



Center Local



Layers

Layout

Hierarchy

Create All

- Main Camera
- Directional Light
- Terrain_main

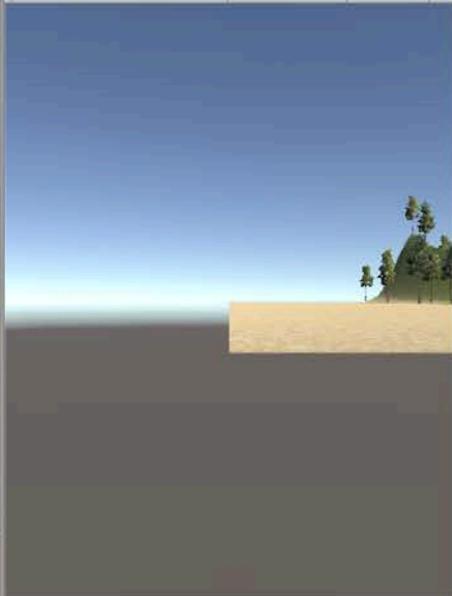
Scene

Shaded 2D Gizmos All



Game

Free Aspect Maximize on Play Mute audio



Inspector

Terrain_main Static

Tag Untagged Layer Default

Transform

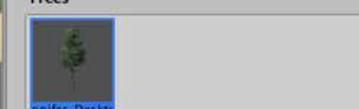
Position	X	0	Y	0	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

Terrain



Place Trees
Hold down shift to erase trees.
Hold down ctrl to erase the selected tree type.

Trees



Mass Place Trees Edit Trees... Refresh

Settings

- Brush Size
- Tree Density
- Tree Height Random?
- Lock Width to Height
- Tree Width Random?
- Color Variation
- Random Tree Rotation

Terrain Collider

- Material None (Physic Material)
- Terrain Data New Terrain 1
- Enable Tree Colliders

Add Component

Console Project

Create All

- All Models
- All Prefabs
- All Scripts

Assets

- Editor
- Standard Assets
- Characters
 - FirstPersonChar
 - Audio
 - Prefabs
 - Scripts
- PhysicsMaterials
- RollerBall

Library

This folder is empty

Textures

Textures should be in the following format to enable ‘tiling’

Square and the power of two

128 x 128

256 x 256

512 x 512

1024 x 1024



Prefabs

pre-fabricated objects

Prefabs store a game object together with its components (transforms, appearances, scripts, etc.) and configurations for easy duplication/reuse.

- trees
- bullets
- characters, and anything else

Unity makes it easy to move around a world interactively (either in a first person or third person perspective) using prefabs.



Prefabs

Object-oriented instances can be **Instantiated** at run time

At **run time** a script can cause a new object instance to be created (instantiated) at a given location with a given set of properties

Prefabs allow functional game objects to be reused in scenes or imported into other projects as external assets.

The First Person Controller



First Person Controller

20. Assets > Import Package > Character Controller
(character in Unity 5)

Project Window > Standard Assets folder

FP Character > Prefabs > FPController

drag the FP Controller onto your scene

Make sure to position the Controller ABOVE the TERRAIN

Preview the game

explore the terrain / look around with your mouse

move with WASD or the arrow keys / jump with the space bar





Create

Reveal in Finder

Open

Delete

Import New Asset...

Import Package

Export Package...

Find References In Scene

Select Dependencies

Refresh

⌘R

Reimport

Reimport All

Run API Updater...

Sync MonoDevelop Project

Custom Package...

2D

Cameras

Characters

CrossPlatformInput

Effects

Environment

ParticleSystems

Prototyping

Utility

Vehicles

Hierarchy

Create

Main Camera
Directional Light
Terrain_main

Console

Project

Create

All Materials
All Models
All Prefabs
All Scripts

Assets
Editor
Standard Assets
Characters
FirstPersonChar
Audio
Prefabs
Scripts
PhysicsMaterials
RollerBall

Assets > Standard Assets > Characters > FirstPersonCharacter > Prefabs
FPSController
RigidBodyFPSController

terrain_main_scene.unity - IntroUnity - PC, Mac & Linux Standalone (Personal)

Play, Pause, Stop buttons

Layers, Layout dropdowns

Game view title

Free Aspect, Maximize on Play, Mute audio, Stop buttons

Inspector view title

Inspector panel content

Scripting

MONO compiler

Scripts can be written in

JavaScript

Majority of introductory tutorials are written in Javascript

C#

Unity can be integrated with the Microsoft Visual Studio editor, to get full benefits of code completion, source version control, intergration, serious developers work in C#

BOO (like Python)

Smaller development in this



Scripting

scripting is Unity's most powerful tool
gives you the ability to customize objects
control how they behave in the environment

- how to create and attach JavaScript scripts to objects in Unity
- Intro to the development environment MonoDevelop

Variables

Functions

Triggers

Collisions

Sounds

Colors



JavaScript vs C#

JavaScript

```
#pragma strict
```

```
var myInt : int = 5;
```

```
function Start ()
```

```
{
```

```
    myInt = MultiplyByTwo(myInt);
```

```
    Debug.Log (myInt);
```

```
}
```

C#

```
using UnityEngine;
```

```
using System.Collections;
```

```
public class VariablesAndFunctions  
    : MonoBehaviour
```

```
{
```

```
    int myInt = 5;
```

```
    void Start ()
```

```
{
```

```
        myInt = MultiplyByTwo(myInt);
```

```
        Debug.Log (myInt);
```

```
}
```

