Diving Into Unity

Sebastian Grygorczuk - STEM Institute at CCNY

Today's Agenda

We're going to be going over Chapter 1: Introduction to Unity, Chapter 2: Game Objects, and Chapter 3: Models, Materials, and Textures

- We're going to get familiar with the layout out all the Unity Views.
- Learn how to navigate through a scene and interact with Game Objects.
- Devel into different Components such as Mesh, and 3D Renderers.

The Unity View

The Unity Engine is made up of a multitude of windows that will be essential to creating your game. The windows we will currently look at are:

[1] Scene View

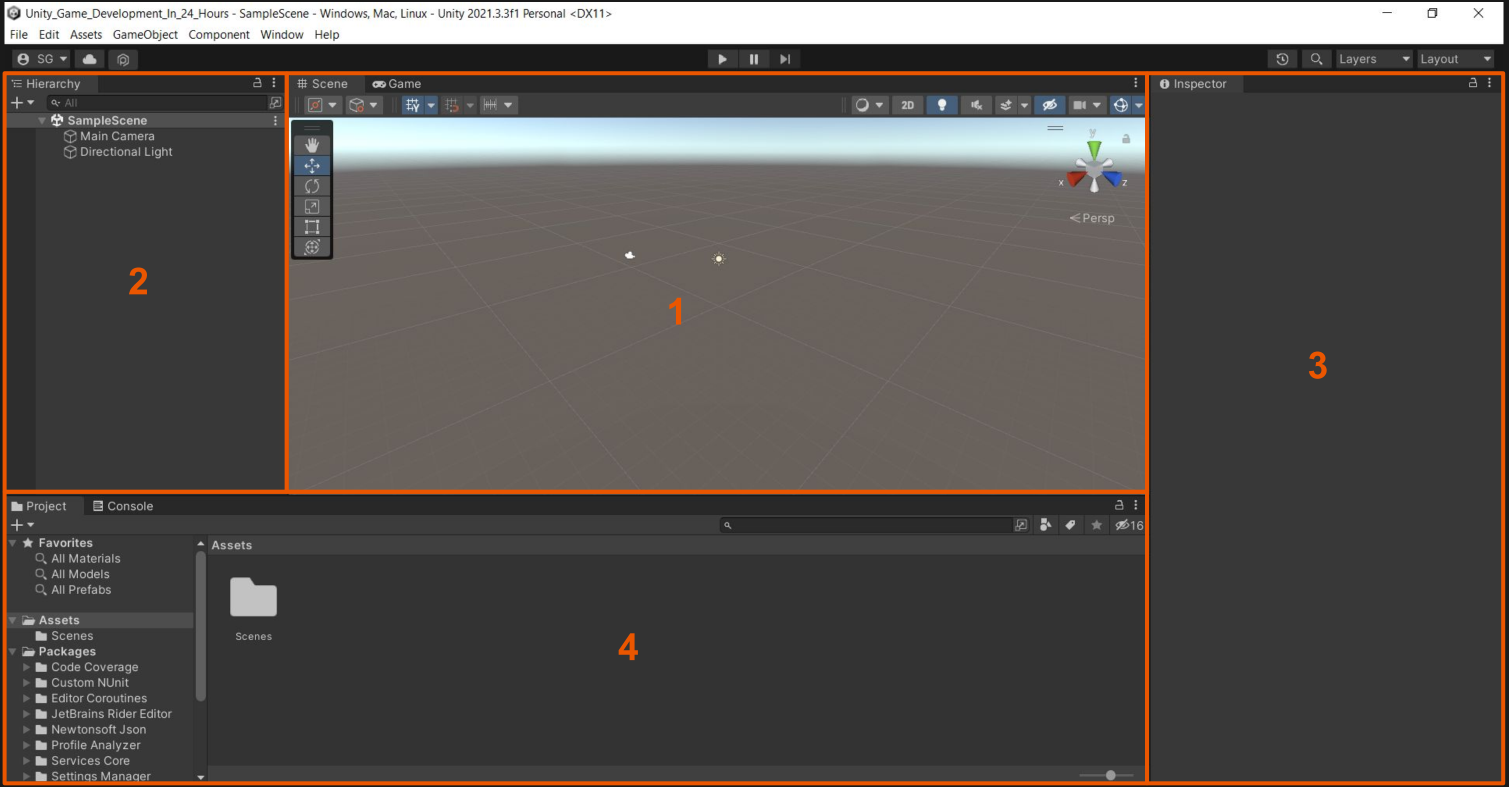
[2] Hierarchy

[3] Inspector

[1] Game View

[4] Project View

[4] Console



Sebastian Grygorczuk - STEM Institute at CCNY

Game Object

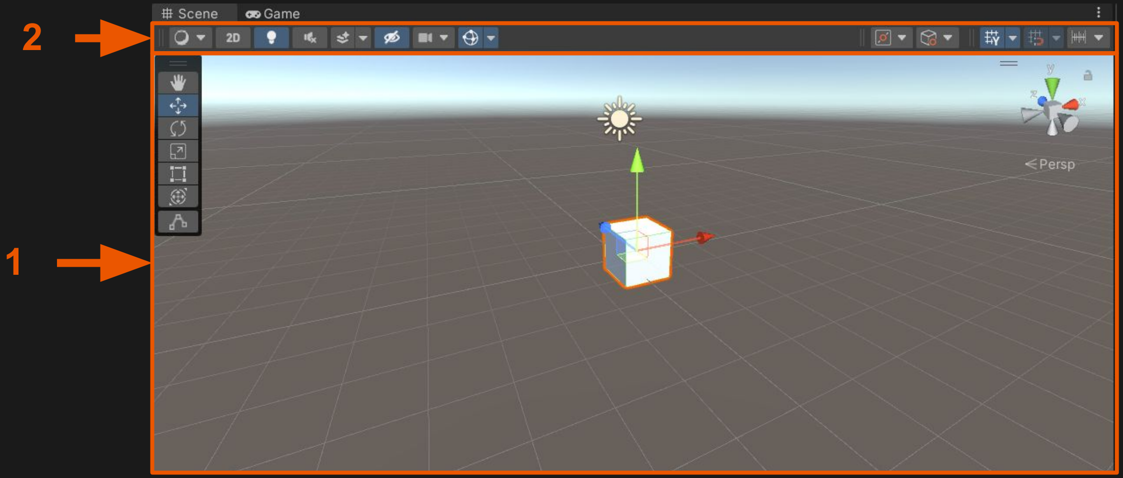
Before we get into the different views I would like to first define what a Game Object is as the two words will come up a lot.

Game Object is a foundational component of anything that will exist in a game. Think of it as an empty container that you can add any property to turing it from nothing into a car, person or anything else you can imagine.

As we make our ways through all of the Views you will be very familiar with Game Objects.

Additional Resources: Page 23 of the Textbook, Game Objects and Components - Unity Official Tutorials

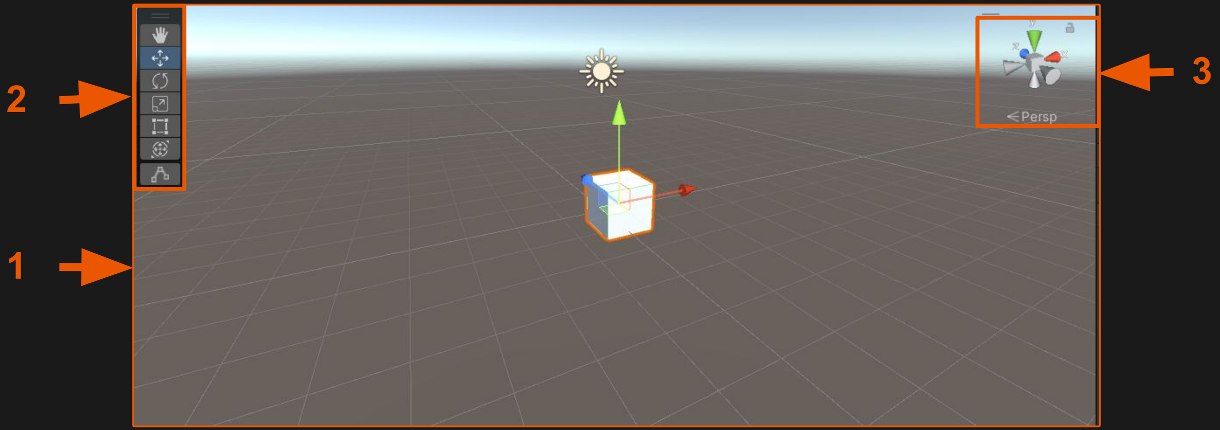
Scene View



The [1] Scene View allows you to visual see the game you are building, and the [2] menu allows you to manually edit the Game Objects in the scene. In this scene we can see two Game Objects, the Cube and the Light Source.

Additional Resources: Pages 14-15 of the Textbook, The Scene View - Unity Official Tutorials

Elements inside the Scene View

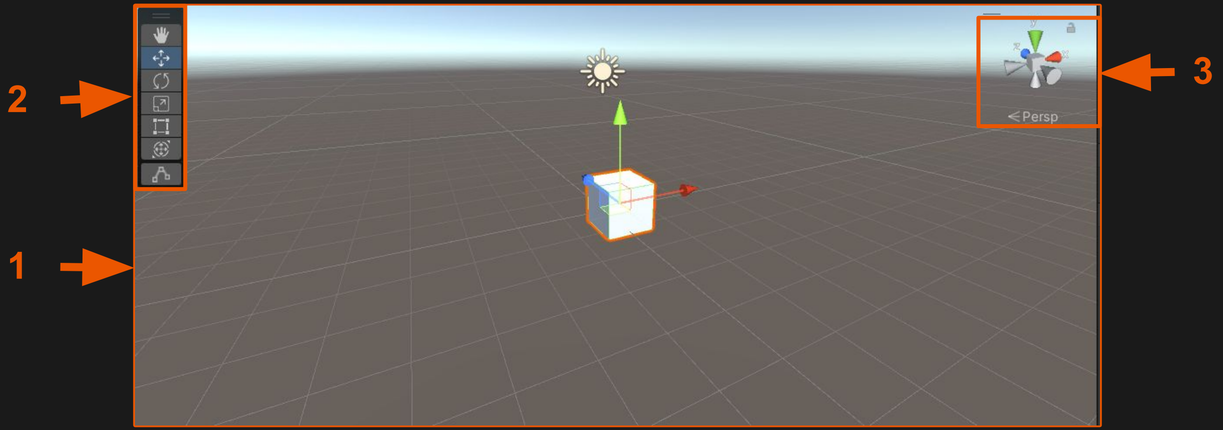


[1] The Scene View, here you can fly/move around the scene or level and edit the position, rotation, and scale of Game Objects using [2] Transformation Tools Bar.

[2] Transform Tools Bar, allow you to manipulate the Game Object that are within the Scene View.

[3] Scene Gizmo, allows you to snap to different planes to see the game at different angles.

Scene View



[2] Transform Tools Bar, this collects different tools that will help you navigate the scene and interact with Game Objects. These tools are are the Hand Tool, the Move Tool, the Rotate Tool, Scale Tool, and Rect Tool.

Hand Tool/Scene Navigation

When you have the Hand Tool Selected, you can freely explore your scene. There are many button combinations that will allow you to change the way you are navigating.

Holding the Left Mouse Button will let you move along the current 2D plane, allowing you to move left,right, up, and down.

Holding the Right Mouse Button will enter you into Fly Mode which will allow you to use WASD to move freely around the world.

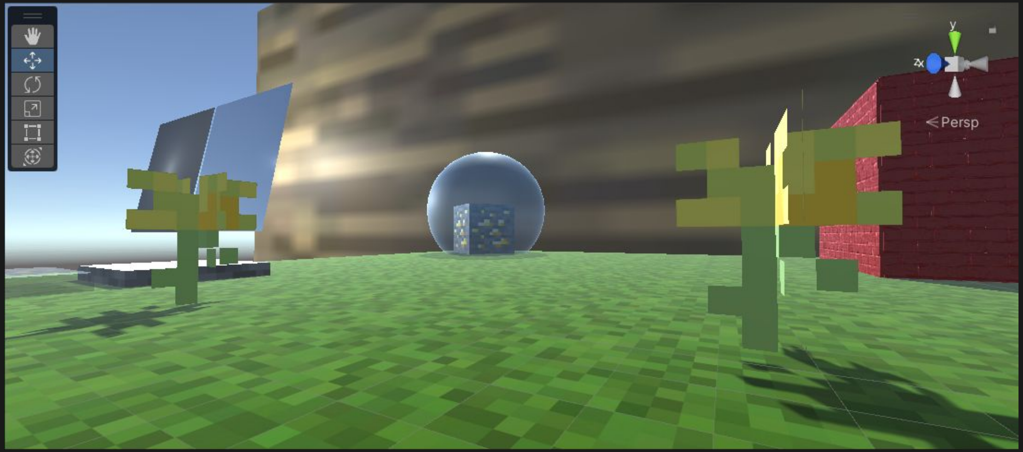
Holding Left Alt locks you in your current position and using the Left Mouse Button you can now rotate around that point or using the Right Mouse Button you can zoom in or out.

You can also zoom in or out using the Mouse Wheel at any point.

Additional Resources: Pages 18-20 of Textbook

Navigation Challenge

Navigate around the Scene and Recreate this Screen Shot

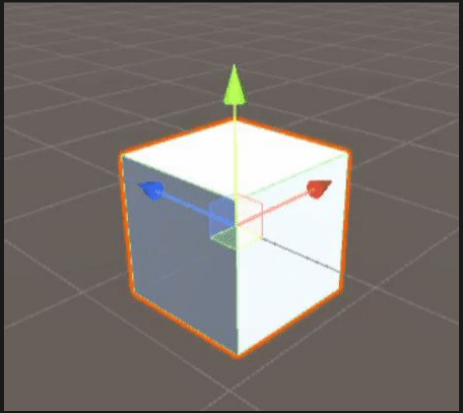


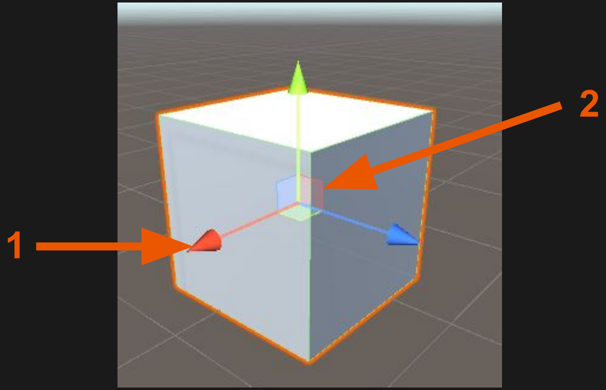
Transform Tools - Move Tool

The move tool allows you to change the position of the Game Object you are currently looking at. You will know what Game Object you are currently view as it will show the handles.

[1] The arrows allow you to move the object along individual axis, X,Y, Z and red, blue, green respectively. This will correspond to the colors on the Scene Gizmo.

[2] The planes will allow you to move the Game Object along the three different plains. XY, XZ, and YZ, and blue, green, and red respectively.





Additional Resources: Pages 32-33 of Textbook

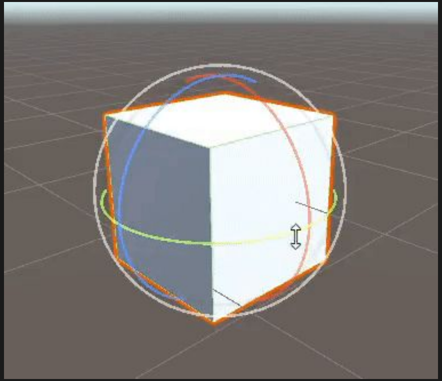
Transform Tools - Rotate Tool

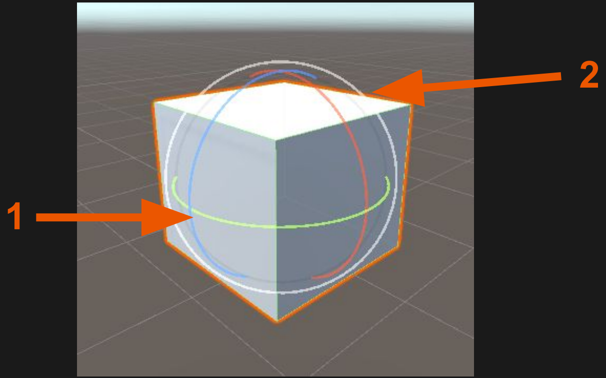
The rotation tool allows you to rotate the object you are working with.

[1] The three rings are broken down into individual colors of red, green and blue, and allow you to individually adjust the rotation of X, Y, and Z respectively.

You will notice a yellow arc show up as you are rotating, it shows you the amount of degrees you rotated from your original rotation.

[2] The White Ring will rotate the around the Scene View Z Axis





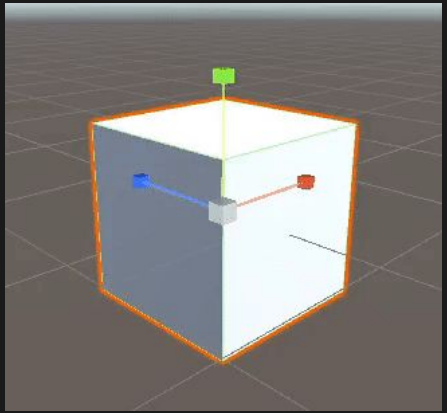
Additional Resources: Pages 33-34 of Textbook

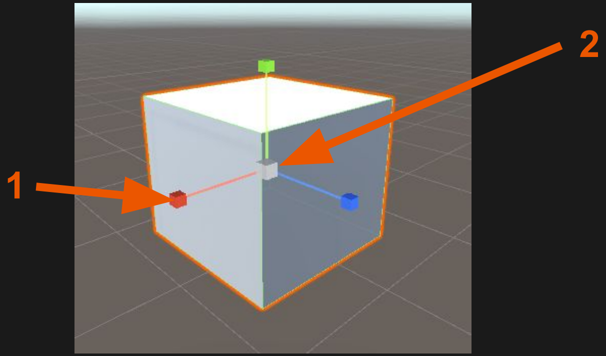
Transform Tools - Scale Tool

The scale tool allows you to modify the size of each side of the Game Object that you are currently looking at.

[1] The Arrows allow you to modify each of the axis individually X, Y, Z with red, green and blue representing each respectively.

[2] The Middle Notch will allow you to scale all three axis at the same rate.





Additional Resources: Pages 34-35 of Textbook

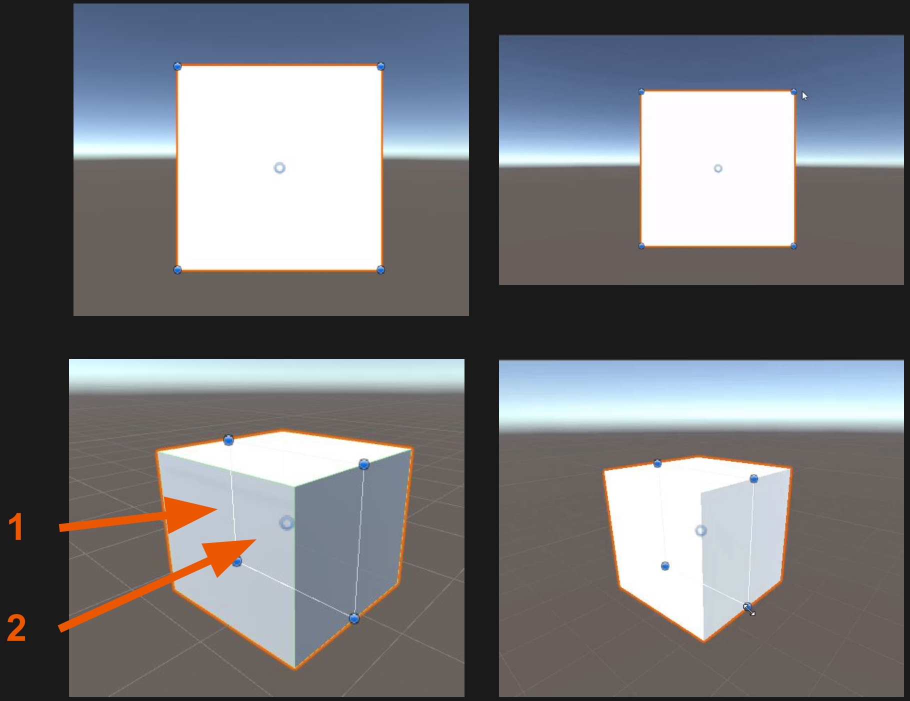
Sebastian Grygorczuk - STEM Institute at CCNY

Transform Tools - Rect Tool

The Rect Tool combines all three of the previous tools into one, allowing you to translate, rotate and scale the object all at once.

[1] As you can see however the Rect Tool works on a plane, so you will only be modifying the object in XY, XZ, or ZY planes therefore it's relegated for use on 2D elements such as 2D game object or UI elements.

[2] Pivot Point is the center of the object and is the point the object rotates around.

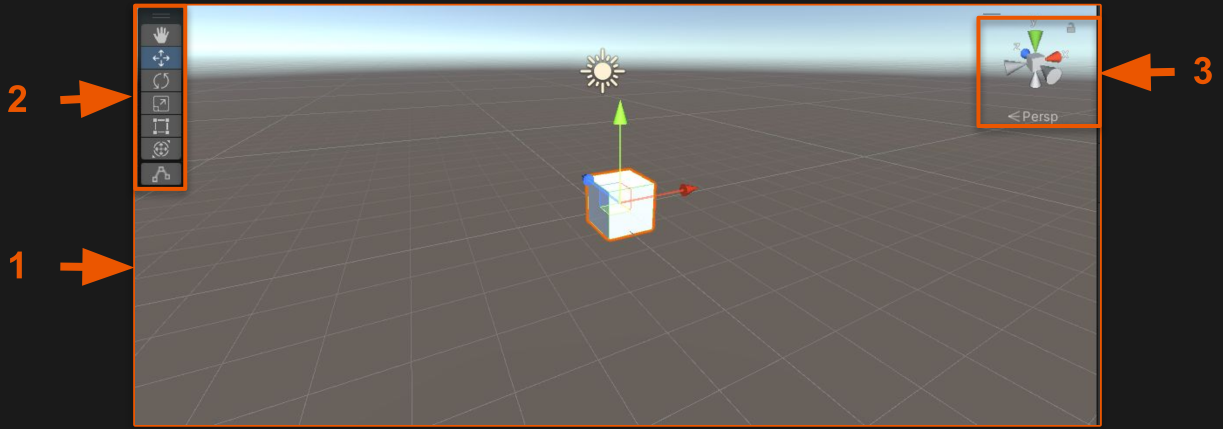


Tool Hotkeys

All of these functions are mapped to Hotkeys. If you would rather quickly swap between settings instead of selecting from the menu, here's a helpful chart.

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

Scene View



[3] Scene Gizmo allows you to quickly snap to different planes XY, XZ, ZY, -XY, -XZ, and -ZY in addition to changing the camera view from perspective to isometric and likewise. This can be incredibly helpful in transforming Game Objects as manipulating them in 3D space

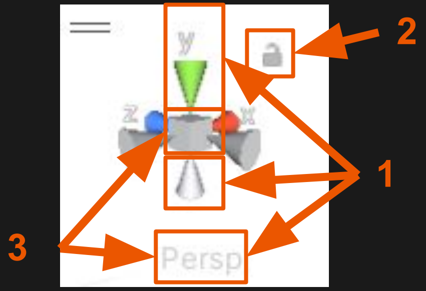
Scene Gizmo - Six Planes

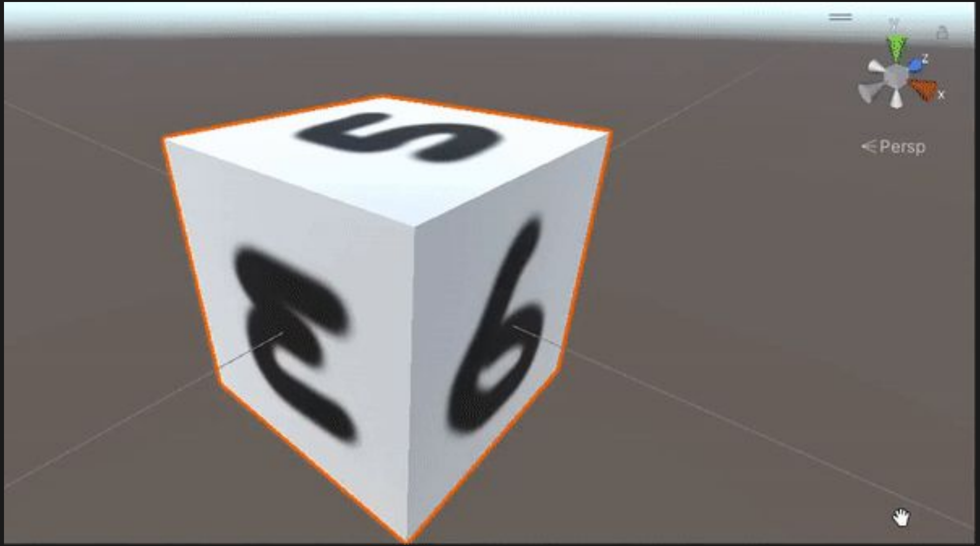
Transforming Game Objects in 3D space can be very hard to orient and translate. Thus snapping to a 2D axis could allow us to much easier move the object to desired position, rotation or size.

[1] There are six cones, ones for the positive X, Y, and Z axes and ones for the negative X, Y, and Z axes. Clicking on the Y cone would snap you to the XZ plane.

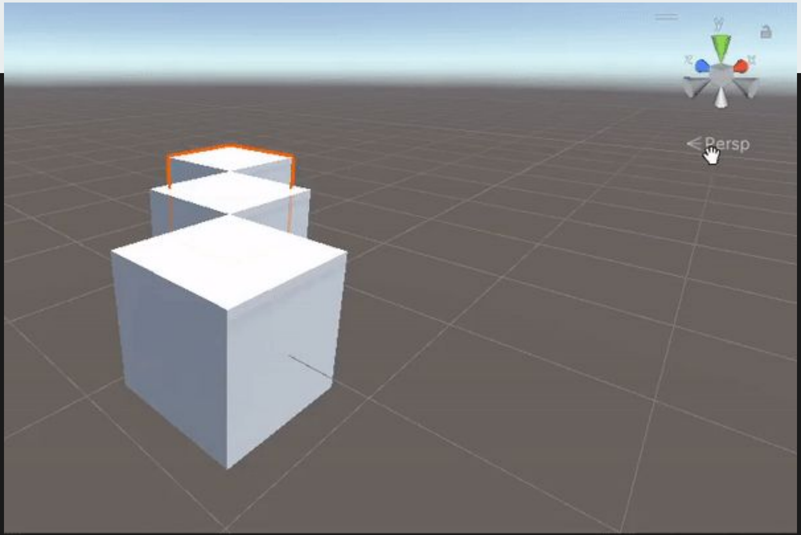
There is a text indicator that helps you figure out the orientation, while performing the snapping you will go through Top (ZX), Front (-XY), Back(XY), Left (-ZY), Right (ZY) and Bottom (-ZX).

