Intro to Scripting part 1 & 2

Variables
Functions
Syntax
Arithmetic operators
If statement

Sounds
Colors
Materials
Interaction
-button input
-key input
Project organization

Sounds

Supported Audio Formats

MPEG layer 3 .mp3
Ogg Vorbis .ogg
Microsoft Wave .wav
Audio Interchange File Format .aiff / .aif
Ultimate Soundtracker module .mod
Impulse Tracker module .it
Scream Tracker module .s3m
FastTracker 2 module .xm

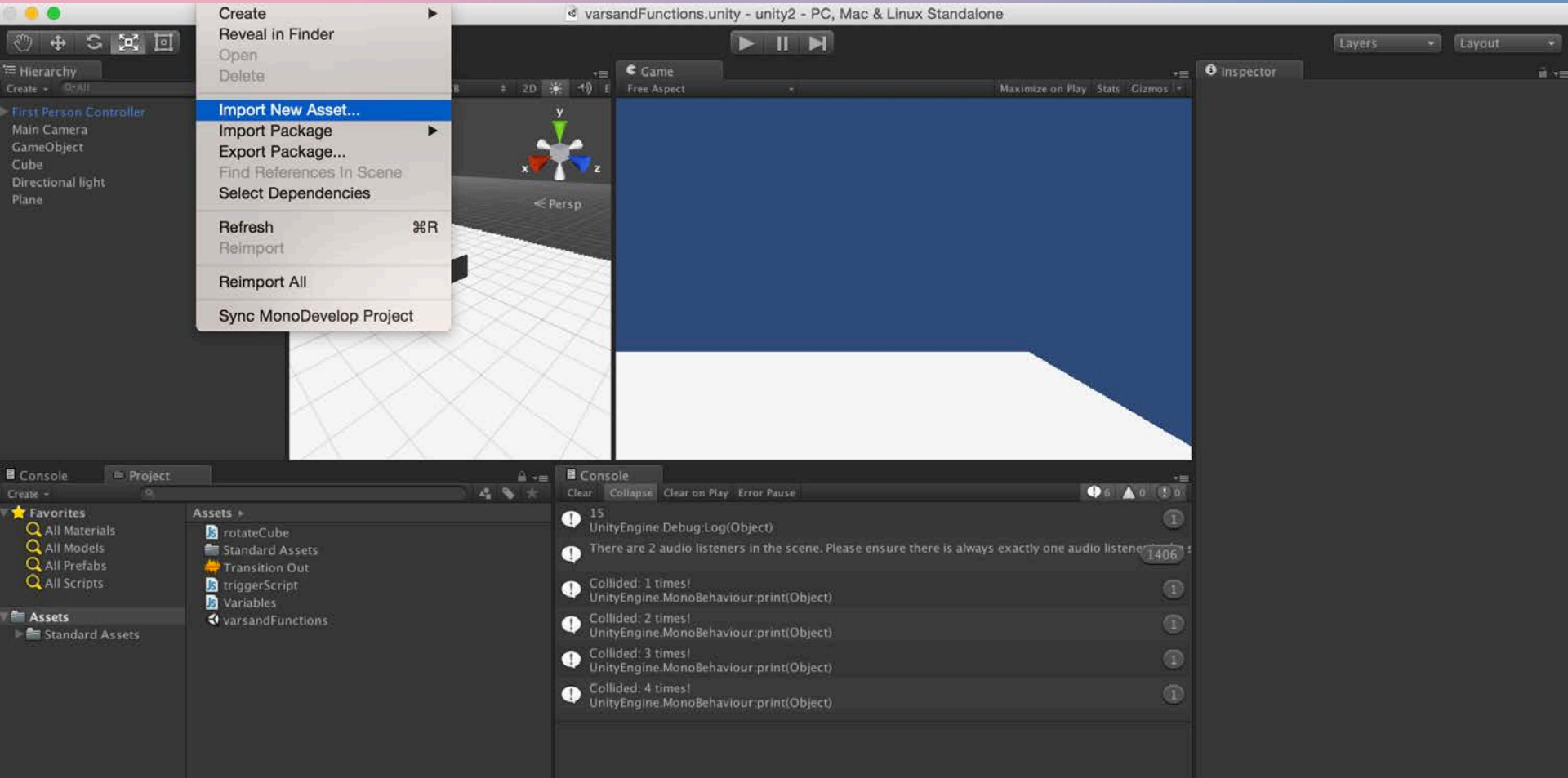
Sounds

- Import new Asset (sound effect/s)
- Add Audio Source to the Cube (Inspector>Add Component >Audio Source)
- Uncheck button “Play On Awake”

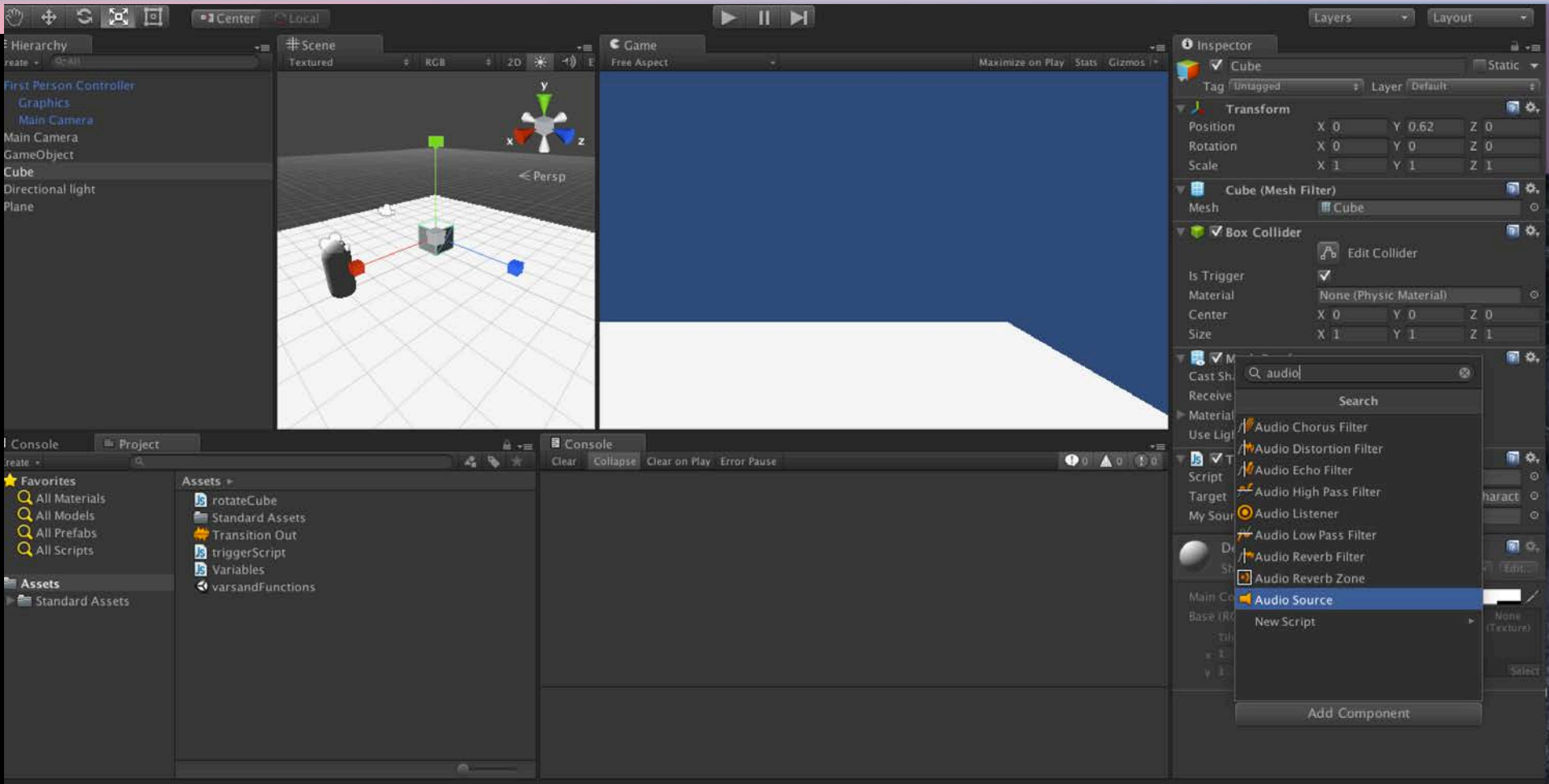
Sounds

```
public class ifstatement : MonoBehaviour {
        public AudioClip mySound;
void Update()
 {
        if (Input.GetKeyDown(KeyCode.Y))
        {
        GetComponent<Renderer>().material.color = Color.yellow;
        GetComponent<AudioSource>().PlayOneShot(mySound);
        }
 }
}
```