Scripting

You can use both C# and Javascript in one project!
(one way communication only)

My Scripts Folder (Outside)
(Compiled last)

Script
Script
script

JavaScript

Standard Assets
(Compiled first))

Script
Script
Script

C#



JavaScript Variables

- A variable is a storage location and an associated symbolic name (an identifier) which contains some known or unknown quantity or information, a value
- variables are used to store information about any aspects of a project's state



JavaScript Variables

begin with a lowercase letter

no special characters, numbers, (#, %, etc.)

cannot contain reserved keywords such as “if”, “while”, etc.

case sensitive

descriptive

no spaces

Declaration/ **Type**/ Initialization

```
var myVarBool : boolean = true;
```

```
var myVarInt : int = 10;
```



Data Types

Float	0.75
Int	10
String	“Hello”
Boolean	true / false

```
var myVarBool : boolean = true;
```

```
var myVarInt : int = 10;
```

```
Var myFloat : float = 1.4;
```



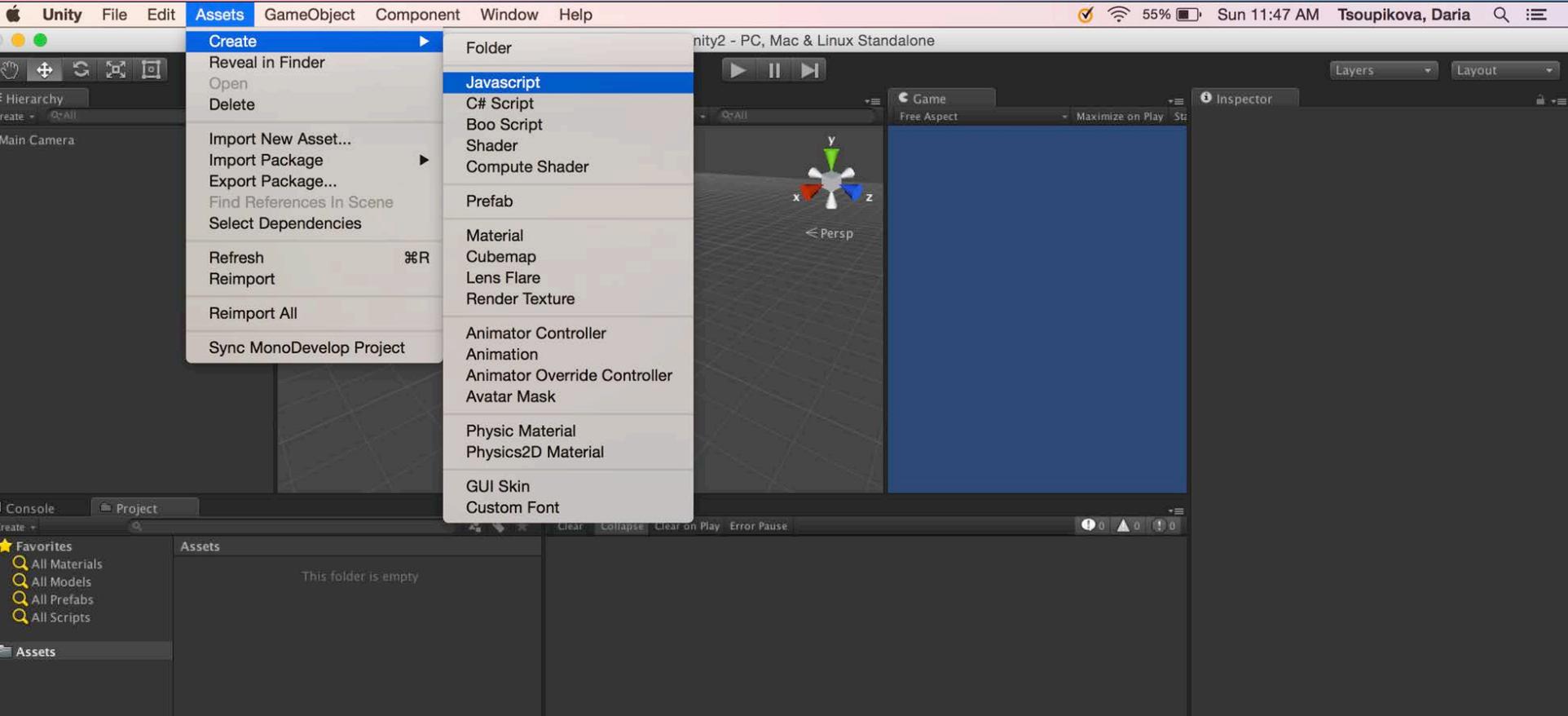
Creating scripts in Unity

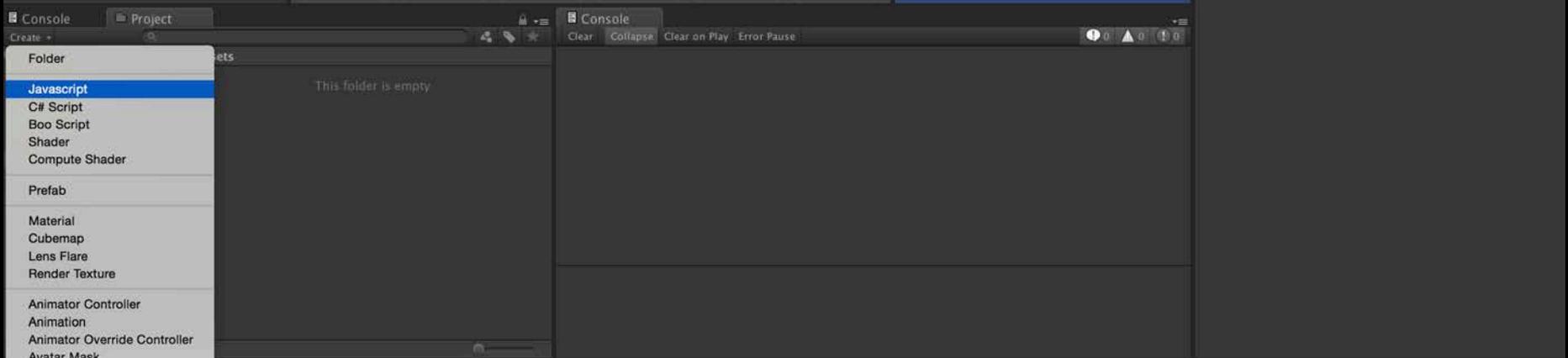
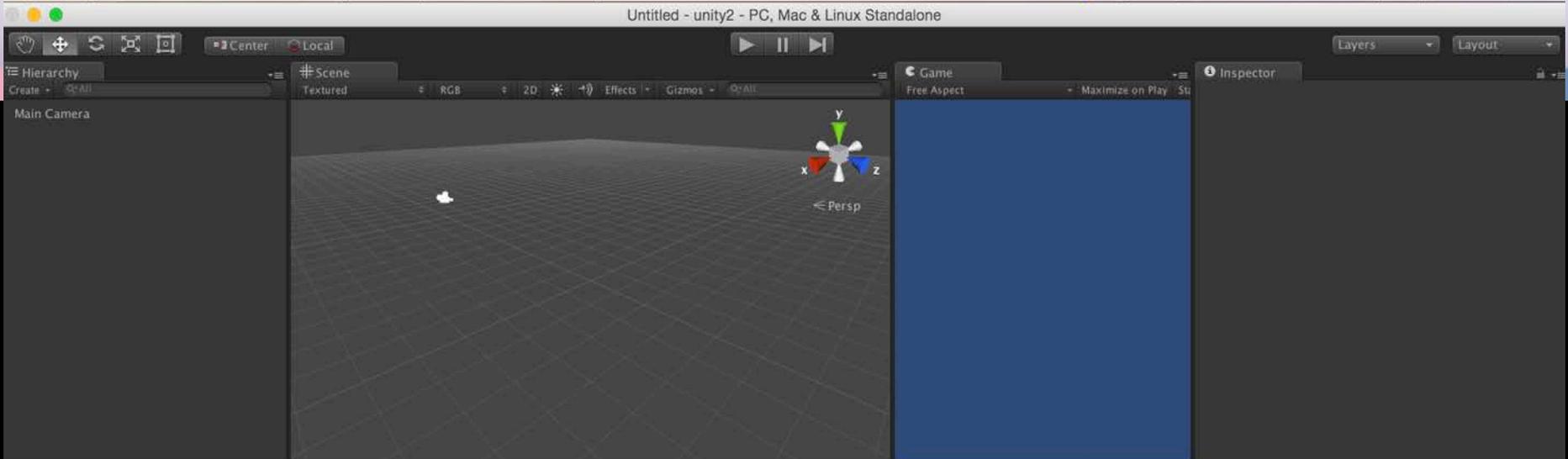
- Project menu >Create > JavaScript
- Main Menu > Assets > Create Javascript
- Project window >RMC > Create > JavaScript
- Inspector >Add script
- Name the script in the Project/Assets window