[2] You can lock the view to the current rotation in case you don't want to accidentally move the scene camera.





Isometric Vs Perspective

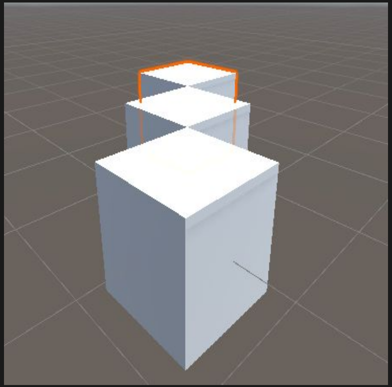


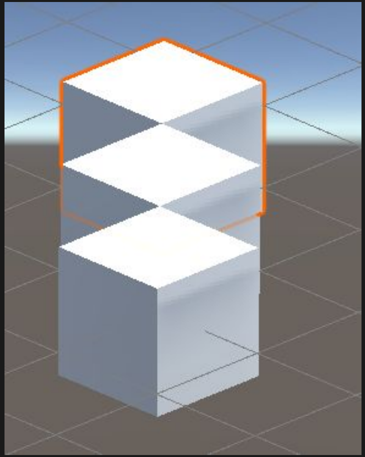
[3] You may have noticed that before we snapped to a single plane the Scene Gizmo indicated that we were in Persp, or Perspective mode.

Perspective is the way we view the world, the further a object is from us the more foreshortened are it's dimension are in relation to us.

Isometric on the other hand does away with that distortion by keeping the projects on all three axis equal. This is helpful when we want to have a detailed drawing but we lose Depth in the process.

To switch between these you can tap on the Cube in the middle of the Scene Gizmo or on the text.

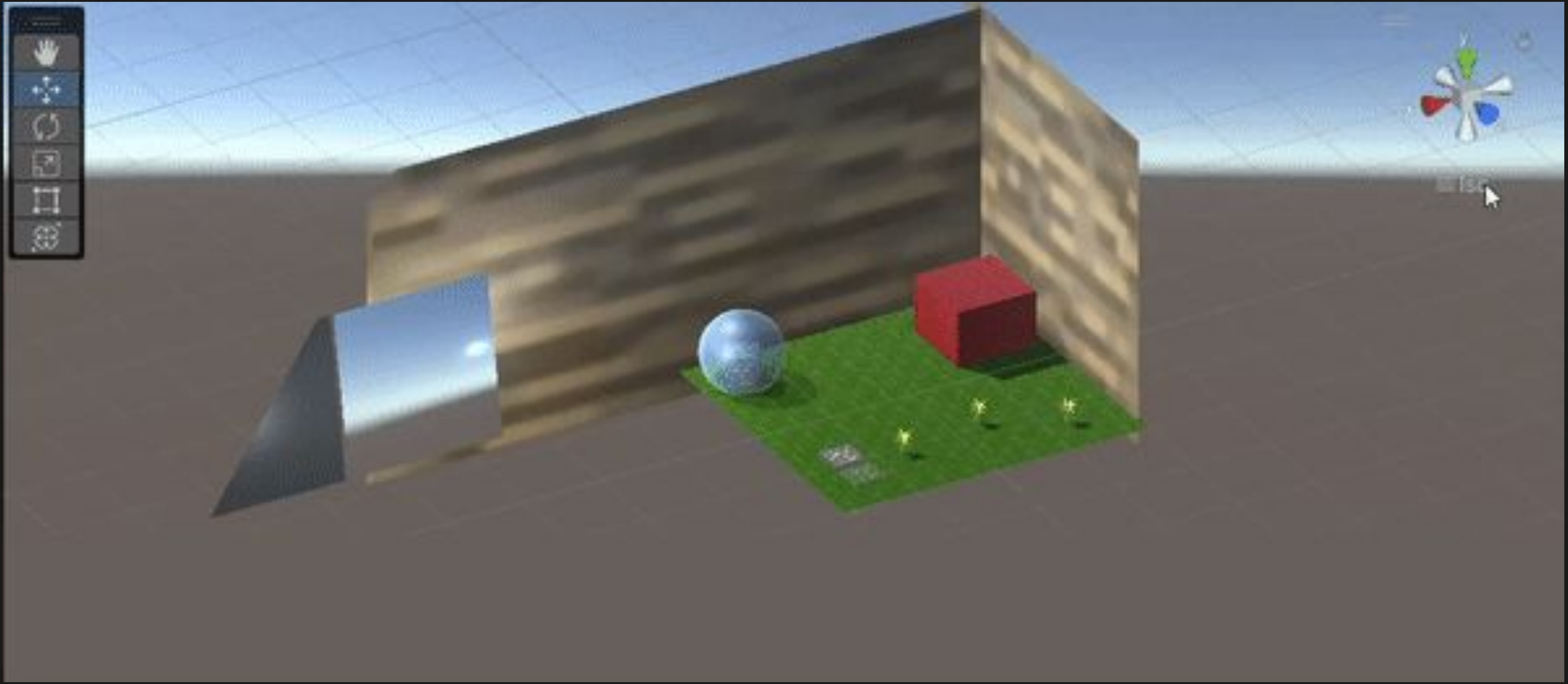




Additional Resources: Isometric vs Perspective

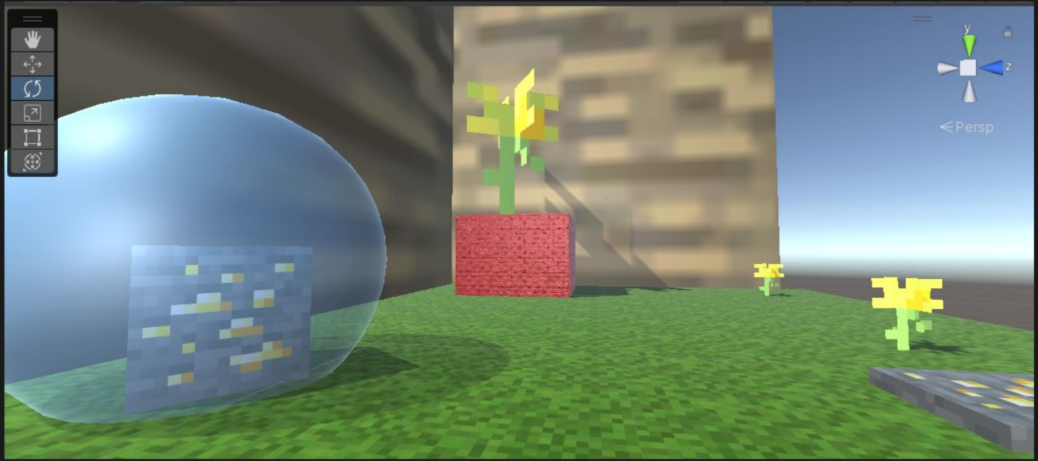
Sebastian Grygorczuk - STEM Institute at CCNY

Isometric Vs Perspective Example

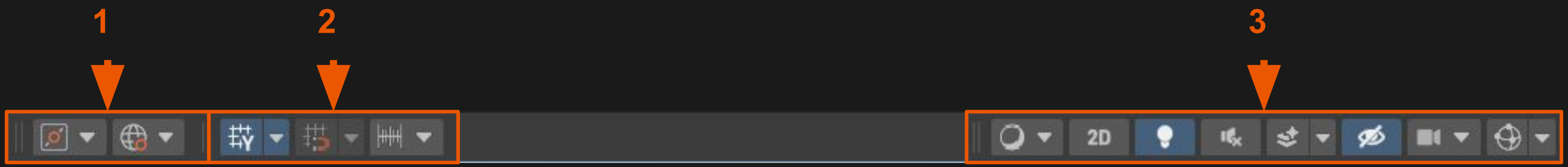


Challenge - Transformation

Scale one of the flowers to be much bigger, Rotate it and Translate on top of the building.



Scene View Control Bar



The Scene View Control Bar breaks down into three tools sets, these will further help us manipulate the object to our desire and display the scene that is best at the moment.

[1] Tool Settings, Modifies how the tools work

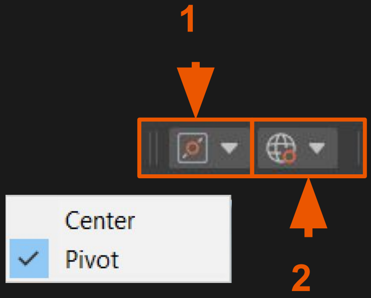
[2] Grid and Snap, Controls the Grid

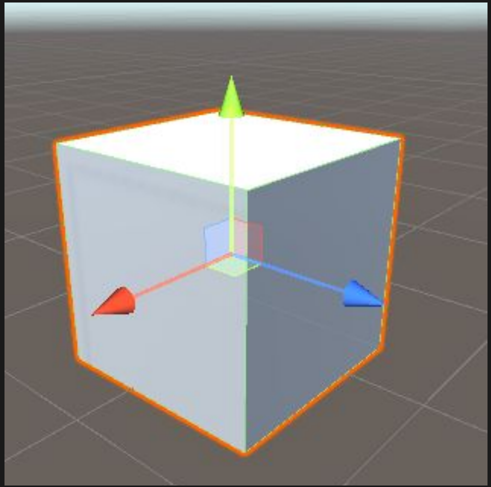
[3] View Options, Controls how the Scene is show to you

Tool Settings: Pivot

[1] Handle Position Toggle lets you switch where you'd like the Transform Gizmo to show up on the Game Object. So far we have seen it always show up in the center of the Game Object, this act as the origin point for the Game Object.

However, if you import an item from a 3D modeling software it may have a pivot point that's not it's center. Unity allows you to use either one as your preferred origin point.

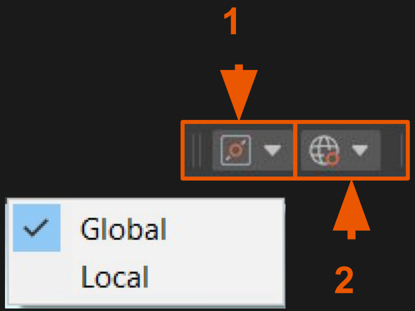


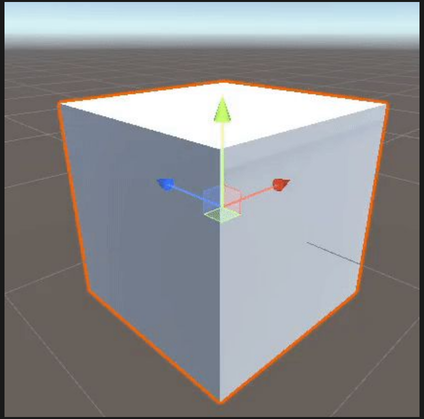


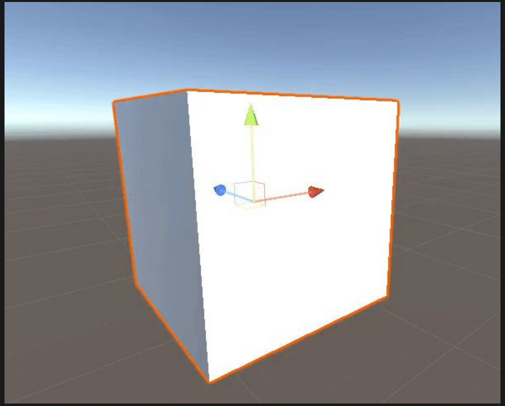
Tool Settings: World vs Local Coordinate

[2] As we have been transforming our Game Objects they always referred to their Origin Point as (0,0,0). This means they are abiding by the World Coordinate System. Each object also has their own coordinate they keep track of, mainly in respect to its rotation.

This is a very useful feature for character control. Rather than having to adjust which way you're moving in accordance to the world you can just program to move forward and using the Local Coordinates you just move in the axis you deemed as forward.




World Coordinates


Local Coordinates

Additional Resources: Pages 28-29 of Textbook, Unity Tutorial - Local Space Vs World Space in Unity

Sebastian Grygorczuk - STEM Institute at CCNY

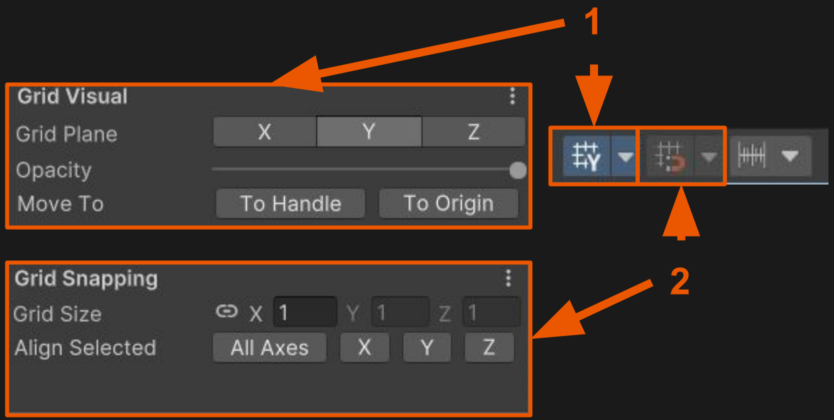
Grid and Snap

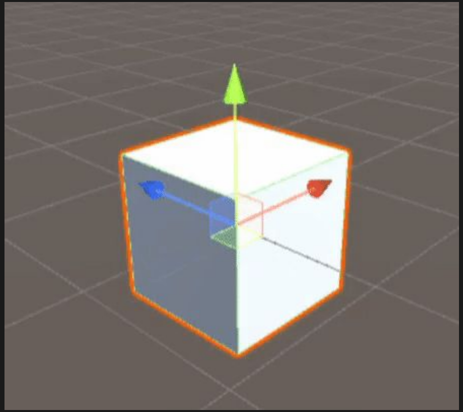
[1] The Grid Visual allows you to toggle the Grid On and Off, set the axis on which you'd like it to show up on and adjust the opacity [transparency] of the grid.

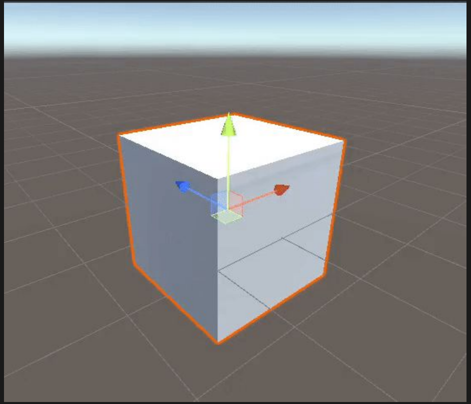
[2] Grid Snapping allows you to toggle on and off if you'd like your Move Tool to snap to the points along the grid. You can change the distance between each point.

Grid Snapping is only available while you are translating the Game Object in World Coordinates.

Using the grid you can have better time placing items precisely.

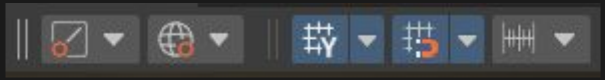



Snapping Off

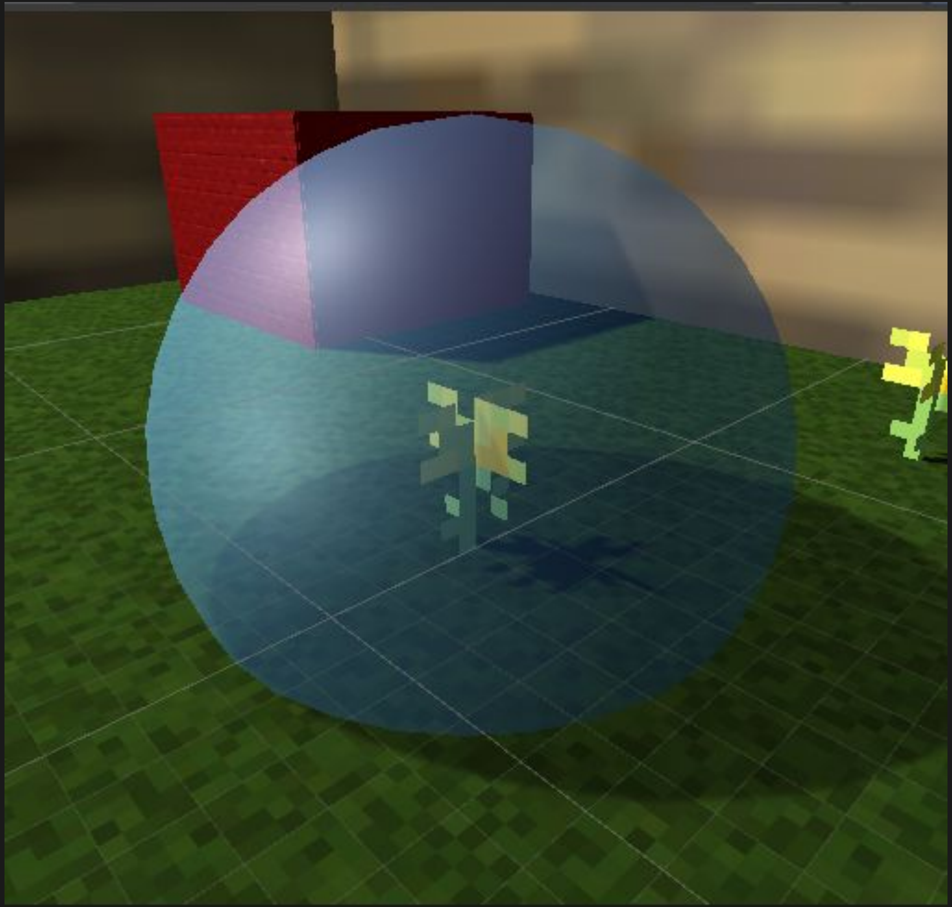

Snapping On

Sebastian Grygorczuk - STEM Institute at CCNY

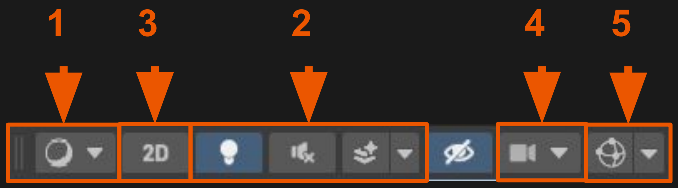
Challenge - Turn on Grid



1. Make sure you’re set in the Global Coordinate System
2. Select the Move Tool
3. Make sure that the Grid Toggle and Grid Snapping are Toggled on,
4. Move the Glass Sphere over one of the Flowers



View Options



The View Options give you full control over how and what is displayed in the scene.

[1] Draw Mode picks how the objects should be rendered.

[2] Light, Audio and Other Visual Effects Toggles checks how the scene works.

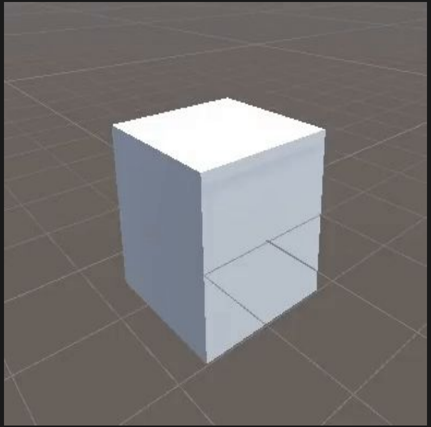
[3] 2D Toggle flips between using the 3D space or using 2D space.

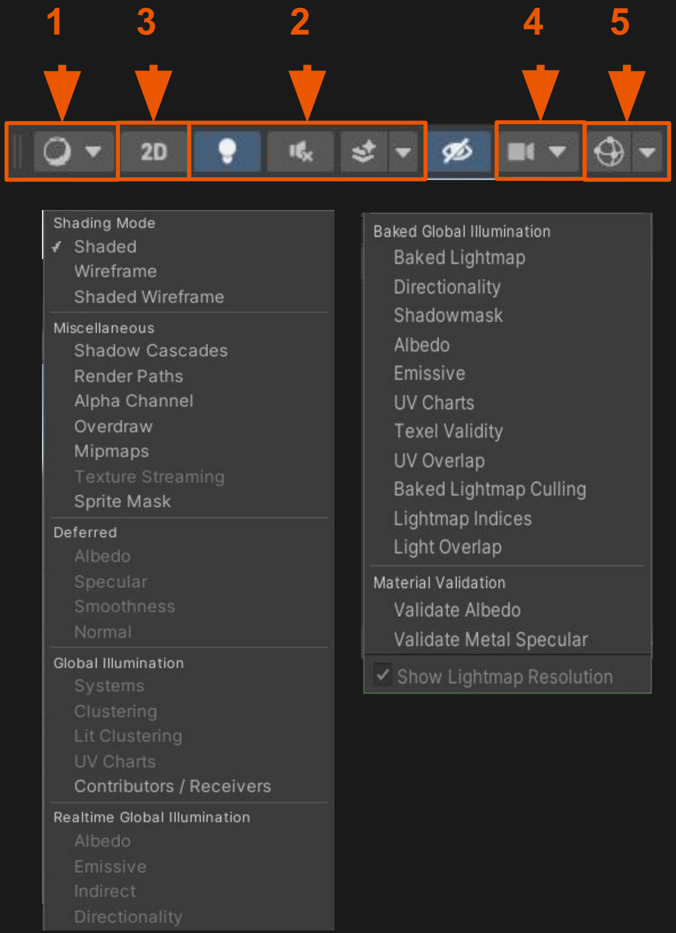
[4] View Scene Camera Options are camera settings that affect how we move around in the Scene View.

[5] Gizmo Visibility Toggle controls Icon display

View Options - Draw Mode

[1] Draw Mode allows you to switch between different version of rending the objects in the scene. For the most part we will stay in the default Shade Model option as all of these option pertain to advanced lighting ideas.





Additional Resources: Scene View Draw Modes For Lighting

Sebastian Grygorczuk - STEM Institute at CCNY

View Options - Lighting, Audio

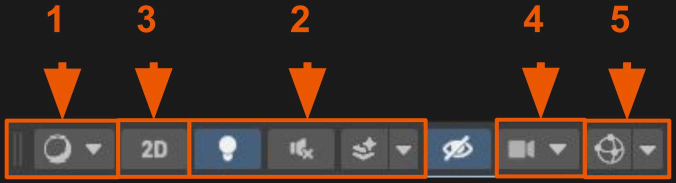
[2] These three toggles are pretty simple, you can toggle on and off

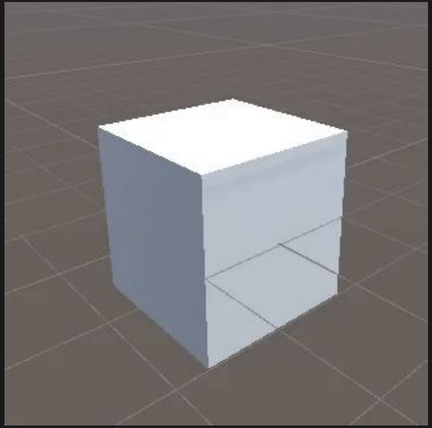
Lighting Effect

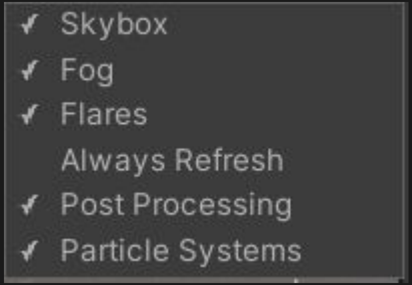
Audio Effects

Visual Effects

This can allow you to see what the Game Objects look like when unaffected by the environment.







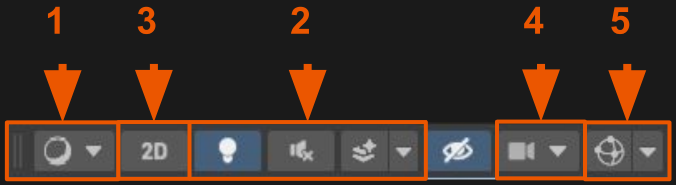
Turning Off/On Lighting and Visual Effect

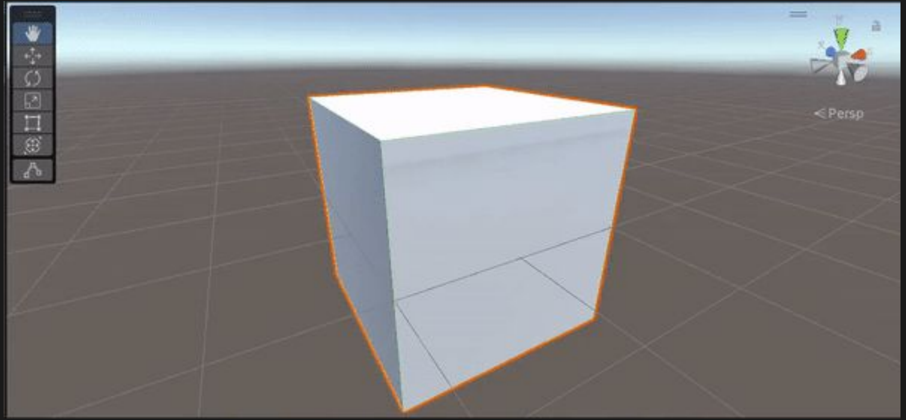
View Options - 2D Mode

[3] 2D On/Off Toggle turning this on will flatten your view against the 2D XY plane.

This will remove your ability to use the Scene Gizmo to snap or go into Isometric View.

This option will be turned on whenever creating a 2D game or working on UI.

