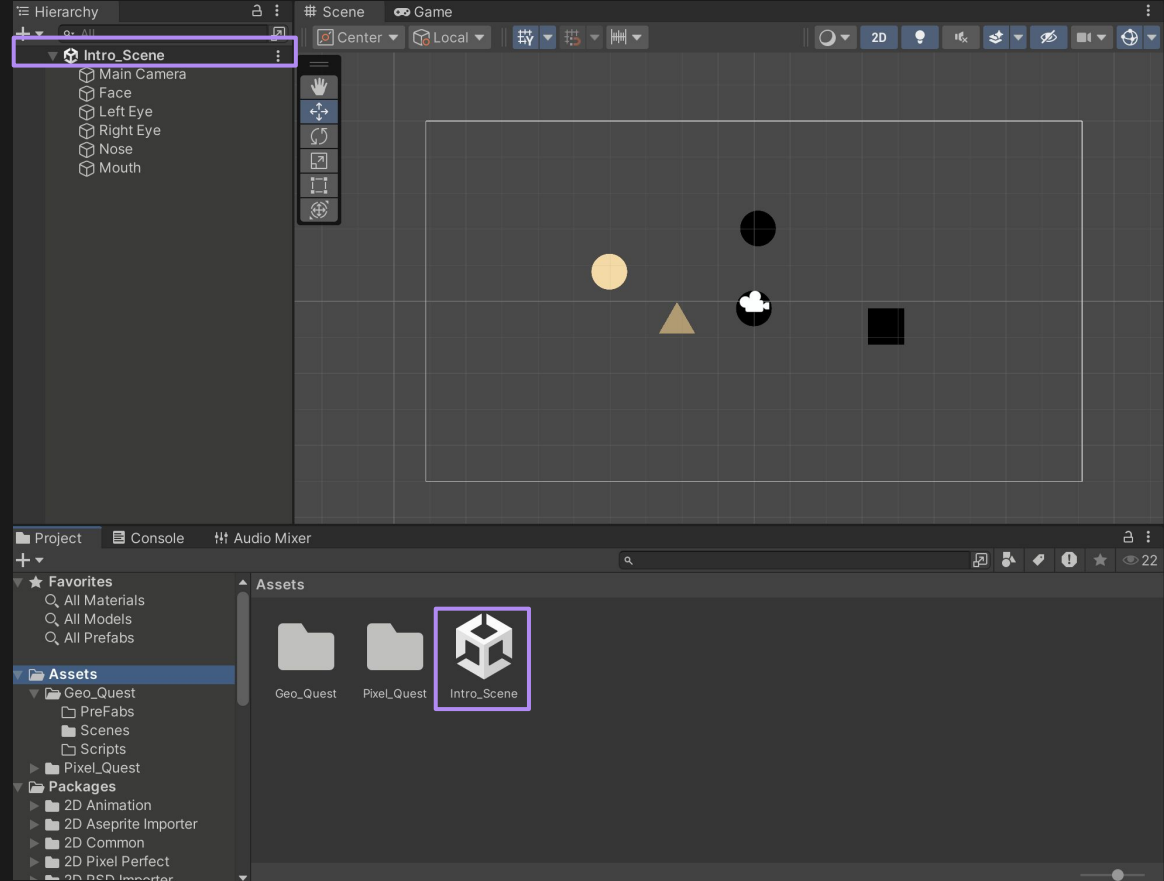


Intro Scene

Before making any changes, ensure you are in the correct workspace.

When Unity first opens, it may load an empty untitled scene. To begin, navigate to the Assets folder and double-click on Intro_Scene.

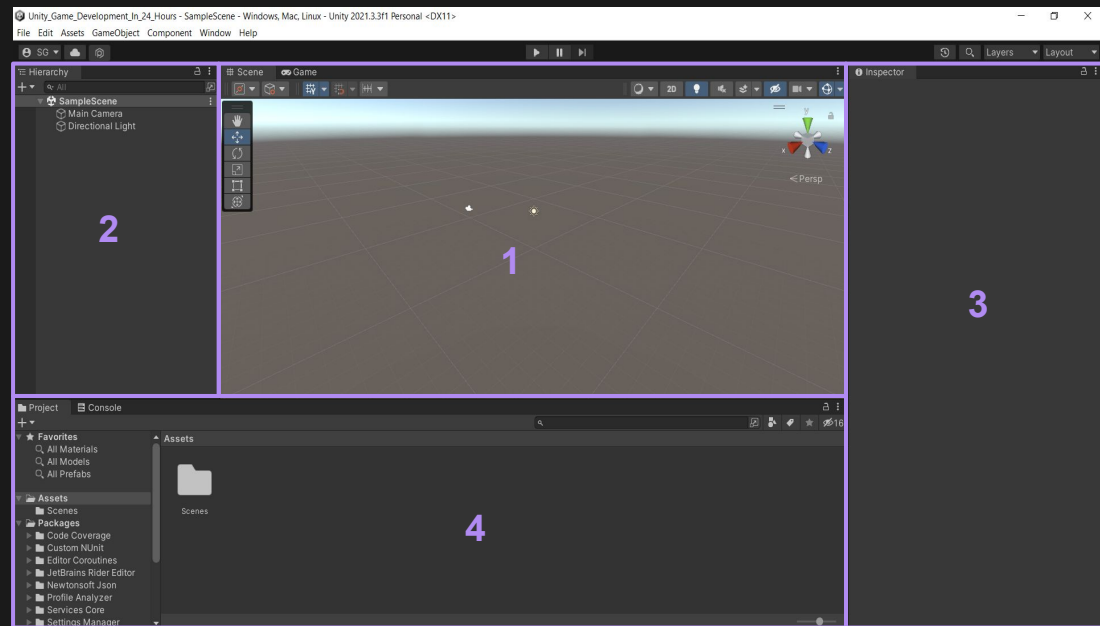
This is a Scene Game Asset, which functions as a level in Unity. Double-click to open it.



The Unity View

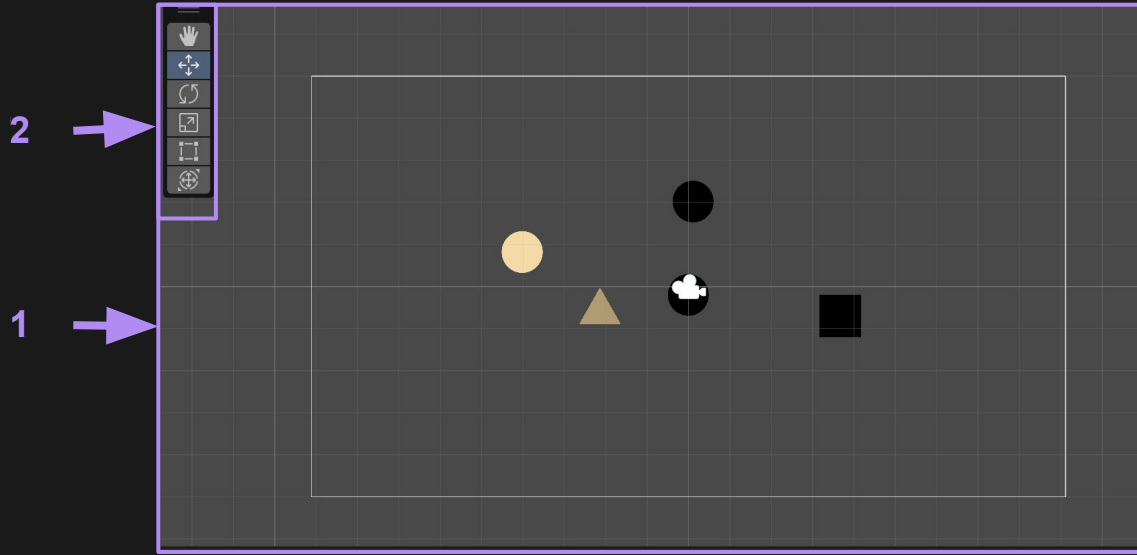
Unity's interface consists of multiple essential views that aid in game development. The core views we will focus on are:

1. **Scene View** – Provides a visual representation of the game world.
1. **Game View** – Displays how the game will look when played.
2. **Hierarchy** – Lists all objects present in the scene.
3. **Inspector** – Allows you to edit properties of selected objects.
4. **Project View** – Organizes all assets used in the project.
4. **Console** – Displays errors, warnings, and debug messages.



Scene View + Navigation & Game Object Manipulation

Scene View



The **Scene View** allows you to navigate through the scene or level and edit the **position, rotation, and scale** of Game Objects using the **Transform Tools Bar**.

The **Transform Tools Bar** provides a set of tools to manipulate Game Objects within the **Scene View**, enabling precise adjustments to their transformation properties.

Scene View

The **Transform Tools Bar** contains various tools that assist in scene navigation and Game Object manipulation. These tools include the **Hand Tool**, **Move Tool**, **Rotate Tool**, **Scale Tool**, and **Rect Tool**, each serving a specific function in transforming objects within the **Scene View**.

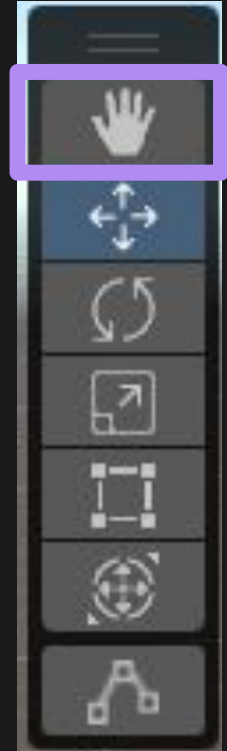


Hand Tool/Scene Navigation

When the **Hand Tool** is selected, you can freely explore your scene using various navigation controls:

- **Left Mouse Button** – Moves along the current 2D plane (left, right, up, and down).
- **Right Mouse Button** – Activates **Fly Mode**, allowing movement with the **WASD** keys.
- **Left Alt + Left Mouse Button** – Locks your position and enables rotation around a fixed point.
- **Left Alt + Right Mouse Button** – Zooms in or out.
- **Mouse Wheel** – Zooms in or out at any time.

These controls allow for flexible scene navigation, making it easier to position and adjust objects efficiently.

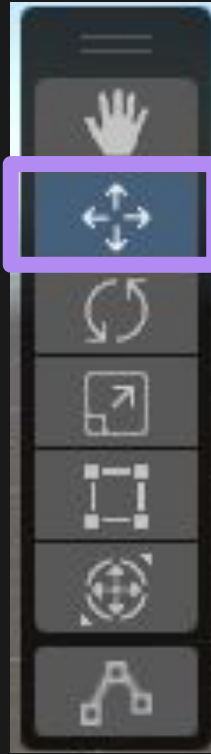
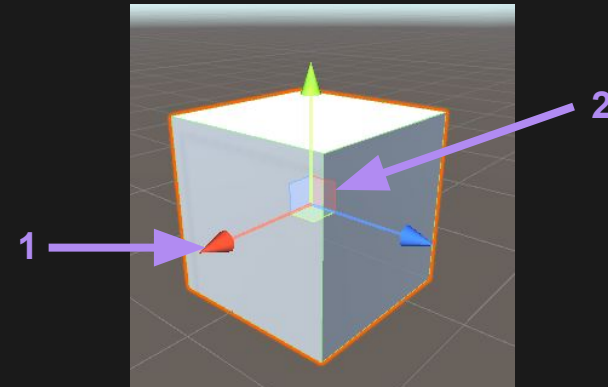
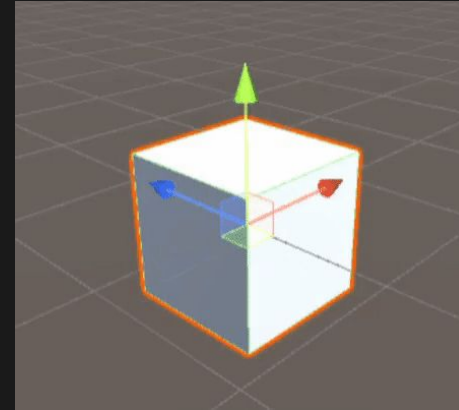


Transform Tools - Move Tool

The **Move Tool** allows you to adjust the position of a selected Game Object. The currently selected object will display **move handles**, indicating that it is active for transformation.

- **Arrows** – Move the object along individual axes: **X (red)**, **Y (green)**, and **Z (blue)**, matching the Scene Gizmo colors.
- **Planes** – Enable movement along two axes at a time: **XY (blue)**, **XZ (green)**, and **YZ (red)**, allowing for constrained adjustments.

Using these handles, you can precisely position objects within the scene.



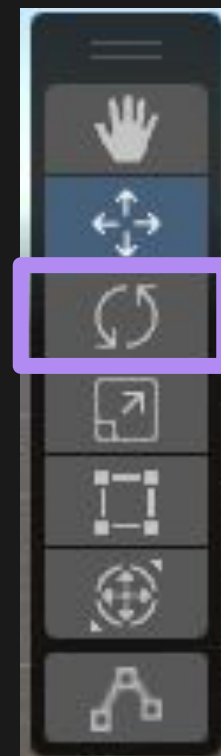
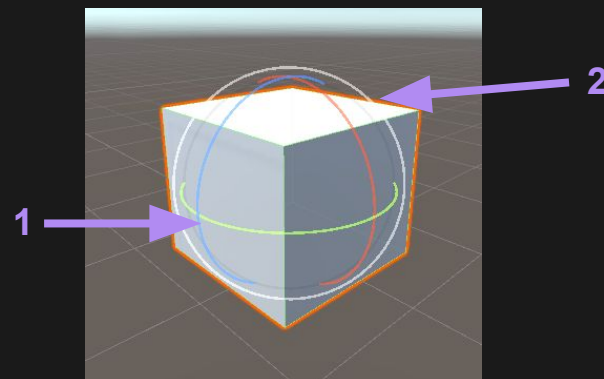
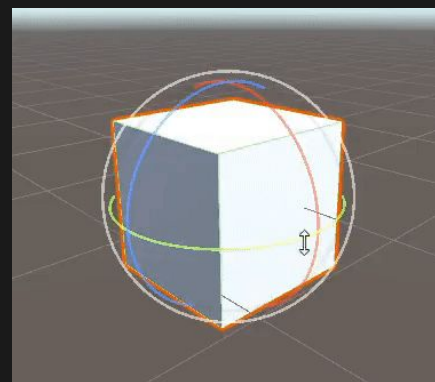
Transform Tools - Rotate Tool



The **Rotation Tool** allows you to adjust the orientation of a selected Game Object.

- **Colored Rings (Red, Green, Blue)** – Control rotation along the **X, Y, and Z** axes respectively.
- **Yellow Arc** – Displays the degree of rotation from the object's original position.
- **White Ring** – Rotates the object around the **Scene View's Z-axis** for free rotation.

This tool provides precise control over object orientation, helping to align and position elements accurately in the scene.



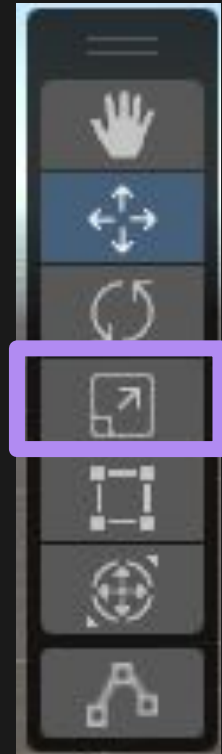
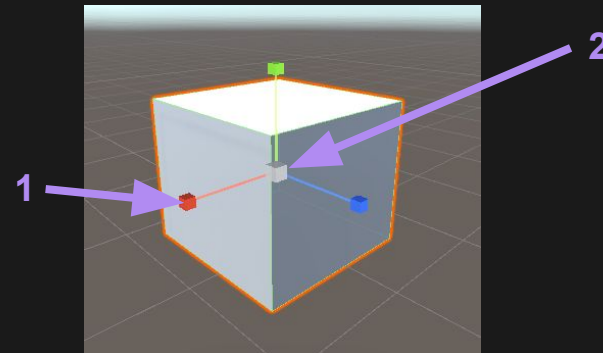
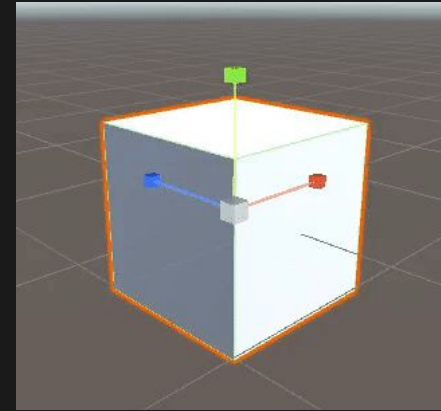
Transform Tools - Scale Tool



The **Scale Tool** allows you to modify the size of a selected Game Object along each axis.

- **Arrows (Red, Green, Blue)** – Scale the object individually along the **X, Y, and Z** axes.
- **Middle Notch** – Scales the object uniformly across all three axes at the same rate.

With these controls, you can easily adjust the size of objects in the scene, either along specific axes or uniformly.