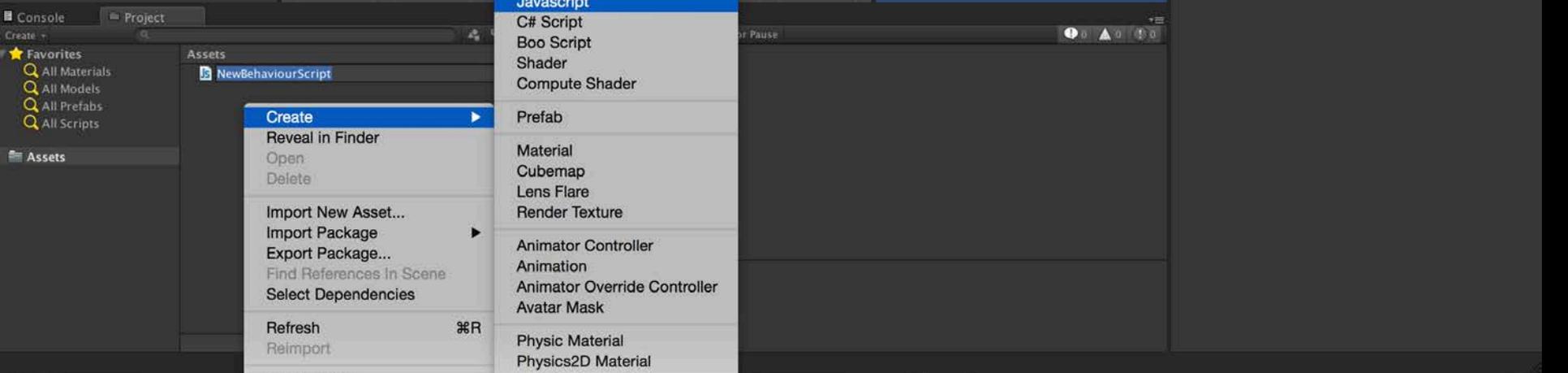
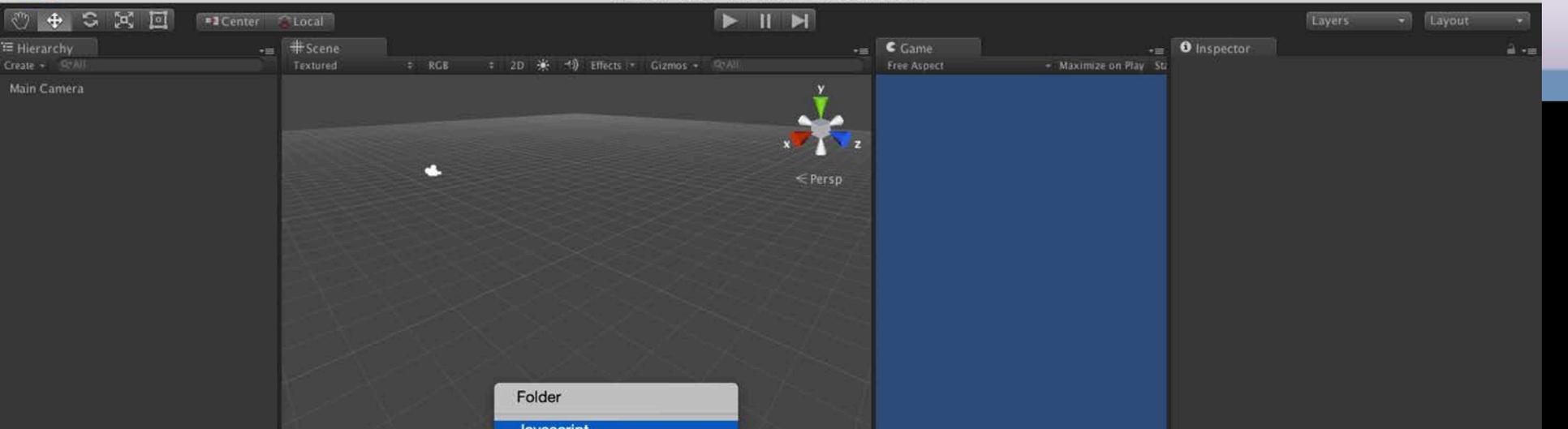
- Assign the script to an object (drag and drop)
- Run and test
- Fix compiler errors



Creating scripts in Unity







Click to add notes

If you would like to render the trigger zone invisible just uncheck the game object's

Creating scripts in Unity

The screenshot shows the MonoDevelop-Unity IDE interface. The title bar reads "MonoDevelop-Unity" and "Assembly-UnityScript - Variables.js - MonoDevelop-Unity". The menu bar includes "File", "Edit", "View", "Search", "Project", "Build", "Run", "Version Control", "Tools", "Window", and "Help". The status bar at the top right shows system icons for Wi-Fi, 55% battery, and the date "Sun 11:47 AM" with the name "Tsoupikova, Da".

The Solution Explorer on the left shows a project structure with folders like "Demo2010", "Assembly-CSharp-firstpass", "Assembly-UnityScript", "unity2", and "References". The "Variables.js" file is selected under "Assembly-UnityScript".

The main editor window displays the following JavaScript code in "Variables.js":

```
1 #pragma strict
2
3 function Start () {
4
5 }
6
7 function Update () {
8
9 }
```



Creating scripts in Unity

The screenshot shows the MonoDevelop-Unity IDE interface. The top menu bar includes File, Edit, View, Search, Project, Build, Run, Version Control, Tools, Window, and Help. The title bar indicates the current project is 'Assembly-UnityScript - Variables.js - MonoDevelop-Unity'. The Solution Explorer on the left shows a project structure with folders like 'Assembly-CSharp-firstpass', 'Assembly-UnityScript', and 'Assembly-UnityScript-firstpass', and a file named 'Variables.js'. The main editor window displays the following JavaScript code:

```
1 #pragma strict
2
3 var myInt : int = 5;
4
5
6 function Start ()
7 {
8     myInt = MultiplyByThree(myInt);
9     Debug.Log (myInt);
10 }
11
12
13 function MultiplyByThree (number : int) : int
14 {
15     var ret : int;
16     ret = number * 3;
17     return ret;
18 }
```

Center Local

Hierarchy

Main Camera
GameObject

Scene

Textured

RGB 2D Effects Gizmos

Game

Free Aspect Maximize on Play

Inspector

Layers Layout



All compiler errors have to be fixed before you can enter playmode!

Console

Project

Assets

Variables

Console

Clear Collapse Clear on Play Error Pause

10
UnityEngine.Debug.Log(Object)

Assets/Variables.js(18,13): BCE0005: Unknown identifier: 'MultiplyByThree'

Unity Editor interface showing a 3D scene, console, and project view.