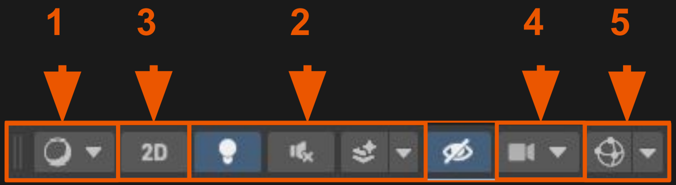


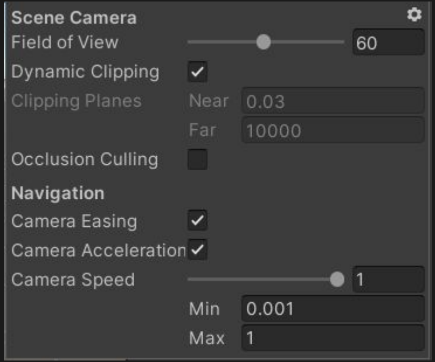


View Options - View Scene

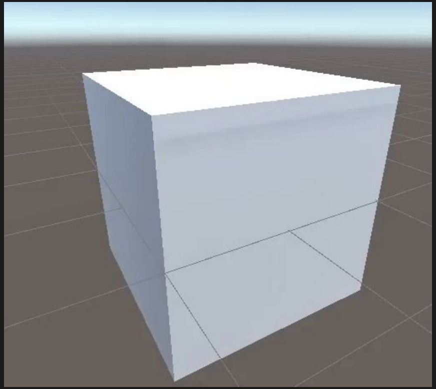
[4] View Scene Camera, allows you to control the Field of View. This dictates how near or far the vanishing point is. The smaller the Angle the closer to Isometric view we are the further the angle the more distorted the object is. Default is 60 degrees.

Navigation settings are there to help you with Fly Mode, Easing and Acceleration make the movement feel natural and you might want to set the max and current speed to whatever is comfortable It for you.





Scene Camera		
Field of View		60
Dynamic Clipping	✓	
Clipping Planes	Near	0.03
	Far	10000
Occlusion Culling		
Navigation		
Camera Easing	✓	
Camera Acceleration	✓	
Camera Speed		1
	Min	0.001
	Max	1



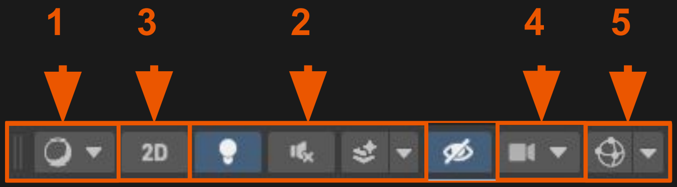
Additional Resources: Focal Length and Field Of View

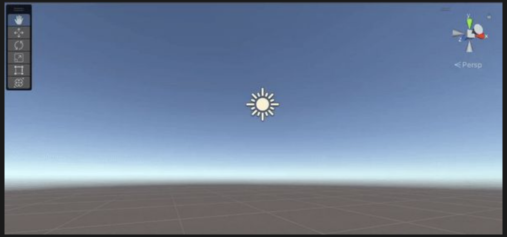
Sebastian Grygorczuk - STEM Institute at CCNY

View Options - Gizmo Toggle

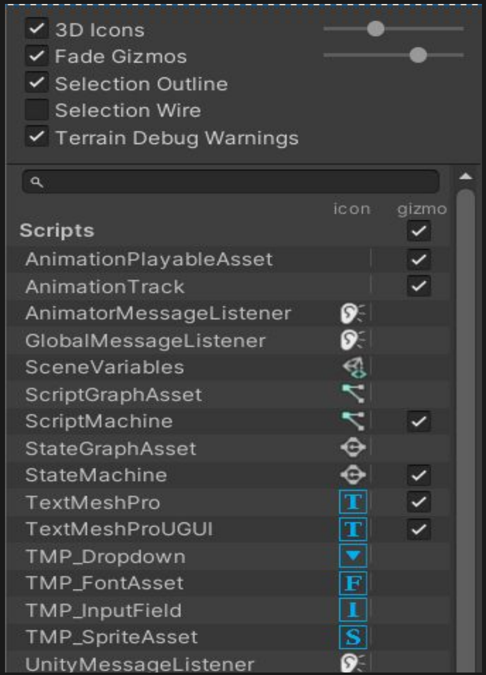
[5] Gizmo Toggle allows you to turn off and on visual for game object that may not have visual presence. You've already seen two of those, the camera and the light source both have icons that display their location in the scene.

As you can see there are many icons that could be on screen and you have full ability to change which are on and off.





One setting you might want to be aware of is the 3D Icon, toggling it on and off will leave them ay a standard view no matter where you are or keeping their size dependent on your position to it. a

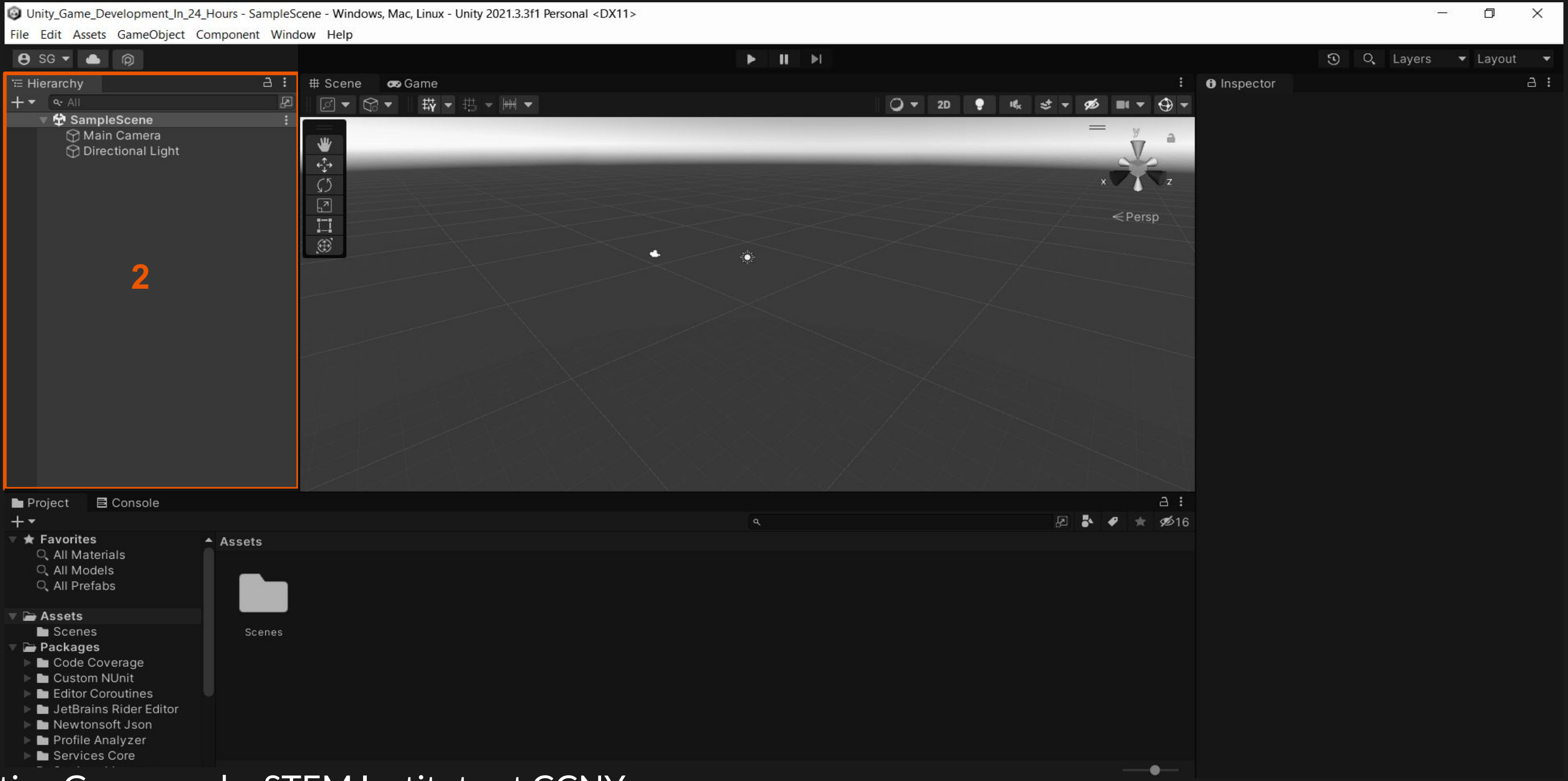


- ✓ 3D Icons
- ✓ Fade Gizmos
- ✓ Selection Outline
- Selection Wire
- ✓ Terrain Debug Warnings

	icon	gizmo
Scripts		✓
AnimationPlayableAsset		✓
AnimationTrack		✓
AnimatorMessageListener		
GlobalMessageListener		
SceneVariables		
ScriptGraphAsset		
ScriptMachine		✓
StateGraphAsset		
StateMachine		✓
TextMeshPro	T	✓
TextMeshProUGUI	T	✓
TMP_Dropdown	▼	
TMP_FontAsset	F	
TMP_InputField	I	
TMP_SpriteAsset	S	
UnityMessageListener		

Sebastian Grygorczuk - STEM Institute at CCNY

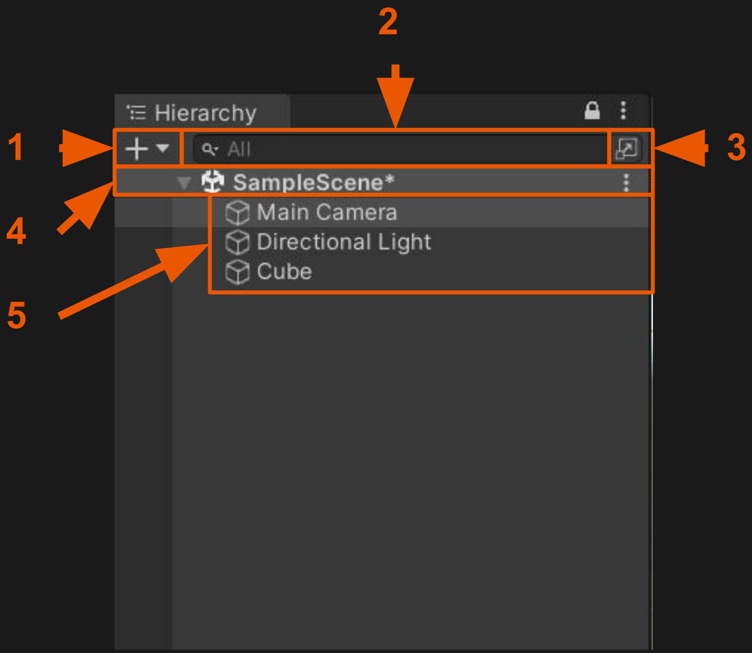
Hierarchy



Sebastian Grygorczuk - STEM Institute at CCNY

Hierarchy View

Hierarchy View displays a list of every Game Object that's currently in the scene. This View is invaluable when looking for anything as searching for Game Object through the Scene View could be almost impossible once the amount of Game Objects in a scene is large enough.



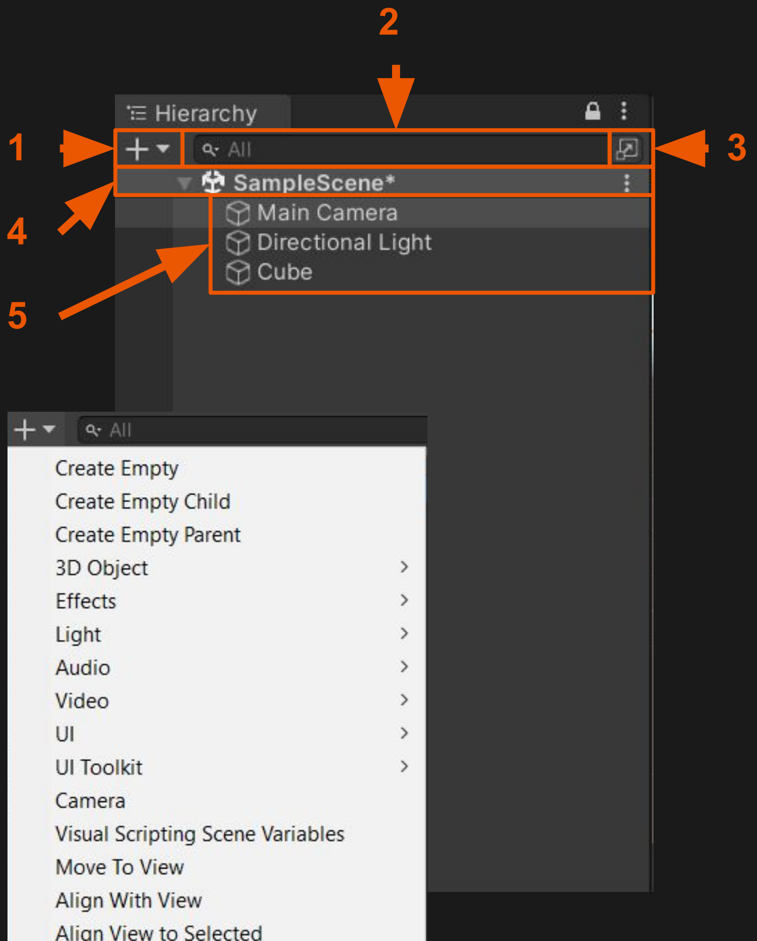
Additional Resources: Page 12 of the Textbook, The Hierarchy Window - Unity Official Tutorials

Sebastian Grygorczuk - STEM Institute at CCNY

Hierarchy View

[1] Create, this drop down menu allows you to create a new Game Object. You can create an Empty one or a Unity Premade One with select components. We will create many of these throughout this course.

You can also create a Game Object by right clicking on the empty space in the Hierarchy or clicking the Component button in the File Menu Header.

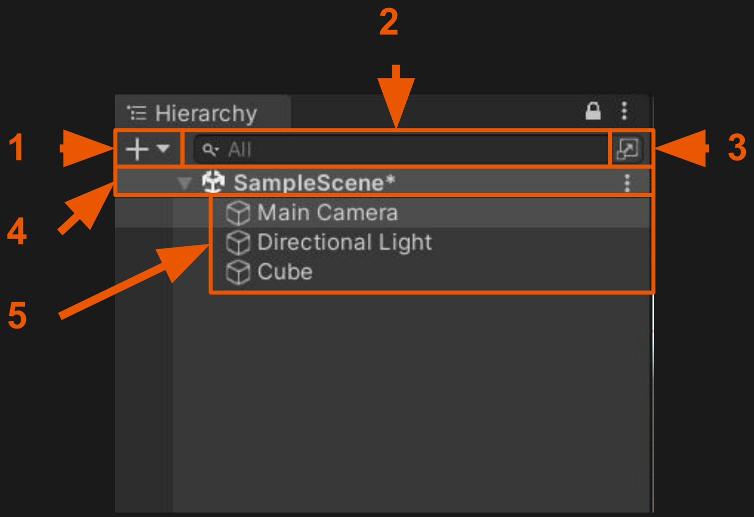


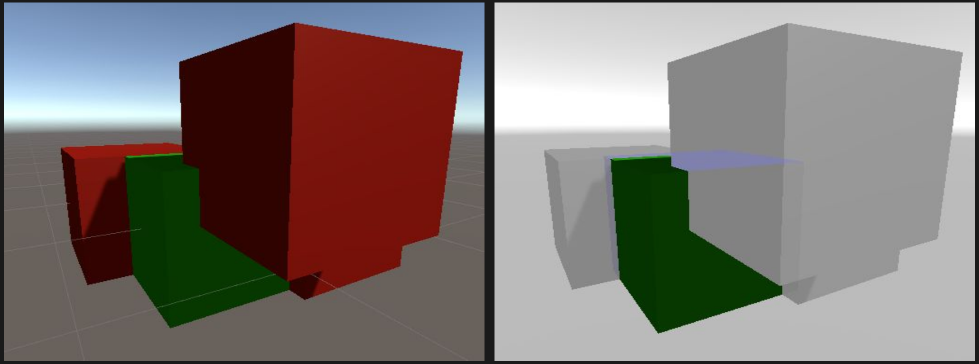
Sebastian Grygorczuk - STEM Institute at CCNY

Hierarchy View

[2] Search Bar, will help you find the Game Object, help you find anything that contains what you type into it. Thus if you type in "Cube", it will show you every Cube you have in the Scene. If you have a very specific Cube in mind, it's best to give your Game Objects descriptive names.

A wonderful feature of the search bar is that it work in conjunctions with the Scene View and will grey out any object that doesn't contain the word typed.

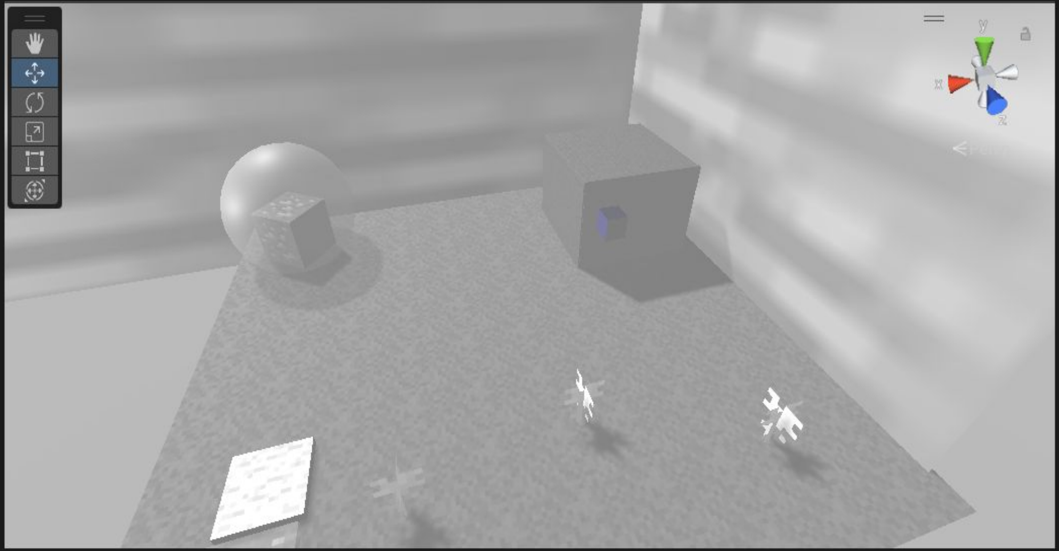




Sebastian Grygorczuk - STEM Institute at CCNY

Challenge Search

Using the Search Bar find a "Cube" and put it in the middle of the grassy field.

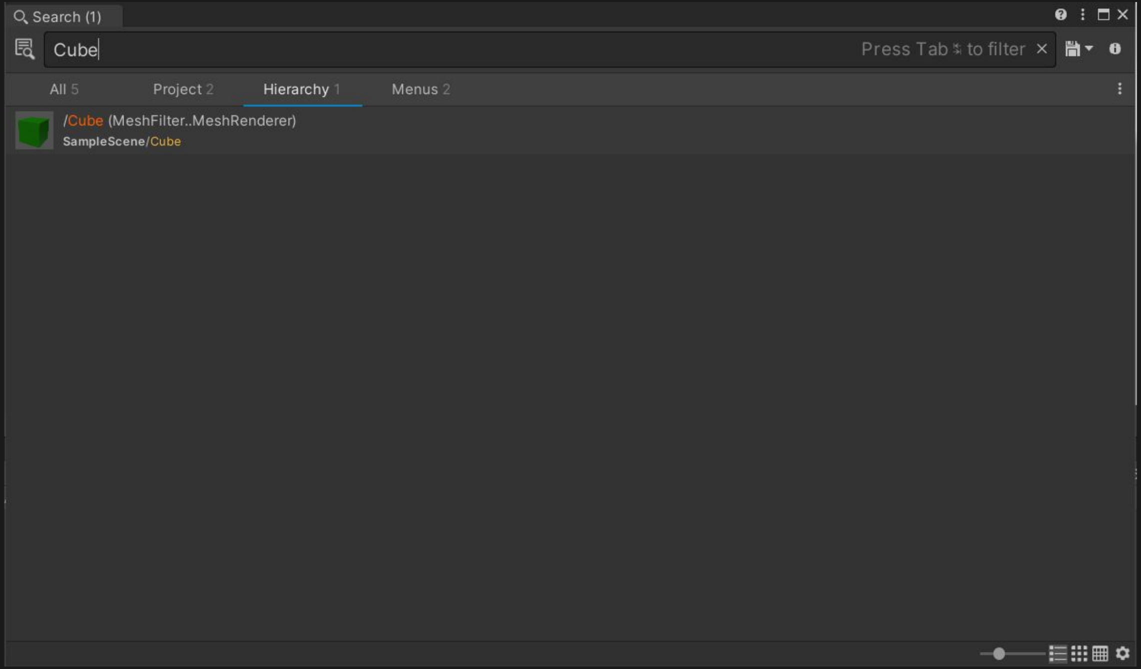


Sebastian Grygorczuk - STEM Institute at CCNY

Hierarchy View

[3] Advanced Search, clicking the button next to the search bar will bring up the Advanced Search Window. Here you can look for the Object across Projects, Scene, Hierarchy, and more.

For projects on our scale we won’t have to rely too much on this tool but it’s good to know about it.



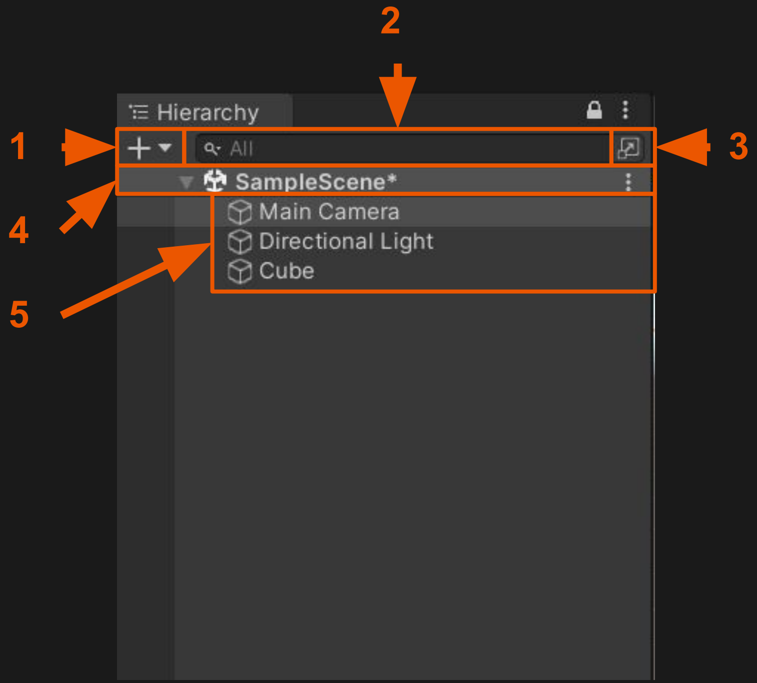
Hierarchy View

[4] Scene Header, this tells us which scene the objects are connected to. For the most part we will work with individual scenes, however in the future you may notice a new header popup when we work with data that's cross scene.

You may have noticed that there's a * next to the SampleScene, that means you've made changes that have not been saved.

Make sure to save your progress often as Unity doesn't have a built in auto-save function and if it crashes all your progress will be gone.

[5] Game Object List, shows you all of the Game Objects in the Scene.



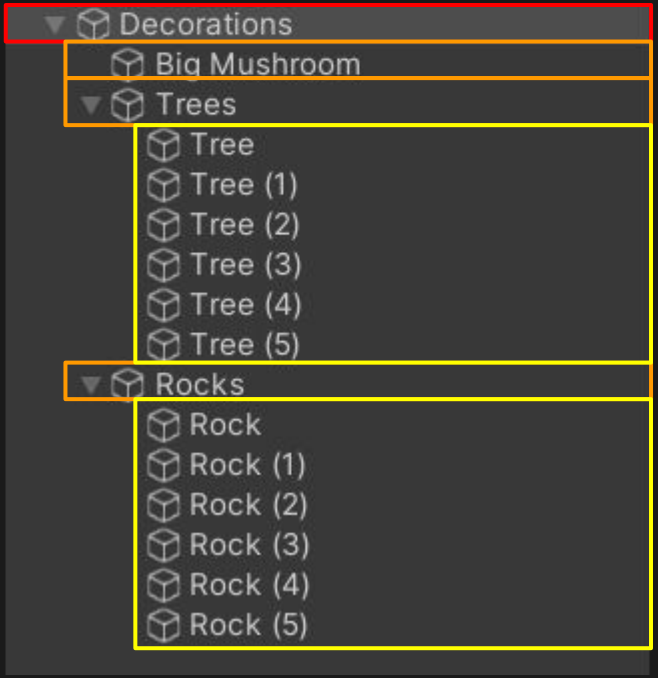
Nesting Game Objects

Nesting is the process of connecting Game Objects to other Game Objects. You can achieve this by dragging one object on top of another. You will know that the Object is nested as it will indent below it's parent.

The Game Object that we connected to is called the Parent and the Object that got connected is called the Child. So in this case Big Mushroom is a child of Decorations. Similarly Trees is a parent and has children Tree, Tree (1) … Tree (5).

Tree, Tree (1) … Tree (5), aren't direct children of the base of this Nesting, so they are called Descendants of Decorations.

Bring back World vs Local, when you nest a Game Object the parent object now becomes the world Origin point for it rather then the standard scene view one.



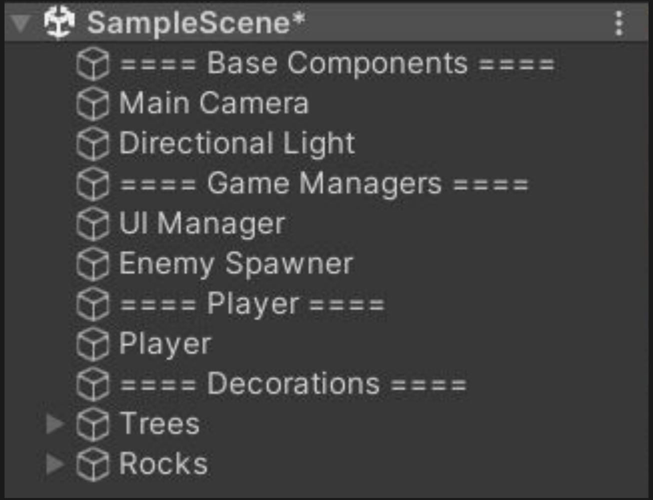
Keeping Hierarchy Organized

Beyond just just nesting a good practice is to put up dividers. These are just empty objects with some kind of symbol to make it easier for the eye to distinguish. It's not necessary for you to do it but your future self will thank you when you can find that one game object at a glance.