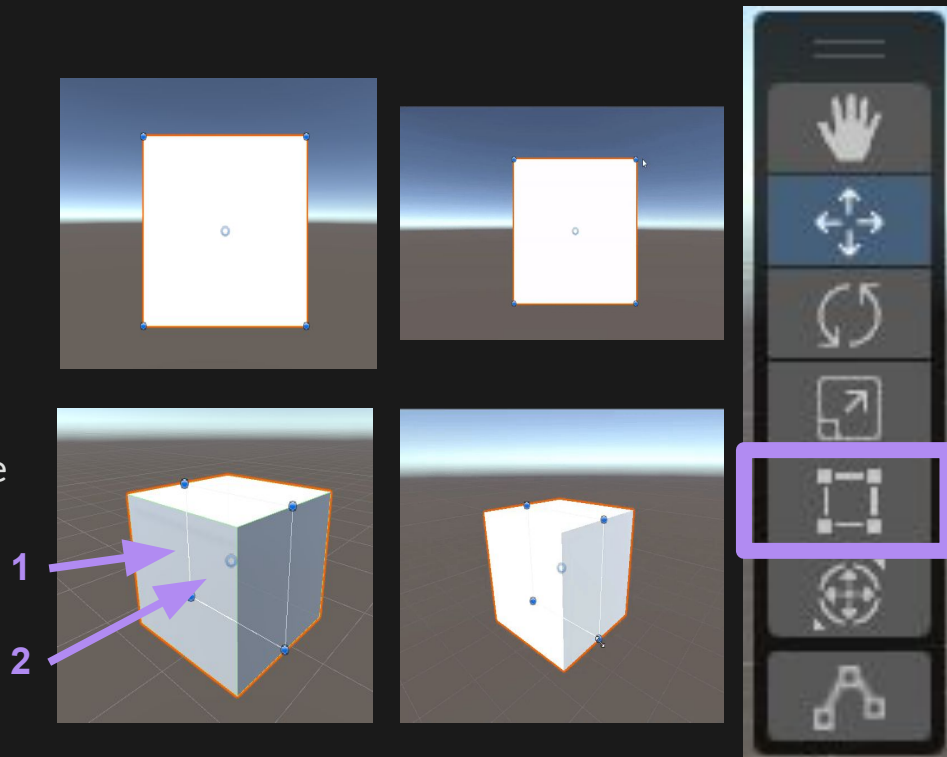
Transform Tools - Rect Tool



The **Rect Tool** combines the functionality of the **Move**, **Rotate**, and **Scale** tools into one, allowing you to translate, rotate, and scale an object simultaneously.

- **Plane Restriction** – The **Rect Tool** operates on a 2D plane, meaning it modifies the object only along the **XY**, **XZ**, or **ZY** planes.
- **Pivot Point** – The **Pivot Point** is the center of the object and the point around which the object rotates when transformations are applied.

This tool is particularly useful for 2D scene manipulation and UI element adjustments.



Tool Hotkeys

All of these functions have **hotkeys** assigned for quick access. If you'd prefer to switch between settings without navigating through the menu, you can use the hotkeys for faster workflow.

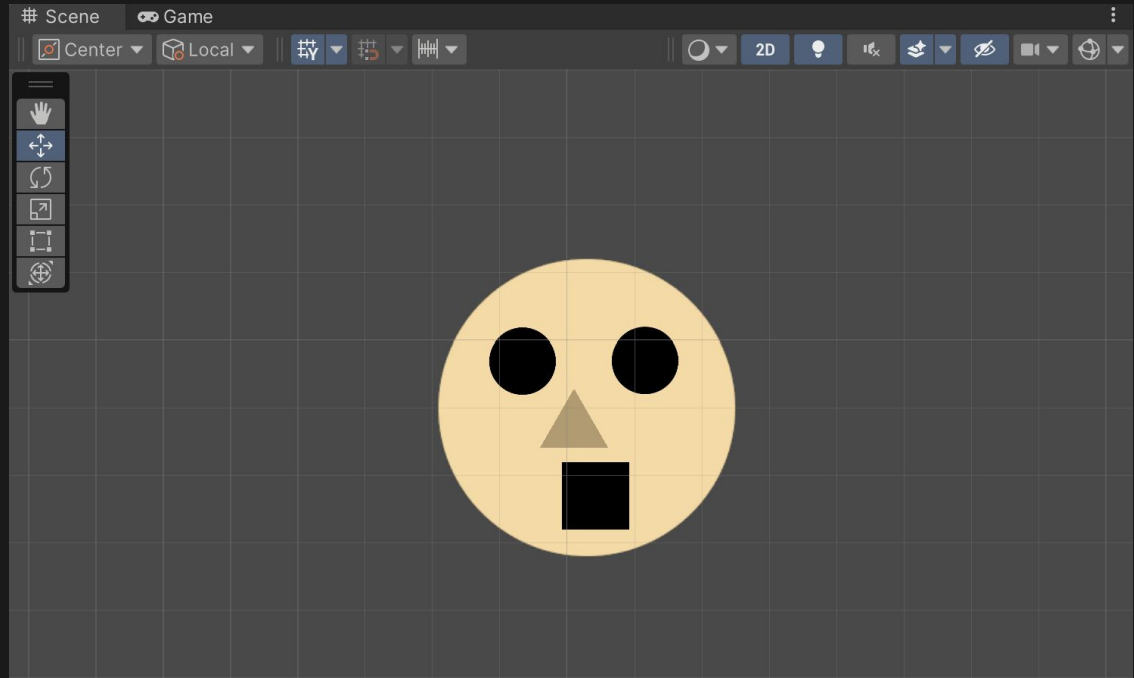
Tool	Hotkey	Function
Hand	Q	Navigate in the Scene
Move	W	Translate selected object
Rotate	E	Rotate selected object
Scale	R	Resize selected object
Rect	T	Manipulate 2D object

Challenge: Make a Face

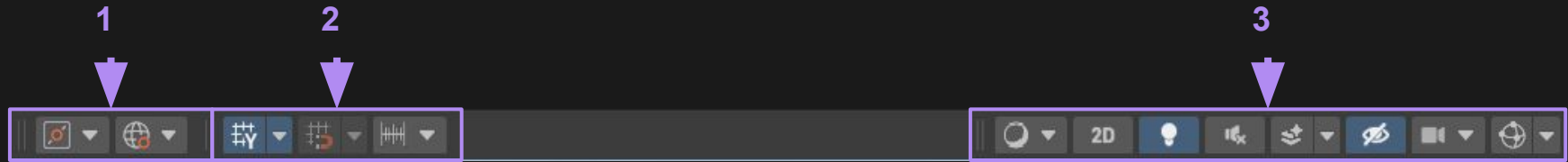
Now, let's put your skills to the test!

Using the Game Objects in this scene, manipulate them to create a simple face.

Try moving, rotating, and scaling the objects to form eyes, a nose, and a mouth. This will help you practice using the **Move Tool**, **Rotate Tool**, and **Scale Tool** effectively.



Scene View Control Bar



The **Scene View Control Bar** is divided into three tool sets that help manipulate objects and adjust how the scene is displayed:

1. **Tool Settings** – Modifies how the tools behave and interact with objects.
2. **Grid and Snap** – Controls the grid system, allowing for precise placement and movement of objects.
3. **View Options** – Adjusts how the scene is shown, offering different perspectives and display settings.

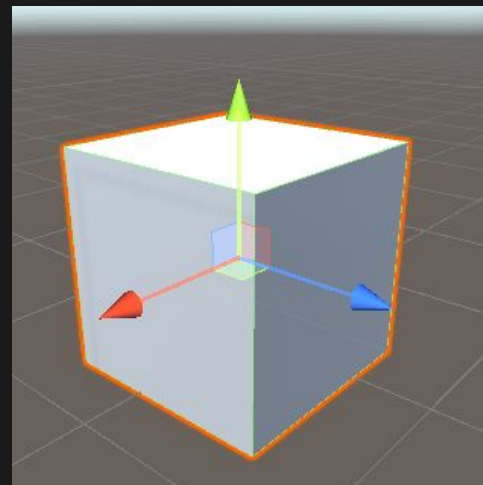
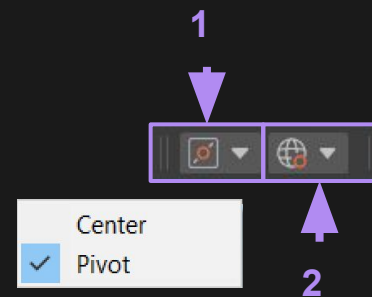
These options give you greater control over both your object manipulation and scene visualization.

Tool Settings: Pivot

The **Handle Position Toggle** lets you choose where you want the **Transform Gizmo** to appear on a Game Object. By default, it shows up at the center of the object, which acts as its origin point.

However, if you import an asset from a 3D modeling software, it may have a pivot point that's different from its center.

Unity allows you to choose either the center or the pivot point as your preferred origin for transformations.

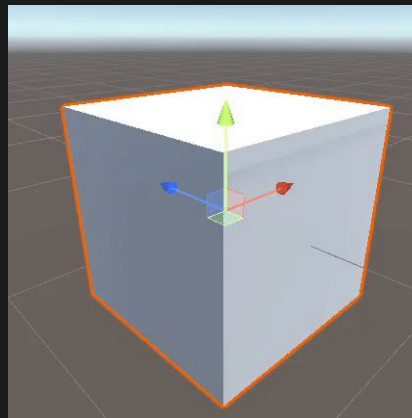
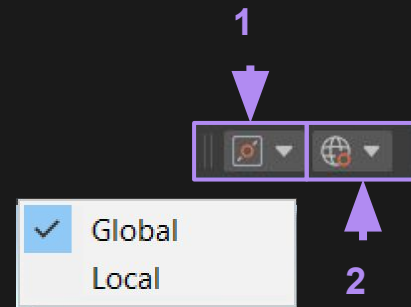


Tool Settings: World vs Local Coordinate

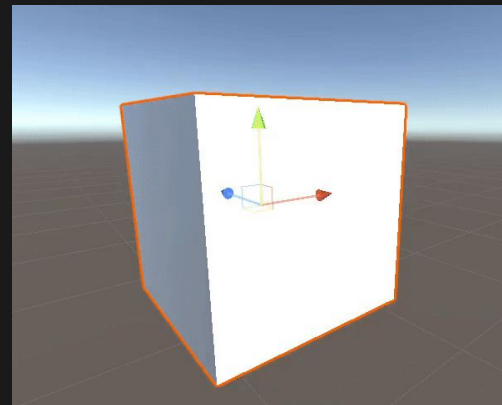
As we've been transforming our Game Objects, they've been referencing their **Origin Point** (0,0,0), meaning they're aligned with the **World Coordinate System**.

Each object also has its own **Local Coordinate System**, which it uses to track its position and rotation. This is especially useful for character control.

Instead of adjusting movement relative to the world's axes, you can simply program the character to move "forward" along its own local axes, allowing for more intuitive movement controls, regardless of the object's rotation in the world.



World Coordinates



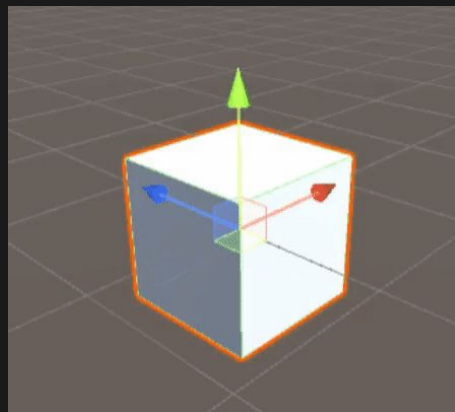
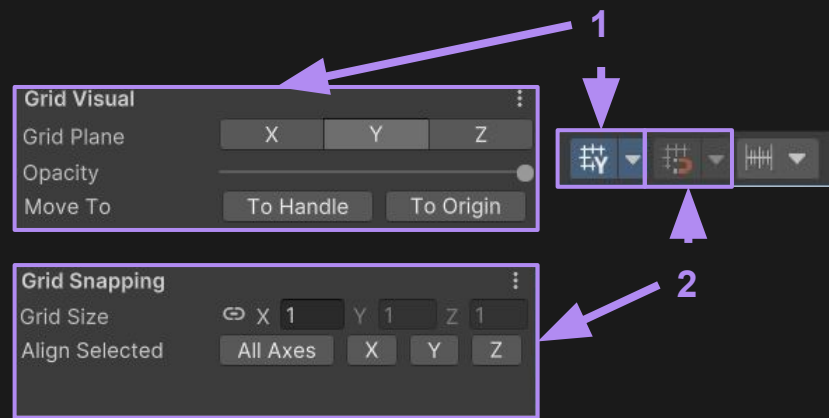
Local Coordinates

Grid and Snap

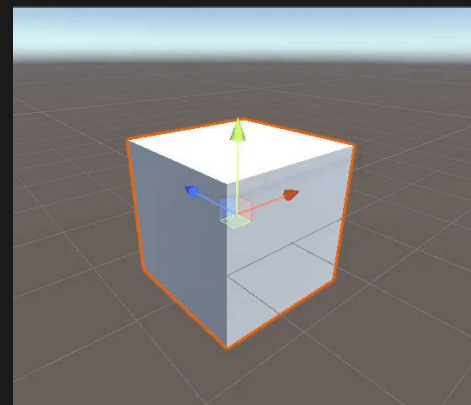
The **Grid Visual** allows you to toggle the grid on or off, select the axis on which you'd like it to appear, and adjust its opacity (transparency).

Grid Snapping enables you to snap the **Move Tool** to specific points along the grid. You can adjust the distance between each point for more precise movement.

Grid snapping is only available when translating Game Objects in **World Coordinates**, but it's a great way to ensure more accurate and consistent placement of items in your scene.



Snapping Off



Snapping On

View Options



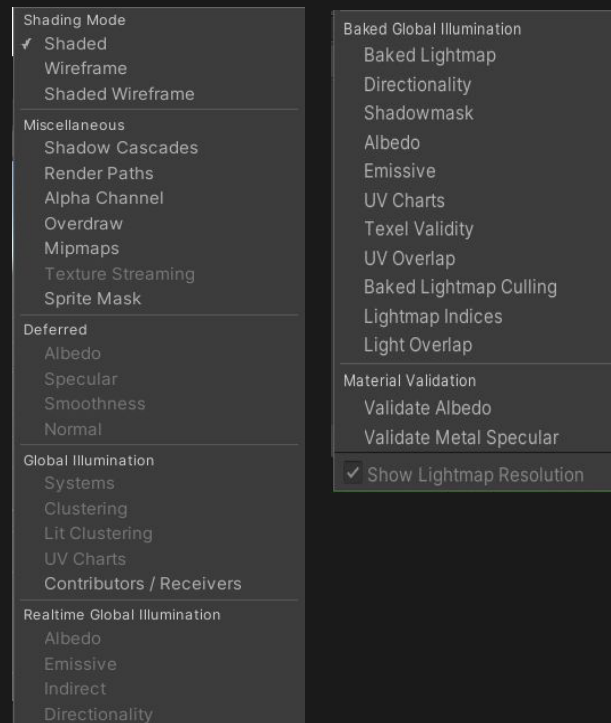
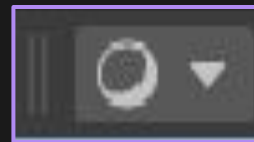
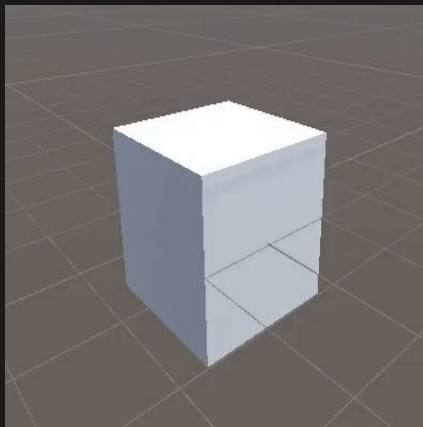
The **View Options** provide full control over how and what is displayed in the scene.

1. **Draw Mode** – Selects how the objects are rendered in the scene.
2. **Light, Audio, and Other Visual Effects Toggles** – Controls how lighting, audio, and other effects behave in the scene.
3. **2D Toggle** – Switches between **3D** space and **2D** space, allowing you to work with 2D elements more easily.
4. **Gizmo Visibility Toggle** – Controls whether or not the icons and visual cues (such as the **Transform Gizmo**) are displayed in the scene.

These options help fine-tune your scene's appearance and make navigation and object manipulation easier.

View Options - Draw Mode

Draw Mode allows you to switch between different rendering modes for the objects in the scene. For the most part, we'll stay in the default Shaded mode, as the other options are typically used for advanced lighting and rendering techniques.



Additional Resources: [Scene View Draw Modes For Lighting](#)

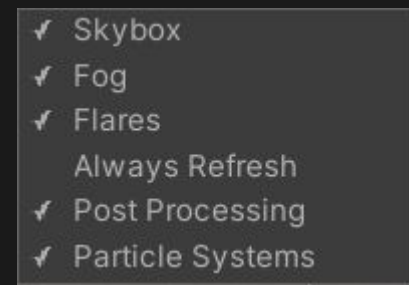
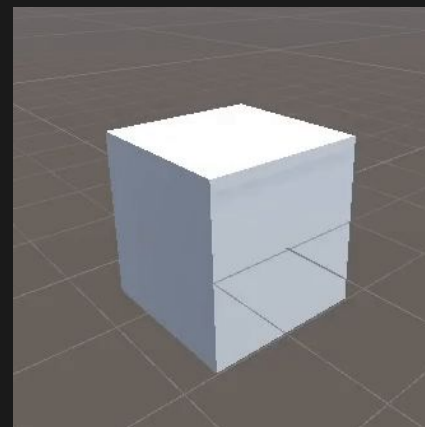
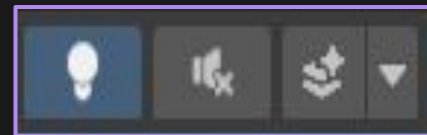
View Options - Lighting, Audio

These three toggles are straightforward:

- **Lighting Effects**
- **Audio Effects**
- **Visual Effects**

You can toggle them on or off to see how the Game Objects look when unaffected by these environmental factors.

This can be helpful for focusing on the objects themselves without the distractions of lighting or other scene effects.