Scene View: Displays a 3D grid with a white cube. A message reads: "All compiler errors have to be fixed before you can enter playmode!". A camera control overlay shows axes (x, y, z) and a "Persp" button.

Console: Shows two error messages:

- 10 UnityEngine.Debug.Log(Object)
- Assets/Variables.js(8,13): BCE0005: Unknown identifier: 'MultiplyByThree'.

Project View: Shows a folder named "Variables" under the "Assets" folder.

Inspector: Currently empty.

Hierarchy: Shows "Main Camera" and "GameObject".

Game View: Currently blank (blue).

Bottom Status Bar: Assets/Variables.js(8,13): BCE0005: Unknown identifier: 'MultiplyByThree'

Functions

Function is a collection of statements to perform a task
Functions are blocks of code which are written once and can then be reused as often as needed.
begin with an uppercase letter

```
function FuncName ()  
    {  
        statement1;  
        statement 2;  
    }
```



JavaScript Functions

Calling a function:

```
FuncName ();
```

```
myInt = MultiplyByThree(myInt);
```



Function Parameters

```
function MultiplyByThree (number : int) : int
{
    var ret : int;
    ret = number * 3;
    return ret;
}
```

Calling a function – `myInt = MultiplyByThree(myInt);`



Functions

Default functions

Start ()

executed only once before gameplay begins
helpful for initialization

Update()

executed every frame
for as long as the gameplay continues



Functions

```
var myInt : int = 5;
```

```
function Start ()
```

```
{  
    myInt = MultiplyByThree(myInt);  
    Debug.Log (myInt);  
}  
function MultiplyByThree (number : int) : int  
{  
    var ret : int;  
    ret = number * 3;  
    return ret;  
}
```



Arithmetic Operators

+	addition
-	subtraction
/	division
*	multiplication
++	increment
--	decrement
%	modulus



Functions

- 1) Create 3D object cube
- 2) create new Javascript “rotateCube”
- 3) Assign the script to the cube (drag and drop)

```
#pragma strict  
var speed = 5.0;
```

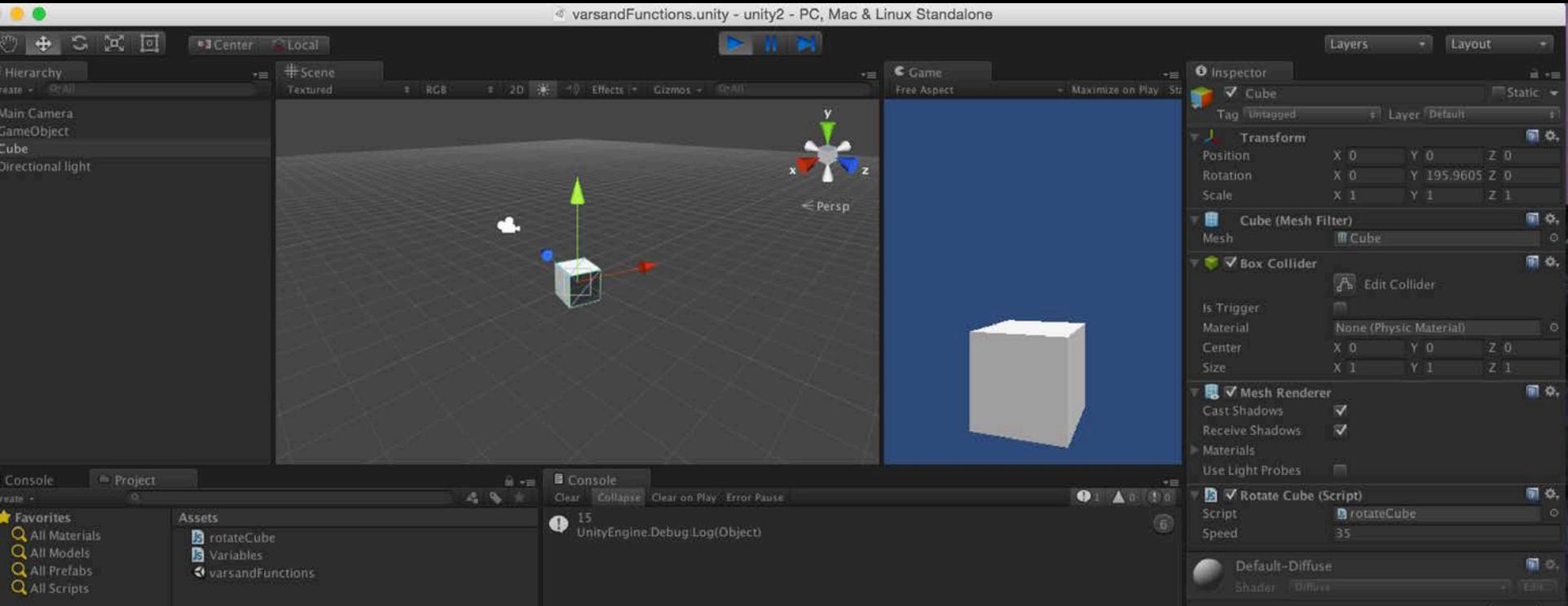
```
function Start () {  
}
```

```
function Update () {  
transform.Rotate(0, speed*Time.deltaTime, 0);  
}
```



Functions

- 4) Change the value of var speed in the Inspector window (35)
- 5) Play and test