==== Divider ====

**** Divider ****

---- Divider ----

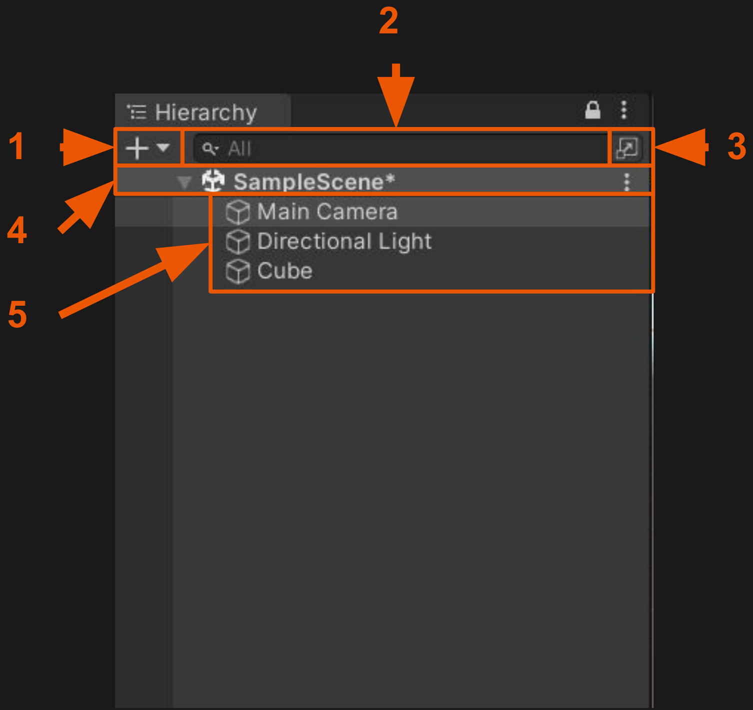


```
SampleScene*
    ==== Base Components ====
    Main Camera
    Directional Light
    ==== Game Managers ====
    UI Manager
    Enemy Spawner
    ==== Player ====
    Player
    ==== Decorations ====
  ▶ Trees
  ▶ Rocks
```

Finding Game Object in Scene View

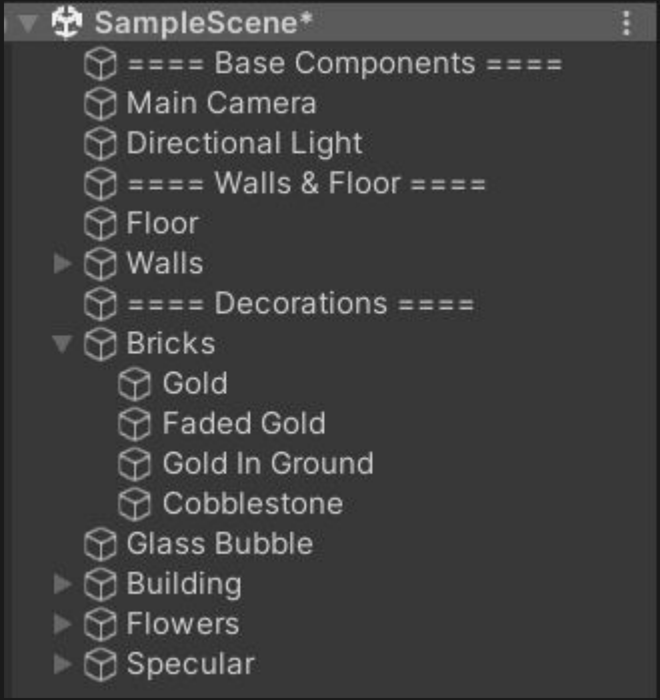
By double clicking on any of the Game Object the Scene View will automatically zoom in to have to object in view.

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



Hierarchy Challenge

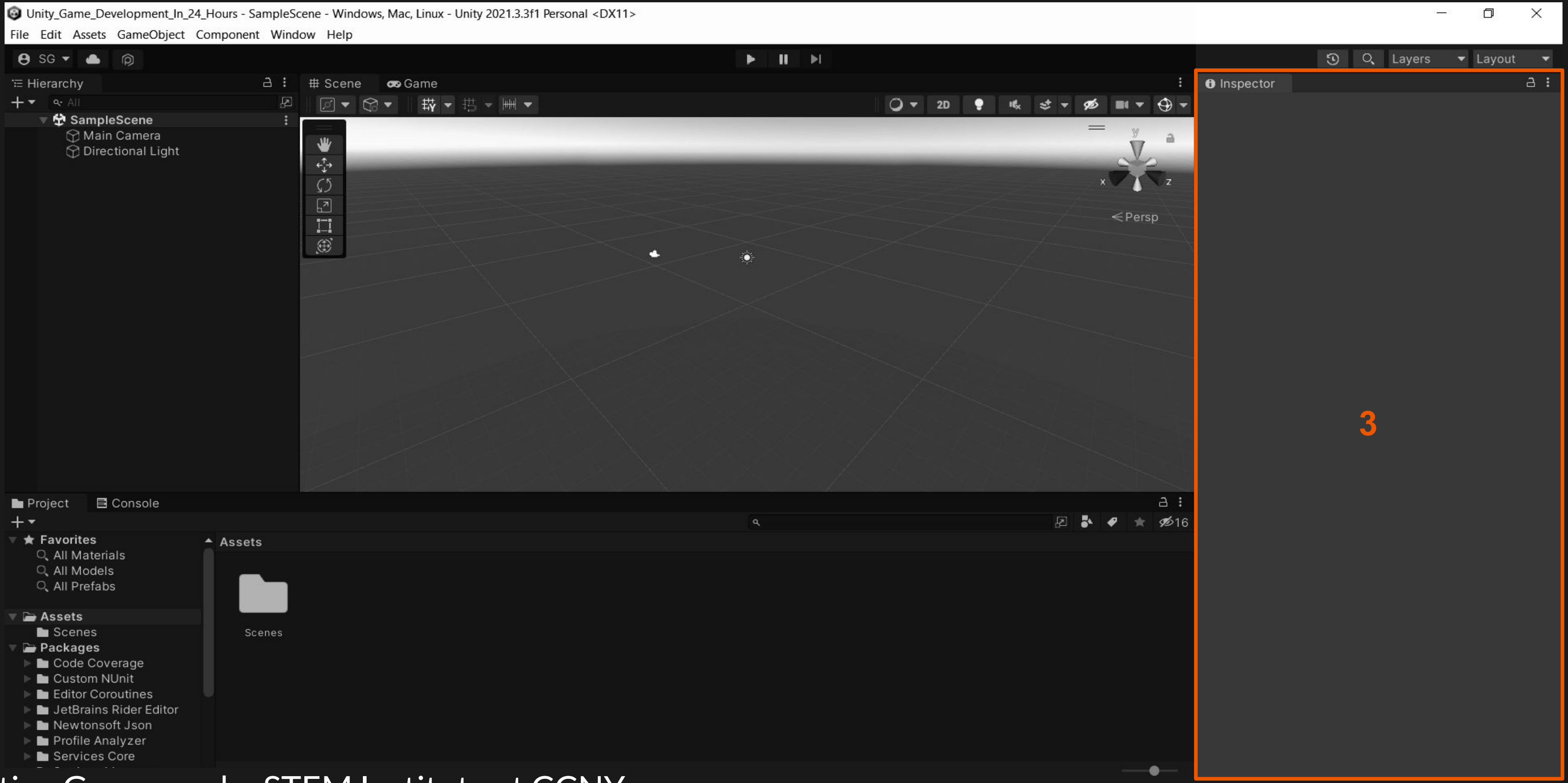
1. Make your Hierarchy looking similar to this and then
2. Using the double click method or by selecting a Game object and clicking F while hovering the mouse over the Scene View find Gold In Ground Game Object



SampleScene*
==== Base Components ====
Main Camera
Directional Light
==== Walls & Floor ====
Floor
Walls
==== Decorations ====
Bricks
Gold
Faded Gold
Gold In Ground
Cobblestone
Glass Bubble
Building
Flowers
Specular

Sebastian Grygorczuk - STEM Institute at CCNY

Inspector



Sebastian Grygorczuk - STEM Institute at CCNY

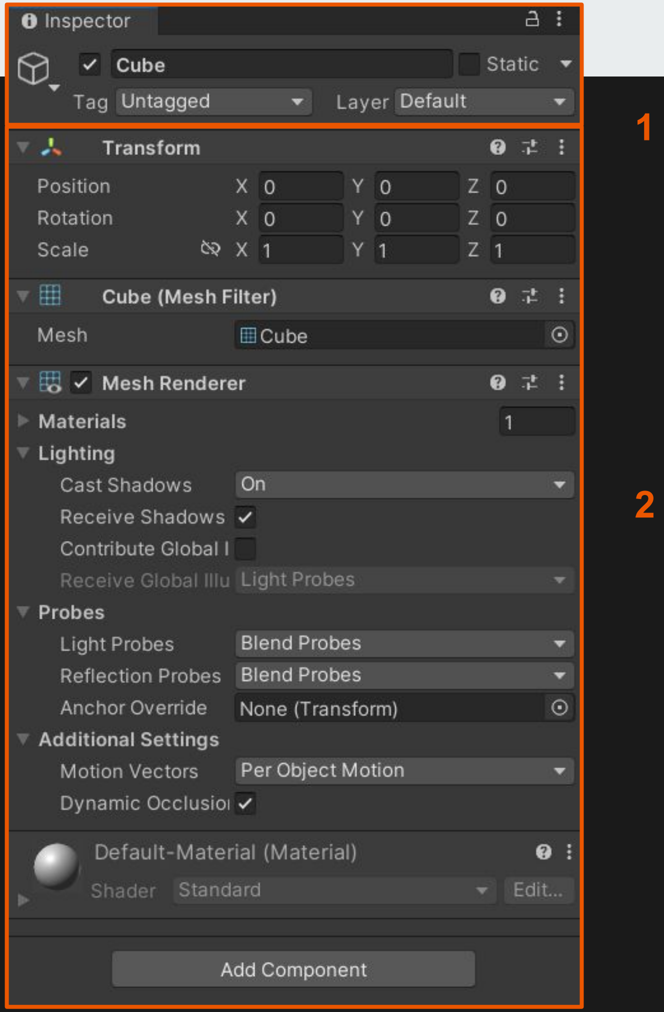
Inspector View

The Inspector View shows all of the properties of the selected object. In this instance it's a Cube.

With this we can see the the:

[1] Inspector View Overhead

[2] Game Objects Component List.



Inspector

✓ Cube — Static

Tag Untagged — Layer Default

Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

Cube (Mesh Filter)

Mesh: Cube

✓ Mesh Renderer

Materials: 1

Lighting
- Cast Shadows: On
- Receive Shadows: ✓
- Contribute Global I: ☐
- Receive Global Illu: Light Probes

Probes
- Light Probes: Blend Probes
- Reflection Probes: Blend Probes
- Anchor Override: None (Transform)

Additional Settings
- Motion Vectors: Per Object Motion
- Dynamic Occlusio: ✓

Default-Material (Material)

Shader: Standard — Edit...

Add Component

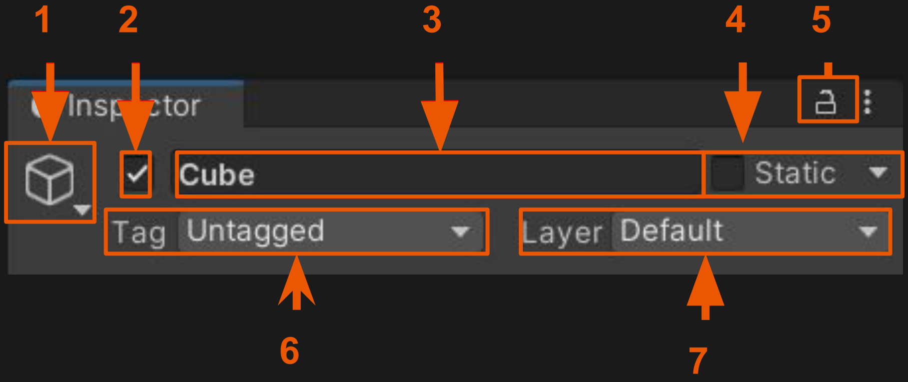
Additional Resources: Pages 13-14 of the Textbook, The Inspector Window - Unity Official Tutorials

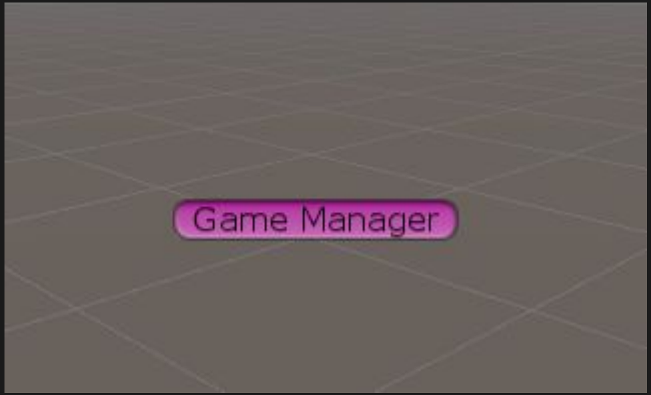
Sebastian Grygorczuk - STEM Institute at CCNY

Inspector View Header

There's a lot of thing here but don't worry most of them are very simple and very useful to us.

[1] Icon, all game objects can have an icon, but start without one; that's what the empty cube means. Given that game objects don't necessarily have a visual thing to represent them an Icon can be attached to find. These are mostly used for things like Game Managers and Spawn Points to help you visualize the scene better.

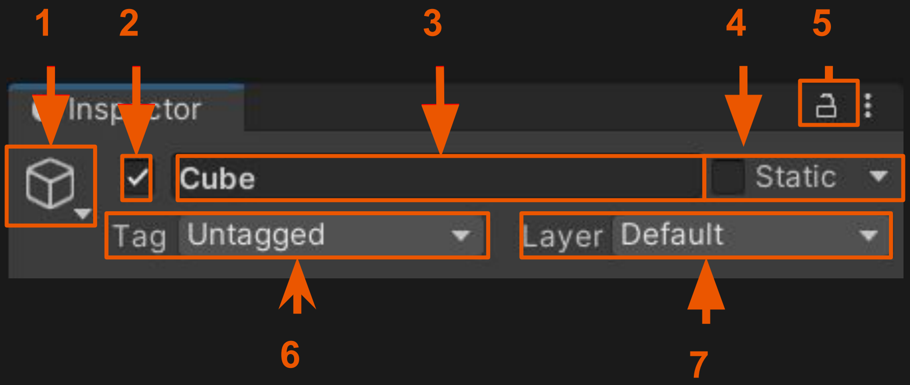




Inspector View Header

[2] Enable, the check mark represents that the Game Object is enabled, meaning it’s active in the scene. If the toggle is uncheck that means the Game Object will not appear in the scene and interact with the level.

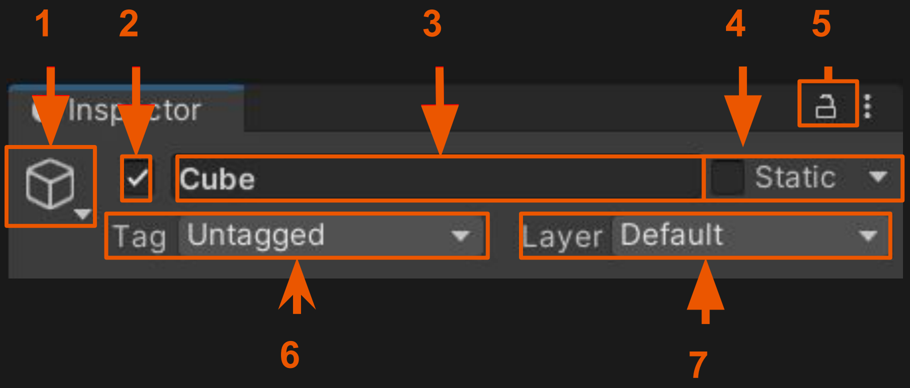
There are many reasons to turn of the Game Objects such a performance; if the physics and Rendering are turned off the computer will have an easier time calculating everything else happening in the game.



[3] Name, you can edit the name of object from here and it will be directly change in the Hierarchy View. Keep the names simple and direct to the point so that you can find it later when editing the Scene.

Inspector View Header

[4] Static, means the Game Object should be ignored by built in Unity systems such as lighting or navigation. You can set it to all or toggle specific systems you want to be ignore. We'll put a pause on this tool as it's not necessary for anything we'll be doing.



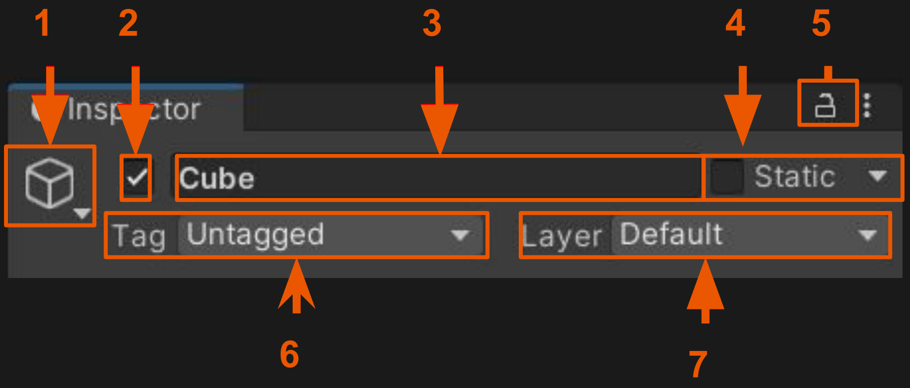
[5] Lock, this will lock the Inspector to the Game Object you are currently viewing. This is useful in situations where you might be working two or more objects and in so would end up having the Inspector switch between them but you want consistently edit the property on a light for example.

Sebastian Grygorczuk - STEM Institute at CCNY

Inspector View Header

These two allow us to distinguish what the Object is and how it should interact with the given environment.

[6] Tag, using tags we can use our Scripts to identify game objects and apply wanted behaviors; very effective for checking collisions and applying behaviors if any are required.

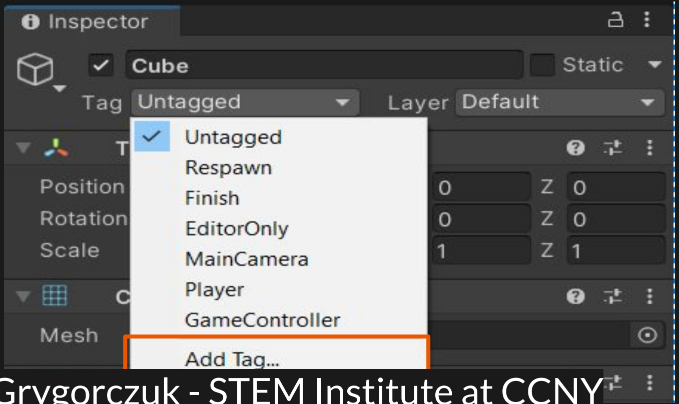


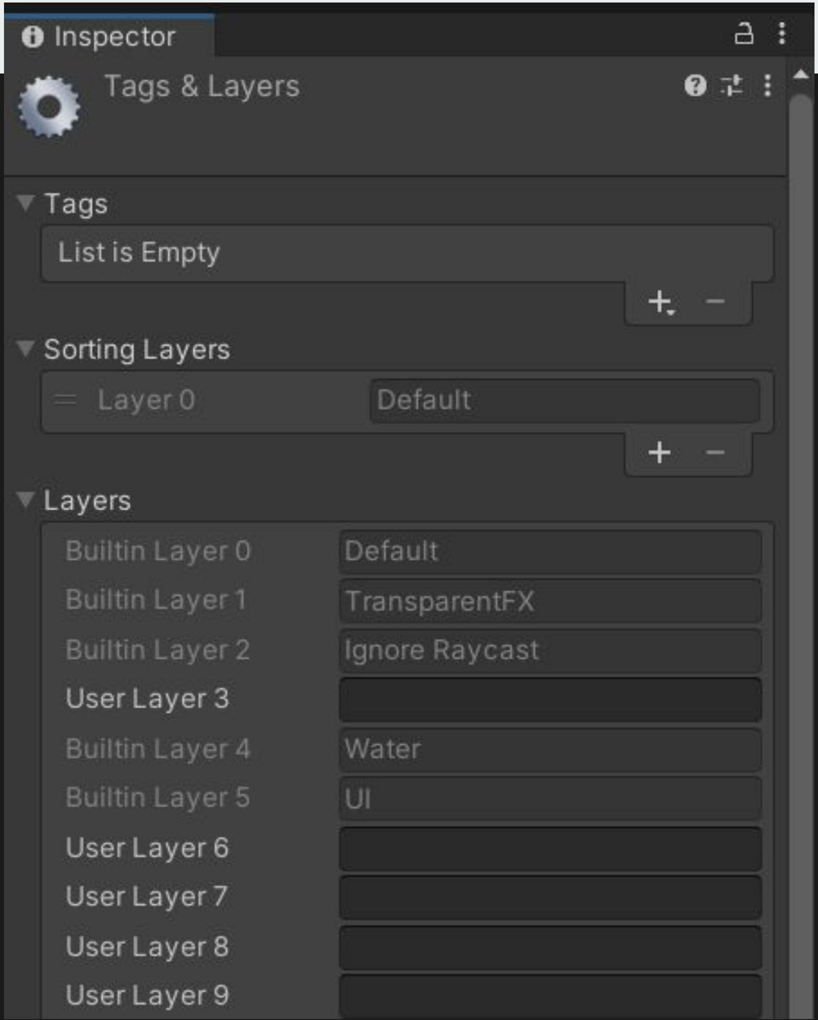
[7] Layer, are similar to Tags but rather than identifying Game Object for our script it's used for built in Unity tools such as lighting or navigation.

Inspector View Header

It's important to note that even though both Tags and Layers come with a predefined lists you can add your own categories.

You can access this list by clicking the drop down menus on the Tag or Layer drop list and selecting "Add Tag" or "Add Layer".





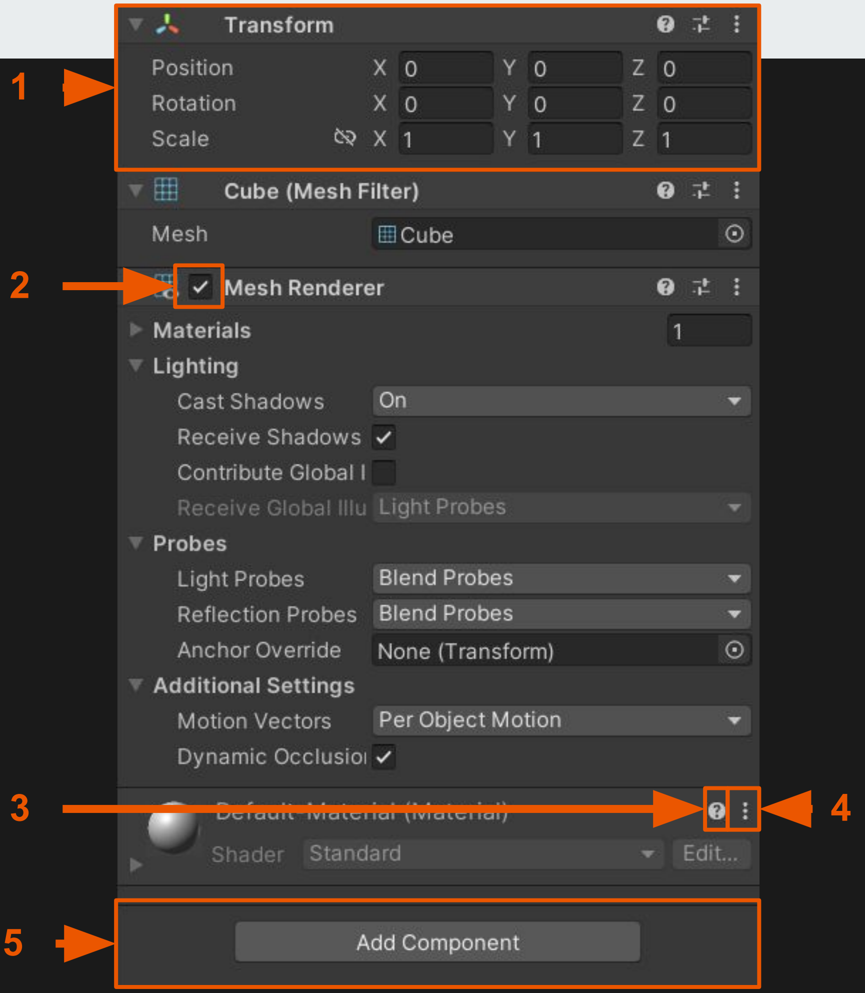
Builtin Layer 0	Default
Builtin Layer 1	TransparentFX
Builtin Layer 2	Ignore Raycast
User Layer 3	
Builtin Layer 4	Water
Builtin Layer 5	UI
User Layer 6	
User Layer 7	
User Layer 8	
User Layer 9	

Components List

The Component List will show you everything that's attached to the Game Object, from Renders, to Lights, to your own Scripts.

As you see the Cube has three game objects. The Transform, Mesh Filter, and Mesh Rendered. It might look like The Material underneath Mesh Render is a Component but that's part of the Mesh Render.

You can tell because all of the Components have a slight different color Header that holds a little arrow next to the name of the Component allowing you to collapse it if the properties aren't important to you at the moment.