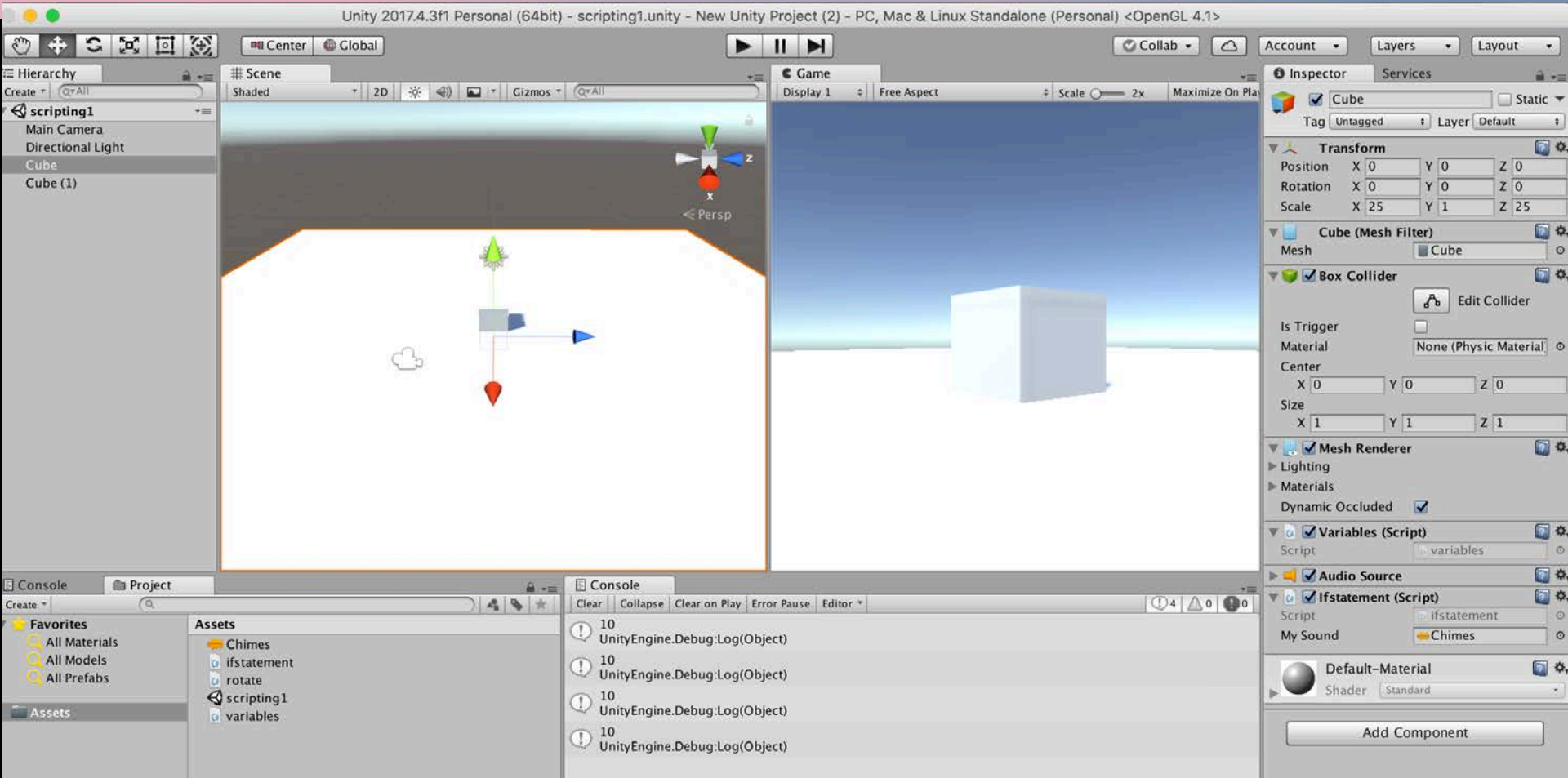
Sounds


Create
Reveal in Finder
Open
Delete
Import New Asset...
Import Package
Export Package...
Find References In Scene
Select Dependencies
Refresh ⌘R
Reimport
Reimport All
Sync MonoDevelop Project
varsandFunctions.unity - unity2 - PC, Mac & Linux Standalone
Hierarchy
First Person Controller
Main Camera
GameObject
Cube
Directional light
Plane
Game
Free Aspect
Maximize on Play
Stats
Gizmos
Persp
Inspector
Layers
Layout
Console
Project
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
Standard Assets
rotateCube
Standard Assets
Transition Out
triggerScript
Variables
varsandFunctions
Clear
Collapse
Clear on Play
Error Pause
15
UnityEngine.Debug:Log(Object)
There are 2 audio listeners in the scene. Please ensure there is always exactly one audio listene
1406
Collided: 1 times!
UnityEngine.MonoBehaviour:print(Object)
Collided: 2 times!
UnityEngine.MonoBehaviour:print(Object)
Collided: 3 times!
UnityEngine.MonoBehaviour:print(Object)
Collided: 4 times!
UnityEngine.MonoBehaviour:print(Object)