Triggers and Collisions

Triggers are methods to detect collisions

Triggers are useful for triggering other events in your project

- teleportations

- automatic door openings

- displaying messages

- changing levels

- responsive events

- and many more



Triggers and Collisions

- 4) select the game object in the Hierarchy window
click on the little gear on the top right corner of the script property
select “remove component”
- 5) Create new script “triggerScript”

```
var target : Collider;  
function OnTriggerEnter(cubeTrigger : Collider)  
{  
if (cubeTrigger == target)  
{  
print("Collision");  
}  
}
```



Triggers and Collisions

- 6) Assign script to our cube
- 7) Check property “Is Trigger” in the Inspector
- 8) Create 3D plane
- 9) Import Character Controller Package
- 10) Drag FPC controller to the scene
- 11) Drag and drop the FPC from the Hierarchy window onto the variable Target in the Inspector



Triggers and Collisions

checks if the position of the FPC
intersects with the position of the trigger zone (the cube)
prints out “Collision”



Triggers and Collisions

To add a counter to collision

Checks how many times collision happened

```
var target : Collider;
```

```
private var counter : int = 0;
```

```
function OnTriggerEnter(cubeTrigger : Collider)  
{  
  if (cubeTrigger == target)  
  {  
    counter = counter + 1;  
    print("Collided: " + counter + " times!");  
  }  
}
```



Triggers and Collisions

The screenshot displays the Unity game engine interface. The top-left pane shows the Hierarchy panel with the following objects: First Person Controller, Main Camera, GameObject, Cube, Directional light, and Plane. The top-middle pane shows the Scene view in a perspective view, featuring a character, a cube, and a plane. The top-right pane shows the Game view in a free aspect view, displaying a simple grey cube on a white ground plane against a blue sky. The bottom-left pane shows the Console window with the following log messages:

```
15  
UnityEngine.Debug:Log(Object)  
There are 2 audio listeners in the scene. Please ensure there is always exactly one audio listener in the scene.  
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)
```

The bottom-right pane shows the Inspector panel for the selected Cube component. The Inspector is set to the Box Collider component, which is configured as follows:

- Tag: Untagged
- Layer: Default
- Transform: Position (X: 0, Y: 0.62, Z: 0), Rotation (X: 0, Y: 0, Z: 0), Scale (X: 1, Y: 1, Z: 1)
- Cube (Mesh Filter): Mesh: Cube
- Box Collider: Edit Collider, Is Trigger: (checked), Material: None (Physic Material), Center (X: 0, Y: 0, Z: 0), Size (X: 1, Y: 1, Z: 1)
- Mesh Renderer: Cast Shadows: (checked), Receive Shadows: (checked), Materials: Use Light Probes: (unchecked)
- Trigger Script (Script): Script: triggerScript, Target: First Person Controller (Character)
- Default-Diffuse Shader: Main Color: [White], Base (RGB): Y: 0, Z: 0

An "Add Component" button is visible at the bottom of the Inspector panel.

Triggers and Collisions

to create an invisible trigger zone

Select the object >

Inspector > remove Mesh Renderer Component

The object will be invisible but still allow collision detection



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

13) Import new Asset (sound effect/s)

14) Add Audio Source to the Cube (Inspector>Add Component >Audio Source)

15) Uncheck button “Play On Awake”