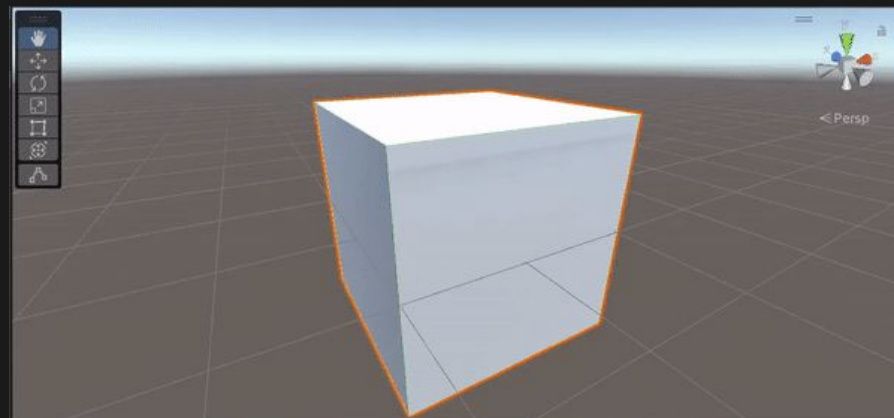
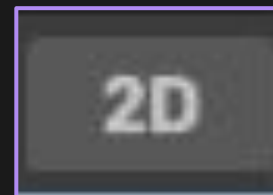


Turning Off/On Lighting
and Visual Effect

View Options - 2D Mode

The **2D On/Off Toggle** flattens your view against the **2D XY plane** when enabled. This removes the ability to use the **Scene Gizmo** for snapping or switching to **Isometric View**.

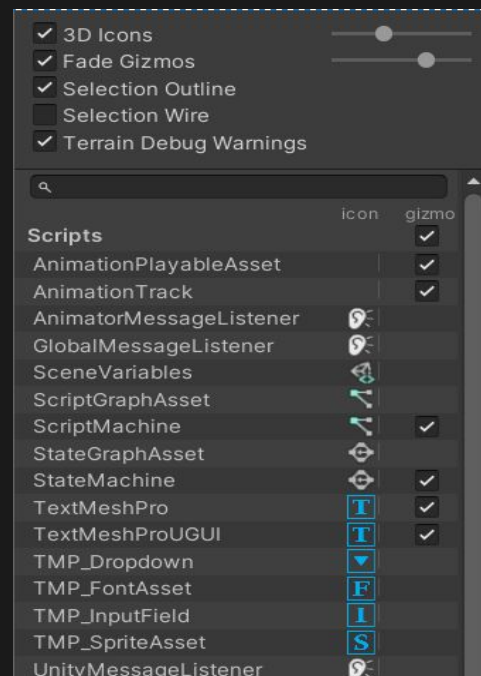
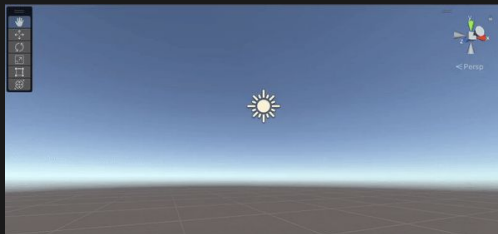
You'll typically use this option when creating a **2D game** or working on **UI elements**, as it simplifies the workspace by restricting movement to a 2D plane.



View Options - Gizmo Toggle

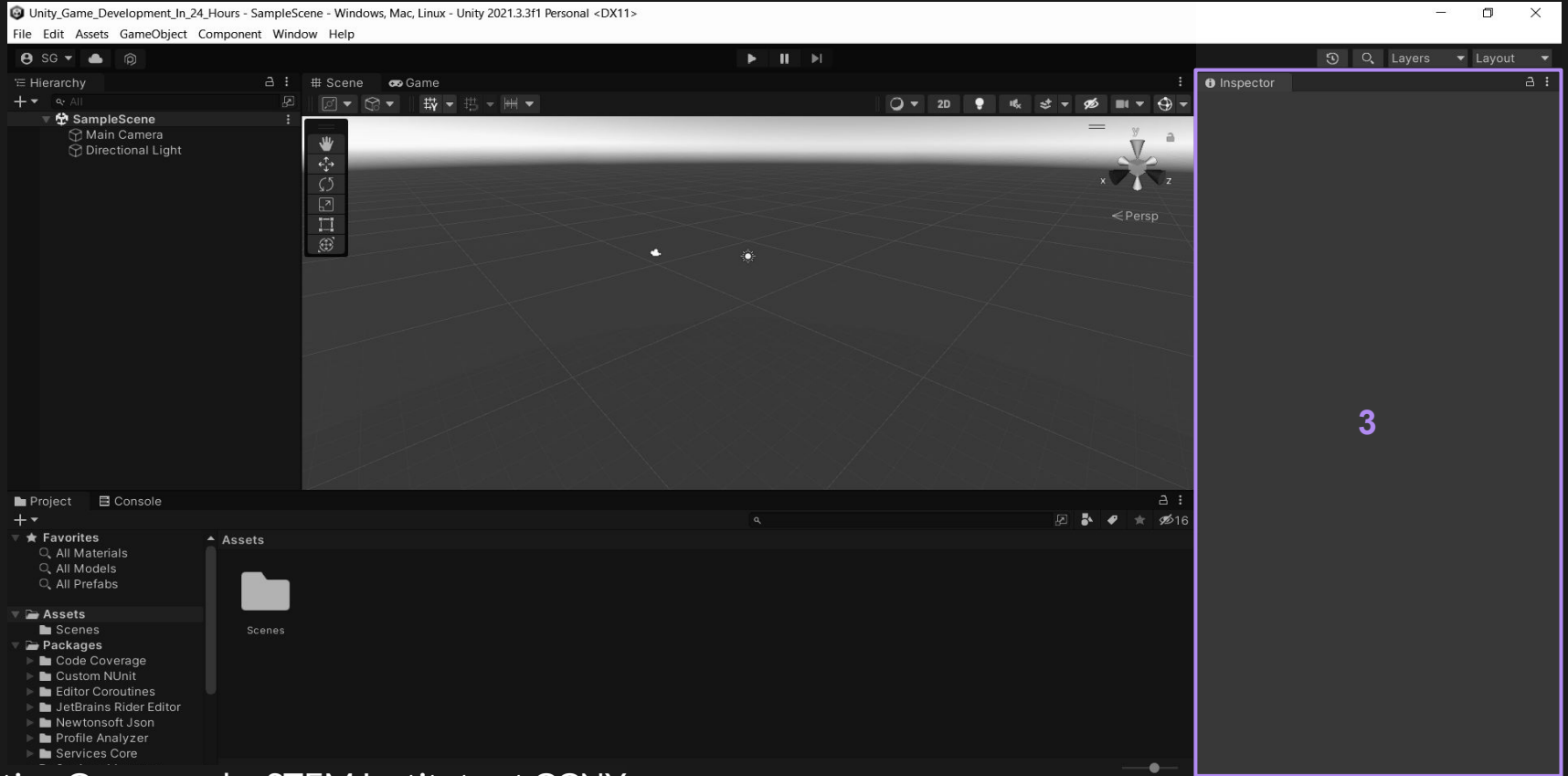
The **Gizmo Toggle** allows you to turn on or off the visual icons for Game Objects that may not have a direct visual presence in the scene. For example, you've already seen the **camera** and **light source**, both of which display icons that indicate their location.

There are many other icons that could appear on the screen, and you have full control over which ones are visible, helping you keep the scene organized and focused on what's important.



Inspector + Components

Inspector

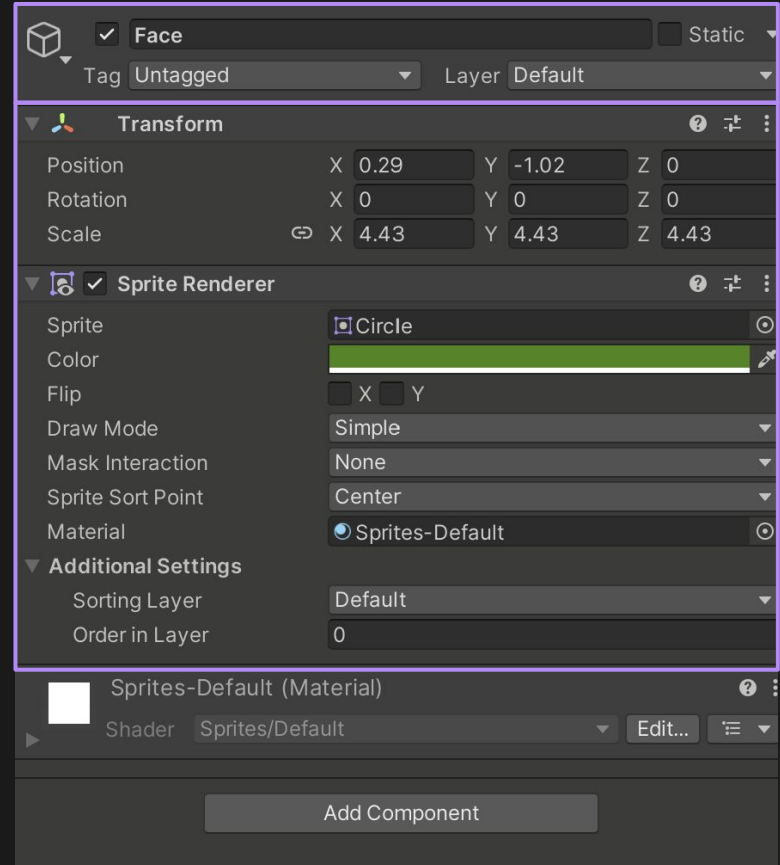


Inspector View

The **Inspector View** shows all of the properties of the selected object. In this instance, it's a **Sprite Renderer**.

1. **Inspector View Overhead** – Provides a summary of the selected object's details.
2. **Game Object's Component List** – Displays a list of all components attached to the Game Object, such as the **Sprite Renderer**, and other components that control the object's appearance and behavior.

This view lets you modify the properties of the object, such as changing the sprite or adjusting other settings within the **Sprite Renderer** component.



1

2

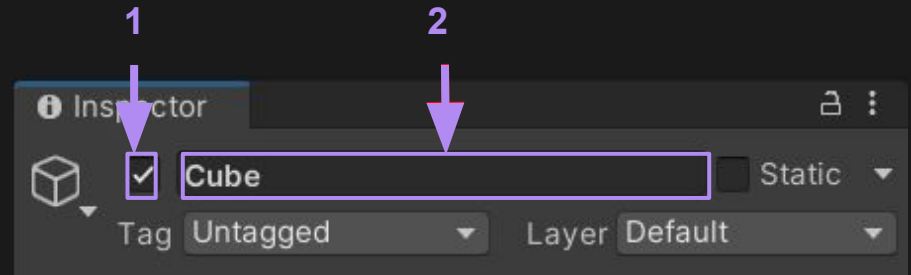
Inspector View Header

Enable – The checkmark indicates that the Game Object is **enabled**, meaning it's active and will appear in the scene and interact with the level.

If the toggle is unchecked, the Game Object will be **disabled**, which means it won't appear in the scene or interact with other elements.

Name – You can edit the name of the Game Object directly from the **Inspector View**, and it will instantly update in the **Hierarchy View**.

It's a good practice to keep the names simple and descriptive so that you can easily identify them later when editing the scene. Clear naming helps keep your project organized and improves workflow.

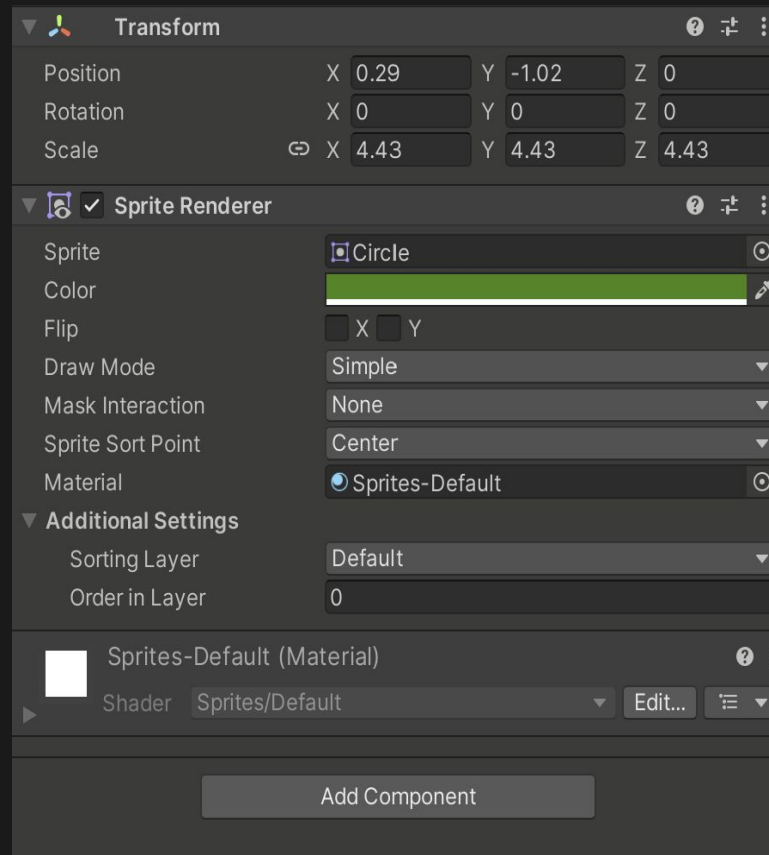


Components List

The **Component List** shows everything that's attached to the Game Object.

In this example, the object has two components: **Transform** and **Sprite Renderer**. You can identify components by their colored headers, each with a small arrow next to the name.

Clicking the arrow allows you to collapse the component if you don't need to access its properties at the moment, helping keep the view clean and organized.

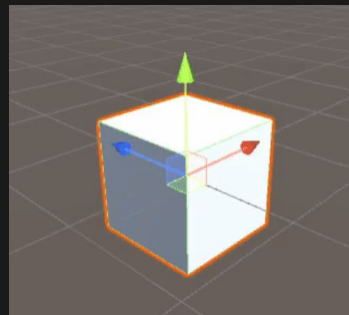
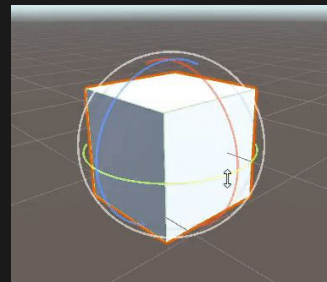
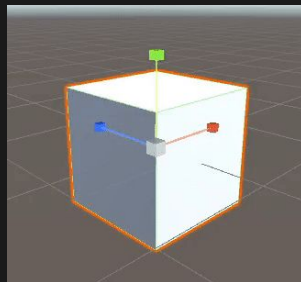
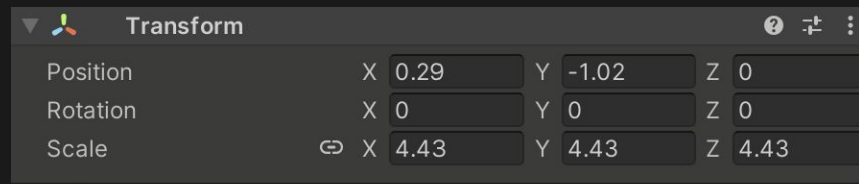


Components List

Transform – Every Game Object comes with a **Transform** component, which defines its position, rotation, and scale in the scene.

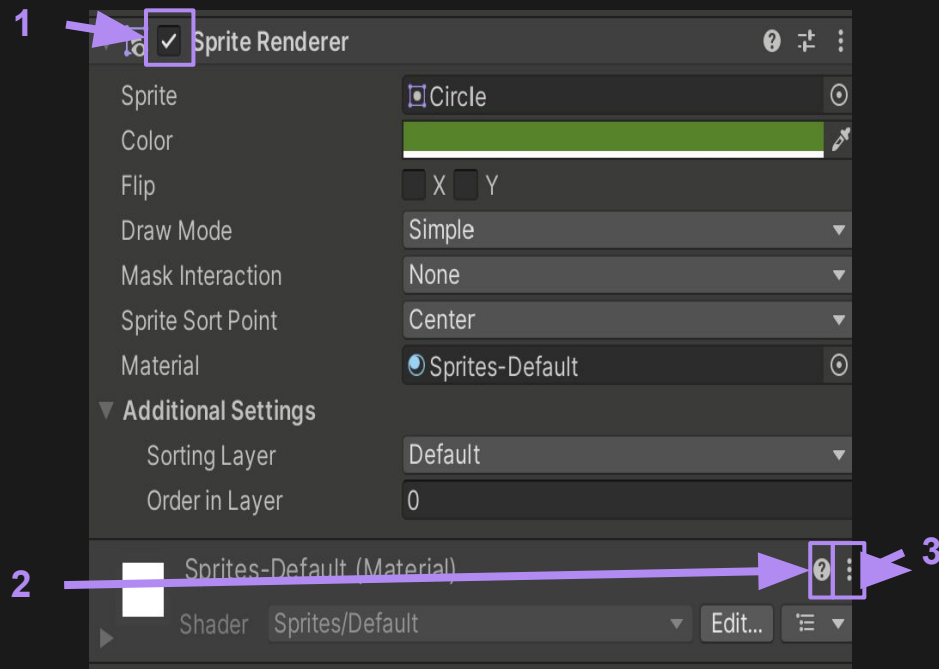
Even when working on **2D games**, the **Z-axis** will still be available to edit, as every transform is inherently 3D.

This gives you full control over the Game Object's placement in the scene, regardless of the game's perspective.



Components List

1. **Enable** – Some components, like the Sprite Renderer, also have an Enable toggle, similar to the one on the Game Object itself.
2. **Information Link** – Clicking this link will take you directly to the web page for the Unity Manual related to the selected component, providing detailed information and documentation.
3. **Component Properties** – In this section, you can:
 - Reset the component to its default state.
 - Move the component up or down the list (also possible by dragging the component header).
 - Copy and Paste values between components.
 - Delete the component from the Game Object.