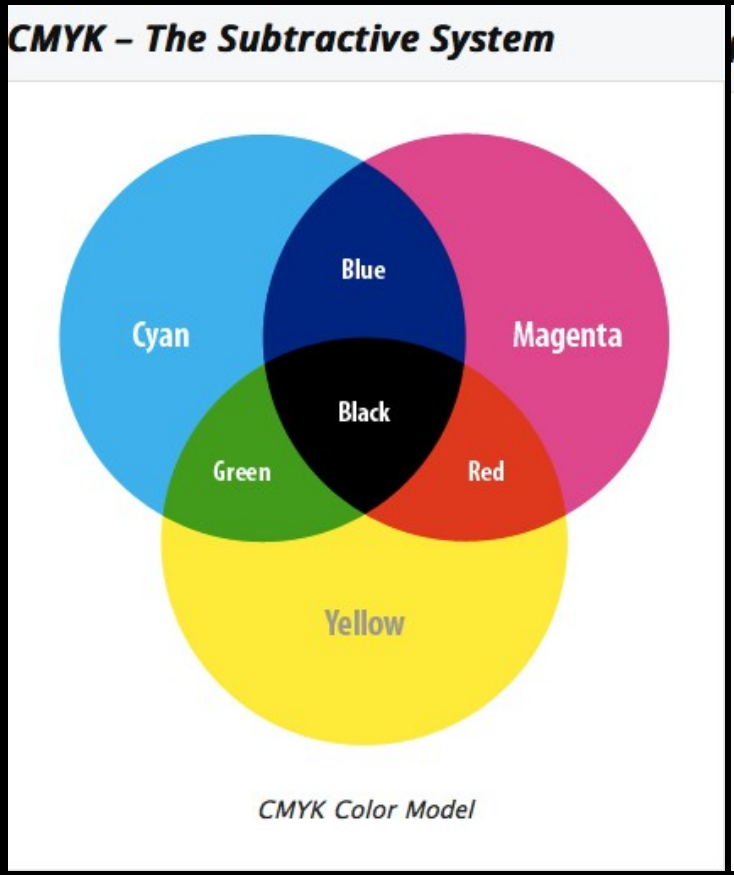
Sounds

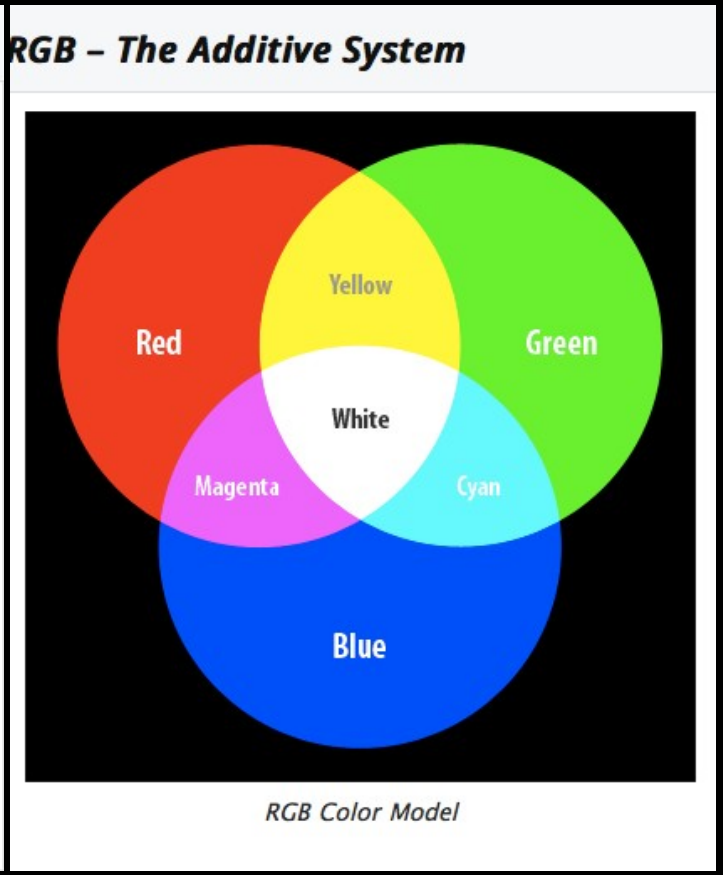

Hierarchy
First Person Controller
Graphics
Main Camera
Main Camera
GameObject
Cube
Directional light
Plane
Scene
Persp
Game
Free Aspect
Inspector
Cube
Transform
Position
Rotation
Scale
Cube (Mesh Filter)
Mesh
Box Collider
Edit Collider
Is Trigger
Material
None (Physic Material)
Center
Size
audio
Search
Audio Chorus Filter
Audio Distortion Filter
Audio Echo Filter
Audio High Pass Filter
Audio Listener
Audio Low Pass Filter
Audio Reverb Filter
Audio Reverb Zone
Audio Source
New Script
Add Component
Console
Project
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
Standard Assets
rotateCube
Standard Assets
Transition Out
triggerScript
Variables
varsandFunctions

Sounds

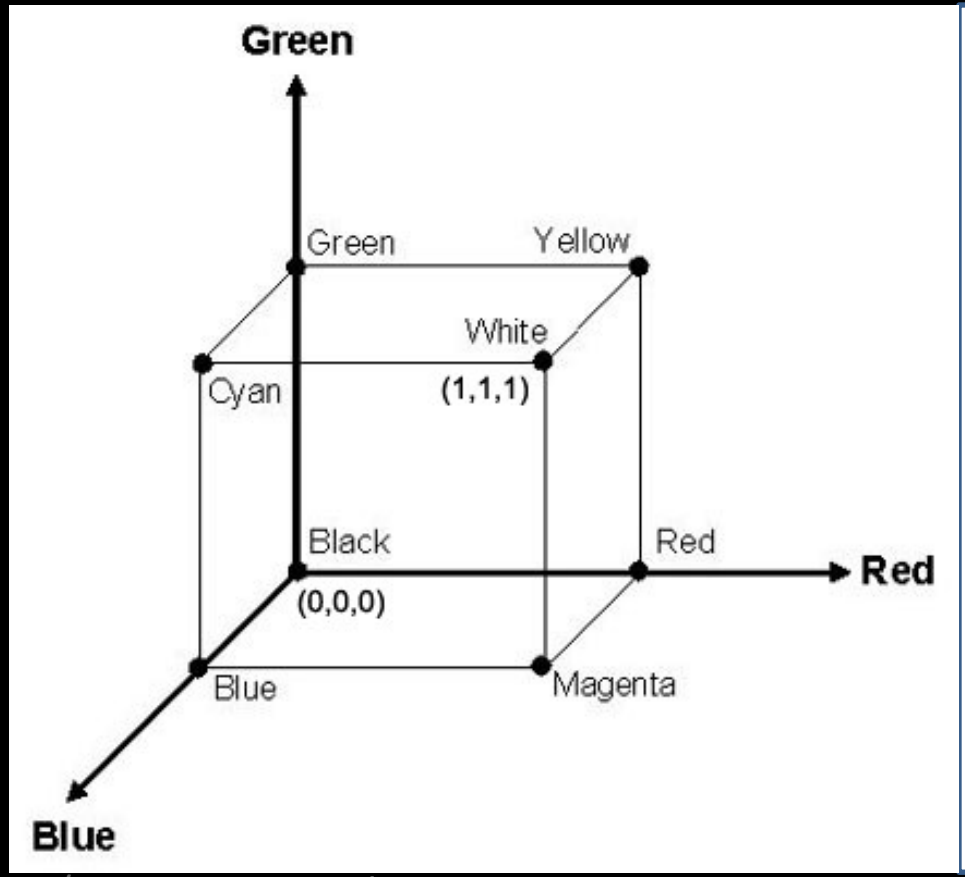

Unity 2017.4.3f1 Personal (64bit) - scripting1.unity - New Unity Project (2) - PC, Mac & Linux Standalone (Personal) <OpenGL 4.1>
Hierarchy
scripting1
Main Camera
Directional Light
Cube
Cube (1)
Scene
Persp
Game
Inspector
Services
Cube
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 25 Y 1 Z 25
Cube (Mesh Filter)
Mesh Cube
Box Collider
Edit Collider
Is Trigger
Material None (Physic Material
Center X 0 Y 0 Z 0
Size X 1 Y 1 Z 1
Mesh Renderer
Lighting
Materials
Dynamic Occluded
Variables (Script)
Script variables
Audio Source
Ifstatement (Script)
Script ifstatement
My Sound Chimes
Default-Material
Shader Standard
Add Component
Console
Project
Favorites
All Materials
All Models
All Prefabs
Assets
Chimes
ifstatement
rotate
scripting1
variables
Clear
Collapse
Clear on Play
Error Pause
Editor
10
UnityEngine.Debug:Log(Object)