	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

Mesh: Cube

Materials: 1

Cast Shadows: On
Receive Shadows: ✓
Contribute Global I
Receive Global Illu: Light Probes

Light Probes: Blend Probes
Reflection Probes: Blend Probes
Anchor Override: None (Transform)

Motion Vectors: Per Object Motion
Dynamic Occlusion: ✓

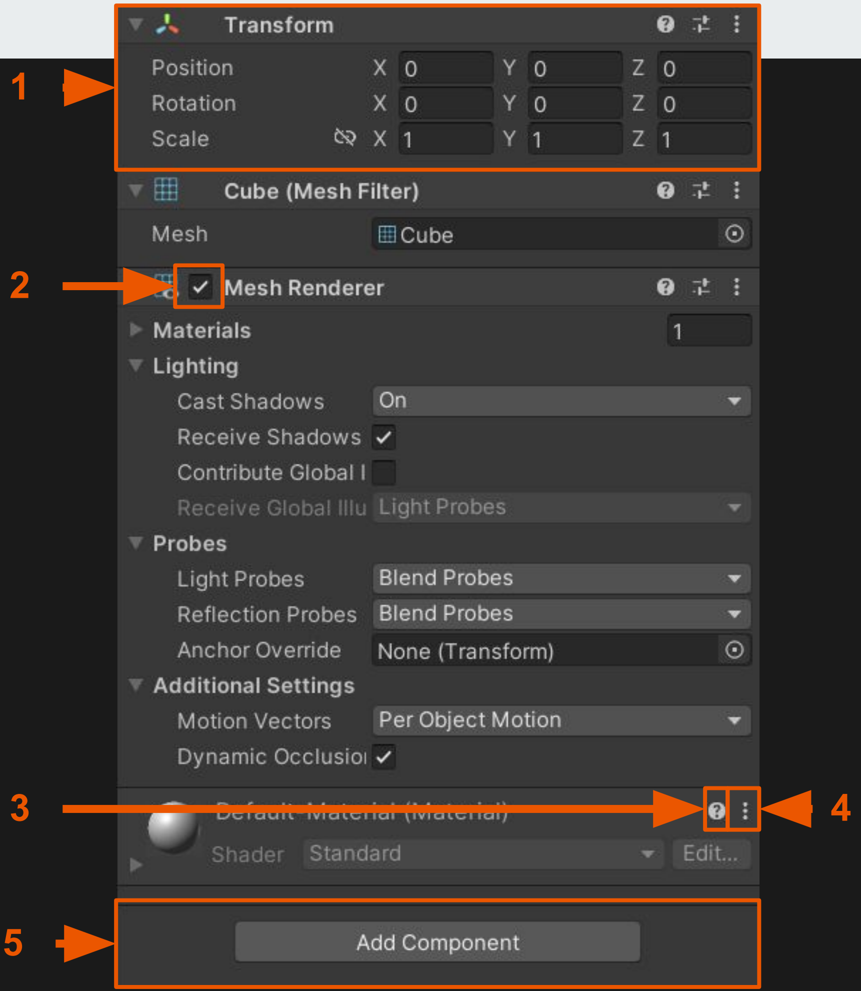
Shader: Standard — Edit...

Sebastian Grygorczuk - STEM Institute at CCNY

Components List

[1] Transform, every Game Object will come with a transform. It defines where it is in the Scene, which way it's rotated and how large of a space it takes up. Every Transform is also 3D, even when building 2D games the Z axis will be available to edit.

[2] Enable, notice that certain Components such as the Mesh Render have a similar Enable toggle like the Game Object has, this could allow you to keep the Render of a Game Object but turn of it's Collision, such that you can see it but nothing will touch it.



Transform

Position	X	0	Y	0	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

Cube (Mesh Filter)

Mesh: Cube

✓ **Mesh Renderer**

Materials: 1

Lighting

Cast Shadows	On
Receive Shadows	✓
Contribute Global I	
Receive Global Illu	Light Probes

Probes

Light Probes	Blend Probes
Reflection Probes	Blend Probes
Anchor Override	None (Transform)

Additional Settings

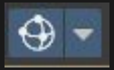
Motion Vectors	Per Object Motion
Dynamic Occlusion	✓

Default-Material (Material)

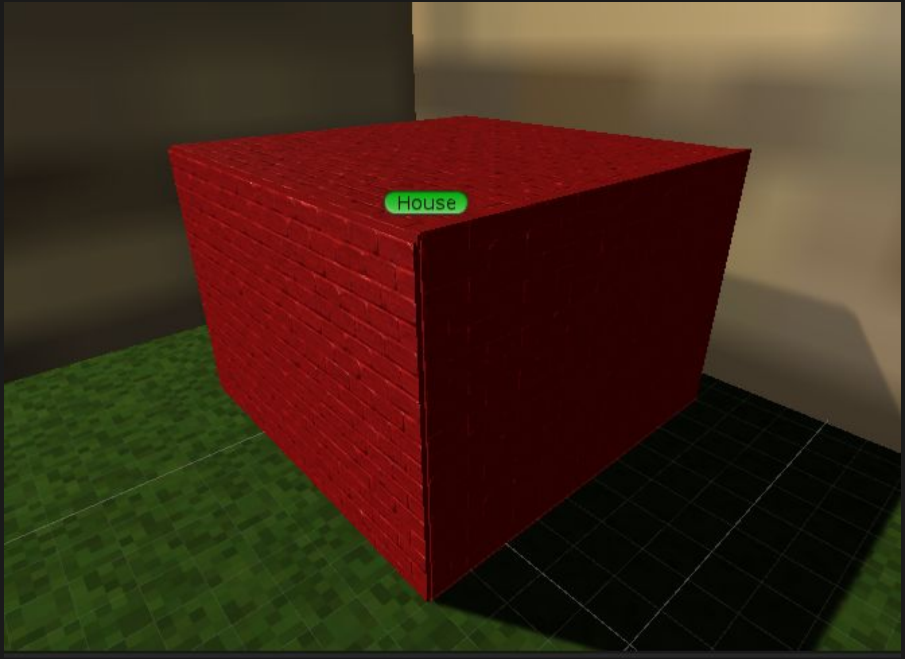
Shader: Standard — Edit...

Sebastian Grygorczuk - STEM Institute at CCNY

Challenge: Create and Move



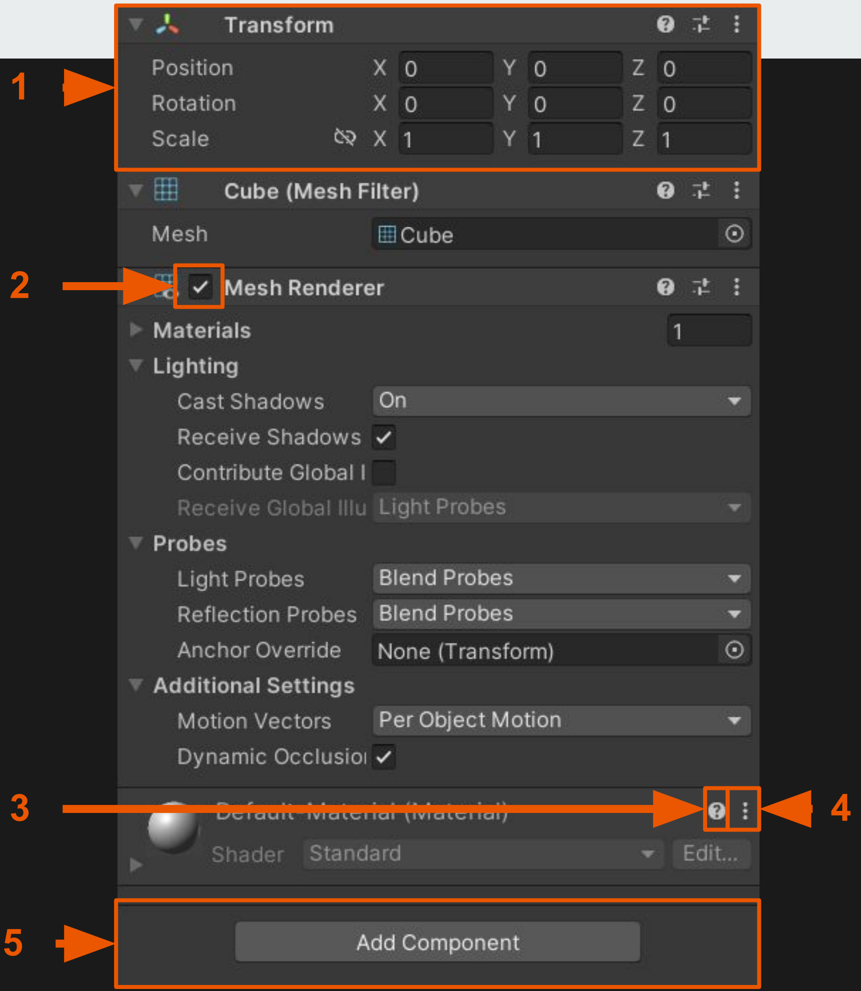
1. Create a new Blank Game Object name it “House” and give it a Icon
2. Make sure you have the Gizmos Toggled on otherwise the Icon won’t show up
3. Place it over the brick house and adjust its coordinates using Transform Component.
4. Click the play button and notice that the Icon isn’t there



Components List

[3] Information Link, clicking this will bring you to the Web Unity Manual page for the given Component.

[4] Component Properties, allows you to Reset the Component to default state, allows you to move the Component up or down the list [Also possible by dragging the Component Header], Copy and Paste the values and altogether delete the Component from the Game Object.



Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Mesh: Cube

Materials: 1

Cast Shadows: On
Receive Shadows: ✓
Contribute Global I
Receive Global Illu: Light Probes

Light Probes: Blend Probes
Reflection Probes: Blend Probes
Anchor Override: None (Transform)

Motion Vectors: Per Object Motion
Dynamic Occlusion: ✓

Shader: Standard

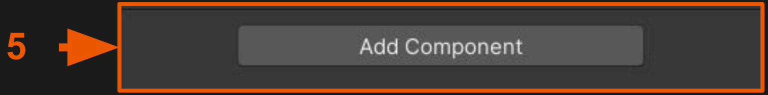
Sebastian Grygorczuk - STEM Institute at CCNY

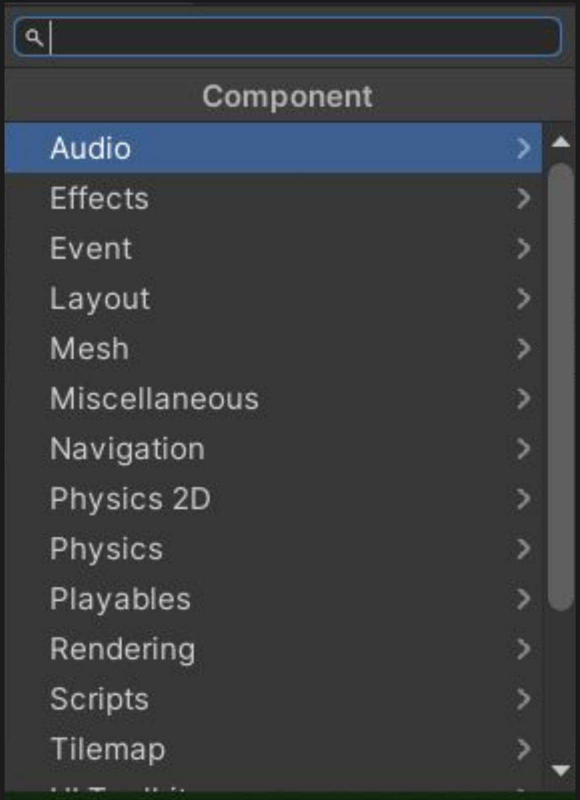
Components List

[5] Add Component, this will be a highly used feature. It allows you to bring in any Component that is pre built by Unity or any custom ones you may create such as Scripts. You can also add Components by dragging them from the Project View into the Component List or using the drop down menu from the File Menu Header named Component.

As you can see there are many categories, we will go over many of the Components held by them by not all.

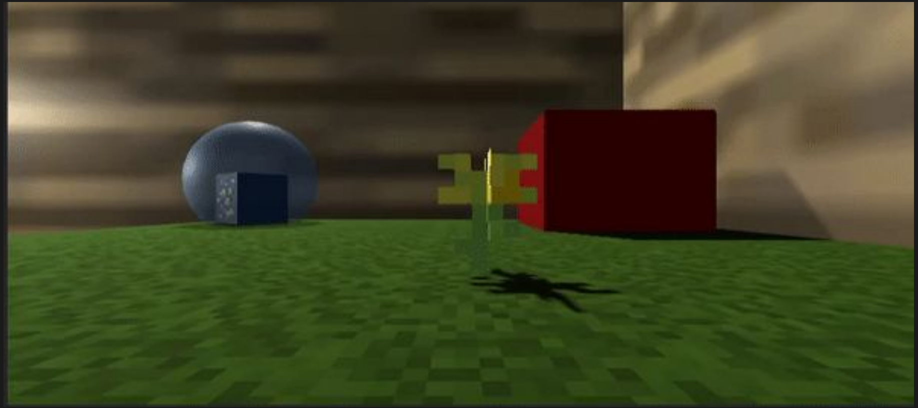
Feel free to use the Information Link, to look up any we may miss through the course of this class.





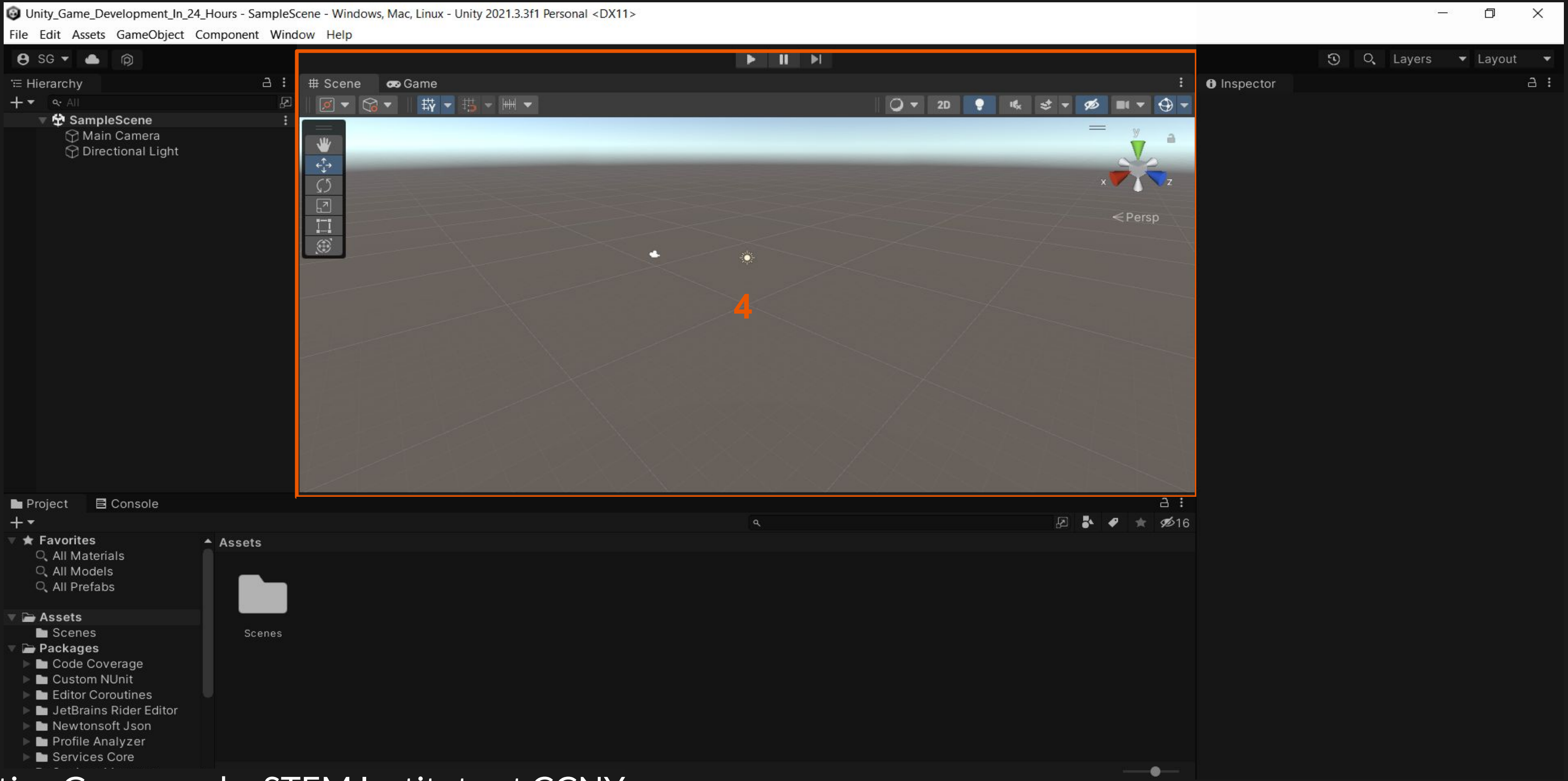
Game Object Challenge

1. Using the Hierarchy View, right click and create a new Cube Game Object
2. Notice that the Game Object has a Collision Component already on it. That allows it to touch other object that have Collision Components.
3. Using the Inspector View add the Rigidbody Component, [not Rigidbody 2D]. This gives the Game Object Physics.
4. Place the new Game Object over the Floor and click the Play Button on the top of the screen.
5. You should see your new made block fall



Sebastian Grygorczuk - STEM Institute at CCNY

Game View

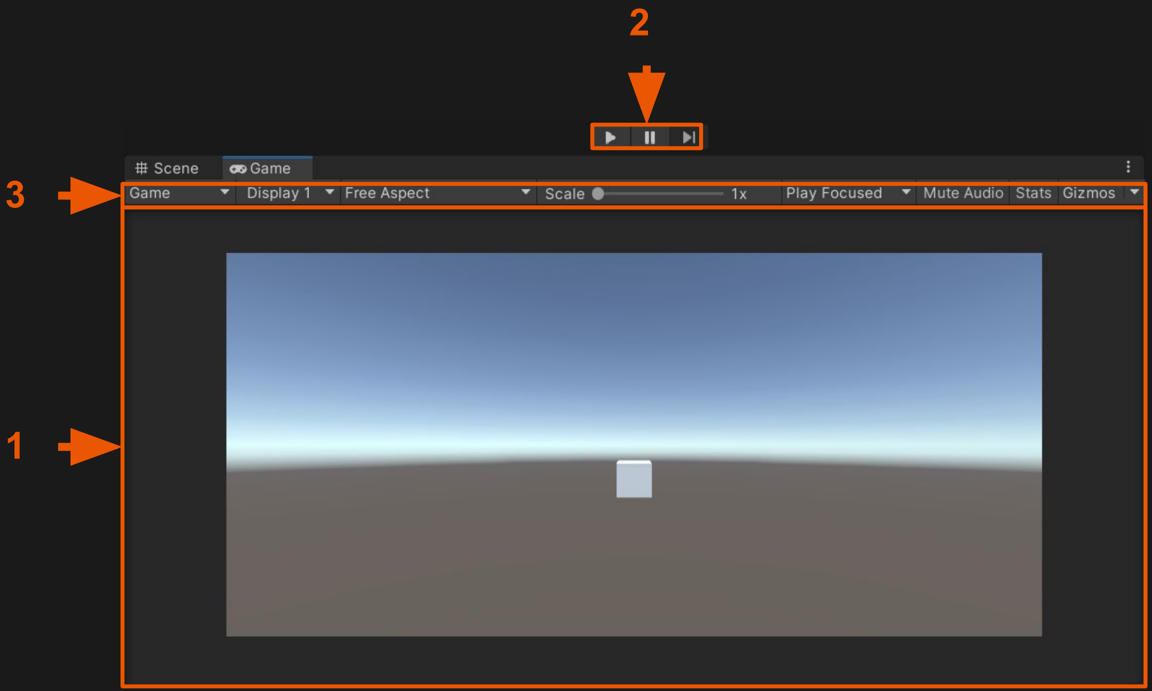


Sebastian Grygorczuk - STEM Institute at CCNY

Game View

Game View shows you the final product, here you can play test the game and all of the mechanics that have been built into it.

[1] The Game View, here you see the game through the Camera Game object that was created alongside the project or a new one you may have predefined.



Additional Resources: Pages 16-17 of the Textbook, The Game View - Unity Official Tutorials

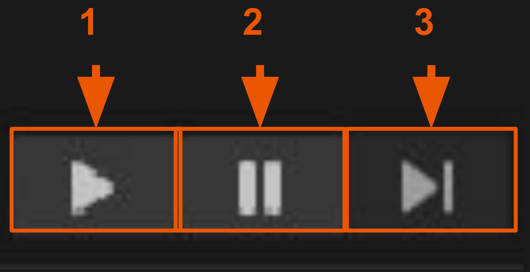
Play Bar

The Play Bar is always available to you, it consists of three buttons

[1] Toggles Play/Off, clicking the grayed out version the the play button will assemble the game and have it available for you to play. This can also be called Running or Executing the game.

Toggling the Play button Off, will reset everything to the way it was before you played. Any enemies you may have spawned will be gone and any progress you may have made in the level will be reset to the start.

Also while in Play Mode you are free to change and move any of the Game Obejct through the Scene View or the Inspector, however be caution that the changes will revert to their original state once Play Mode is off.



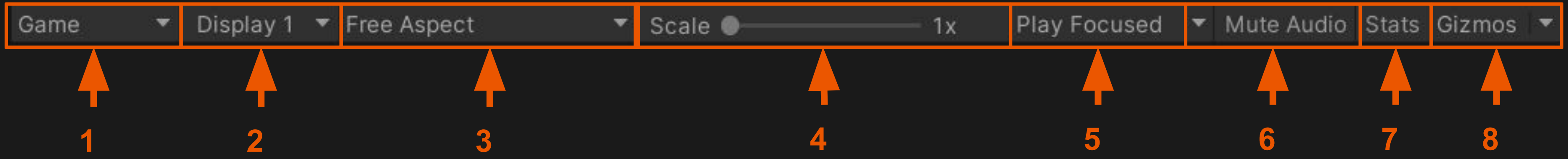
[2] Pause, stops the execution of the game but keeps it at the progress you've made in the level. This is very useful when something isn't acting the way you intended so you can execute the game have it run for a bit, pause when the problem occurs and move to the Scene View to examine the game objects that are cause the problem.

[3] Step, similarly to pause is a great way to debug your game, each click progresses the game by a step so you can see what occurs every step or tick of time that passes in the game.

Challenge Play Game

1. Click the Play Button the game and allow the Cube to fall down once again
2. Click the Pause Button and go to the Scene View Tab. Translate the cube up again while making it larger and rotating it using Transform Component in the Inspector View
3. Go back to the Game View Tab and using the Step Button let the cube fall a bit.
4. Stop the game and see that the changes didn’t stay.

Game View Toolbar



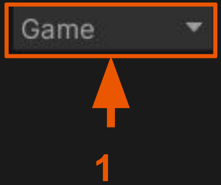
The Game View Toolbar will allow you to change the game is presented to you, allowing you to see what it would look like on different screen and resolutions.

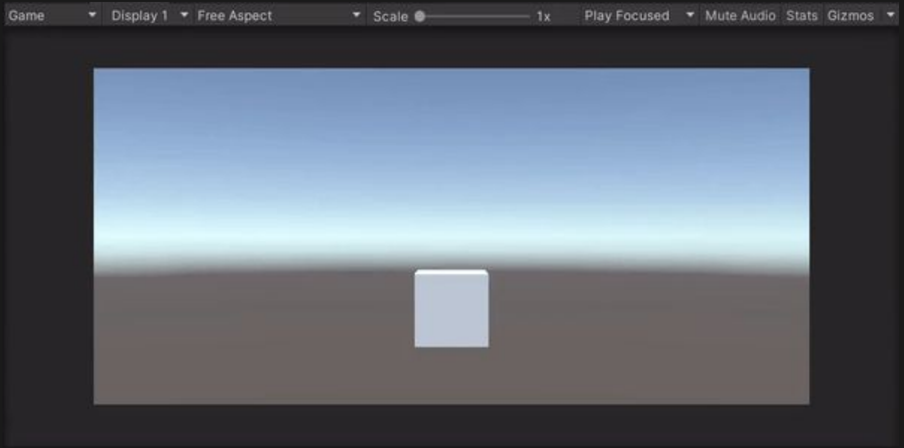
The way you game is seen by you and the player is important as it will dictate how much and what you can show on the screen at once. Creating hard Game Design Decisions.

Game Drop Down

[1] The Game-Simulator Dropdown, will allow you to switch between a game view that is built for Desktop/Console vies to those available on the different mobile devices.

This also changes the available settings on the Toolbar, however we won't be covering Mobile Games in this class so it's good to know but not necessary for rest of the class.





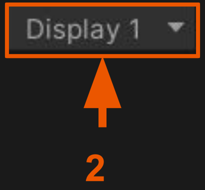
Display DropDown

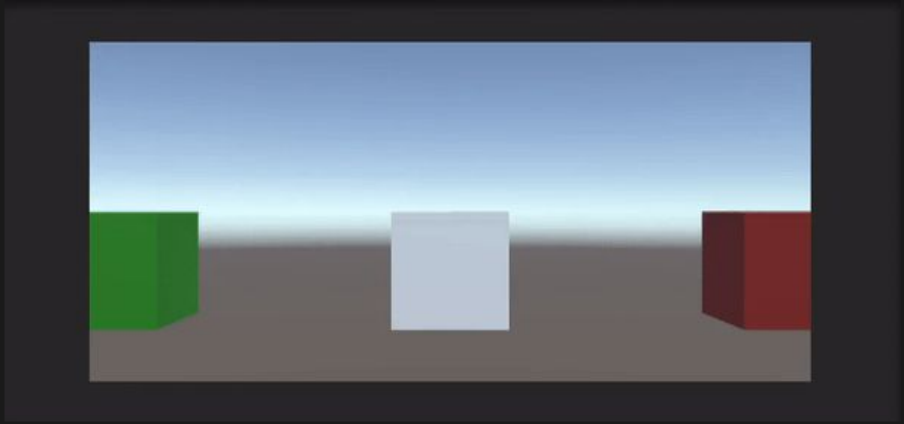
[2] The Display Drop Down, allow you to set up to eight Displays or monitor that the game would run on.

Not to be confused by Split Screen or camera that would be positioned at a different character or location the game you can switch to.

This is for the games being connected to two or more monitors, for example the racing games at Arcades that have two to four player seats and are all connected each monitor following their own player.

We won't be working with feature.





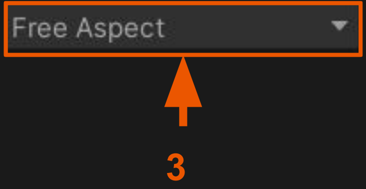
Aspect Dropdown

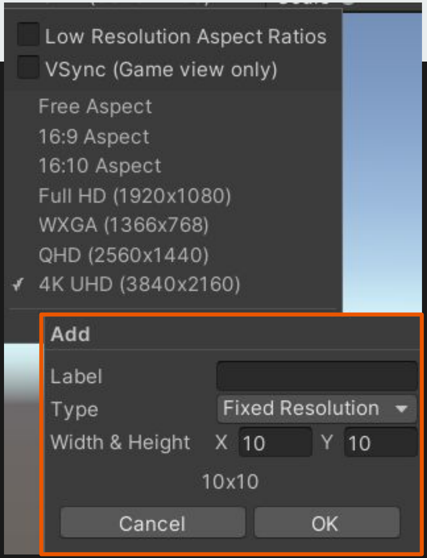
[3] Aspect Dropdown, allows you to set the Aspect Ratio at which you will be viewing the game through the Game View.