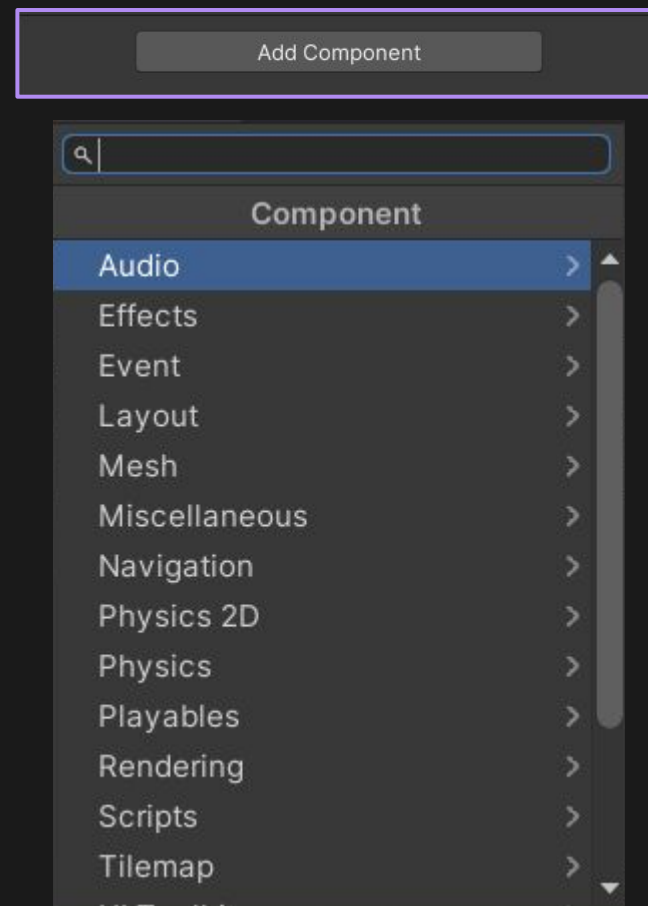
Components List

Add Component – This will be a frequently used feature. It allows you to add any **pre-built Unity components** or custom ones, like your own **Scripts**.

You can also add components by dragging them from the **Project View** directly into the **Component List**, or by using the **drop-down menu** under the **Component** option in the **File Menu Header**.

As you can see, there are many categories of components. We'll cover a good number of them throughout the course, but not all.

If you want to learn more about any components we don't cover, feel free to use the **Information Link** to check out the **Unity Manual** for more details.



Sprite Renderer Component

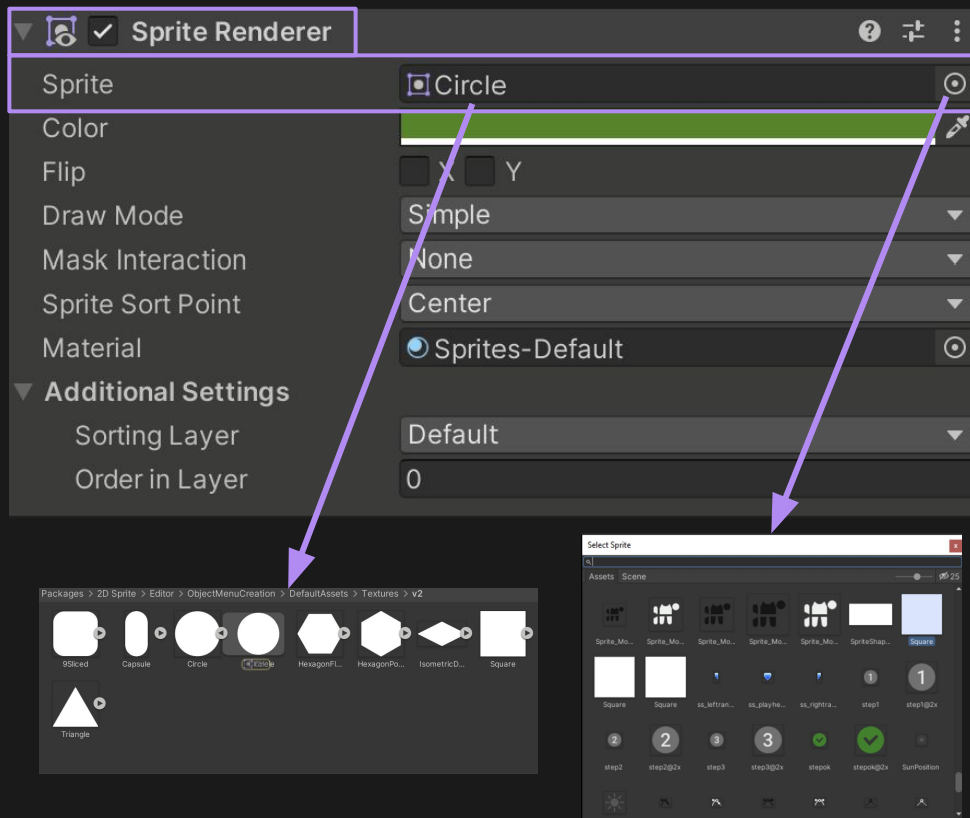
Now, let's take a look at a component we'll be using a lot: the **Sprite Renderer**. It has several options, but for now, we'll focus on four of them.

1. **Enable/Disable Toggle** – This option allows you to turn the sprite on or off in the scene. If disabled, the sprite won't be visible, but it can still exist in the scene.
2. **Sprite** – This is where you connect the image you want to display. Currently, it's showing the **Circle** image.

You can change the image in a few ways:

- Drag an image from the **Project View** into the **Sprite** slot.
- Click the **circle icon** next to the slot to see all the images in your project and select one from there.
- You can also locate the image directly in your **Project View** by clicking on the name of the image in the **Sprite** slot.

These options give you flexibility in choosing and updating the sprite displayed by your Game Object.

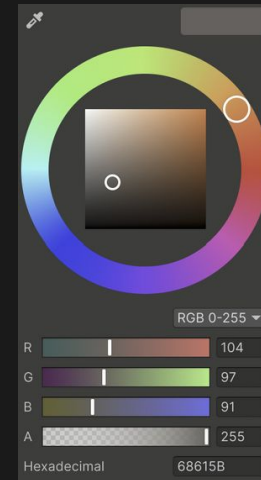
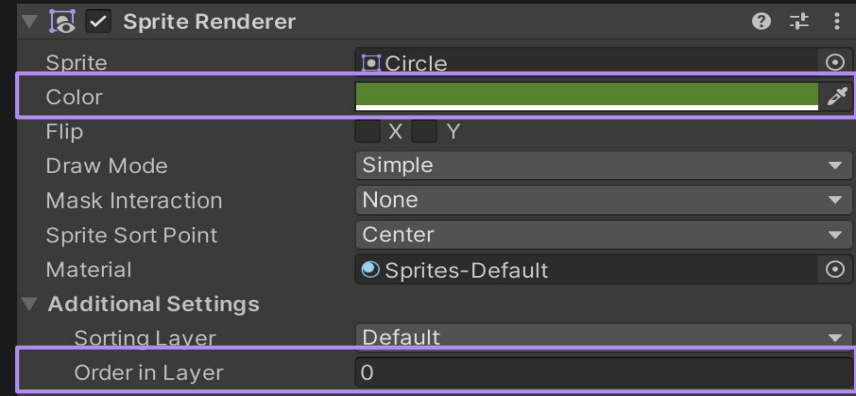


Sprite Renderer Component - Color

Color – This option allows you to change the color of the sprite. Clicking on the color box opens the color wheel. The circle lets you adjust the hue (choose the color), and the square in the middle adjusts the saturation and brightness. The color is broken down into four values:

- R: Red
- G: Green
- B: Blue
- A: Alpha (opacity)

The first three values determine the intensity of each color, while the Alpha value controls the transparency. A value of 0 means completely transparent, and a value of 255 means fully opaque.

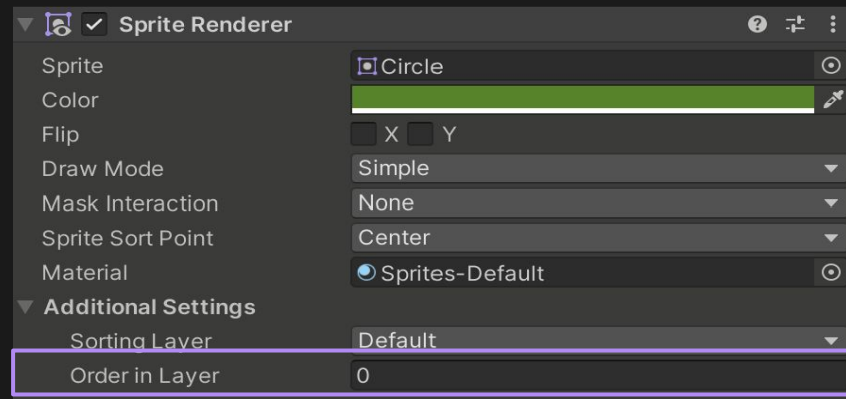


Sprite Renderer Component - Order in Layer

Order in Layer – This setting is especially important for 2D games because, unlike in 3D games, there's no natural depth or automatic front/back layering.

A higher Order in Layer value brings the sprite closer to the camera (on top of other objects), while a lower value pushes it further away (behind other objects).

Adjust this to ensure the correct layering of sprites in the scene, especially when you have overlapping elements.



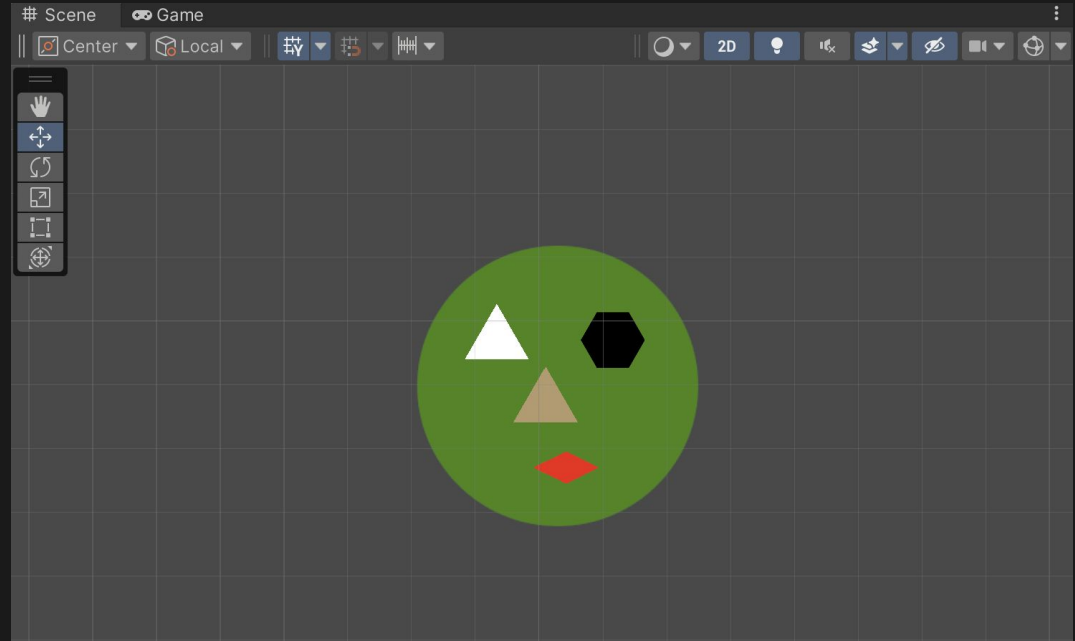
Sprite Renderer Challenge

Now, let's put your skills to the test!

Using the **Sprite Renderers** in this scene, modify the sprites by changing their images and colors.

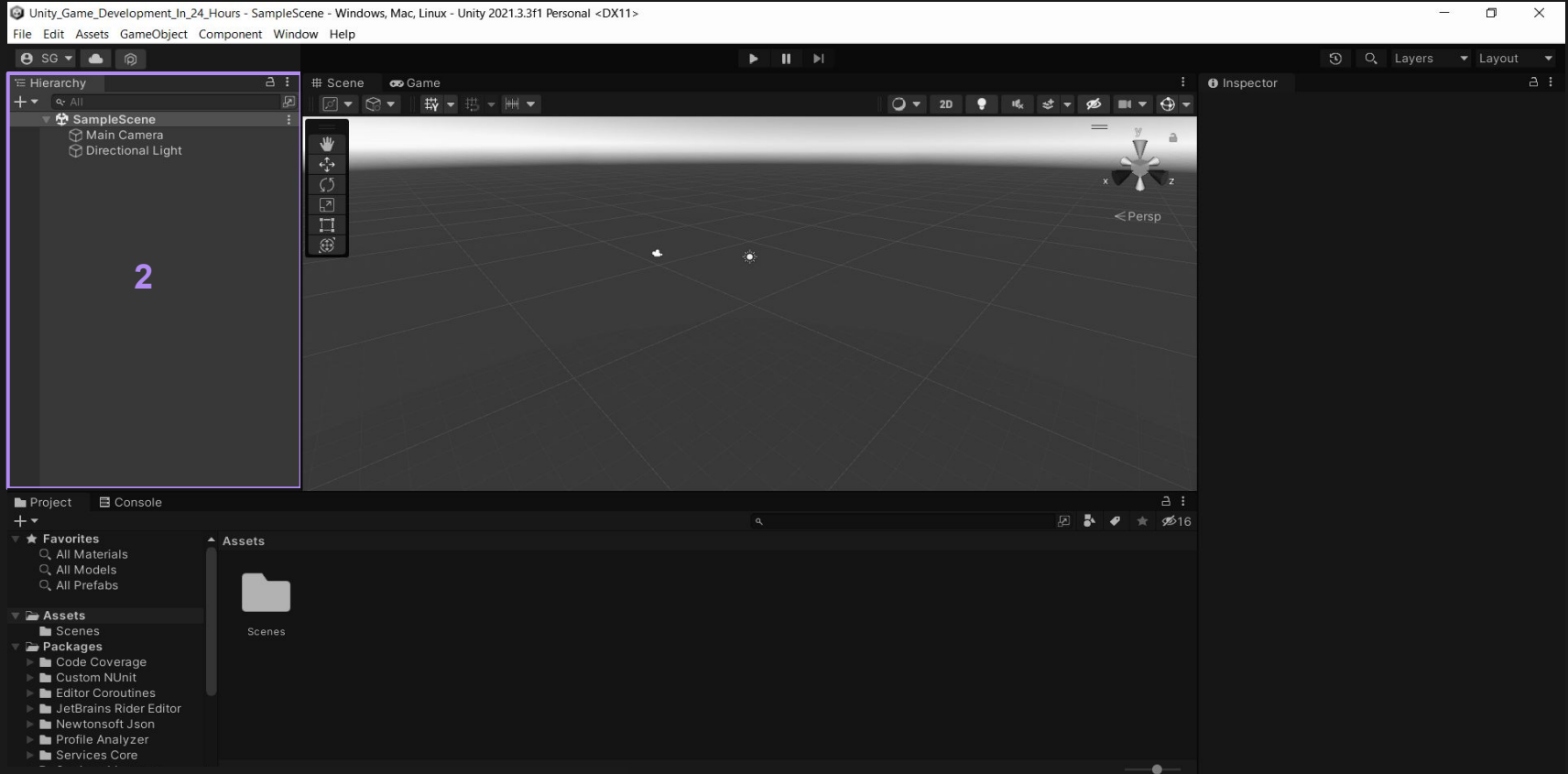
You can try replicating the example shown, or feel free to create your own unique design.

Adjust the **Order in Layer** to control the layering of the sprites and make sure everything is visually arranged how you want it.



Hierarchy View

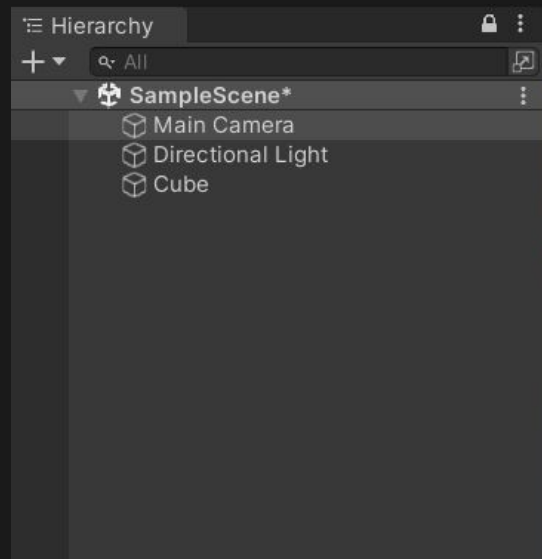
Hierarchy



Hierarchy View

The **Hierarchy View** displays a list of every Game Object that's currently in the scene.

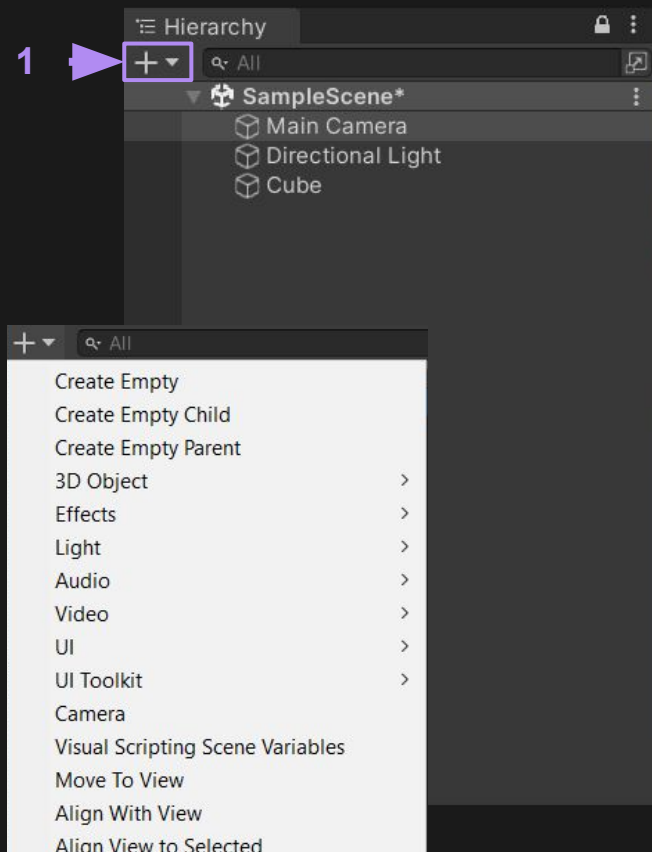
This view becomes incredibly helpful as your scene grows and fills up with more objects. Searching for a specific Game Object directly in the Scene View can become difficult or even impossible once you have a large number of Game Objects.