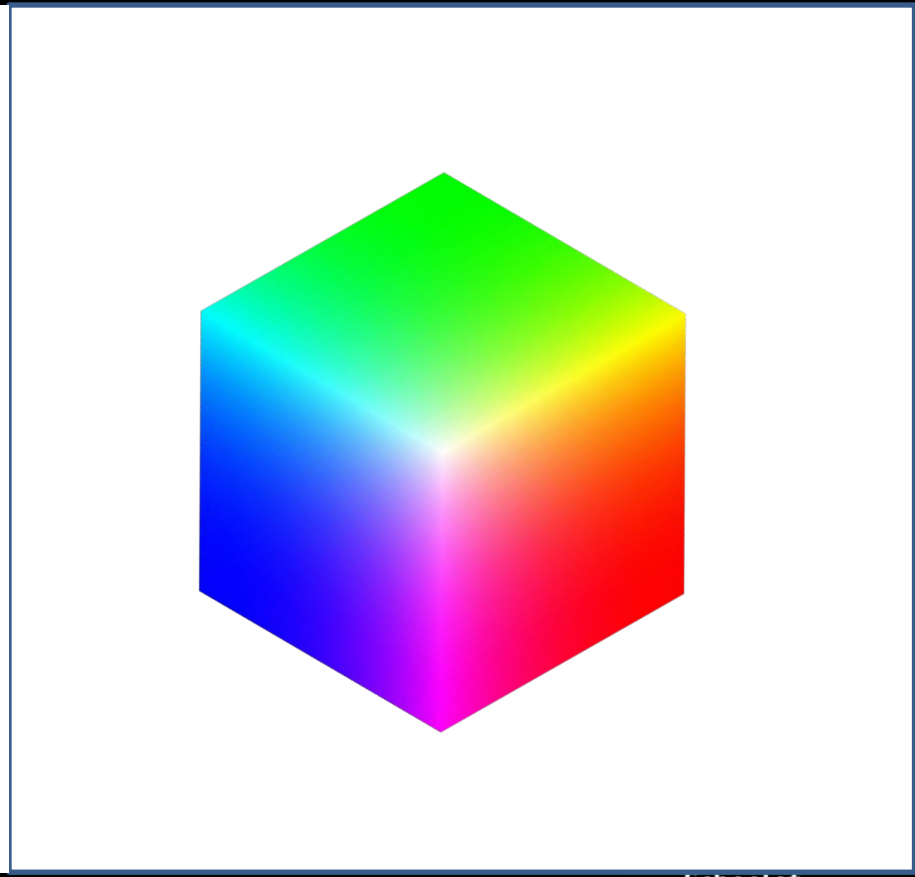
Color Systems





Color Cube


Green
Green
Yellow
White
Cyan
(1,1,1)
Black
Red
Red
(0,0,0)
Blue
Magenta
Blue



RGBA Color

Representation of RGBA colors

Values for red, green, blue and alpha are floating point values with a range from 0 to 1 (ex. 0.3, 0.77)

Alpha component (a) defines transparency

alpha of 1 is completely opaque, alpha of zero is completely transparent

Black RGBA is (0, 0, 0, 1)

Blue RGBA is (0, 0, 1, 1)

Gray RGBA is (0.5, 0.5, 0.5, 1)

Clear Completely transparent. RGBA is (0, 0, 0, 0)

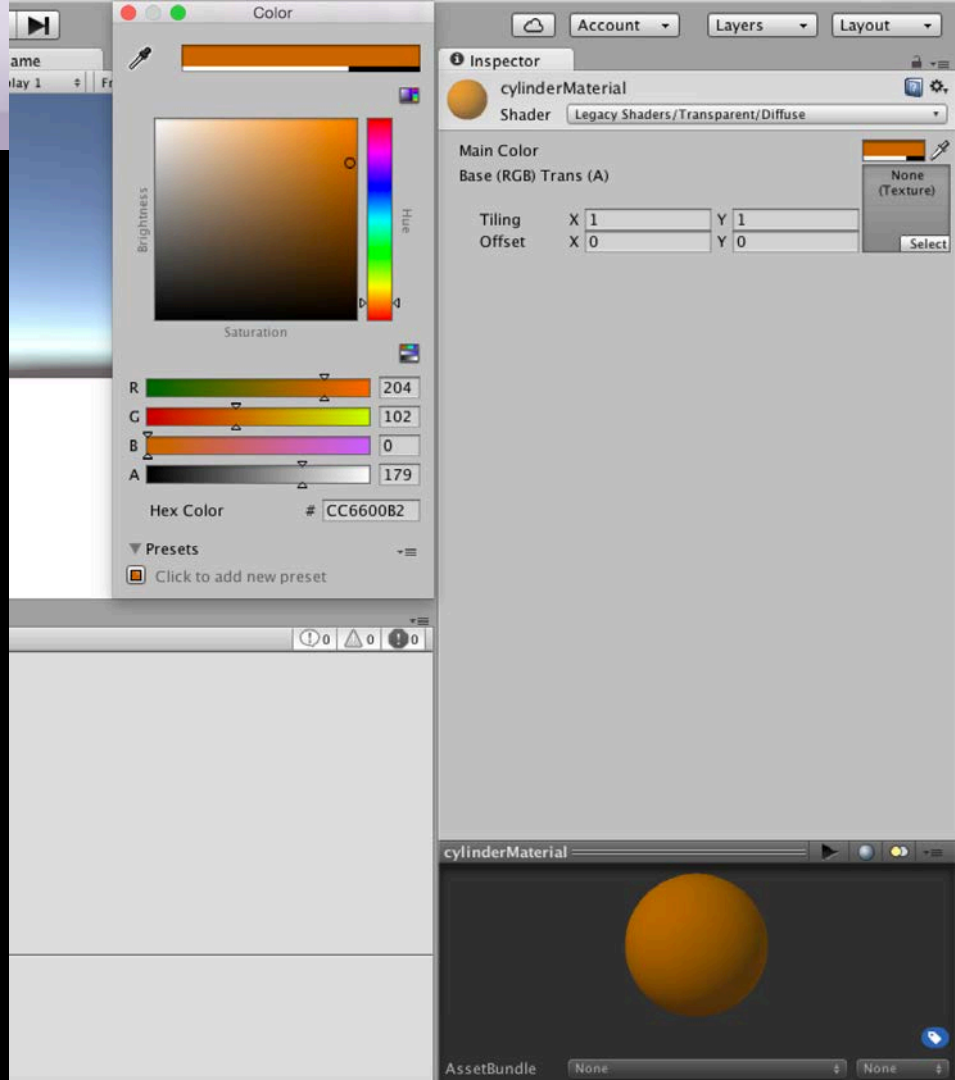
RGBA Color

Black RGBA is (0, 0, 0, 1)
Blue RGBA is (0, 0, 1, 1)
Gray RGBA is (0.5, 0.5, 0.5, 1)
Clear Completely transparent. RGBA is (0, 0, 0, 0)

Magenta?
Yellow?
cyan?