16) Drag sound effect to the Inspector > Trigger Script >My sound

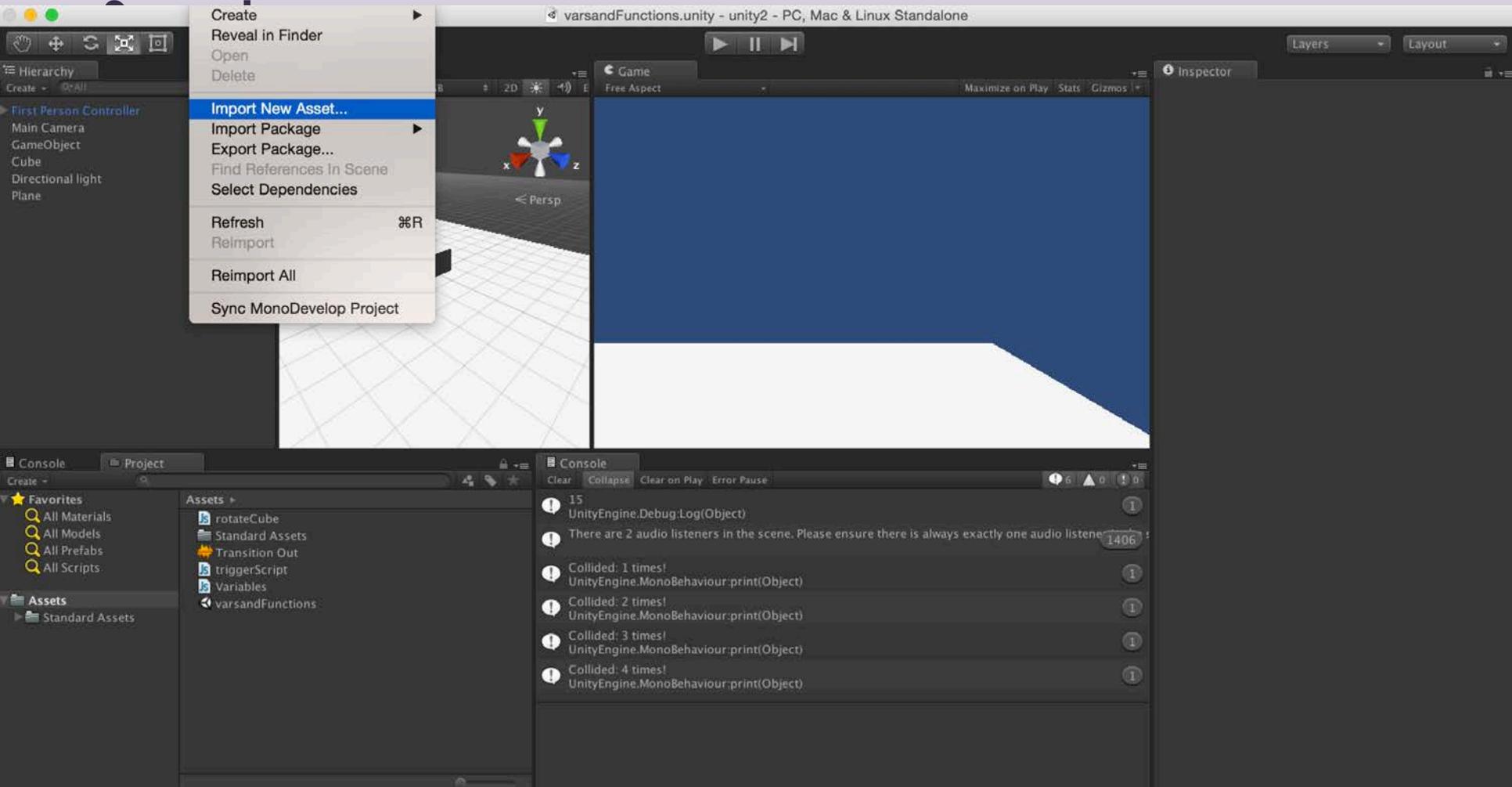


Sounds

```
var target : Collider;
private var counter : int = 0;
var mySound : AudioClip;

function OnTriggerEnter(cubeTrigger : Collider)
{
    if (cubeTrigger == target)
    {
        GetComponent.<AudioSource>().PlayOneShot(mySound);
        counter = counter + 1;
        print("Collided: " + counter + " times!");
    }
}
```





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

Layers Layout

Inspector

Game

Free Aspect

Maximize on Play Stats Gizmos

Console Project

Favorites

- All Materials
- All Models
- All Prefabs
- All Scripts

Assets

Standard Assets

Assets

- rotateCube
- Standard Assets
- Transition Out
- triggerScript
- Variables
- varsandFunctions