Hierarchy View - Create Game Objects

Either click the **Plus Button** in the top left of the **Hierarchy**, or **right-click** in the empty space. This will open a drop-down menu where you can create your desired Game Object.

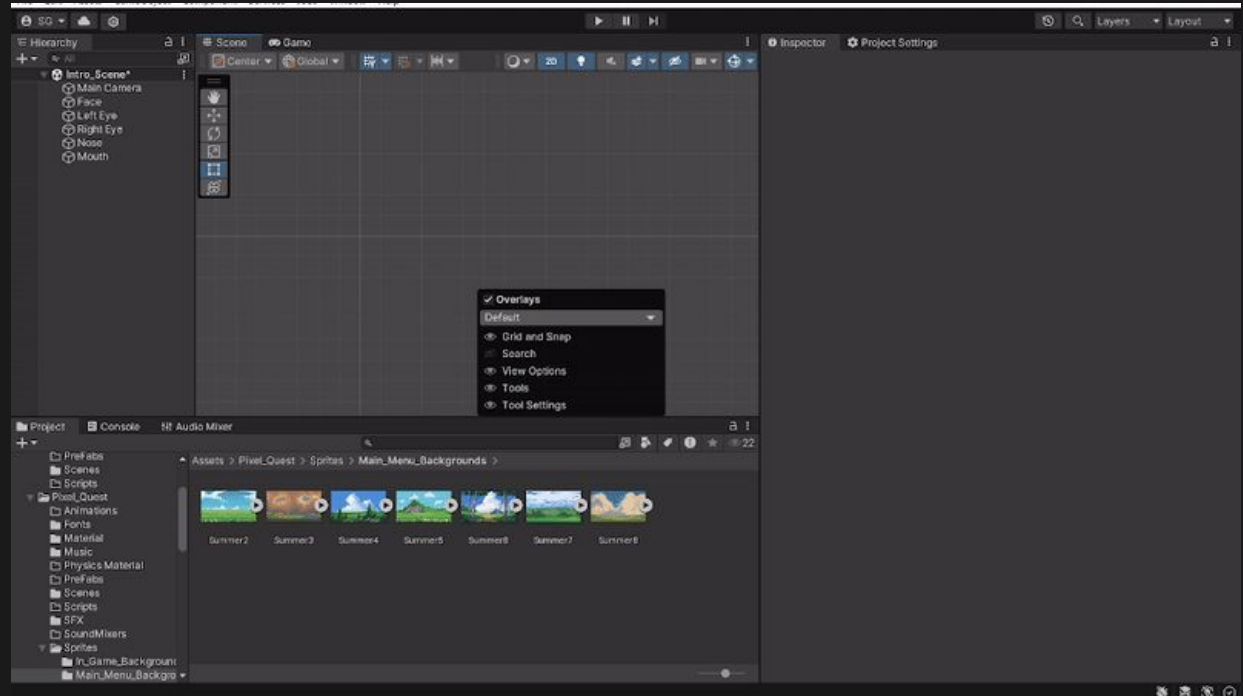
This drop-down menu allows you to create a new **Game Object**. You can choose to create an **empty Game Object** or a **Unity premade one** with selected components.



Drag to Create

Another way to create a Game Object is by simply dragging it directly from the **Project View** into the **Scene**.

This method is quick and convenient, allowing you to instantly add assets to your scene without needing to go through the **Hierarchy View**.

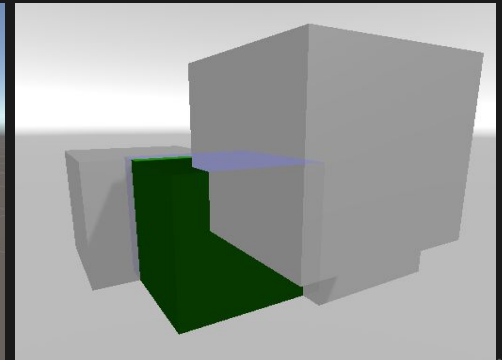
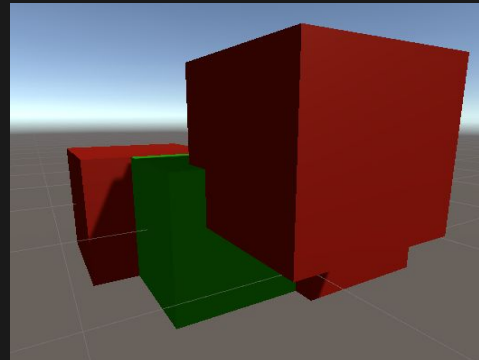
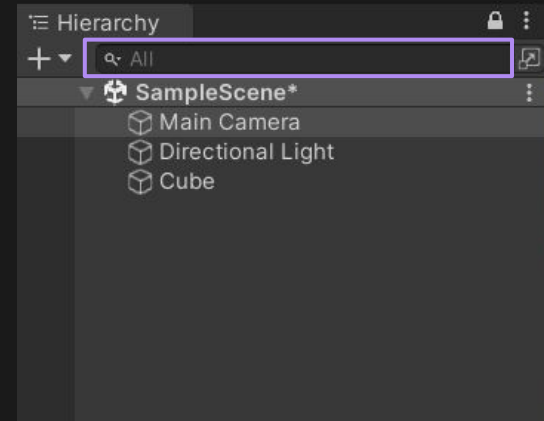


Hierarchy View

Search Bar: The search bar helps you quickly find a Game Object by filtering the items based on what you type.

For example, if you type "Cube," it will show you every Game Object in the scene that contains the word "Cube." To make your search even more effective, it's a good idea to give your Game Objects descriptive names.

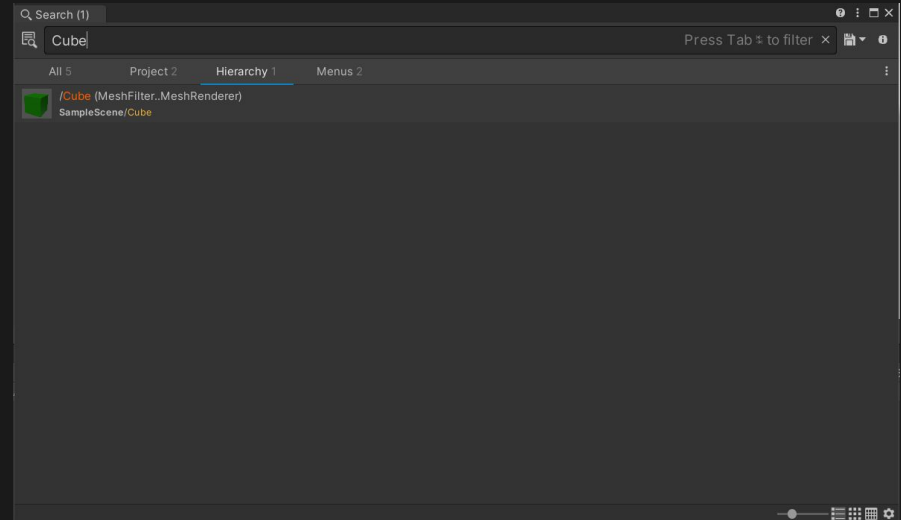
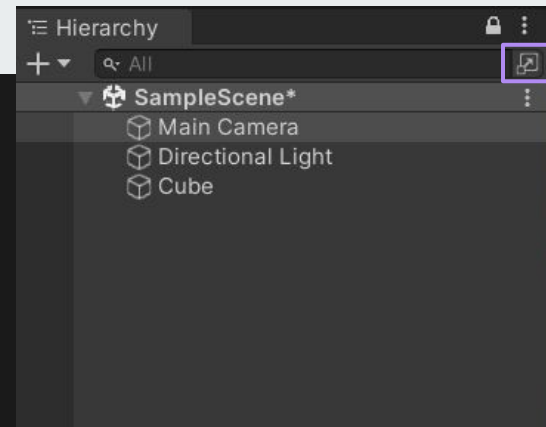
A great feature of the search bar is that it works in conjunction with the **Scene View**, greying out any objects that don't match your search term, making it easier to focus on the specific Game Objects you're looking for.



Hierarchy View

Advanced Search: Clicking the button next to the search bar will bring up the **Advanced Search** window. This feature allows you to search for Game Objects across the **Project, Scene, Hierarchy, and** more.

For smaller projects, you may not need to rely on this tool often, but it's helpful to know about it for when your project grows larger. It allows for a more in-depth search to find exactly what you're looking for across different sections of your Unity project.

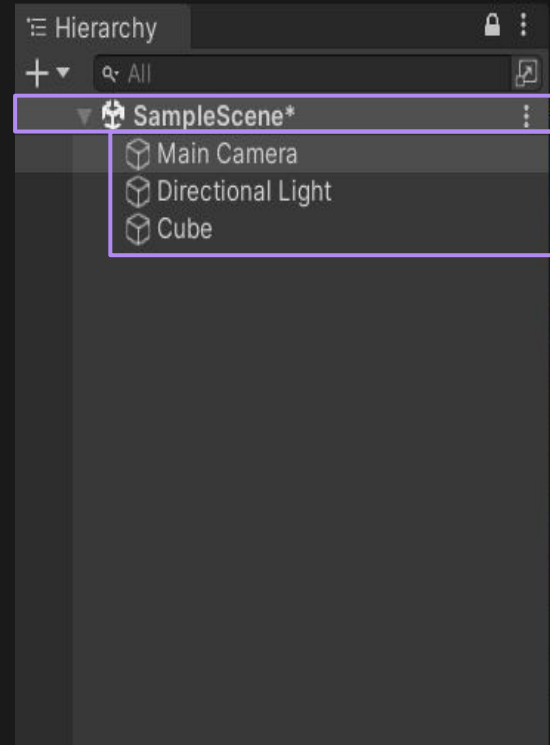


Hierarchy View

Scene Header: This section shows which scene the currently selected objects are part of. We will mostly be working within individual scenes, but as we work with data that spans across scenes, you might notice a new header appearing.

If you see an asterisk (*) next to the scene name (e.g., *SampleScene*), it indicates that changes have been made but not saved. Be sure to save your work frequently, as Unity does not have an autosave feature, and if it crashes, you could lose your progress.

Game Object List: This section displays all the Game Objects present in the current scene, giving you an overview of what is currently placed in your level. It helps you keep track of everything in the scene.



Nesting Game Objects

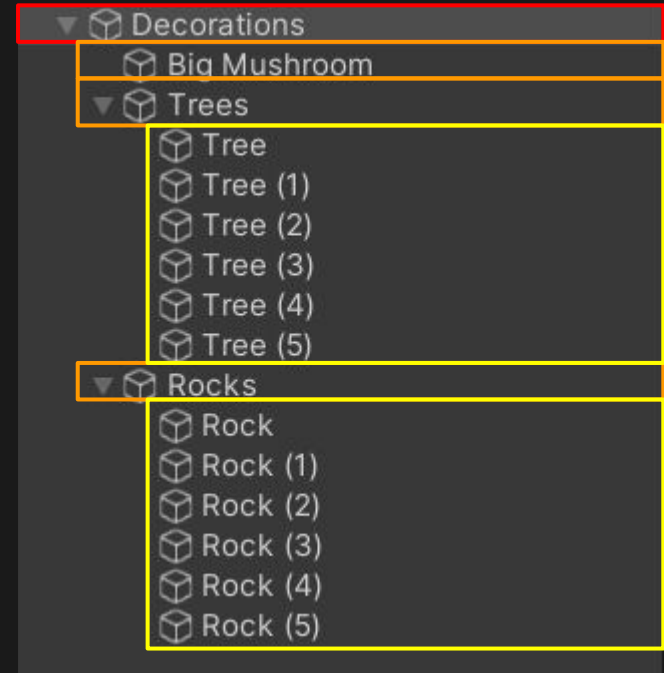
Nesting is the process of linking one Game Object to another, making it a child of the other object. This is done by dragging one object on top of another in the Hierarchy.

When this happens, the nested object (the Child) will indent beneath its parent object (the Parent) in the Hierarchy.

For example, if you drag the "Big Mushroom" onto "Decorations," the Big Mushroom becomes a child of Decorations.

On the other hand, if you have a "Trees" Game Object with multiple individual trees (e.g., Tree, Tree (1), Tree (2), etc.), the "Trees" object is the Parent, and each of the individual trees are children of it.

These individual trees (Tree, Tree (1), etc.) are not direct children of "Decorations," but rather, they are considered Descendants of Decorations if they are nested under Trees.

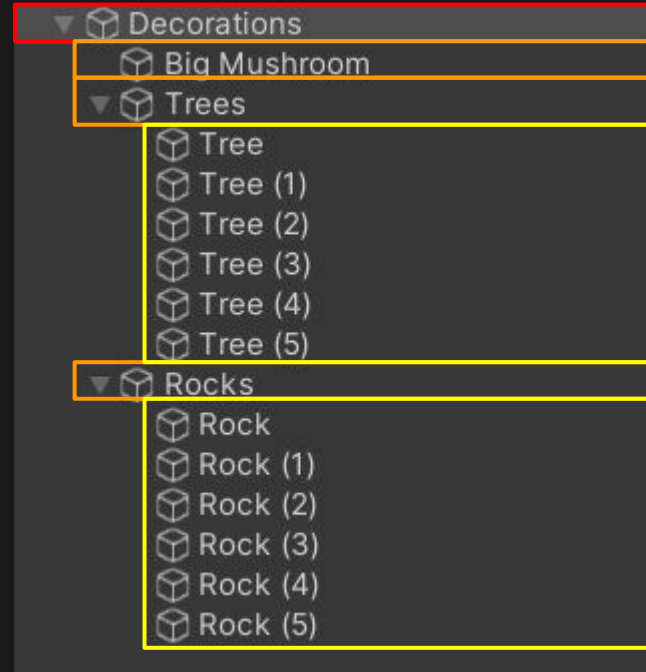


Nesting Game Objects

When a Game Object is nested, the parent object effectively becomes the origin point for the child objects in terms of world coordinates.

This ties back to the **World vs. Local concept**—when you move or manipulate the parent object, the child objects will follow, but they maintain their relative position to the parent.

This means that the child objects' coordinates are now based on the parent's position rather than the global world coordinates.



Keeping Hierarchy Organized

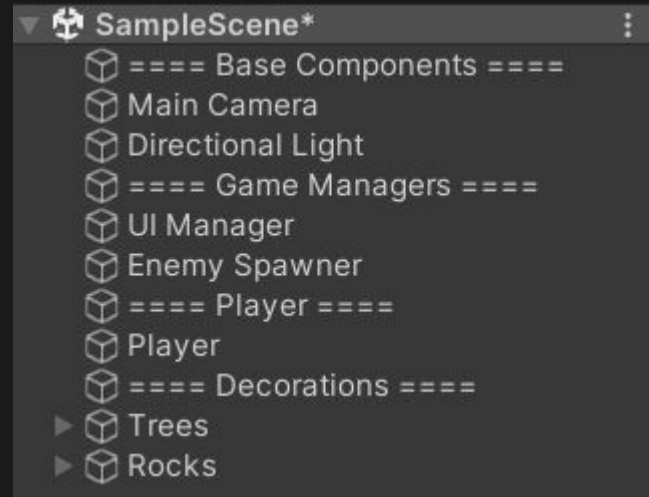
Using **dividers** in your scene hierarchy is a great organizational practice. Dividers are simply empty Game Objects that are used to visually separate and group other Game Objects in the scene.

They help make the hierarchy more readable and easier to navigate, especially as your scene grows larger with more objects.

==== Divider ====

**** Divider ****

---- Divider ----

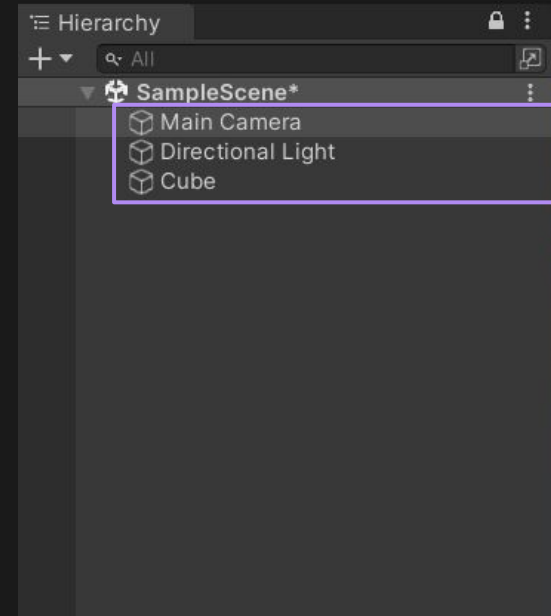


Finding Game Object in Scene View

Double-clicking on any Game Object in the **Hierarchy View** will automatically zoom the **Scene View** in to focus on that object. This makes it easier to work with individual objects, especially if your scene has a lot of elements.

Additionally, you can use the **Hotkey F** while hovering your mouse over the **Scene View** to quickly zoom in on the selected Game Object.