RGBA Color

Unity Color Window



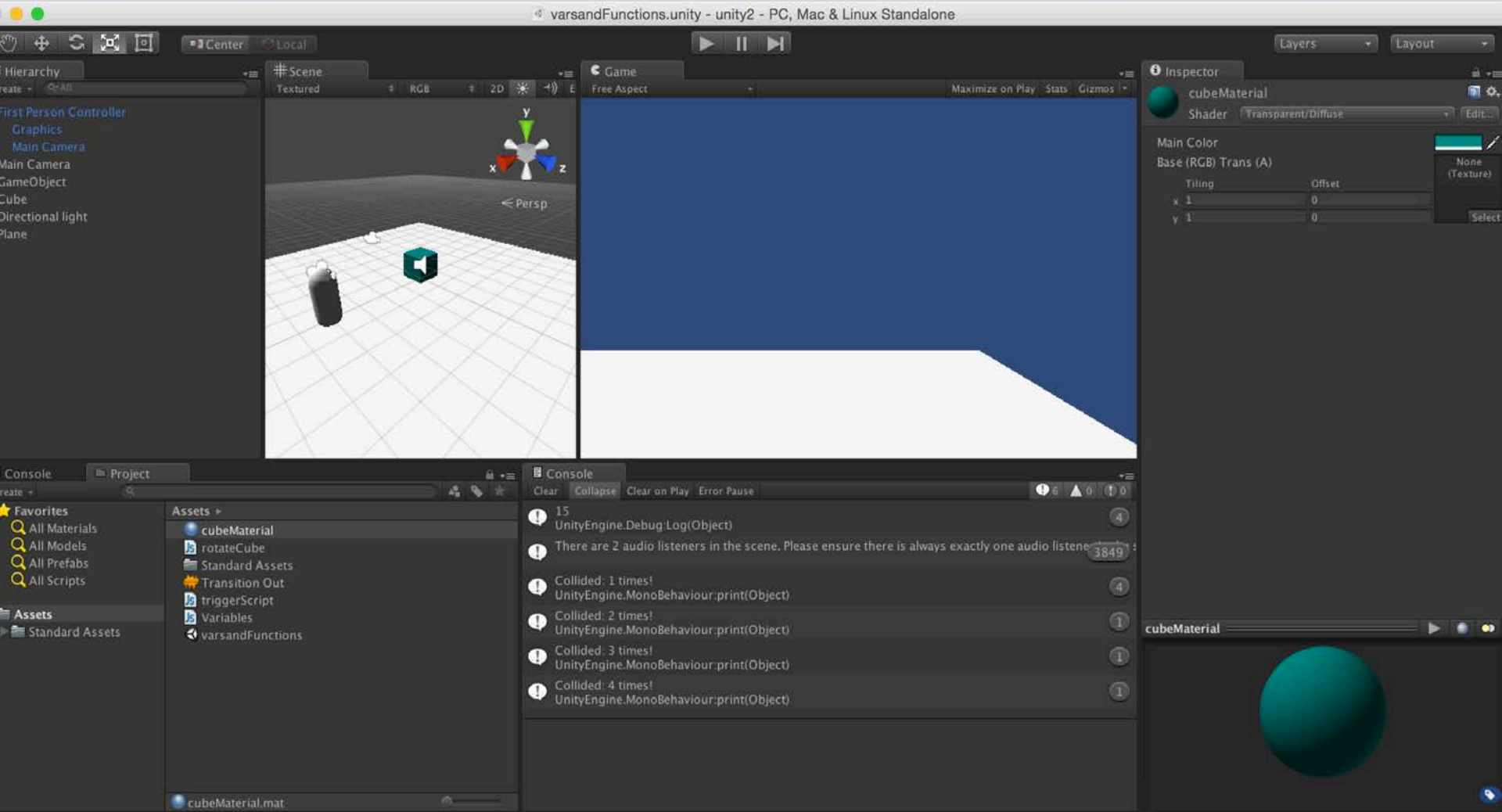
Colors

Use the scene from last lab with the following components:

- 3D Plane
- 3D Cube

Create 2 new materials and add trasnparent/diffuse shader
Assign materials to the cube and the cylinder (drag and drop)

Assets> Create > Material


varsandFunctions.unity - unity2 - PC, Mac & Linux Standalone
Center
Local
Layers
Layout
Hierarchy
First Person Controller
Graphics
Main Camera
Main Camera
GameObject
Cube
Directional light
Plane
Scene
Textured
RGB
2D
Persp
Game
Free Aspect
Maximize on Play
Stats
Gizmos
Inspector
cubeMaterial
Shader
Transparent/Diffuse
Edit...
Main Color
Base (RGB) Trans (A)
None (Texture)
Tiling
Offset
Select
Console
Project
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
Standard Assets
Assets
cubeMaterial
rotateCube
Standard Assets
Transition Out
triggerScript
Variables
varsandFunctions
cubeMaterial.mat
Console
Clear
Collapse
Clear on Play
Error Pause
15
UnityEngine.Debug:Log(Object)
There are 2 audio listeners in the scene. Please ensure there is always exactly one audio listene
3849
Collided: 1 times!
UnityEngine.MonoBehaviour:print(Object)
Collided: 2 times!
UnityEngine.MonoBehaviour:print(Object)
Collided: 3 times!
UnityEngine.MonoBehaviour:print(Object)
Collided: 4 times!
UnityEngine.MonoBehaviour:print(Object)
cubeMaterial

If statement

```
public class ifstatement : MonoBehaviour {
void Update()    {
if (Input.GetKeyDown(KeyCode.C))
{
        GetComponent<Renderer> ().material.color = Color.cyan;
 }
 if (Input.GetKeyDown(KeyCode.M))
{
        GetComponent<Renderer>().material.color = Color.magenta;
 }
 if (Input.GetKeyDown(KeyCode.Y))
{
        GetComponent<Renderer>().material.color = Color.yellow;
}
 }}
```

Color script

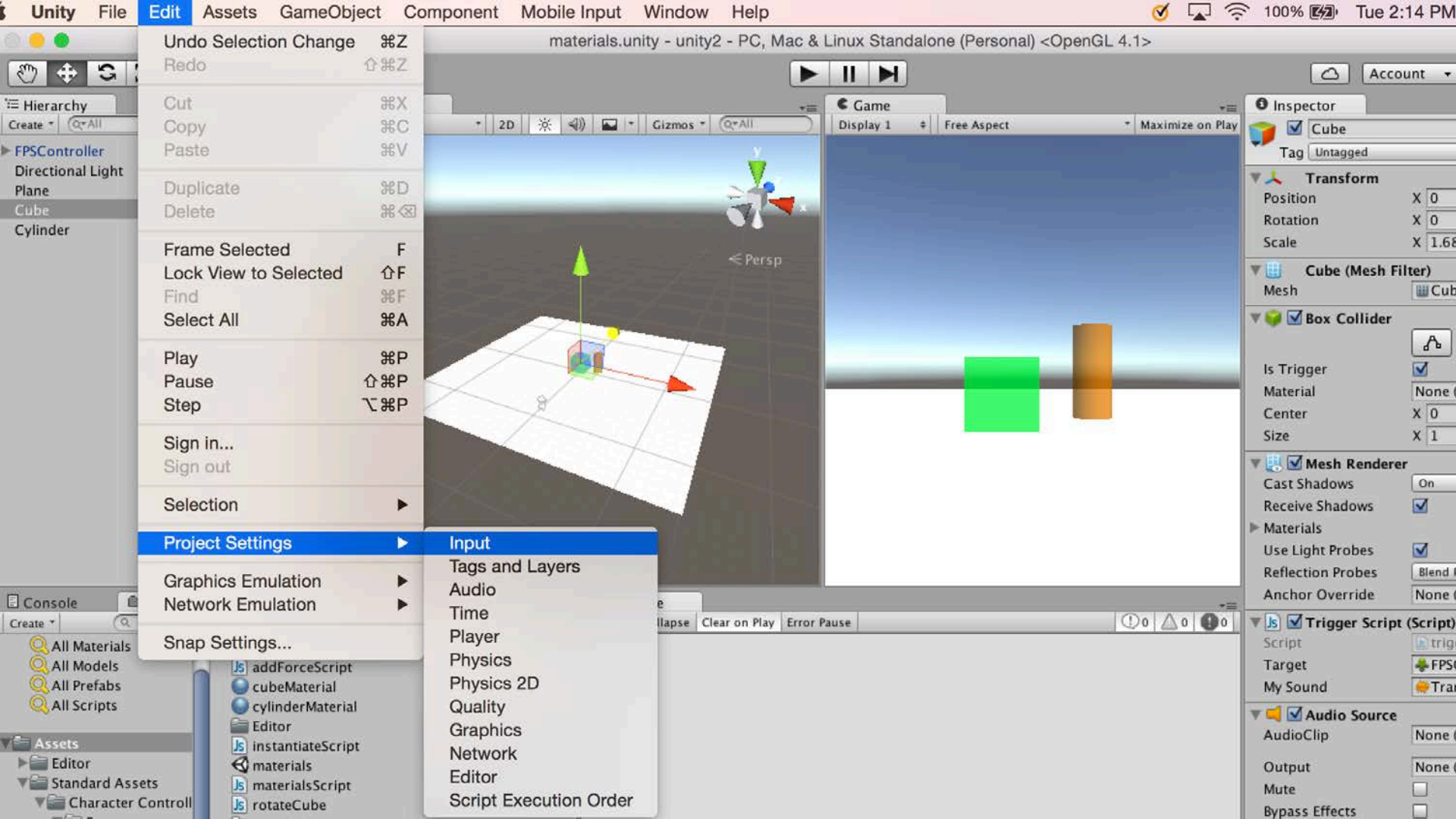
```
public class ifstatement : MonoBehaviour {
void Update()    {
if (Input.GetKeyDown(KeyCode.C))
{
            GetComponent<Renderer>().material.color=new Color(0.0f, 0.5f, 0.2f, 0.9f);
 }
 if (Input.GetKeyDown(KeyCode.M))
{
            GetComponent<Renderer>().material.color = Color.magenta;
 }
 if (Input.GetKeyDown(KeyCode.Y))
{
            GetComponent<Renderer>().material.color = Color.yellow;
}
 }}
```

Interaction – Key and Button input

The input manager allows to name an input and specify a key or button for it.