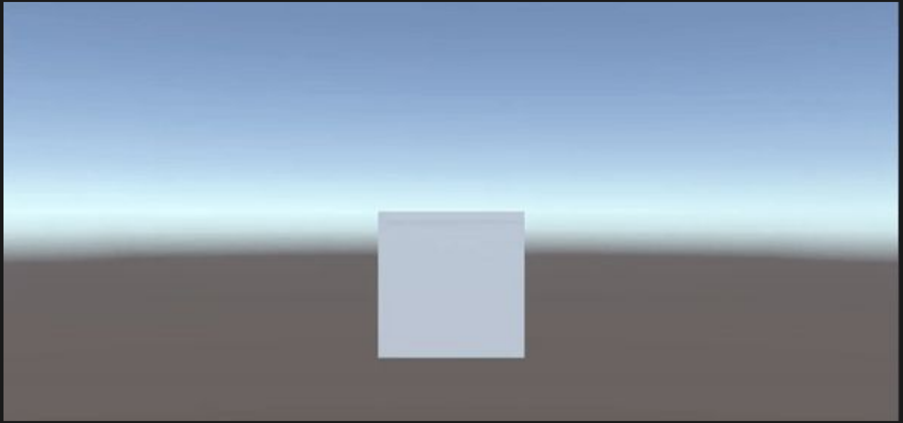
This does not set the Aspect at which the game will be exported out of Unity only how you are currently viewing it.

If you find the options lacking you have the ability create your own aspect ratio.

Depending on what type of game you want to make, you might want to use less of the screen setting up this aspect ratio will let you design your level in accordance with that.







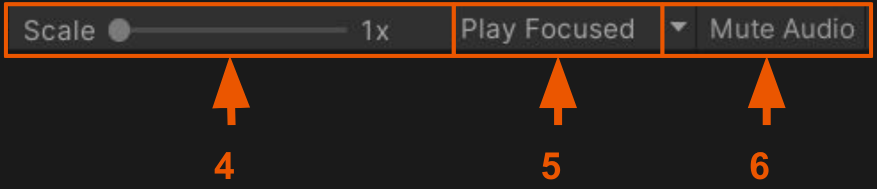
Sebastian Grygorczuk - STEM Institute at CCNY

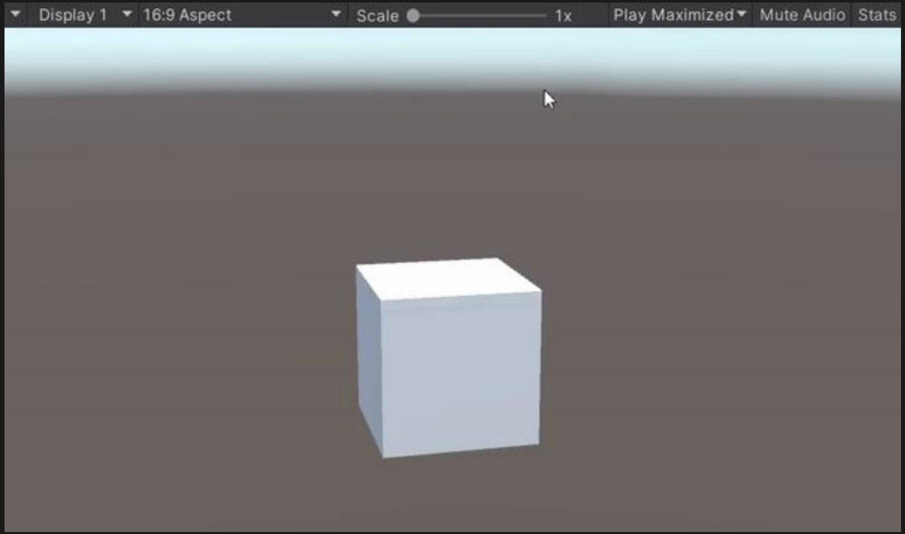
Scale, Play Focused / Maximized, Mute

[4] Scale, allows you to zoom in and out on the game screen. This isn't like navigating close to the Game Object in the Scene view but zooming in on the pixels of the current scene.

[5] Play Focused / Maximized, allows you to set the play setting to take up as much of the screen as possible while keeping the aspect ratio you've selected.

[6] Mute, pretty self explanatory, once you start testing your games for bugs you will be hearing the same sound effects and music over and over and it's good to turn it off when not necessary.

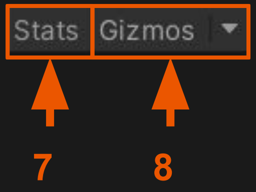


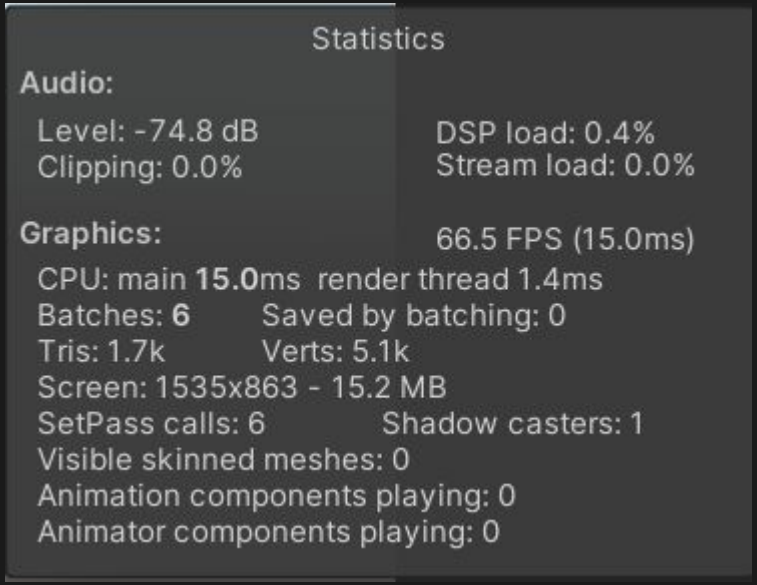


Stats and Gizmos

[7] Stats, a very useful tool telling us how the game is running. It’s broken down into Audio and Graphical strain on your computer. With the scale of our projects you won’t really have to look at this menu unless you’ve really messed up.

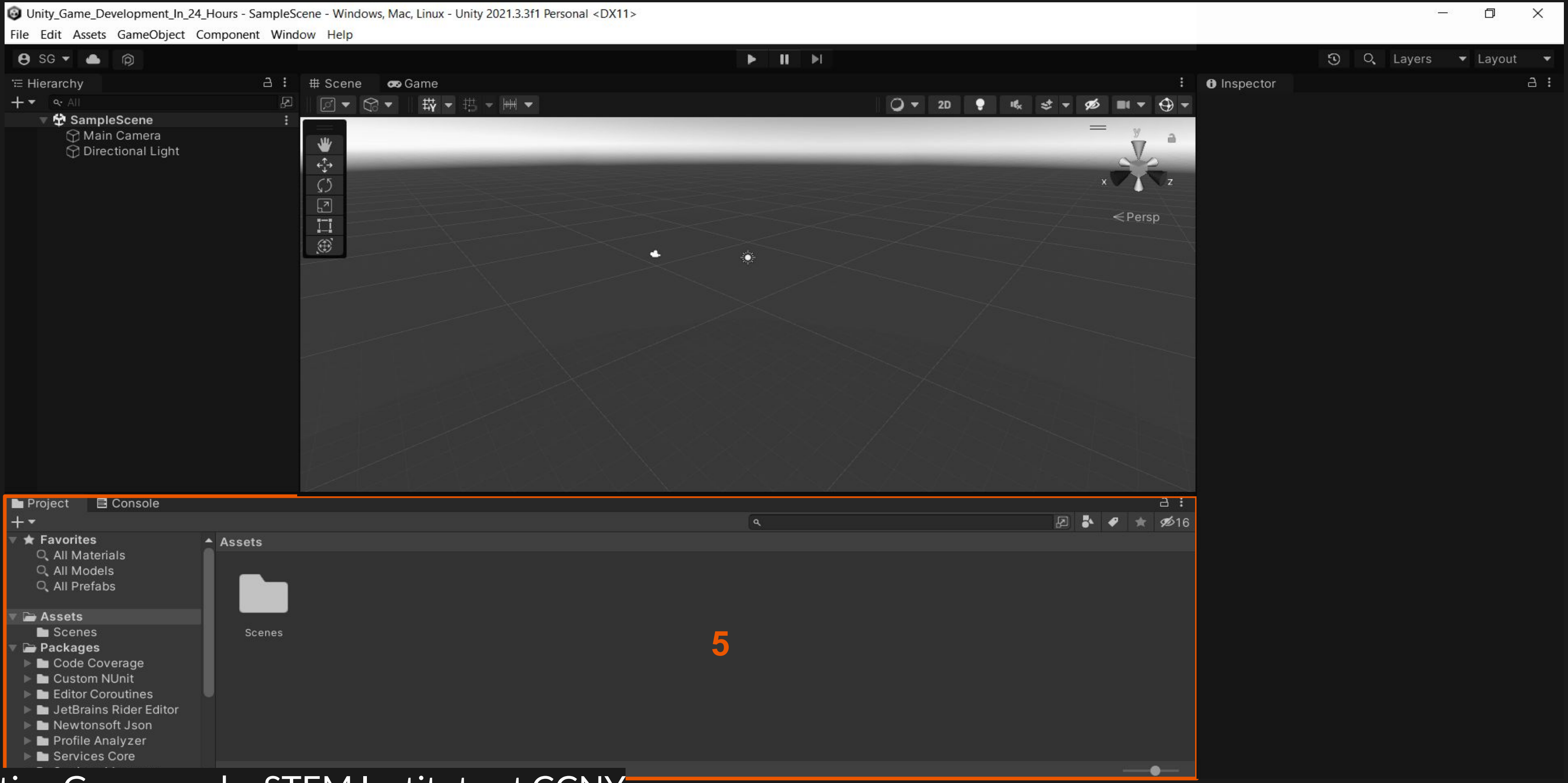
[8] Gizmos, work pretty much exactly like their counterpart in Scene View allowing you to see all Gizmos that are toggled on.




Statistics
Audio:
Level: -74.8 dB   DSP load: 0.4%
Clipping: 0.0%   Stream load: 0.0%
Graphics: 66.5 FPS (15.0ms)
CPU: main **15.0**ms render thread 1.4ms
Batches: **6**   Saved by batching: 0
Tris: 1.7k   Verts: 5.1k
Screen: 1535x863 - 15.2 MB
SetPass calls: 6   Shadow casters: 1
Visible skinned meshes: 0
Animation components playing: 0
Animator components playing: 0

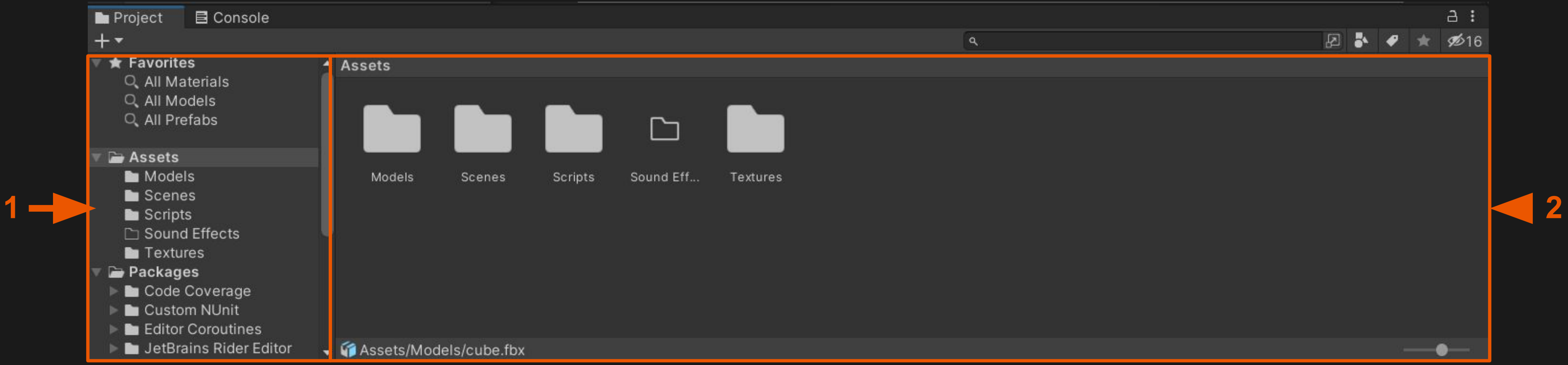
Sebastian Grygorczuk - STEM Institute at CCNY

Project View



Sebastian Grygorczuk - STEM Institute at CCNY

Project View



The Project View will display all of the Game Assets that are available to you, [1] shows you a list of all of the folders available to you, and your favorite searches. [2] Shows the current folder you are looking at.

Additional Resources: Pages 9-11 of the Textbook, [The Project Window - Unity Official Tutorial]

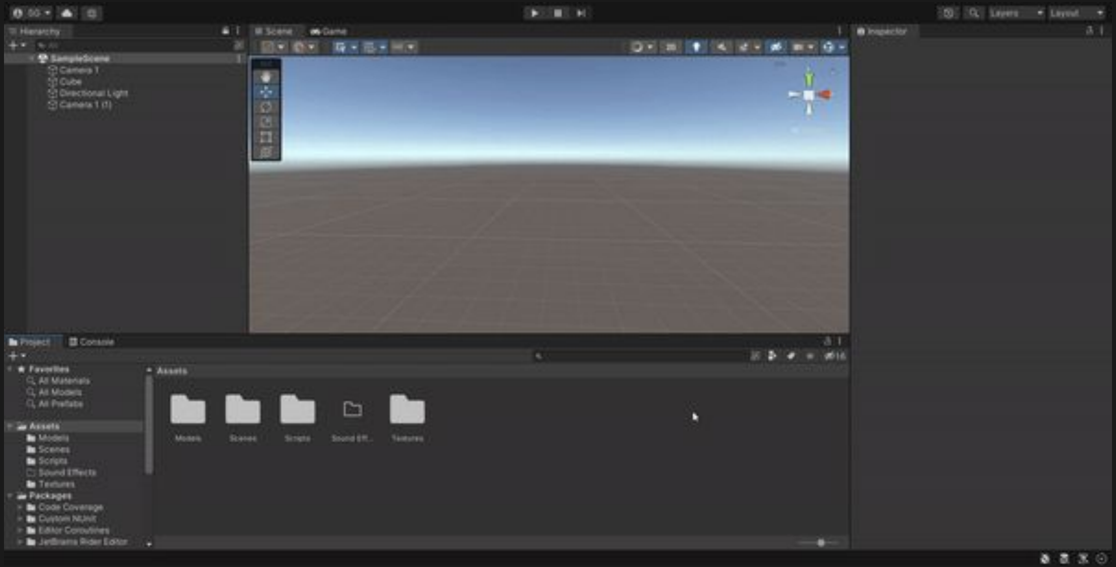
Sebastian Grygorczuk - STEM Institute at CCNY

Game Asset Vs Game Object

A Game Asset is anything that has the potential to be Instantiated or created while Game Object is an instance of a Game Asset.

You can instantiate a Game Asset into a Game Object by dragging it from the Project View to the Scene View. Notice that instantly the Game Object is added to the Hierarchy and the Inspect View is filled with its Components.

Later on we will instantiate Game Objects using code.

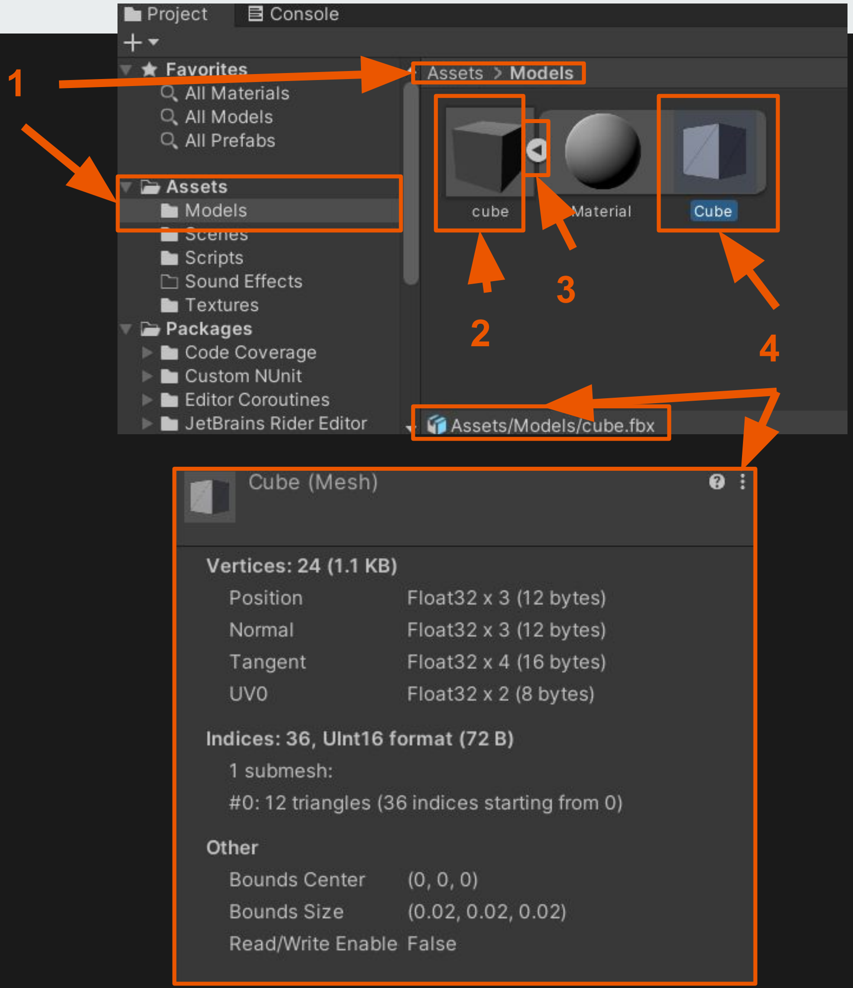


Project View

When looking at Assets in the Project View you will have [1] The Breadcrumb Trail which is the path to the folder we're currently viewing. This will be also displayed in the Folder Column by highing the folder currently accessed.

Next let's look at a [2] Game Asset, here I have a 3D imported cube. More complex Assets such as 3D Models will have hierarchy of sub assets, you can click the [3] Arrow to expand or collapse them.

Finally you can see one of the sub assets for the cube which is it's [4] 3D model, when the asset or sub asset is selected you will see the Path to the asset on the bottom of the Project View and extra details will be shown in the Inspector View.



Vertices: 24 (1.1 KB)

Position	Float32 x 3 (12 bytes)
Normal	Float32 x 3 (12 bytes)
Tangent	Float32 x 4 (16 bytes)
UV0	Float32 x 2 (8 bytes)

Indices: 36, UInt16 format (72 B)

1 submesh:

#0: 12 triangles (36 indices starting from 0)

Other

Bounds Center	(0, 0, 0)
Bounds Size	(0.02, 0.02, 0.02)
Read/Write Enable	False

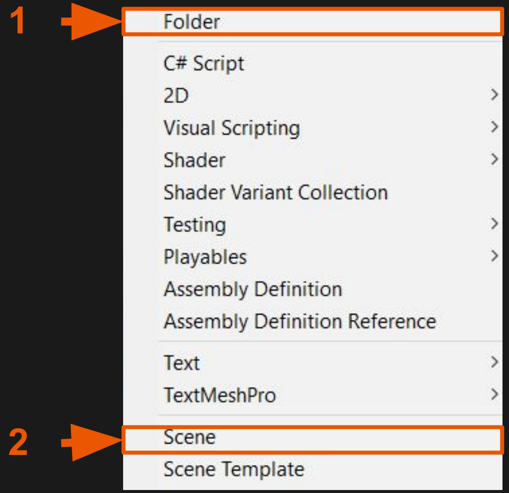
Sebastian Grygorczuk - STEM Institute at CCNY

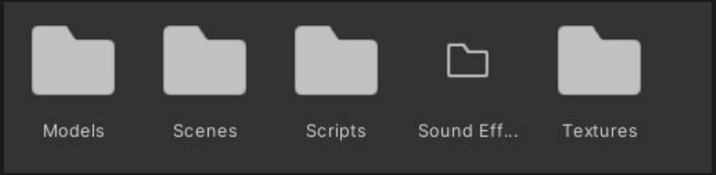
Creating Asset and Organization

There are many Game Assets that you can create directly from the Project View by right clicking on the empty space and selecting Create. This will provide you with a massive list but the main two we are going to focus on for a moment are Folder and Scene.

[1] Folder allows you to create new Folders in the Project View. It's not exciting but just like keeping the Hierarchy View organized is necessary for good game development same goes for the Project View.

[2] Scene Asset allows you to create a new scene or level. Everything we've been doing has been within one scene with this you'll be able to make many level with many scene and later combine them to make a game world.





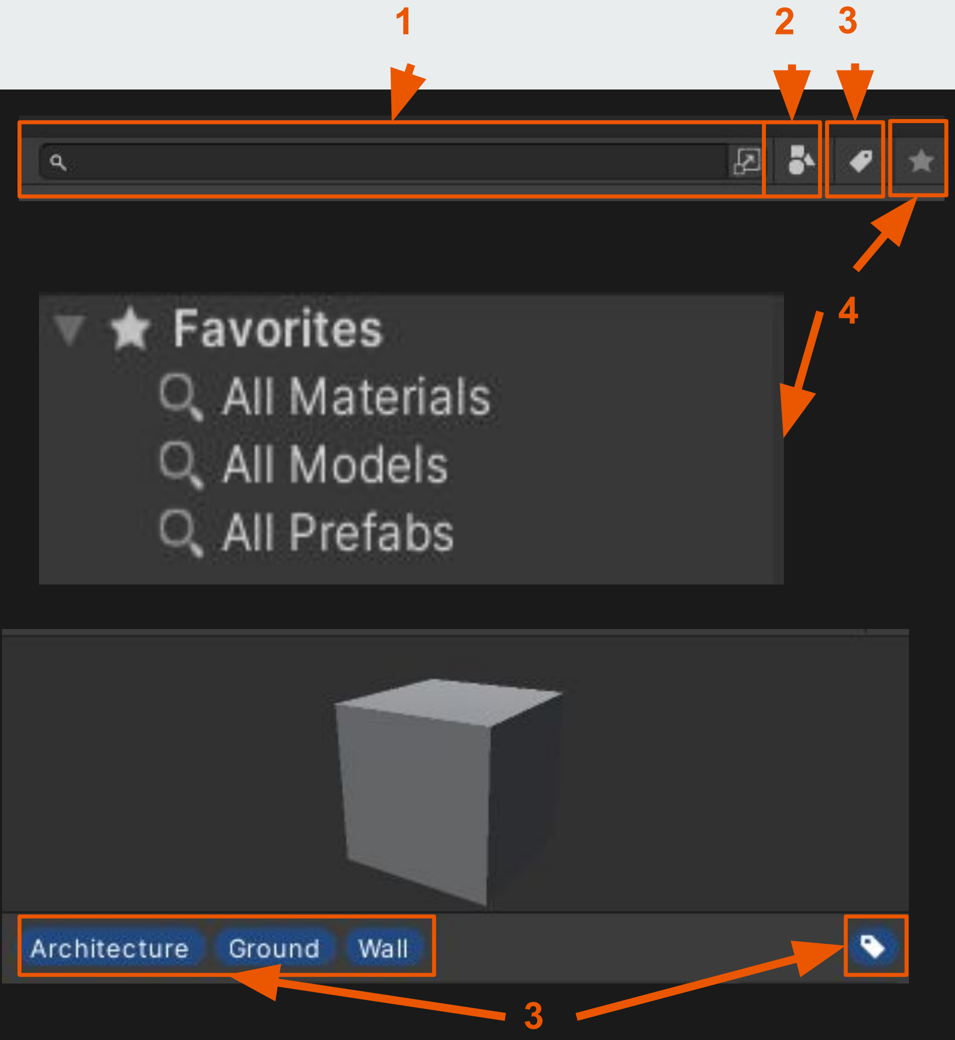
Search & Organization

Finding Assets quickly can shave hours off work. Thus we can use the [1] Search Bar and the Advanced Search just like the one in Hierarchy View. With the next three buttons working as filters for the inputted search.

[2] Allows you to view all items that fit into selected category, Models, Scripts etc. The Project view will look for items that have the correct extensions.

[3] Tags are things you can connect to specific Game Assets through the Inspector View, on the very bottom there will be a tag symbol with a list of descriptors, you can set as many as you'd like and then search by them.

[4] Favorite Search, when you search for a name in Search Bar you can favorite that search and it will pop up as a saved search in the Folder Column.