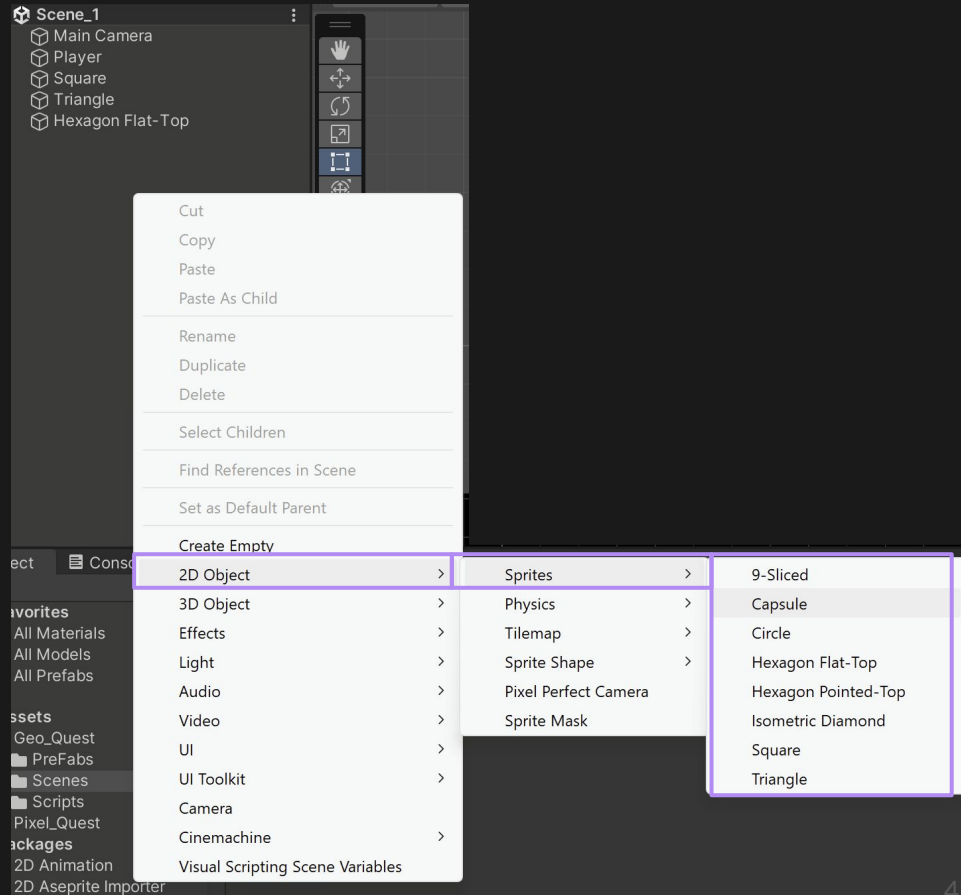
This is a fast way to center and zoom in on the object you're working with, without needing to manually adjust the view.



Hierarchy View - Creating Game Object with Sprite

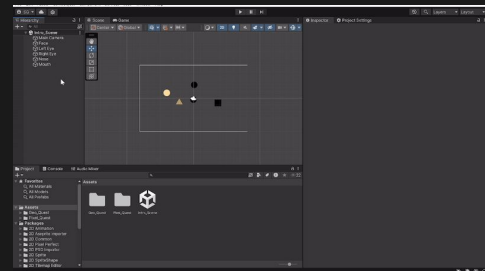
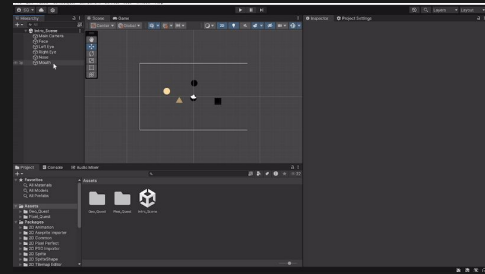
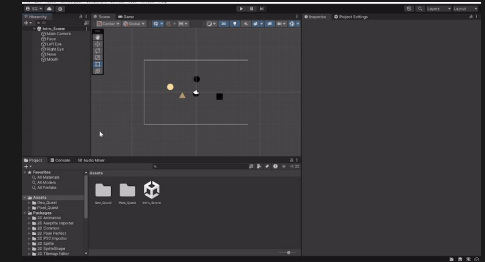
To create a **Game Object** with a **Sprite**, you can either click the **Plus Button** at the top of the **Hierarchy** or **Right-Click** in the **Hierarchy**.

Then, select **2D > Sprite** and choose the shape you wish to create. This will generate a **Game Object** with a **Sprite Renderer Component**, already set with the sprite of the selected shape.



Hot Keys

Hotkey	Function
Ctrl-C	Copies the selected item(s) or text to the clipboard.
Ctrl-V	Pastes the copied item(s) or text from the clipboard to the desired location.
Ctrl-Z	Undoes the last action.
Ctrl-Y	Redoes the last undone action, restoring changes undone by Ctrl + Z.
Shift + Left Click	Selects a range of items. For example, click one item, hold down Shift , and click another—this will select all items in between.
Ctrl + Left Click	Selects or deselects individual items. Use it to select multiple non-adjacent items or deselect something already selected.

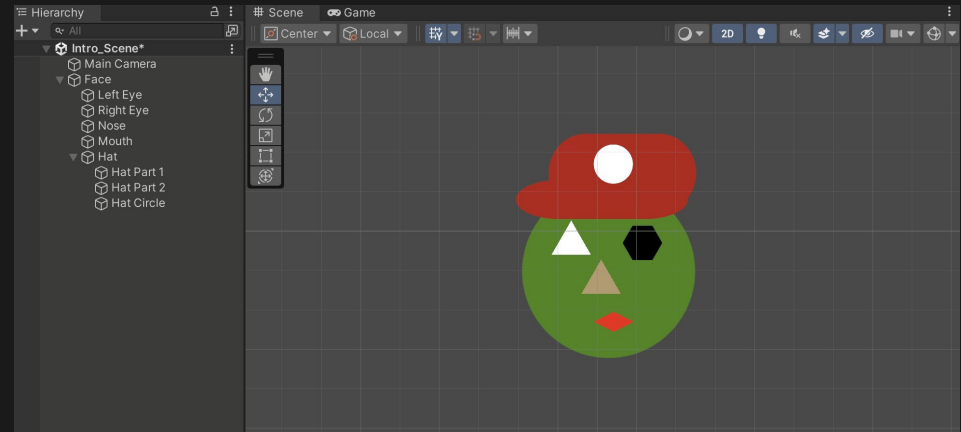


Challenge: Creating & Nesting

Now, let's put your skills to the test!

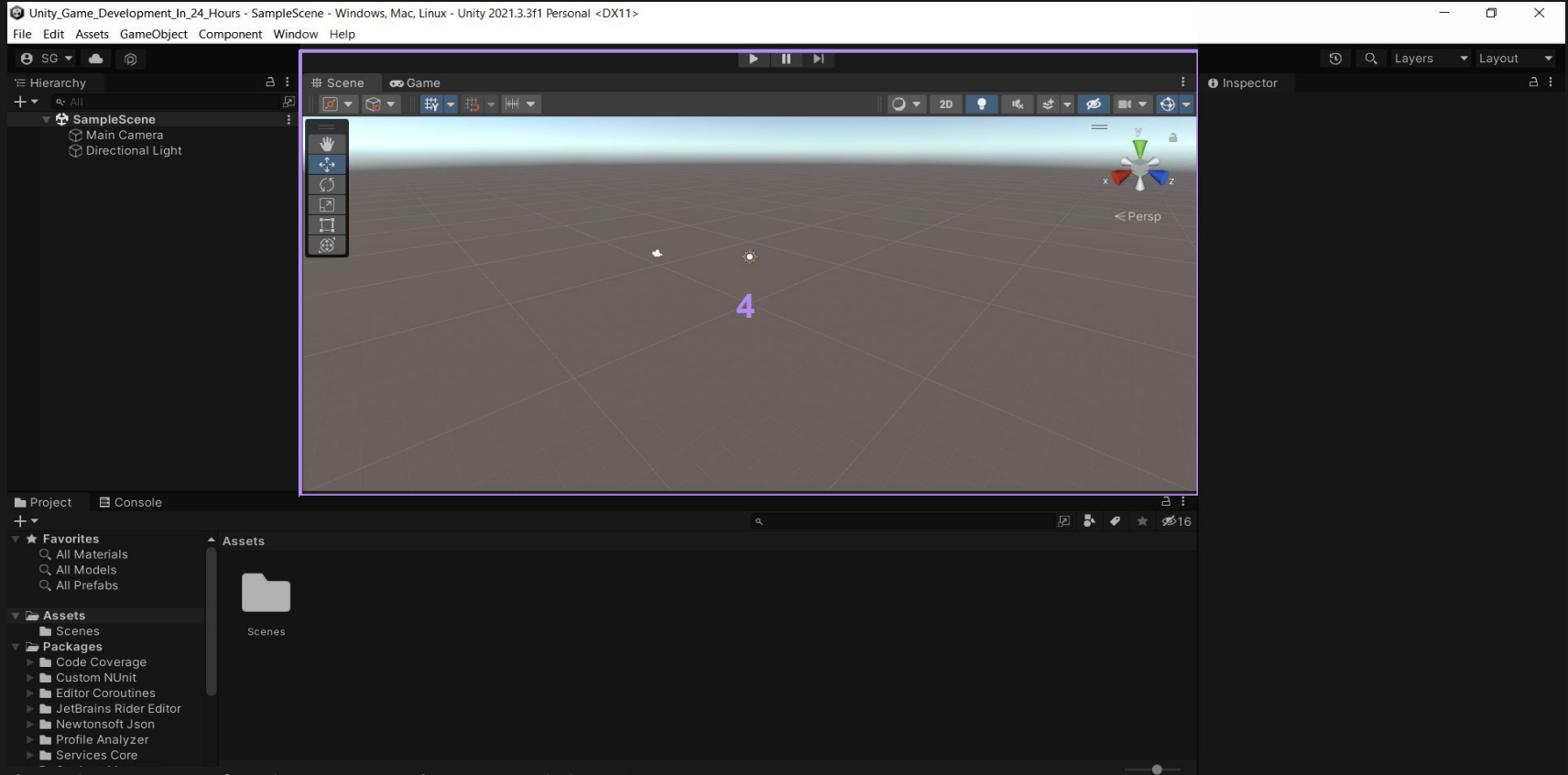
Using the **Hierarchy** and **Inspector**, create and modify Game Objects to give your face a hat. You can design your own hat or get creative with something else!

1. **Create the Hat:** In the **Hierarchy**, right-click and select **2D Object > Sprite** to create a new sprite for your hat.
2. **Modify the Hat:** Use the **Inspector** to adjust the sprite's **position**, **scale**, and **rotation** so it fits properly on top of the face.
3. **Parent the Hat:** After positioning the hat, drag it into the **Face GameObject** in the **Hierarchy** to make it a child object.



Game View

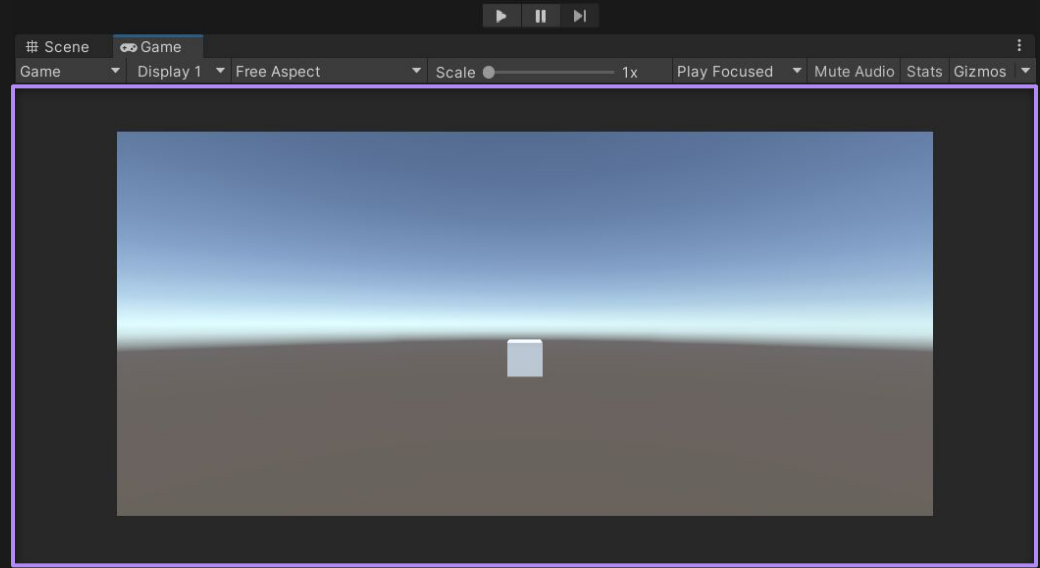
Game View



Game View

The **Game View** displays the final output of your scene as it will appear during gameplay. It's a real-time window that shows the game from the perspective of the **Camera Game Object**.

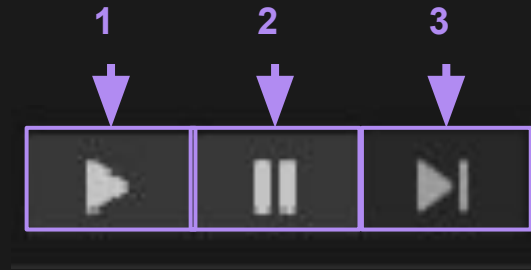
It allows you to play-test your game and see how it looks during actual play, with all the mechanics and interactions you've set up. If you have a different camera setup, you can switch between cameras to test different perspectives.



Play Bar

The **Play Bar** is always accessible and consists of three buttons for controlling gameplay.

1. **Toggles Play/Off:** Clicking the grayed-out play button will assemble and run the game, allowing you to experience it in real-time. This is also known as **Running** or **Executing** the game. When you toggle the play button off, it will reset the game to its original state, undoing any changes or progress made during the session. For example, any enemies spawned will be removed, and any progress in the level will be reset.
2. **Pause:** This button stops the execution of the game while preserving the progress you've made in the level. It's particularly useful when something isn't functioning as expected. You can run the game, pause it when the issue occurs, and then use the Scene View to inspect the Game Objects that might be causing the problem.
3. **Step:** Similar to the Pause button, Step is a valuable tool for debugging. Each click progresses the game by one step, allowing you to observe what happens during each tick of time in the game. This can help pinpoint exactly where issues arise in your game's mechanics.



While in **Play Mode**, you are free to make adjustments to Game Objects through the **Scene View** or **Inspector**. However, be cautious, as these changes will not be saved when you stop Play Mode—they will revert to their original state.

Game View Toolbar



The **Game View Toolbar** allows you to modify how the game is presented, letting you preview how it will appear on different screen sizes and resolutions.

This is important because the way the game is displayed directly affects what the player can see at any given time.

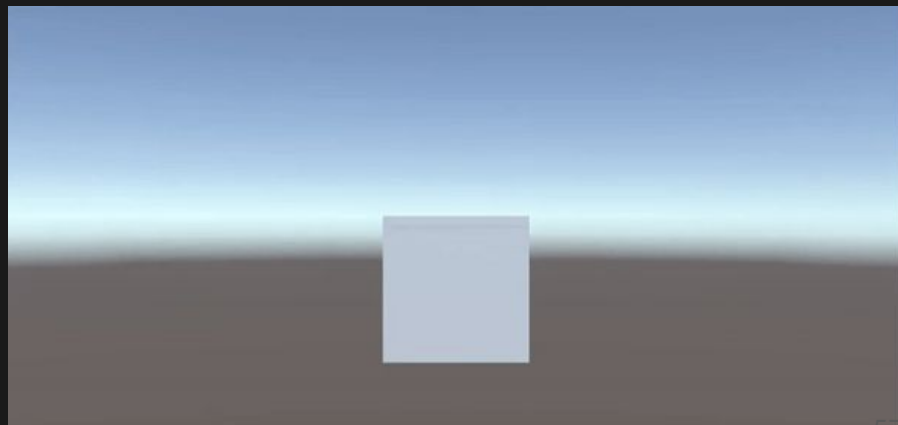
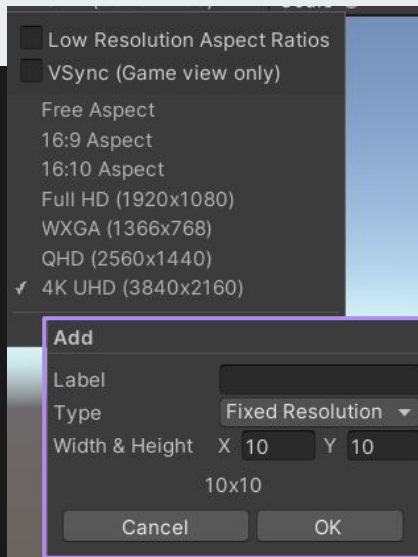
These choices help inform important **Game Design Decisions**, such as how much content fits on the screen and how you structure the user interface and gameplay.

Aspect Dropdown

Aspect Dropdown allows you to set the **Aspect Ratio** for how you view the game in the Game View.