You can access keys on the keyboard and mouse buttons through scripting interface.

Edit > Project Settings >Input

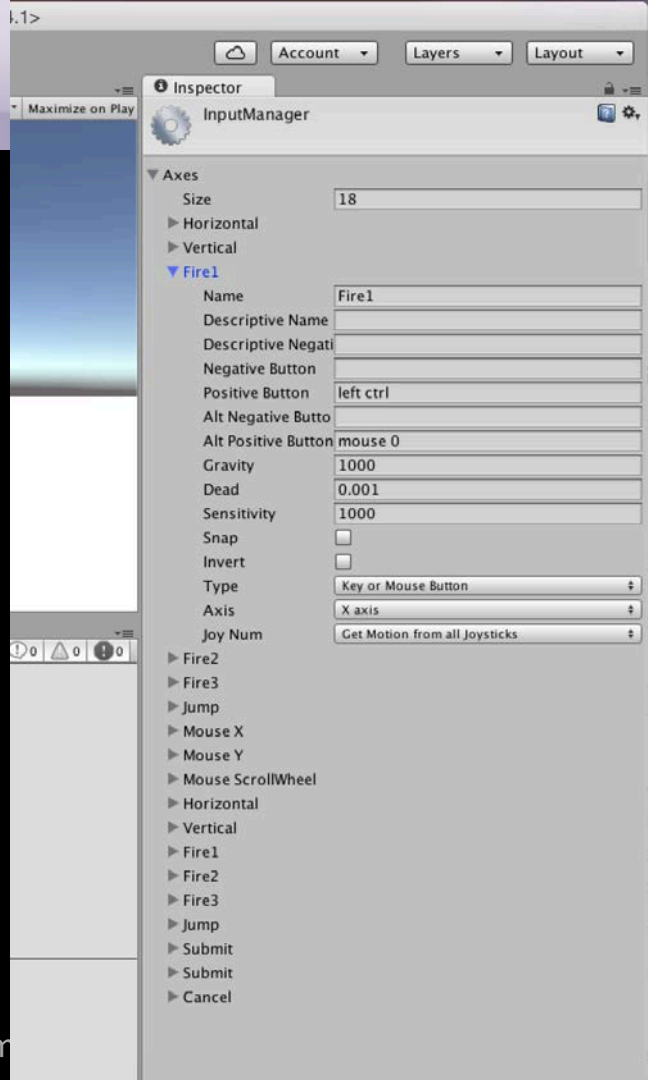

Unity File Edit Assets GameObject Component Mobile Input Window Help
100% Tue 2:14 PM
materials.unity - unity2 - PC, Mac & Linux Standalone (Personal) <OpenGL 4.1>
Undo Selection Change ⌘Z
Redo ⇧⌘Z
Cut ⌘X
Copy ⌘C
Paste ⌘V
Duplicate ⌘D
Delete ⌘⌫
Frame Selected F
Lock View to Selected ⇧F
Find ⌘F
Select All ⌘A
Play ⌘P
Pause ⇧⌘P
Step ⌥⌘P
Sign in...
Sign out
Selection
Project Settings
Graphics Emulation
Network Emulation
Snap Settings...
Input
Tags and Layers
Audio
Time
Player
Physics
Physics 2D
Quality
Graphics
Network
Editor
Script Execution Order
Hierarchy
FPSController
Directional Light
Plane
Cube
Cylinder
Persp
Game
Display 1
Free Aspect
Maximize on Play
Inspector
Transform
Position
Rotation
Scale
Cube (Mesh Filter)
Box Collider
Is Trigger
Material
Center
Size
Mesh Renderer
Cast Shadows
Receive Shadows
Materials
Use Light Probes
Reflection Probes
Anchor Override
Trigger Script (Script)
Audio Source
AudioClip
Output
Mute
Bypass Effects
Console
Clear on Play
Error Pause
All Materials
All Models
All Prefabs
All Scripts
addForceScript
cubeMaterial
cylinderMaterial
Editor
instantiateScript
materials
materialsScript
rotateCube
Assets
Editor
Standard Assets
Character Controll

Interaction – Key and Button input

Input.GetButtonDown("Fire1"))
Mouse button left mouse 0

Input.GetButtonDown("Fire2"))
Mouse button right mouse 1

Input.GetKeyDown("Jump"))
space key space



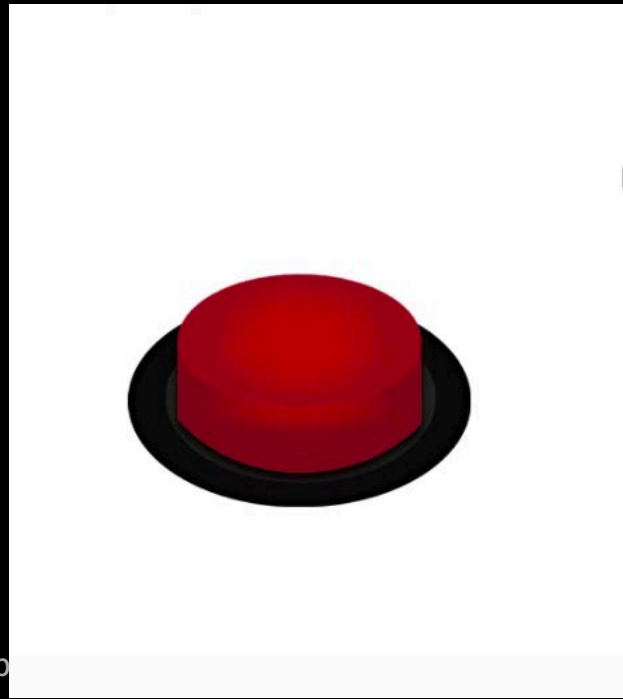
Button Input script

```
public class buttonInput : MonoBehaviour {
void Update()    {
if (Input.GetButtonDown("Fire1"))
{
                GetComponent<Renderer>().material.color=new Color(0.0f, 0.5f, 0.2f, 0.9f);
 }
 if (Input.GetButtonDown("Fire2"))
{
                GetComponent<Renderer>().material.color = Color.magenta;
 }
 if (Input.GetButtonDown("Jump"))
{
                GetComponent<Renderer>().material.color = Color.yellow;
}
 }}
```

Interaction – Key and Button input

GetButtonDown/Up – before any interaction

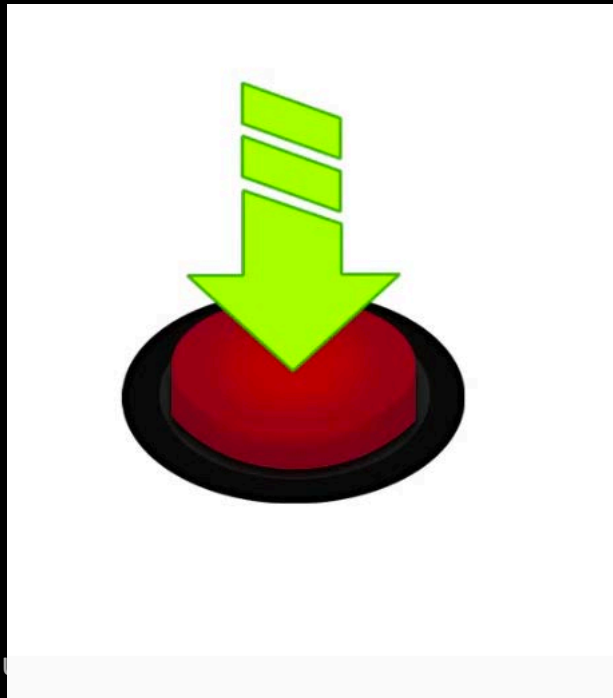
GetButtonDown: false
GetButton: false
GetButtonUp: false