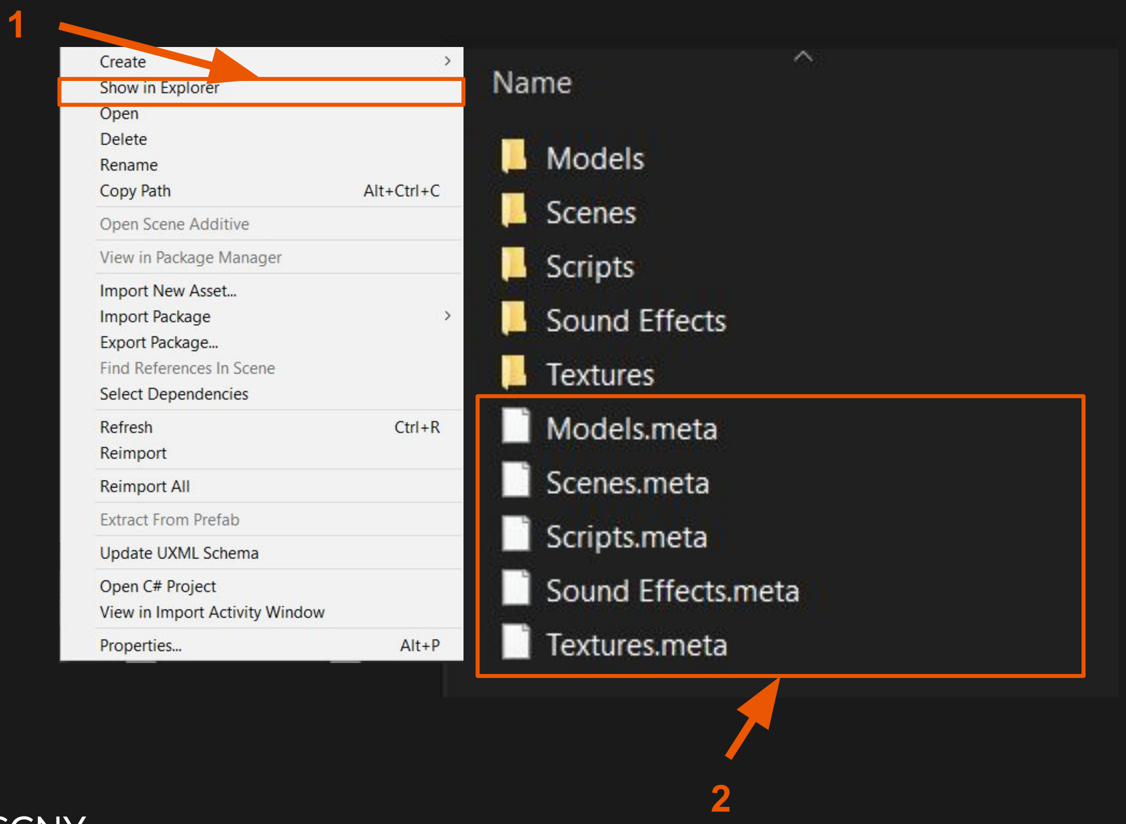
Sebastian Grygorczuk - STEM Institute at CCNY

Files Connect Directly to Project Folder

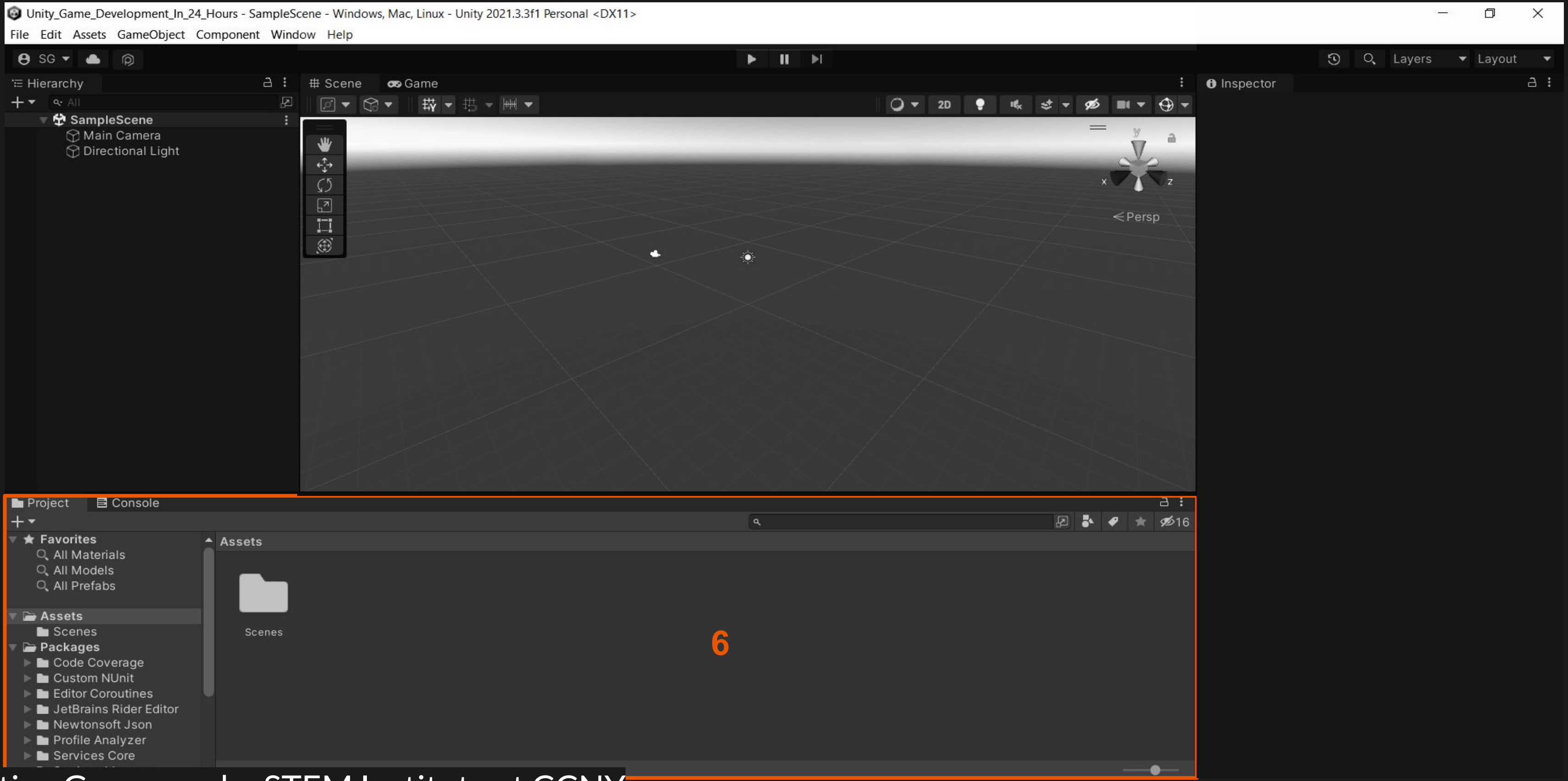
Each folder that's displayed in the Project View is directly connected to a folder in your project files. You can right click any folder and select [1] Show In Explore to open that Window Directly.

You may notice that inside the folder there are [2] meta files. These are files created by Unity that connect the assets to all it's functions in the game. Unity will automatically update them if you move or edit the file in any way.

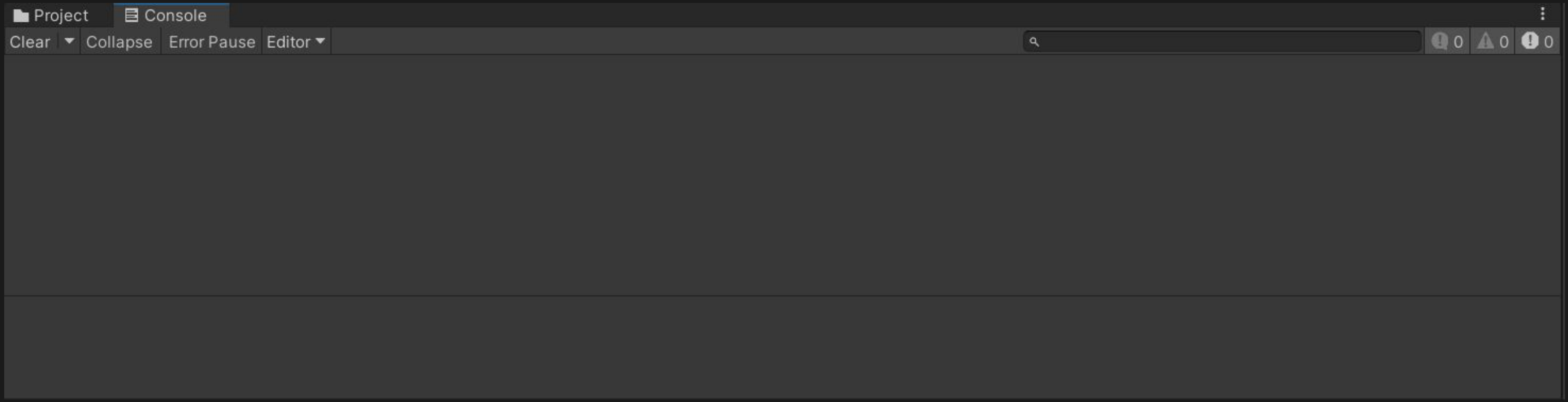
Only edit your files in the Project View as Unity may not update whatever changes you've done to a asset in the widow explore.



Console View



Console



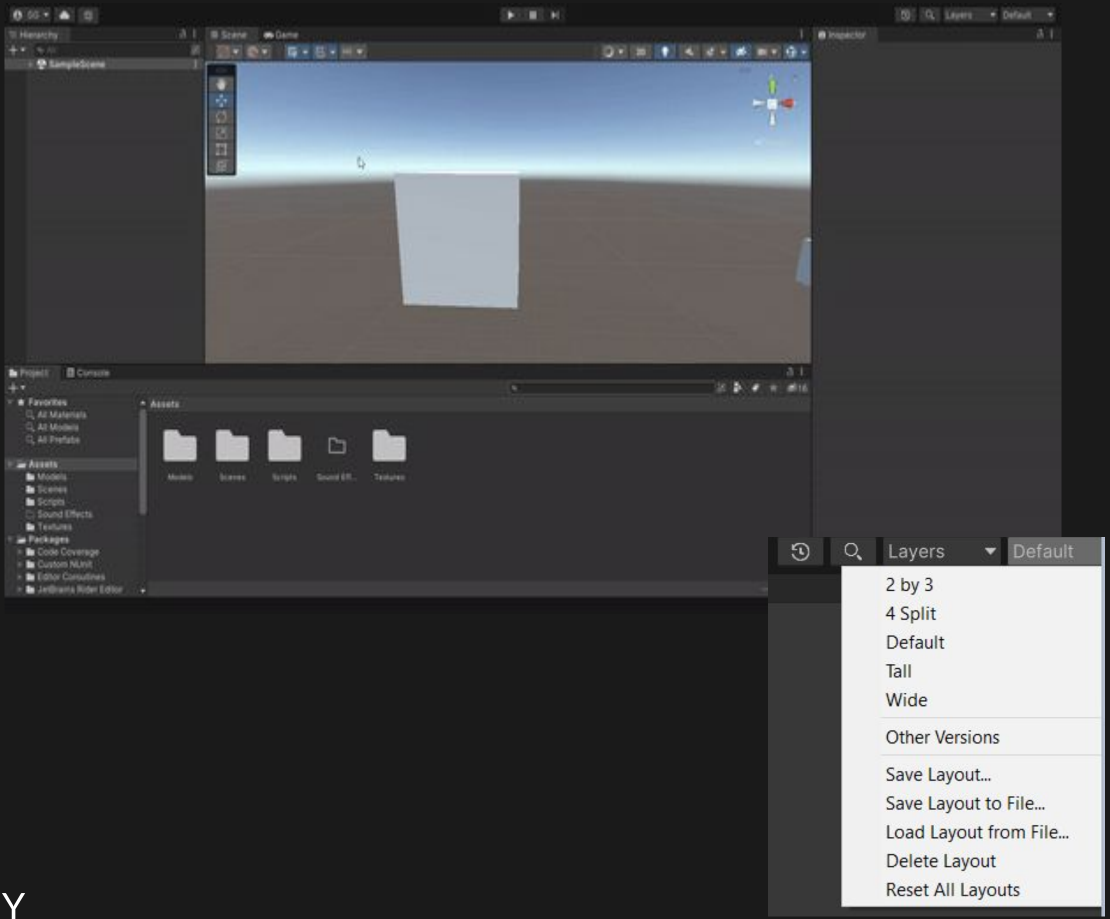
Lastly there's the console, this will be heavily used once we start programming our game. From here you'll be able to print out states of object and look for bugs. We will return to this window in Week 2 once we being programing.

Additional Resources: Page 126

Layouts

One last thing that's important to mention is that every one of the views we've looked at can be moved and placed anywhere in the Unity program. There are many pre made but you also have option to create your own and save them for future use.

As we continue in the class we will introduce more views and depending on the task a new layout may be in order.

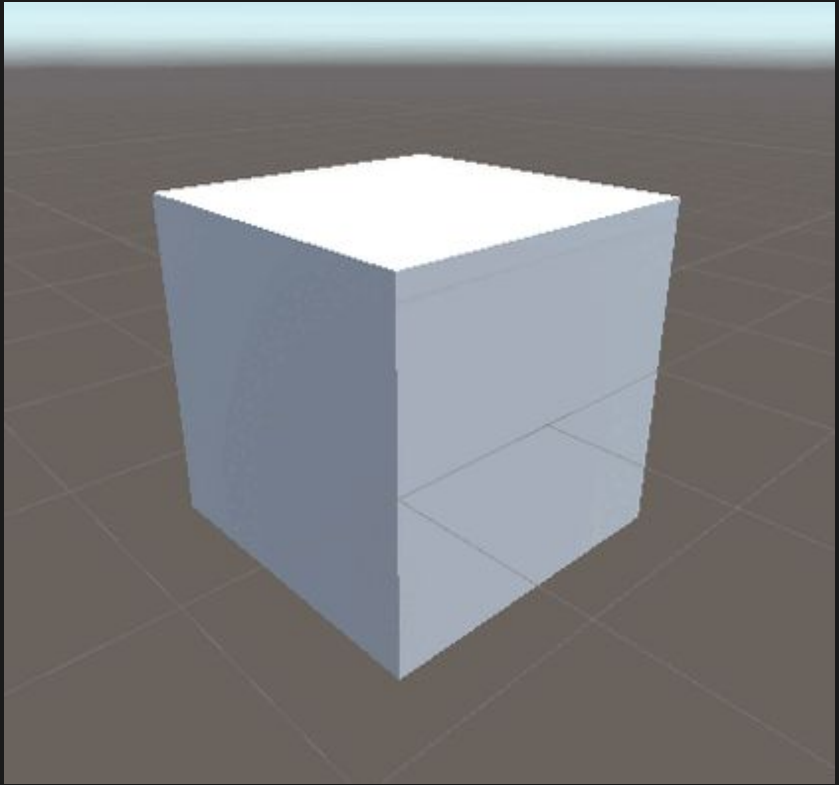


Built-In Models

We've already encountered models or meshes. In our class, the primary model we looked at was the Cube.

Alongside the Cube, Unity has five other Built in Models: Sphere, Capsule, Cylinder, Quad, and Plane.

The difference between the Quad and Plane is number of Polygons. The Quad model consists of two polygons and is better for UI elements while the Plane model has 200 polygons and is used for walls and floors.



Additional Resources: Pages 41 - 42, **Primitive and Placeholder Objects**

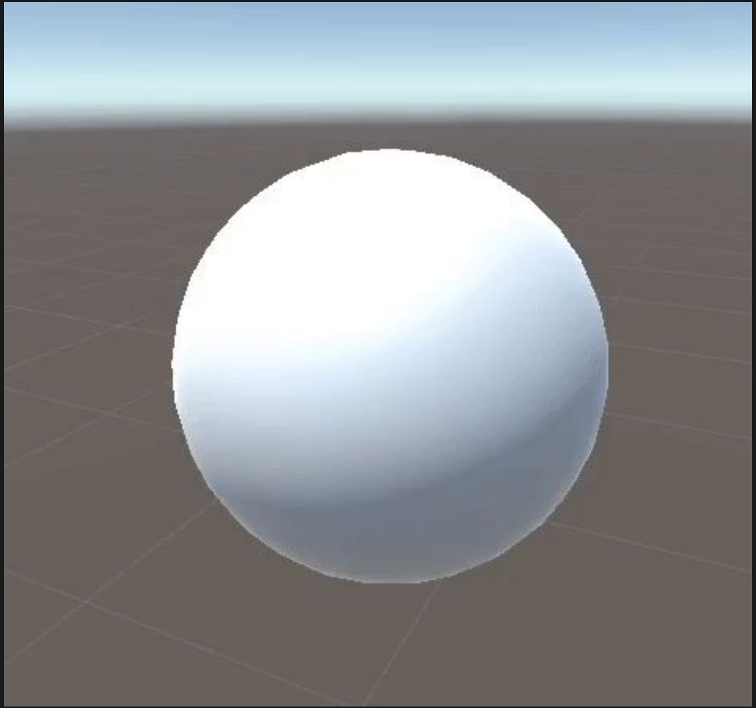
Polygons

Polygons are Triangles that make up 3D Meshes. Every 3D Models is made up of triangles. This is because it is the simplest flat surface you could make.

With a triangle you have three points or vertices and no matter where you put the points in a 3D space they will always make a flat surface.

However, if you were to take four points or more you will have a chance to create a plane or a 3D object such as a pyramid which is much harder to simulate for the computer.

Using polygons allows you to create smoother surfaces that can approximate curves.



Additional Resources: Why Video Games are Made of Tiny Triangles,

Mesh Filter and Mesh Renderer

[1] The Mesh Filter Component tells you what Mesh or Model you want to have attached to the game object. Clicking the circle will bring up a PopUp Menu that will show all of the meshes that are in your Project View.

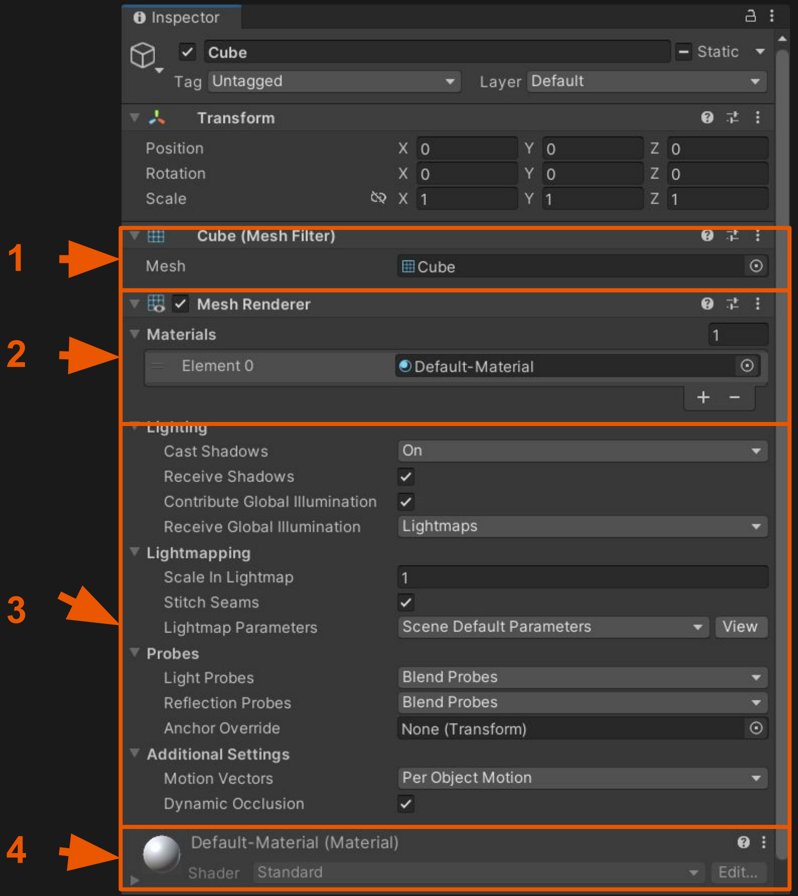
[2] The Mesh Renderer Component draws or renders the mesh with the chosen Material. Material is the way a Mesh is viewed, we will discuss it in the next slide in further detail. You can have multiple Materials connected to a Render but we will not have to worry about that in this class.

[3] The Mesh Render has many lighting setting on how it should interact with lights, we will discuss that in a class focused on lighting.

[4]The Material used in the render will be on the bottom of the inspector in case you want to view it's details.

Additional Resources: Mesh Filter, Mesh Renderer

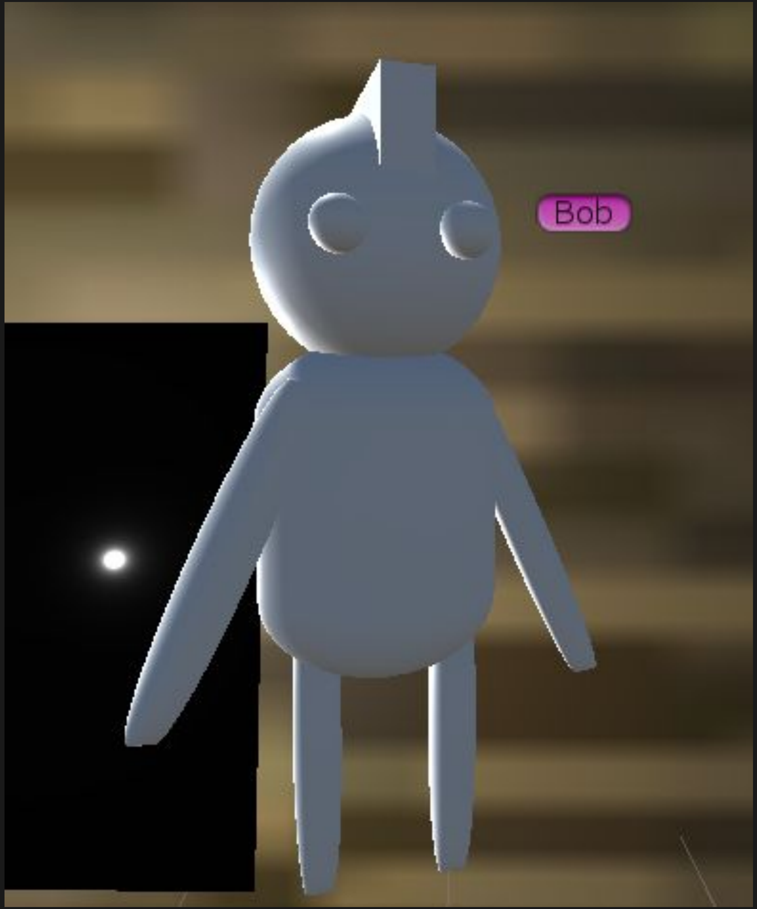
Sebastian Grygorczuk - STEM Institute at CCNY



```
Cube                                   Static
Tag Untagged                           Layer Default

Transform
Position      X 0    Y 0    Z 0
Rotation      X 0    Y 0    Z 0
Scale         X 1    Y 1    Z 1

Cube (Mesh Filter)
Mesh                                   Cube

Mesh Renderer
Materials                              1
  Element 0                            Default-Material
Lighting
  Cast Shadows                         On
  Receive Shadows                      ✓
  Contribute Global Illumination       ✓
  Receive Global Illumination          Lightmaps
Lightmapping
  Scale In Lightmap                    1
  Stitch Seams                         ✓
  Lightmap Parameters                  Scene Default Parameters   View
Probes
  Light Probes                         Blend Probes
  Reflection Probes                    Blend Probes
  Anchor Override                      None (Transform)
Additional Settings
  Motion Vectors                       Per Object Motion
  Dynamic Occlusion                    ✓

Default-Material (Material)
Shader Standard                        Edit...
```

Challenge: Create a Character

1. Using the pomade 3D models let's try to put together a very basic humanoid person.
2. Start off by creating a Empty Game Object and calling it a Person
3. Then Proceed to give them a head, body, arms and legs.
4. After that give them a little extra detail, eyes, fingers, nose hair. It's up to you.



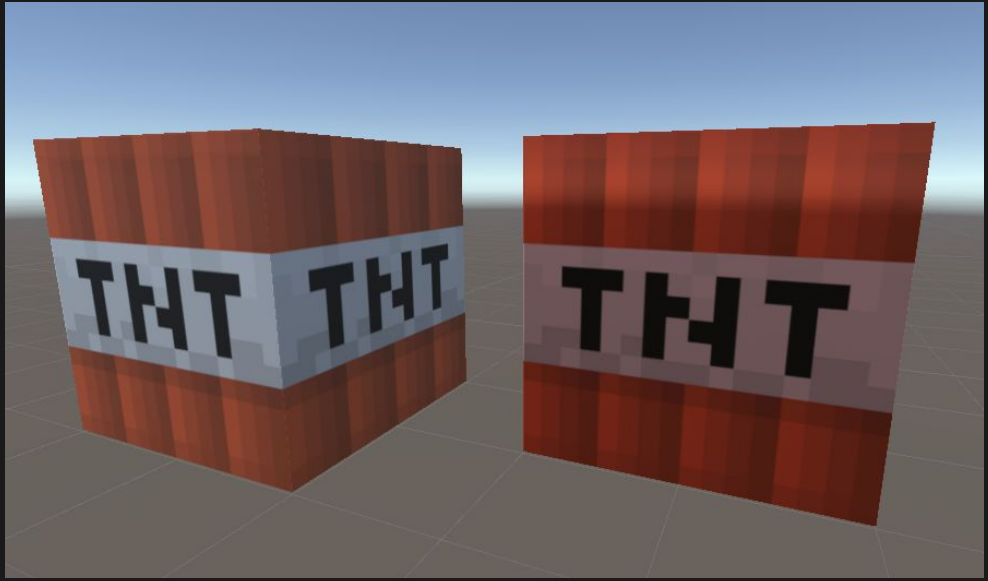
Model Material

To make a 3D asset be viewed it requires something called Model Material, which is made up of two parts

Model Texture - what is drawn on the surface of the 3D Model

Model Shader - how the texture is drawn on the 3D Model

"Imagine you have a piece of wood. The physicality of the wood is its' mesh, or model; the color, the texture, and visible elements are its textures. Now take that piece of wood and pour water on it. The wood still has the same mesh. It is still made of the same substance (wood). It looks different, thought,. Its slightly darker and shiny. In this example you have two "shaders": dry wood and wet wood. " [Page 49]



Additional Resources: Pages 47-50 of the Textbook

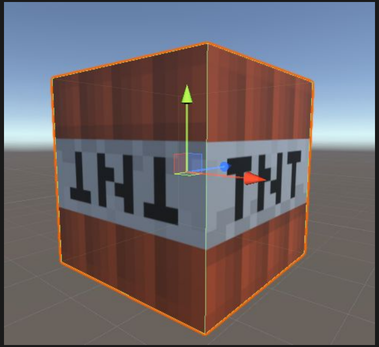
Textures

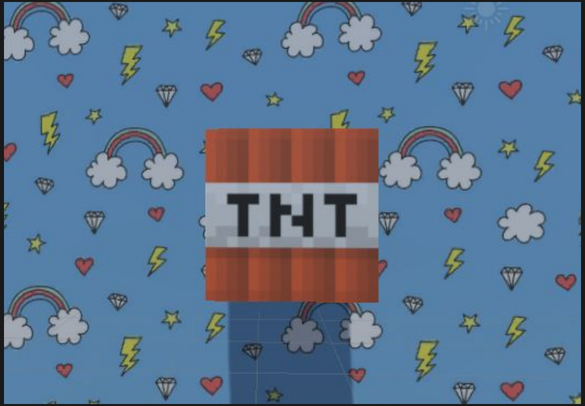
Textures are flat images that get mapped to the 3D Mesh through a process UV Mapping.