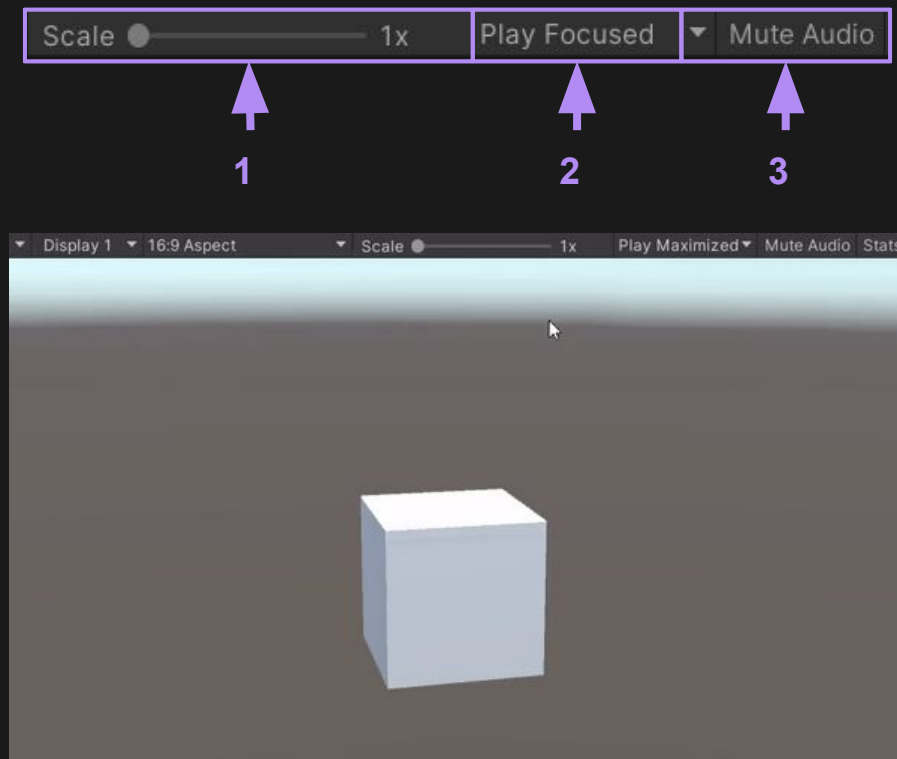
It's important to note that this only affects how you see the game within the Unity Editor—it doesn't determine the final aspect ratio when the game is exported.

Free Aspect ▼



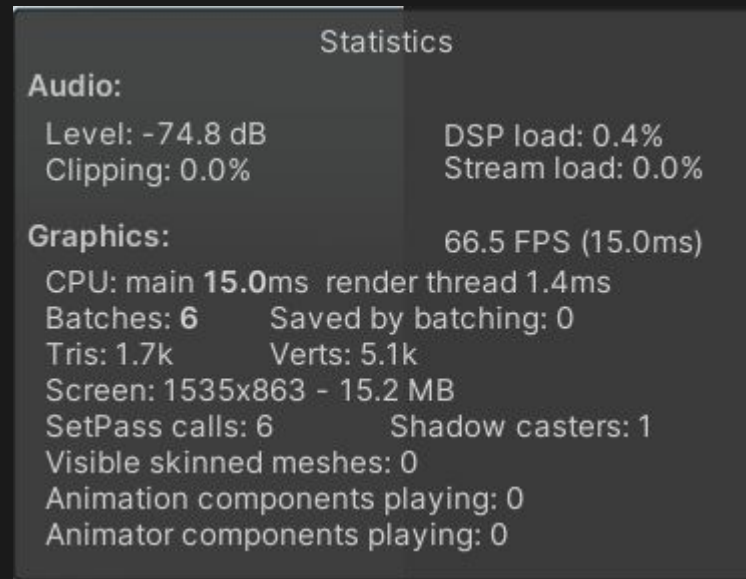
Scale, Play Focused / Maximized, Mute

1. **Scale** allows you to zoom in and out on the game screen, providing a closer or broader view of your game's visual output. This zoom adjusts the display of the game in the Game View, but it's different from navigating close to a Game Object in the Scene View.
2. **Play Focused / Maximized** enables you to maximize the Game View during play mode, making it take up as much of your screen as possible while maintaining the selected aspect ratio.
3. **Mute** is pretty self-explanatory. When you're testing and debugging your game, you might find yourself hearing the same sound effects or music over and over.



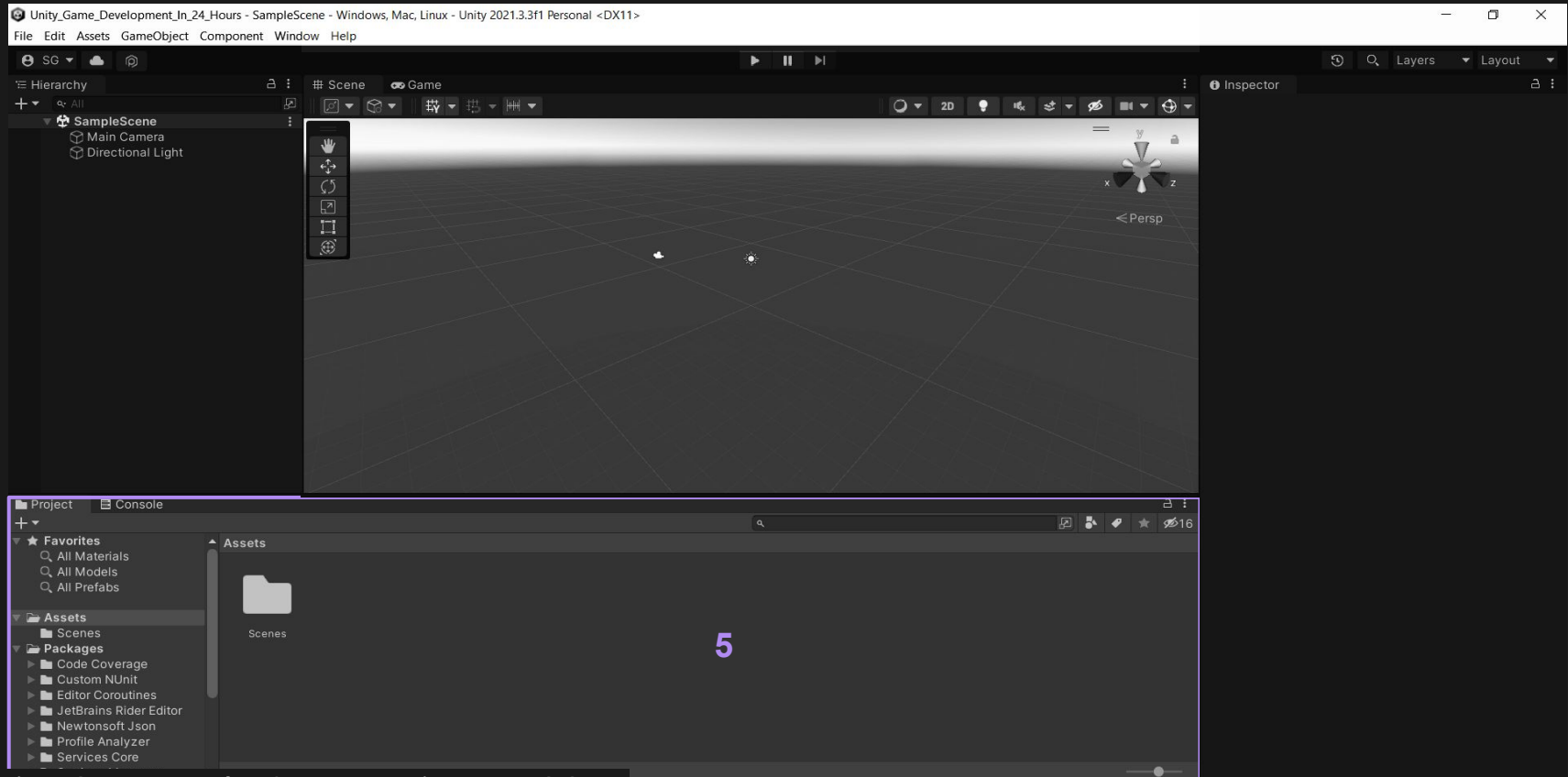
Stats and Gizmos

1. **Stats** is a very useful tool that gives you insights into how your game is running. It breaks down the performance into two main areas: **Audio** and **Graphical** strain on your computer.
2. **Gizmos** work similarly to how they do in the Scene View, allowing you to see the visual cues (such as icons and debug information) that you've toggled on in the Game View.

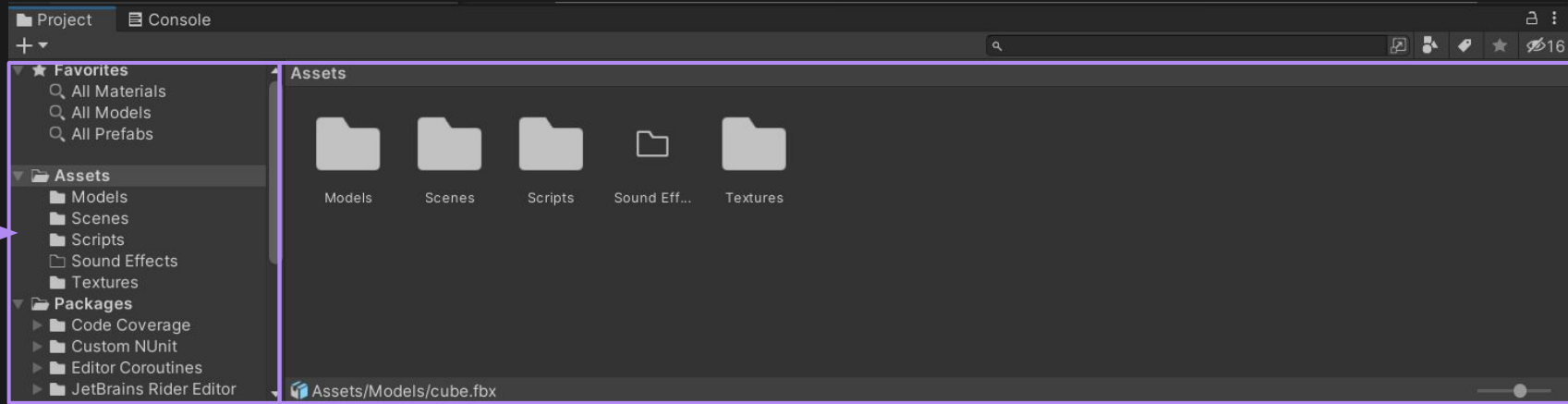


Project View

Project View



Project View



The **Project View** displays all the Game Assets that are available to you for use in your scene.

1. **Folder List** shows you a list of all the folders in your project, making it easy to organize and access the assets. You can also see any saved favorite searches here for quicker access to specific types of files.
2. **Current Folder** shows you which folder you're currently looking at. This is helpful because as your project grows, you might have a lot of assets stored in different folders. You can easily navigate and see which folder you're in at any given time.

Game Asset Vs Game Object

A **Game Asset** is any file or resource that has the potential to be used in the game, such as textures, models, sounds, scripts, and more.

A **Game Object**, on the other hand, is an instance of that Game Asset that exists in the scene and has specific components attached to it (like a Sprite Renderer, Collider, etc.).

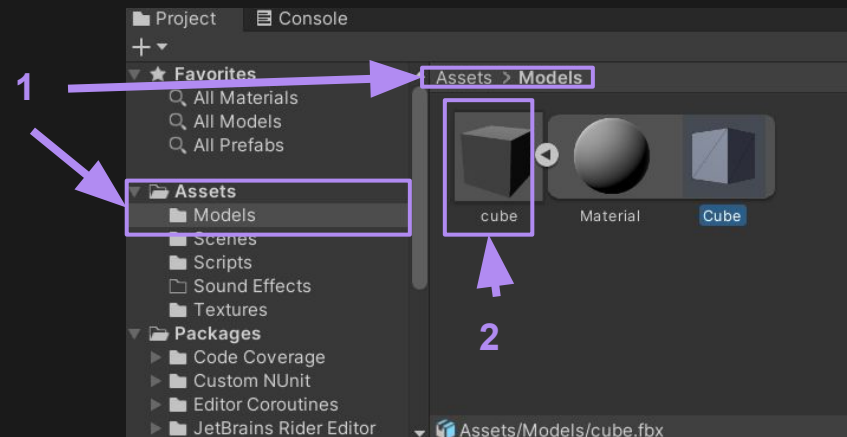
You can **instantiate a Game Asset** into a **Game Object** creating it.



Project View - Organization

When working in the **Project View**, you can explore and manage your Game Assets. Here's a breakdown of the key elements you'll encounter:

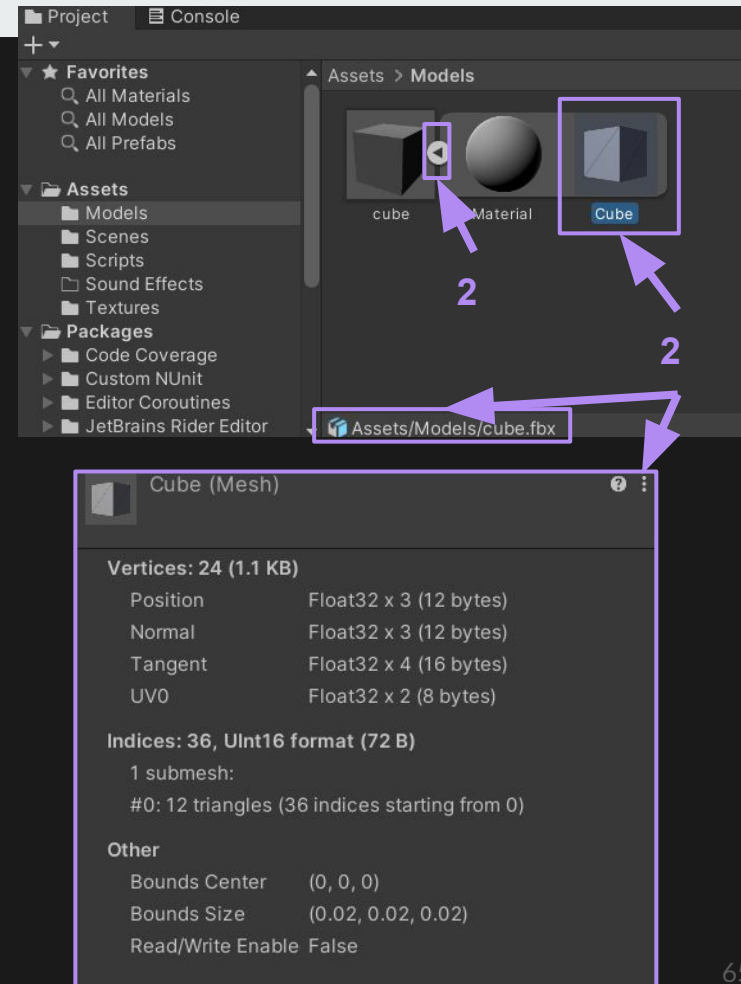
1. **Breadcrumb Trail:** This shows the current path to the folder you're viewing. It's like a map to help you keep track of where you are within the folder structure. You can also see this path in the **Folder Column**, highlighting the folder you're currently accessing.
2. **Game Asset:** This represents the actual file or resource in your project. For example, it could be a 3D model, texture, script, or audio file. In this case, we have a 3D imported cube.



Project View - Organization

1. **Sub-Assets:** More complex assets, like 3D models, often come with a hierarchy of sub-assets. You can expand or collapse these sub-assets by clicking the **arrow** next to the main asset to see additional files, such as textures or meshes associated with the model.
2. **3D Model:** If you select a sub-asset like the 3D model of the cube, you'll see its specific details in the **Inspector View**. The **path** to the asset is displayed at the bottom of the **Project View**, so you always know where it resides in your folder structure.

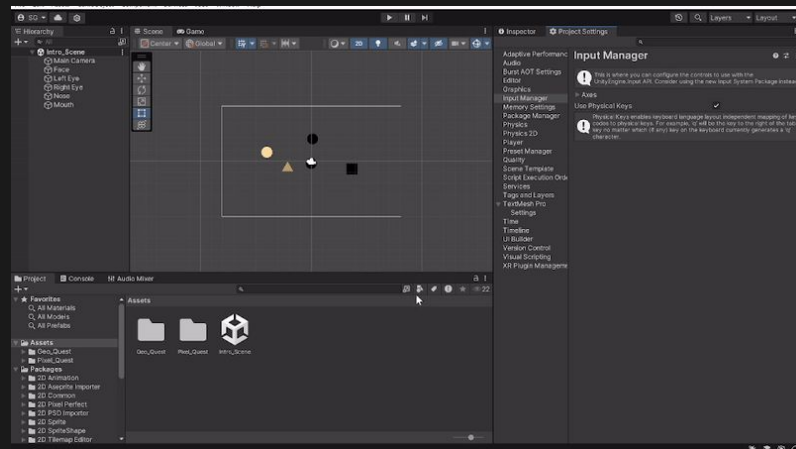
By navigating these elements, you can easily organize and access your assets for use in the game.



Search & Organization

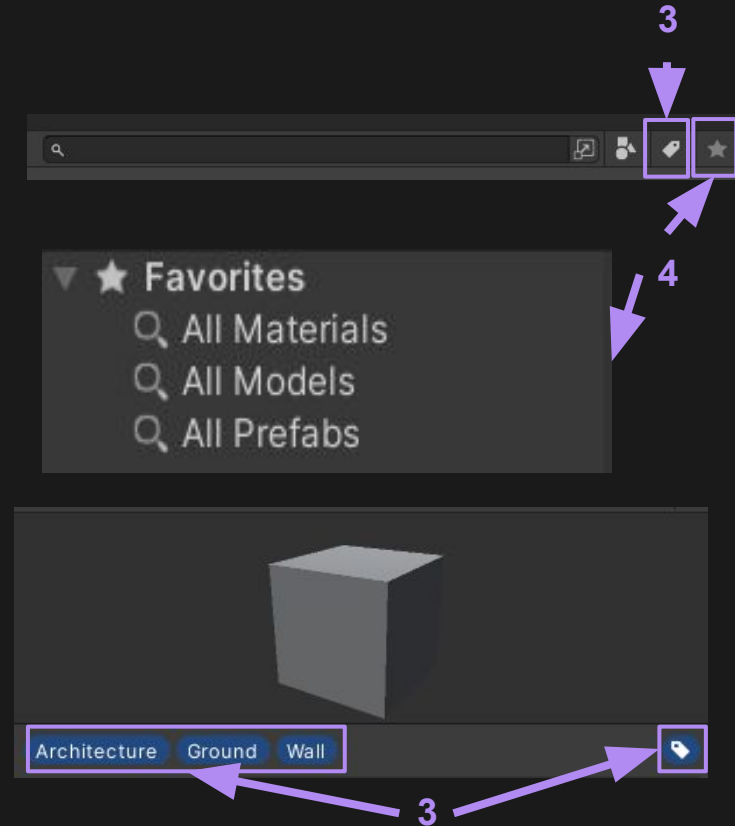
Efficient asset management is key to speeding up your development process, and Unity provides several powerful tools to help you find assets quickly in the **Project View**:

1. **Search Bar:** Just like in the **Hierarchy View**, the **Search Bar** in the **Project View** helps you find specific assets quickly.
2. **Category Filter:** You can filter your search results by category. Clicking on the filter buttons allows you to view items that fit into a specific category, such as **Models**, **Scripts**, and more.