Interaction – Key and Button input

GetButtonDown/Up – on the first frame / on press

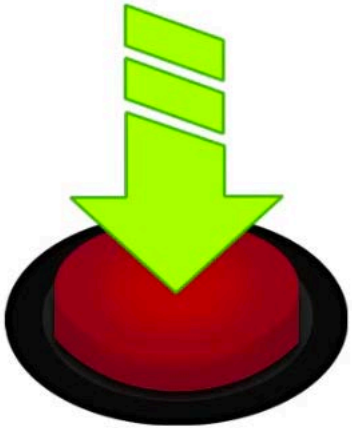
GetButtonDown: true
GetButton: true
GetButtonUp: false



Interaction – Key and Button input

GetButtonDown/Up – after the first frame / holding button down

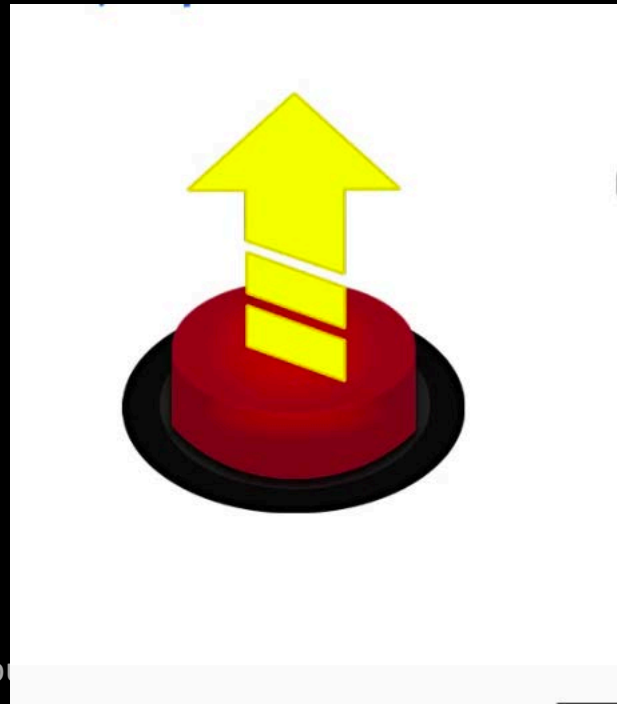
GetButtonDown: false
GetButton: true
GetButtonUp: false



Interaction – Key and Button input

GetButtonDown/Up – on the first frame / on release

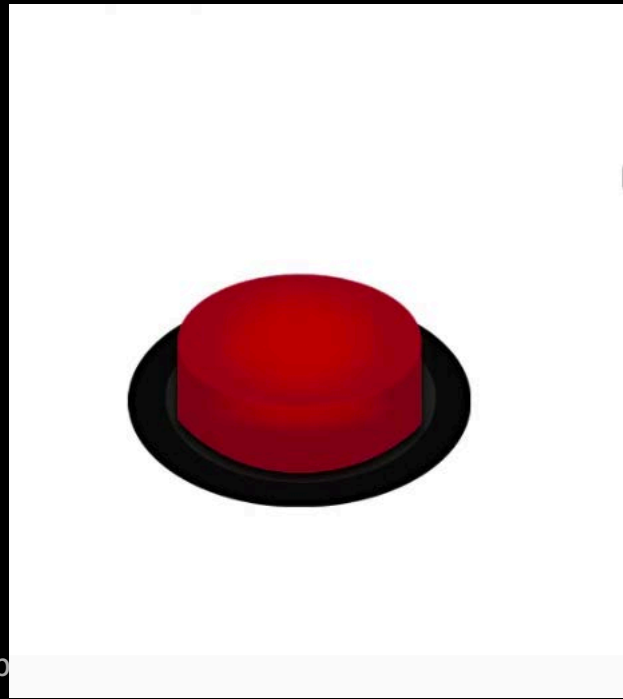
GetButtonDown: false
GetButton: false
GetButtonUp: true



Interaction – Key and Button input

GetButtonDown/Up – after the first frame / on release

GetButtonDown: false
GetButton: false
GetButtonUp: false



Interaction – Key and Button input

Create new object Sphere, add rigidbody component
Create new Script KeyInput.cs
Assign script to the sphere

Key Input

```
void Update () {
  if (Input.GetKey("up"))
   {
        Debug.Log("up arrow key is held down");
  }
   if (Input.GetKey("down"))
  {
        Debug.Log("down arrow key is held down");
   }
  if (Input.GetKeyDown(KeyCode.Space))
  {
GetComponent<Rigidbody>().AddForce(transform.forward * 200f);
        }
    }
```

Interaction – Key and Button input

Uncheck Use Gravity in the inspector
Test the scene

Interaction – Key and Button input

Add useGravity to Rigidbody component in the script

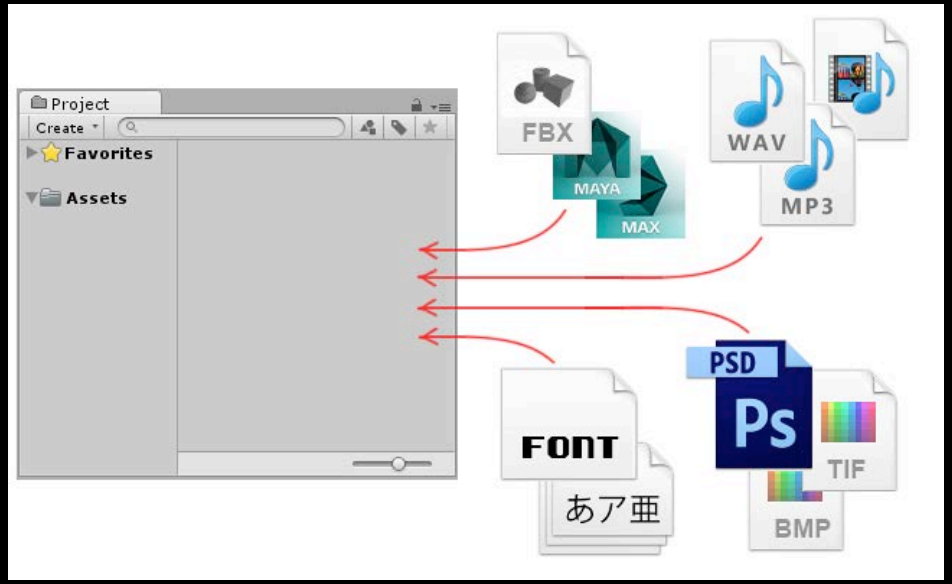
```
    GetComponent.<Rigidbody>().AddForce(transform.forward * 500f);

    GetComponent.<Rigidbody>().useGravity = true;
}
```