In our example we have the Unity standard Cube, it has six faces and just slaps the whole image on its face without much care for orientation.

This may not be best used on a Cube however, given a Tileable Image we could use it to create a convincing floor or wall.

Tileable Image or Seamless Image is an image in which the opposite edges match up allowing you to create a never ending copy of the pattern.



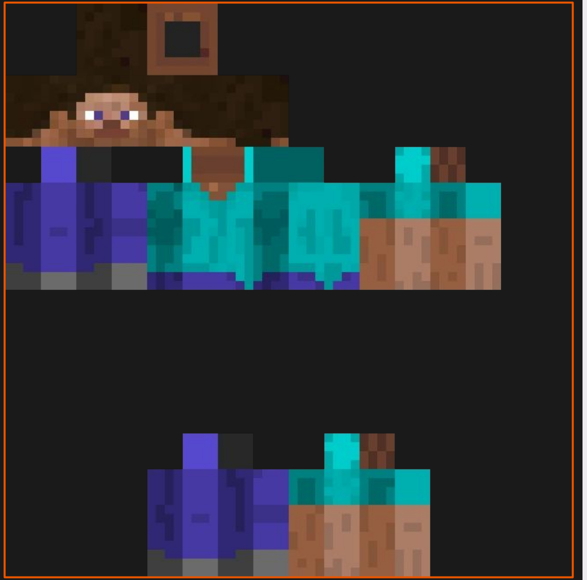


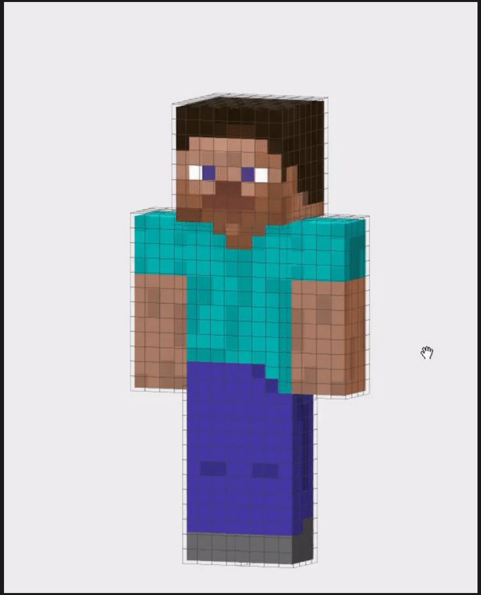
UV Mapping

It's called UV Mapping because the Mesh is in XYZ coordinates the texture is represented in UV coordinates. The mapping comes in play when you divide parts of the texture it can display different parts of an image on different faces of the cube.

Here we can see all the parts of the Steven Texture drawn on a single image, the model has been created to look at those specific parts of the image and map to predefined parts of the mesh.

To do this you'd have to model 3D objects on your own which is beyond the scope of this class, but if you're interested you can download Blendr and try it in your free time.





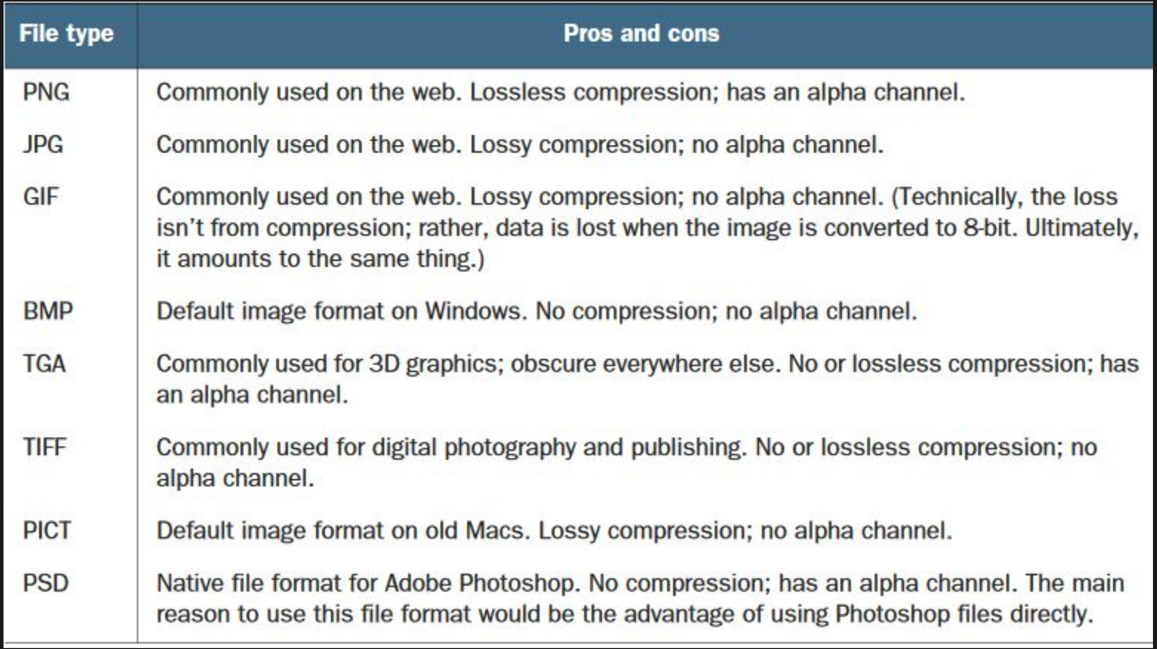
Additional Resources: Introduction to UV Mapping - Learn the Complete Basics, The Skindex

Image Specifications

When getting the images we want as our textures we want them their dimension to be in power of 2^n as computer chips are programing in binary, so sizes such as 32, 64, 128, 256 and so on.

Best image format for the image being a PNG or TGA, as they aren't compressed and have an alpha channel.

Alpha channel allow image to have transparent parts. Every image will have three other channels red, green, and blue.



File type	Pros and cons
PNG	Commonly used on the web. Lossless compression; has an alpha channel.
JPG	Commonly used on the web. Lossy compression; no alpha channel.
GIF	Commonly used on the web. Lossy compression; no alpha channel. (Technically, the loss isn't from compression; rather, data is lost when the image is converted to 8-bit. Ultimately, it amounts to the same thing.)
BMP	Default image format on Windows. No compression; no alpha channel.
TGA	Commonly used for 3D graphics; obscure everywhere else. No or lossless compression; has an alpha channel.
TIFF	Commonly used for digital photography and publishing. No or lossless compression; no alpha channel.
PICT	Default image format on old Macs. Lossy compression; no alpha channel.
PSD	Native file format for Adobe Photoshop. No compression; has an alpha channel. The main reason to use this file format would be the advantage of using Photoshop files directly.

Additional Resources: Chart from Unity in Action pg 83

Sebastian Grygorczuk - STEM Institute at CCNY

Image Import Settings

Every image that you bring into Unity will have Image Import settings that tells Unity how the image should behave. You can get to it by selecting the Image in the Project View.

[1] Is telling Unity how it should treat the image, currently it's set to Texture Type is Default, but we'll use Normal Map and Sprite as well. Texture Shape is 2D keeping it an image but can be turned into others which we'll come back to later.

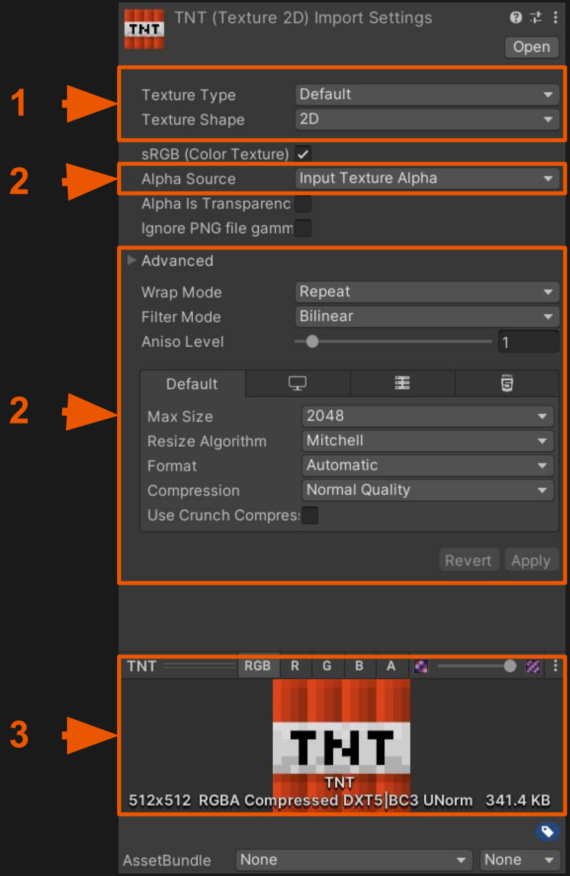
[2] Alpha Source is important for determining what does your image look like if the transparency is lowered. Input Texture Alpha will use the alpha preset with the image while From Gray Scale will gray the image and use that as the alpha channel.

[3] Is how the image should be treated per platform. Some platforms are less powerful and texture need to be compressed for them to operator optimized.

[4] Is the preview of how your choices affect the image.

Additional Resources: Texture Import Settings, Textures - Unity Official Tutorials

Sebastian Grygorczuk - STEM Institute at CCNY

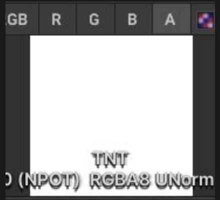


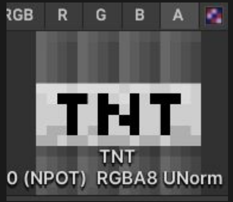
TNT (Texture 2D) Import Settings

Texture Type	Default
Texture Shape	2D
sRGB (Color Texture)	✓
Alpha Source	Input Texture Alpha
Alpha Is Transparenc	
Ignore PNG file gamm	
Wrap Mode	Repeat
Filter Mode	Bilinear
Aniso Level	1
Max Size	2048
Resize Algorithm	Mitchell
Format	Automatic
Compression	Normal Quality
Use Crunch Compres	
AssetBundle	None / None

TNT RGB R G B A

512x512 RGBA Compressed DXT5|BC3 UNorm 341.4 KB





Material

Here we are at the Material Inspection, in the [1] Header you are allowed your Shader type. We only will use Standard and Standard (Specular). The difference between them is that Standard will make the material look metallic while Specular will make it more reflective to light sources. It direct effect what's displayed in [4]

[2] Rendering Mode, there are four modes, Opaque which will show the image as solid, Transparent, Fade and Cutout. Difference between Transparent and Fade is that Transparent even with 0 on the alpha channel will leave behind a reflection of the mesh while Fade will fully disappear. To edit the transparency you will edit it from the color select in Albedo Map [3].

Additional Resources: Standard Shader, Rendering Mode

1

2

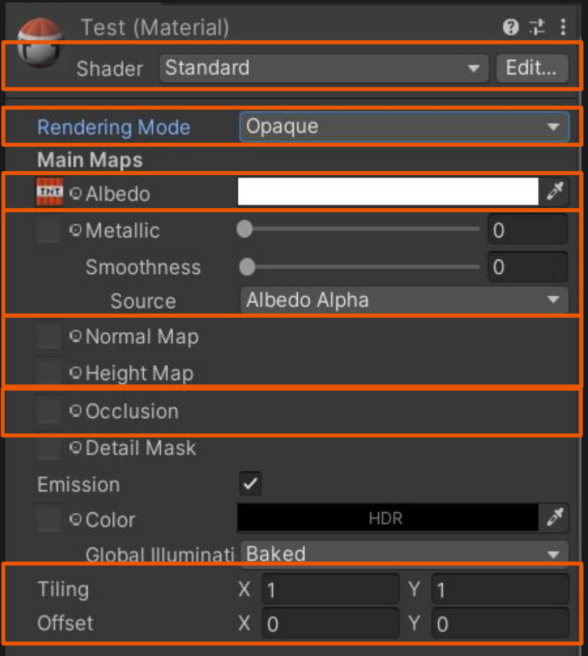
3

4

5

6

7



Test (Material)

Shader Standard Edit...

Rendering Mode Opaque

Main Maps

Albedo

Metallic 0

Smoothness 0

Source Albedo Alpha

Normal Map

Height Map

Occlusion

Detail Mask

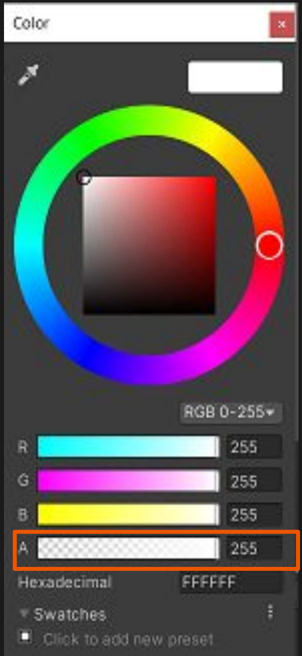
Emission ✓

Color HDR

Global Illuminati Baked

Tiling X 1 Y 1

Offset X 0 Y 0



Color

RGB 0-255

R 255

G 255

B 255

A 255

Hexadecimal FFFFFF

Swatches

Click to add new preset

TNT
TNT
TNT
TNT

Sebastian Grygorczuk - STEM Institute at CCNY

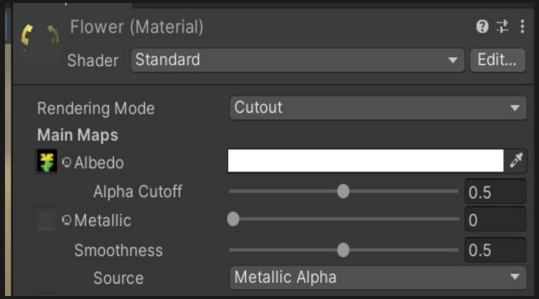
Material

The Flowers are are an example of a Cut Out Material, being two cubes with the Texture drawn over them.

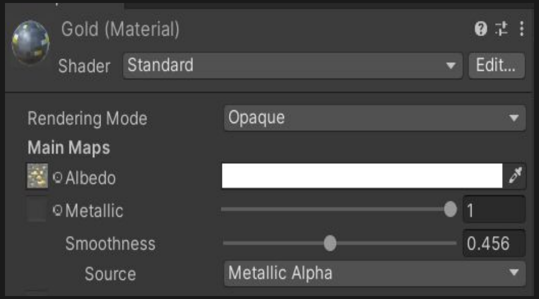
The Gold Brick is a example of Opaque Material. A solid texture influenced by the color.

Gold Fade shows what happens when you have a fades object.

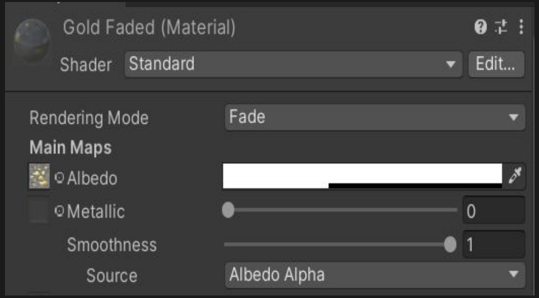
Glass is a good example of a Transparent Material.



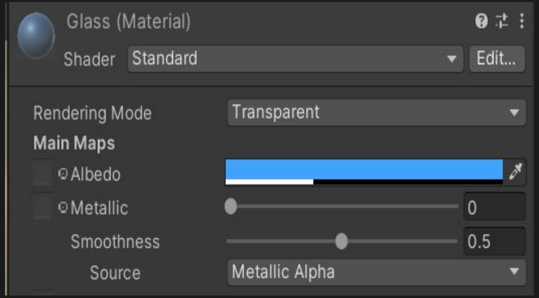
Flower (Material)
Shader Standard Edit...
Rendering Mode Cutout
Main Maps
Albedo
Alpha Cutoff 0.5
Metallic 0
Smoothness 0.5
Source Metallic Alpha



Gold (Material)
Shader Standard Edit...
Rendering Mode Opaque
Main Maps
Albedo
Metallic 1
Smoothness 0.456
Source Metallic Alpha



Gold Faded (Material)
Shader Standard Edit...
Rendering Mode Fade
Main Maps
Albedo
Metallic 0
Smoothness 1
Source Albedo Alpha



Glass (Material)
Shader Standard Edit...
Rendering Mode Transparent
Main Maps
Albedo
Metallic 0
Smoothness 0.5
Source Metallic Alpha

Sebastian Grygorczuk - STEM Institute at CCNY

Challenge: Create Material

1. Go into the Materials Asset Folder and create a brand new Material
2. Using the Lock on the Inspector keep the Material Information available as you go to the Texture Folder and assign the Albedo using one of the provided textures or one you download from the internet.
3. Edit the textures to fit how you'd like them to interact with the world.
4. Then drag that material Asset from the Project View onto your character

Material

[1] Rendering Cutout - Allows us to show portions of the texture by cutting out the darker parts of the alpha channel. It's a great way for creating grass and bushes.

[3] Albedo Map - Determines the color of the material in light, that's where you connect your texture. You can edit the tint and transparency of the original texture here.

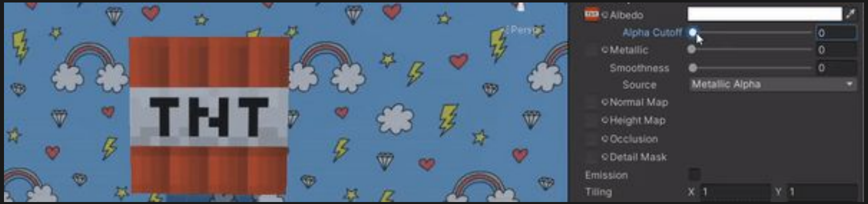
[4] Metallic - Controls how metcalic the object looks like.

[4] Shader Specular - How much does it reflect light sources in the scene.

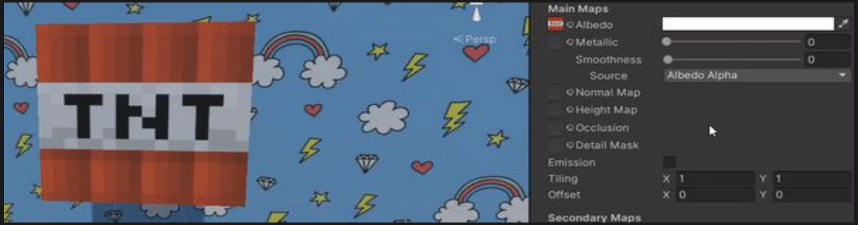
For both metallic and specular you could include an image map of which areas you'd want to be more metallic/specular but that's beyond our scope.

Additional Resources: Albedo Map, Metallic, Specular

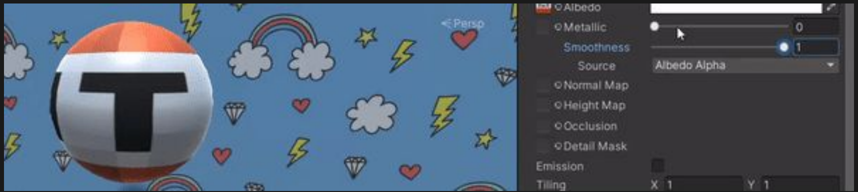
Sebastian Grygorczuk - STEM Institute at CCNY


Persp

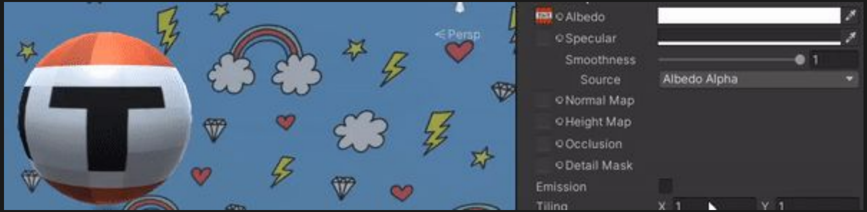
Albedo
Alpha Cutoff 0
Metallic 0
Smoothness 0
Source Metallic Alpha
Normal Map
Height Map
Occlusion
Detail Mask
Emission
Tiling X 1 Y 1


Persp

Main Maps
Albedo
Metallic 0
Smoothness 0
Source Albedo Alpha
Normal Map
Height Map
Occlusion
Detail Mask
Emission
Tiling X 1 Y 1
Offset X 0 Y 0
Secondary Maps


Persp

Albedo
Metallic 0
Smoothness 1
Source Albedo Alpha
Normal Map
Height Map
Occlusion
Detail Mask
Emission
Tiling X 1 Y 1


Persp

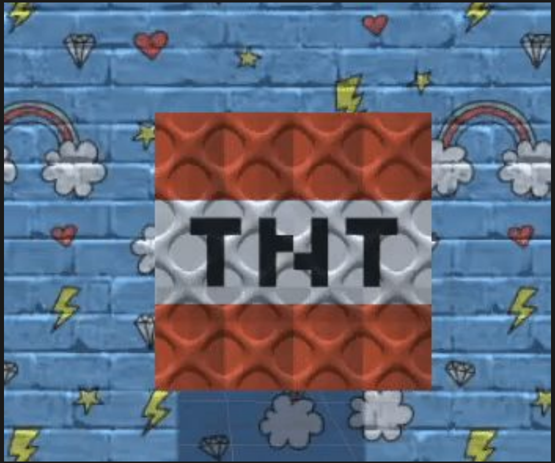
Albedo
Specular
Smoothness 1
Source Albedo Alpha
Normal Map
Height Map
Occlusion
Detail Mask
Emission
Tiling X 1 Y 1

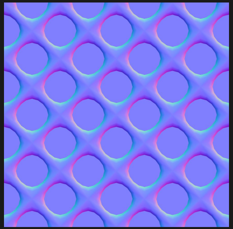
Material

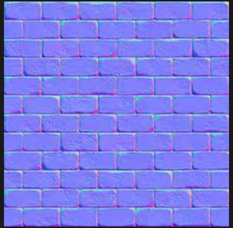
[5] Normal Maps - Also known as a bump map. Allows us to give the illusion of extra geometry by having bumps catching the lighting. This is done with the bluey-purple image where the blue, red and green bits represent which way the light should bounce off in.

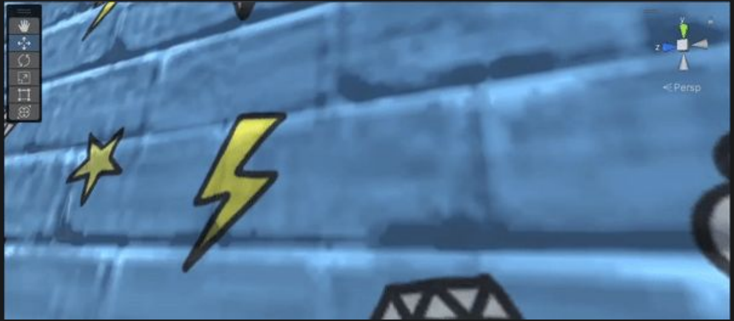
[5] Height Maps - Work in conjunction with Normal Maps they can provide the extra detail by creating depth on the flat surface. They use black and white to high each pixel is from the other, the darker it is the lower it is and the brighter it is the higher it is.

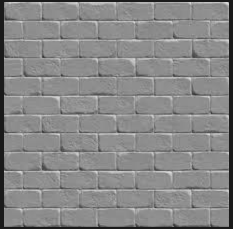
Additional Resources: Normal Map, Height Map











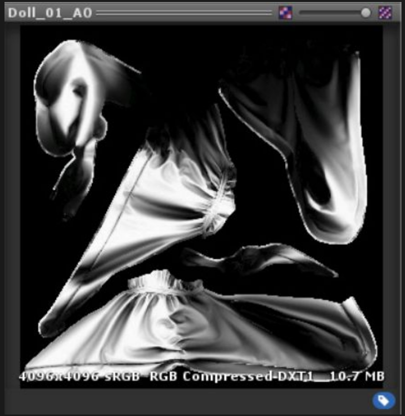
Challenge

[6] Occlusion Map is used to provide information about which areas of the model should receive high or low indirect lighting. Indirect lighting comes from ambient lighting and reflections, so steep concave parts of your model like a crack or a fold would not realistically receive much indirect light." This is beyond our scope but it's good to know.

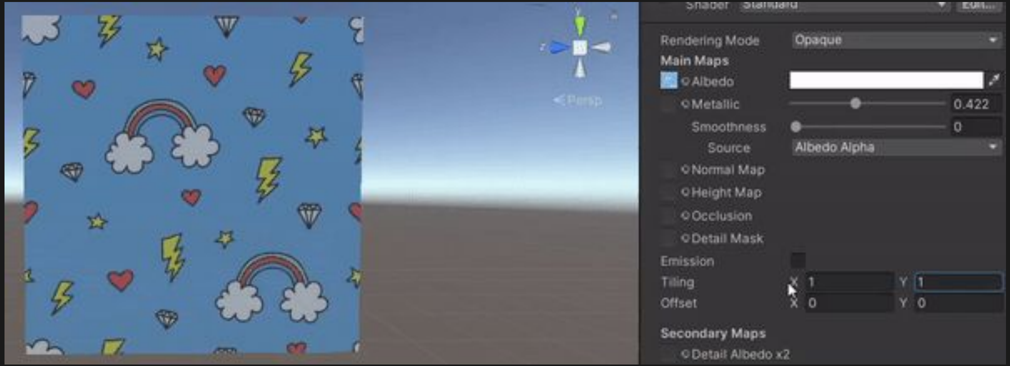
[7] Titling - Allows the texture to be repeated across the X and Y axis and be offset. For best results you want to have Seamless or Tiled Textures that will repeat the pattern when replicated.

Additional Resources: Occlusion Map

Sebastian Grygorczuk - STEM Institute at CCNY


Doll_01_AO

4096x4096 sRGB RGB Compressed-DXT1 10.7 MB



Shader Standard

Rendering Mode Opaque

Main Maps

Albedo

Metallic 0.422

Smoothness 0

Source Albedo Alpha

Normal Map

Height Map

Occlusion

Detail Mask

Emission

Tiling X 1 Y 1

Offset X 0 Y 0

Secondary Maps

Detail Albedo x2

Challenge: Build a House

Your person need a place to live

Using the Normal Maps, Height Maps and Tiling create them a home.