Search & Organization

1. **Tags:** Tags are descriptors you can assign to **Game Assets** in the **Inspector View**.
2. **Favorite Search:** When you search for an asset or asset type, you can "favorite" that search. This saved search will appear in the **Folder Column**, giving you quick access to searches you've frequently used.

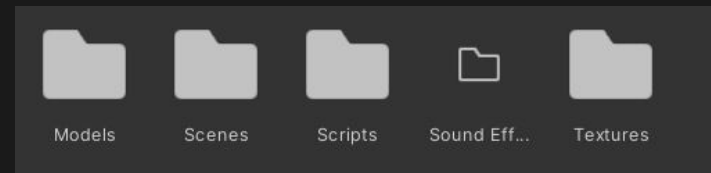
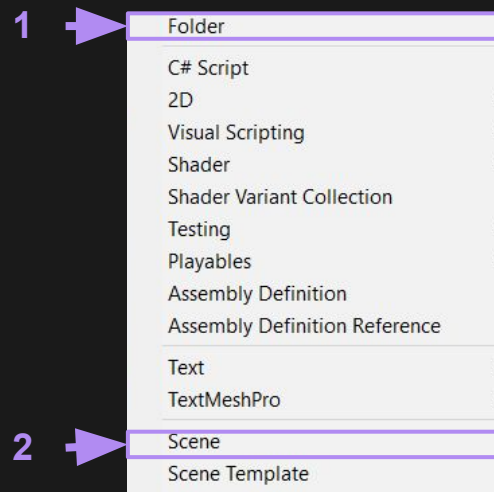


Creating Asset and Organization

In the **Project View**, organizing your assets is key to maintaining a smooth workflow.

You can create new Game Assets by right-clicking on an empty space and selecting **Create**. This brings up a list of different asset types, but for now, let's focus on two important ones:

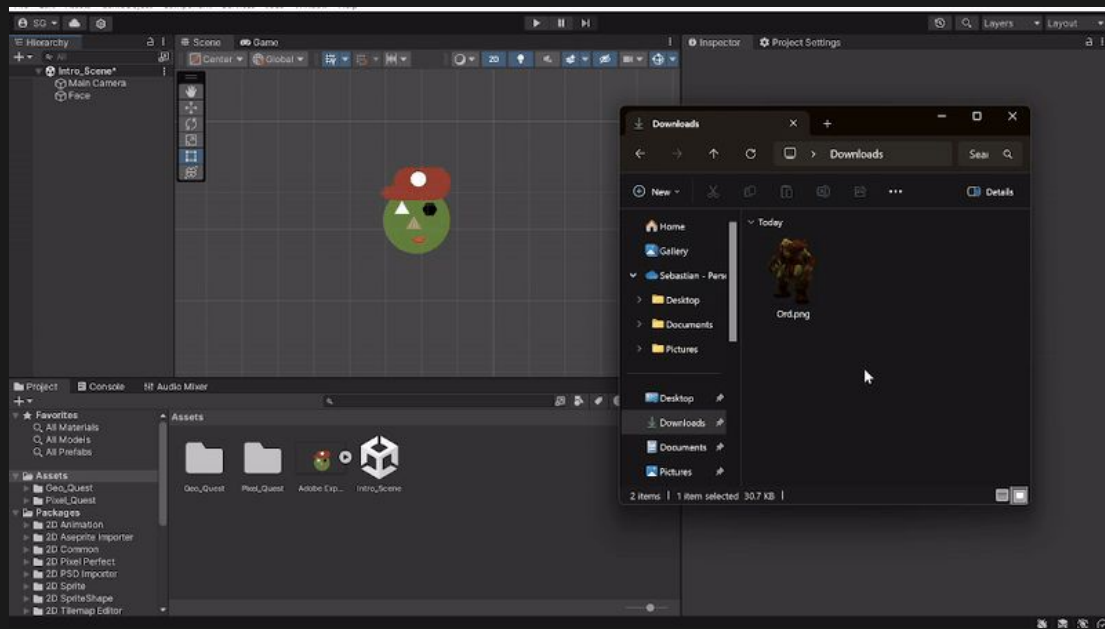
1. **Folder:** This option allows you to create new folders within the Project View. While it may not seem exciting, having an organized folder structure is crucial. Just like you organize the **Hierarchy View**, keeping your **Project View** neat and tidy helps you manage your assets efficiently, especially as your project grows in size.
2. **Scene Asset:** Selecting this option lets you create a new scene or level for your game. So far, we've been working within one scene, but with the **Scene Asset**, you can create multiple levels and scenes.



Importing Assets

The simplest way to import assets into your game is to use drag-and-drop from your file explorer. Here's how you can do it:

1. **Open File Explorer:** Navigate to the folder that contains the asset (image, model, audio file, etc.) that you want to import.
2. **Drag and Drop:** Simply drag the asset from the file explorer and drop it into your **Project View** in Unity. You can drop it directly into the desired folder in the **Project View**, or if you just drag it into the open space, it will automatically appear in the currently active folder.
3. **Drag into Scene:** If you want to immediately add the asset to the scene, you can also drag it directly into the **Scene View**. This will create a new Game Object based on that asset, and it will appear in the currently open folder in your **Project View**.



Challenge Import Image Asset

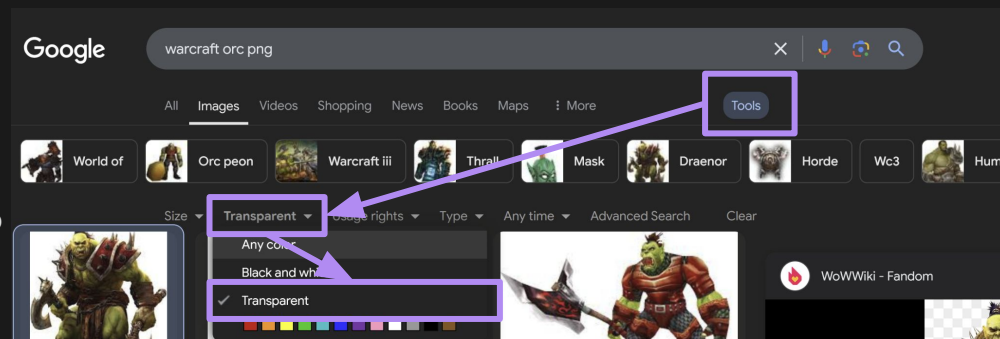
Now, let's put your skills to the test!

Go to the internet and find an image of a character you'd like to use in your scene. Make sure it's class-appropriate and fits the style of your project, download the image to your computer and import it into your project.

If you'd like the character to have a transparent background, follow these steps:

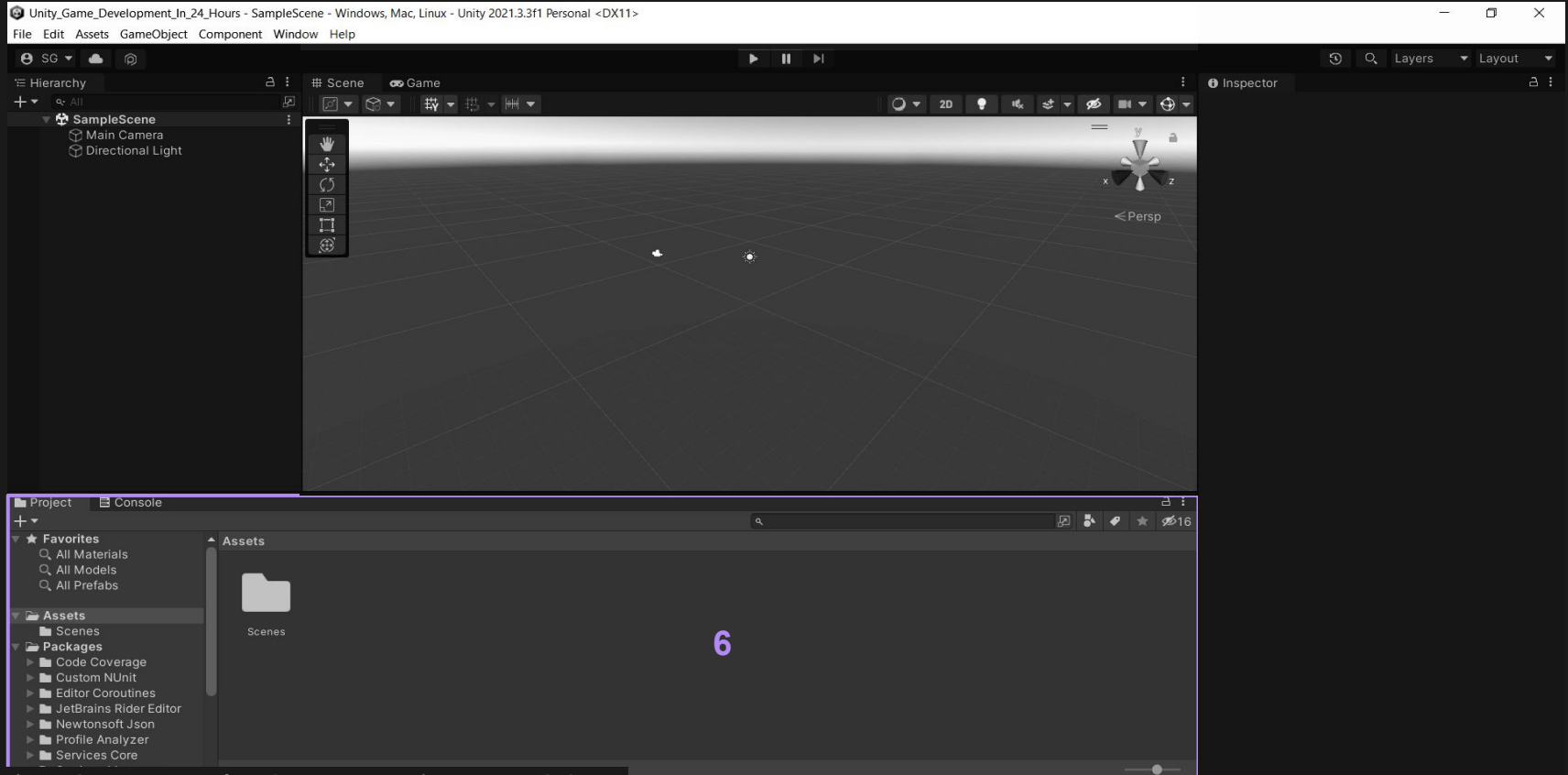
1. Click on Tools in your browser or image editor.
2. Select Color, and set the background to Transparent (if applicable).

When downloading, ensure the file type is .png, as PNG images support transparency. Other image formats (like JPEG) will not maintain the transparent background.

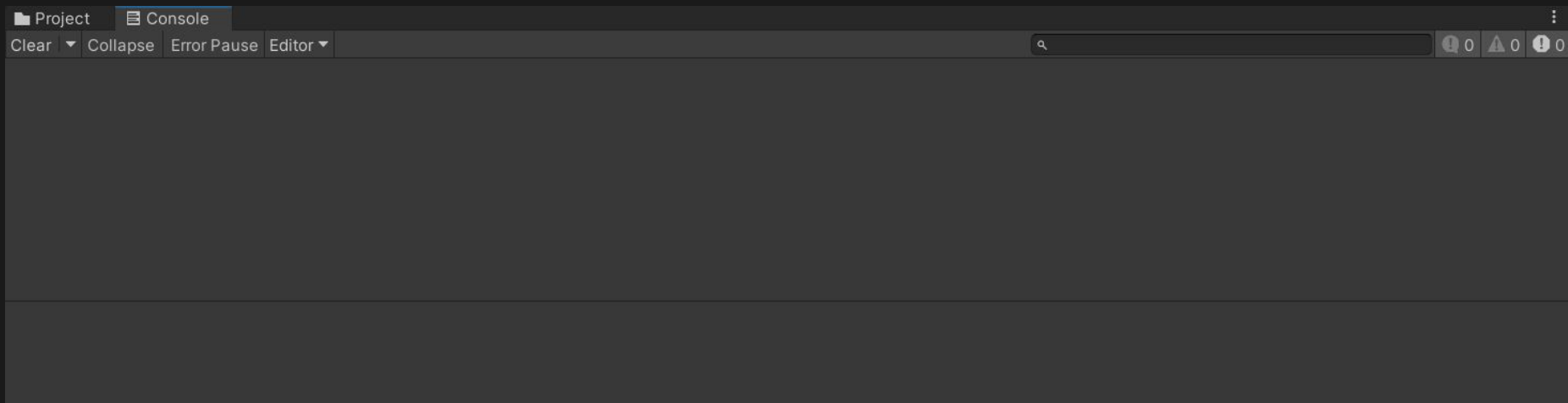


Console View + Layouts

Console View



Console



The Console is an essential tool that will become even more useful once we start programming your game. Here, you'll be able to:

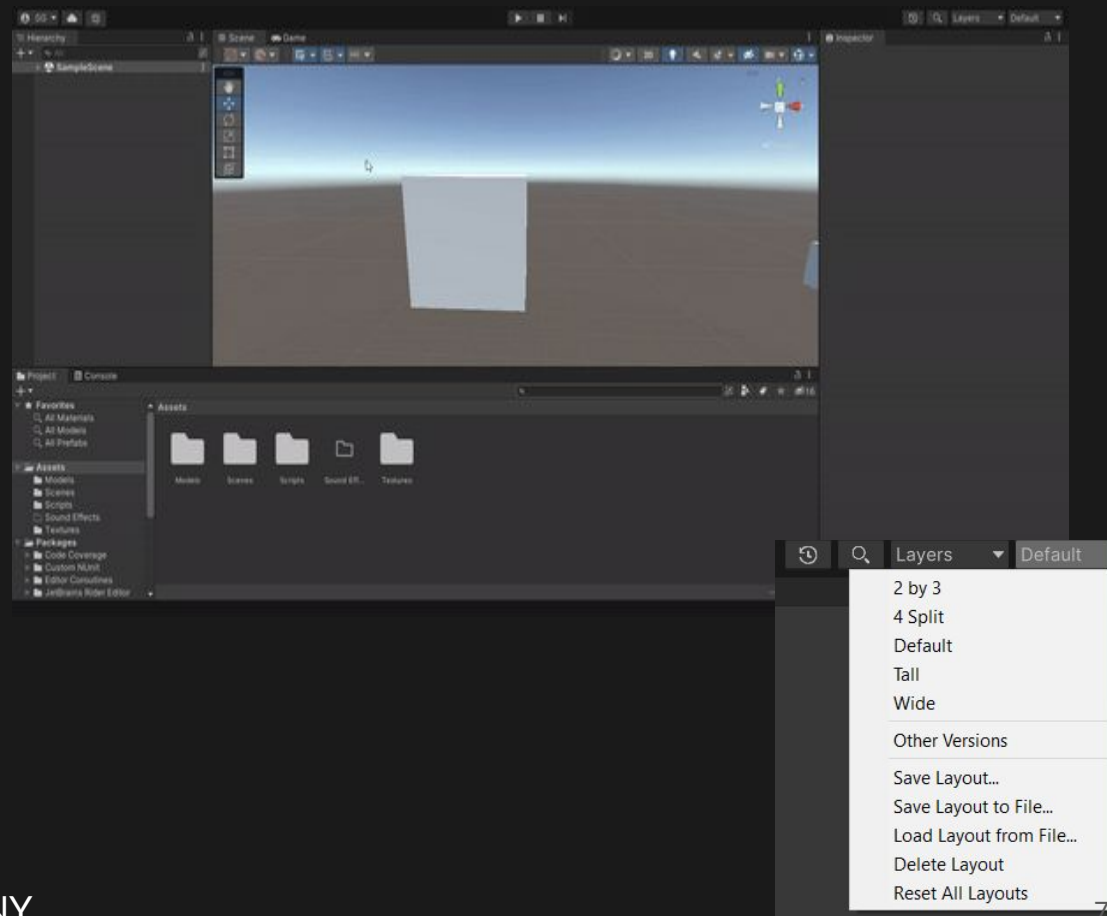
- **Print Debugging Information:** You can display messages, object states, and other data to help track what's happening during gameplay.
- **View Errors and Warnings:** If something goes wrong in your game (like a bug or issue), the Console will show error messages and warnings so you can troubleshoot effectively.

Layouts

Important thing to note is that **all the views** we've explored today—whether it's the Scene View, Game View, Inspector, or Console **can be moved and arranged anywhere** within the Unity window. Unity gives you the flexibility to create a layout that works best for you.

There are a number of **pre-made layouts**, but you also have the option to create and **save your own custom layout**.

And if at any point you find the layout becomes too disorganized, you can always click the **Default** button at the top right and select one of the pre-made layouts to reset everything back to a familiar setup. This makes it easy to get back on track if things start to feel cluttered!



Double Clicking on Window Tabs

At some point, everyone accidentally double-clicks a tab and ends up with a layout they didn't intend. If that happens, don't worry! You can simply **double-click the tab again**, and it will return to your original setup. It's a quick and easy fix to get everything back to normal!