Project Assets

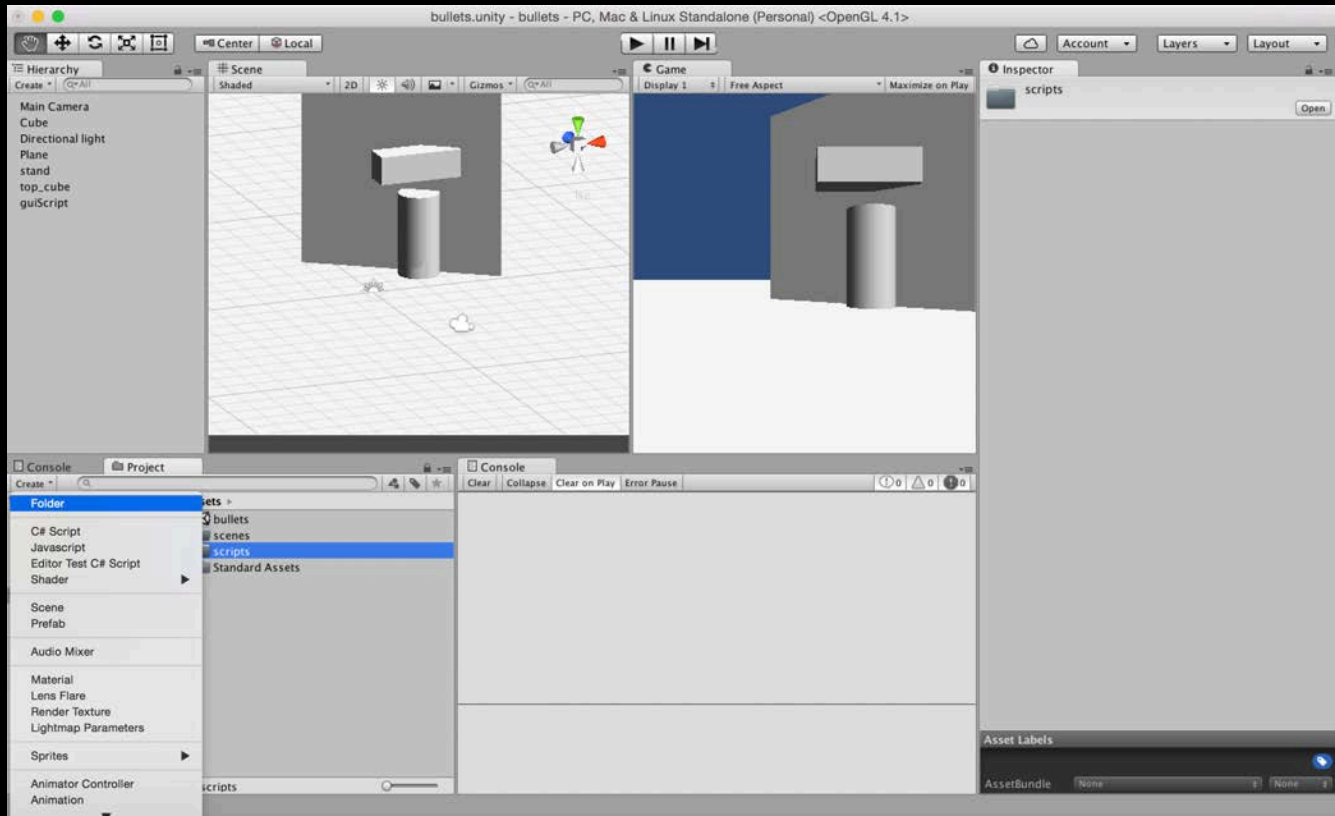
Assets are the models, textures, sounds and all other media files from which you make your 3D project

- Audio Files
- Materials
- Meshes
- Textures
- Prefabs
- Scripts
- Prefabs



Project Organization

Project > Create >Folder

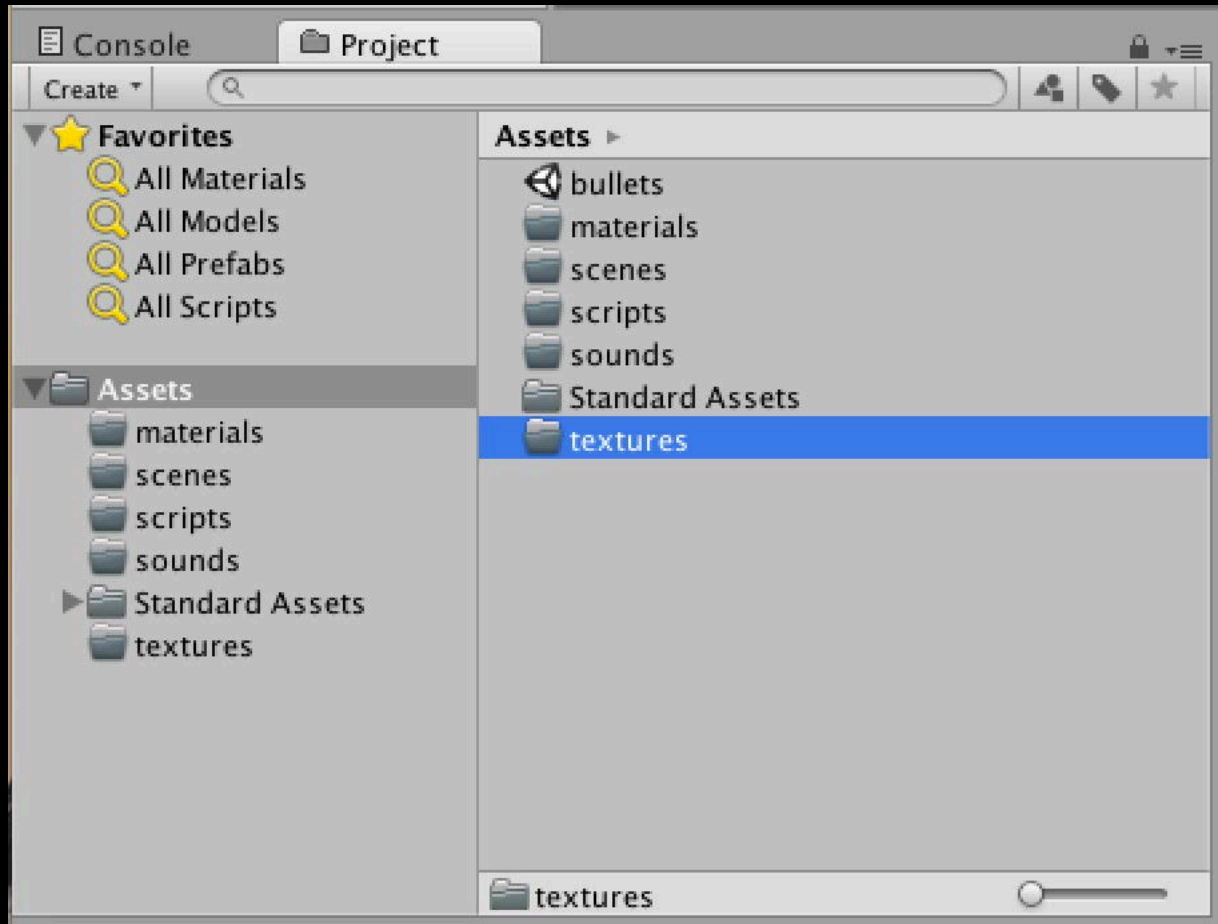


Project Organization

Assets > Folder >

Rename

- Materials
- Scripts
- Scenes
- Prefabs
- Textures
- Sounds



Project Organization

- Keep your assets organized
- Be consistent with naming. If you decide to use camel case for directory names and low letters for assets, stick to that convention.

- Use lowercase folder names

- Use camelCase naming convention if needed

- Sort all the assets in the corresponding folders

Project Organization

- Do not store any asset files in the root directory. Use subdirectories whenever possible.
- Do not create any additional directories in the root directory, unless you really need to.
- Use 3rd-Party to store assets imported from the Asset Store. They usually have their own structure that shouldn't be altered.

Project Organization

- 3rd-Party
- Animations
- Audio
  - Music
  - SFX
- Materials
- Models
- Plugins
- Prefabs
- Resources
- Textures
- Sandbox
- Scenes
  - Levels
  - Other
- Scripts
  - Editor
- Shaders