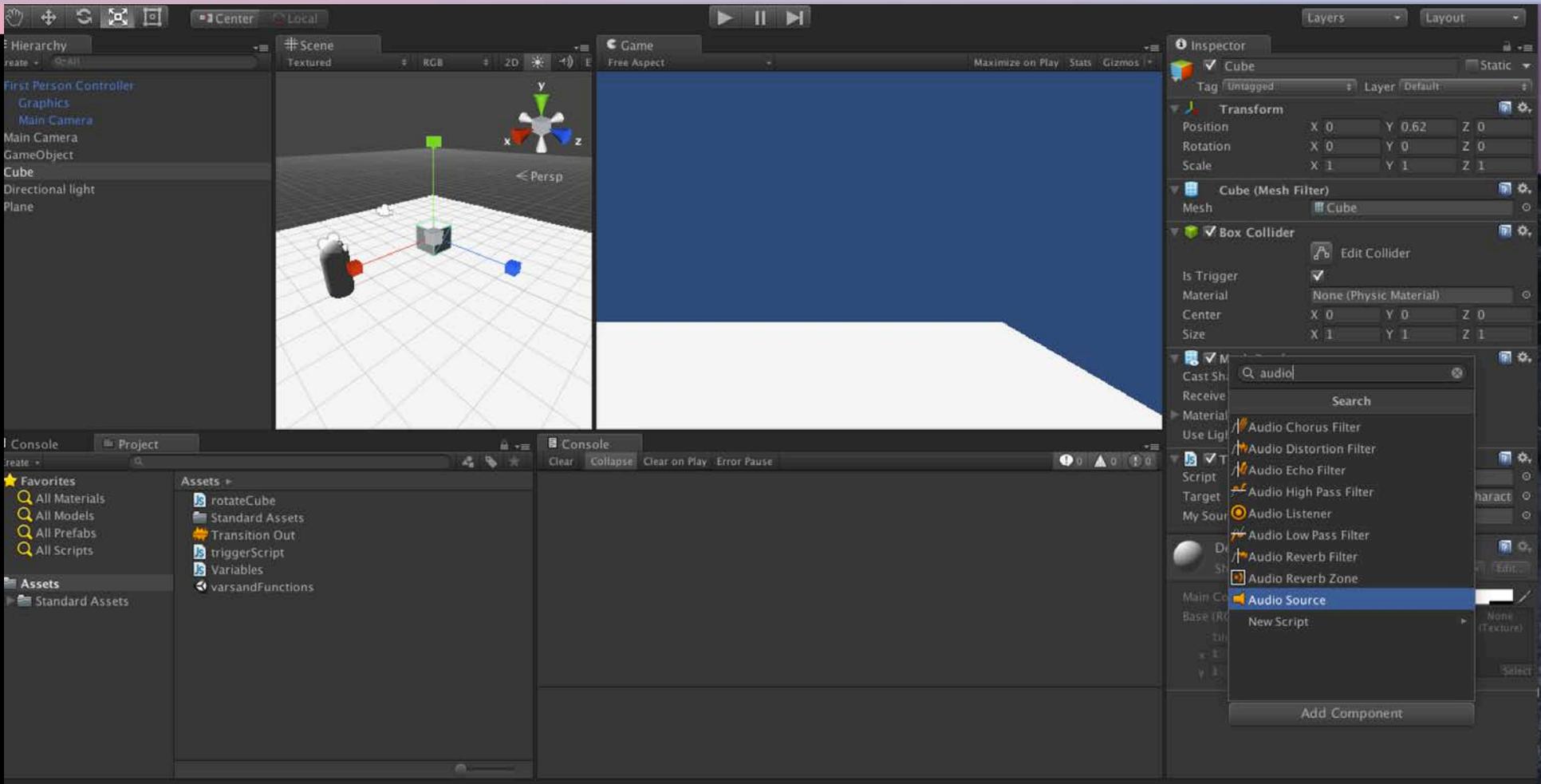
Console

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 listener.
- 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



The screenshot displays the Unity 2017.4.0f1 development environment. The interface is divided into several panels:

- Top Left:** Hierarchy panel showing the scene's object structure, including "First Person Controller", "Graphics", "Main Camera", "Main Camera", "GameObject", "Cube", "Directional light", and "Plane".
- Top Center:** Scene view showing a 3D perspective view of a scene with a grid floor, a blue sky, and various objects like a black capsule, a red cube, a blue cube, and a central cube with a green line extending upwards. A coordinate system (x, y, z) and "Persp" view indicator are visible.
- Top Right:** Game view showing a solid blue background, representing the game's visual output.
- Bottom Left:** Project panel showing the "Assets" folder containing "rotateCube", "Standard Assets", "Transition Out", "triggerScript", "Variables", and "varsandFunctions".
- Bottom Center:** Console panel with "Clear", "Collapse", "Clear on Play", and "Error Pause" buttons.
- Bottom Right:** Inspector panel for the selected "Cube" object, showing properties such as:
 - Transform:** Position (X: 0, Y: 0.62, Z: 0), Rotation (X: 0, Y: 0, Z: 0), Scale (X: 1, Y: 1, Z: 1).
 - Cube (Mesh Filter):** Mesh: Cube.
 - Box Collider:** Edit Collider, Is Trigger (checked), Material: None (Physic Material), Center (X: 0, Y: 0, Z: 0), Size (X: 1, Y: 1, Z: 1).
 - Mesh Renderer:** Cast Shadows (checked), Receive Shadows (checked).
 - Trigger Script (Script):** Script: triggerScript, Target: First Person Controller (Character), My Sound: Transition Out.
 - Audio Source:** Audio Clip: None (Audio Clip), Mute (unchecked), Bypass Effects (unchecked), Bypass Listener Effect (unchecked), Bypass Reverb Zones (unchecked), Play On Awake (checked), Loop (unchecked), Priority: 128, Volume: 1, Pitch: 1.



Drag "transitionOne" to My Sound var in inspector

The screenshot displays the Unity game engine interface. The top-left pane shows the Hierarchy panel with a tree view containing: First Person Controller, Graphics, Main Camera, Main Camera, GameObject, Cube, Directional light, and Plane. The top-center pane shows the Scene view with a 3D perspective view of a scene containing a cube, a capsule, and a blue cube. The top-right pane shows the Game view with a blue background. The bottom-left pane shows the Project panel with a tree view containing: Favorites (All Materials, All Models, All Prefabs, All Scripts) and Assets (rotateCube, Standard Assets, Transition Out, triggerScript, Variables, varsandFunctions). The bottom-center pane shows the Console with several error messages: 15 UnityEngine.Debug:Log(Object), There are 2 audio listeners in the scene. Please ensure there is always exactly one audio listener in the scene. (1271), Collided: 1 times! UnityEngine.MonoBehaviour:print(Object), Collided: 2 times! UnityEngine.MonoBehaviour:print(Object), and Collided: 3 times! UnityEngine.MonoBehaviour:print(Object). The bottom-right pane shows the Inspector panel for a Cube object. The Inspector panel has the following sections: Transform (Position: X 0, Y 0.62, Z 0; Rotation: X 0, Y 0, Z 0; Scale: X 1, Y 1, Z 1), Cube (Mesh Filter) (Mesh: Cube), Box Collider (Is Trigger: checked; Material: None (Physic Material); Center: X 0, Y 0, Z 0; Size: X 1, Y 1, Z 1), Mesh Renderer (Cast Shadows: checked; Receive Shadows: checked; Materials: Use Light Probes: unchecked), and Trigger Script (Script: triggerScript; Target: First Person Controller (Character); My Sound: Transition Out), Audio Source (Audio Clip: None (Audio Clip); Mute: unchecked; Bypass Effects: unchecked; Bypass Listener Effect: unchecked; Bypass Reverb Zones: unchecked; Play On Awake: unchecked; Loop: unchecked; Priority: 128; Volume: 1).

Colors

17) Create new material and add LegacyShaders>transparent>diffuse shader

18) Add material to the cube

19) Modify the script to add new material:

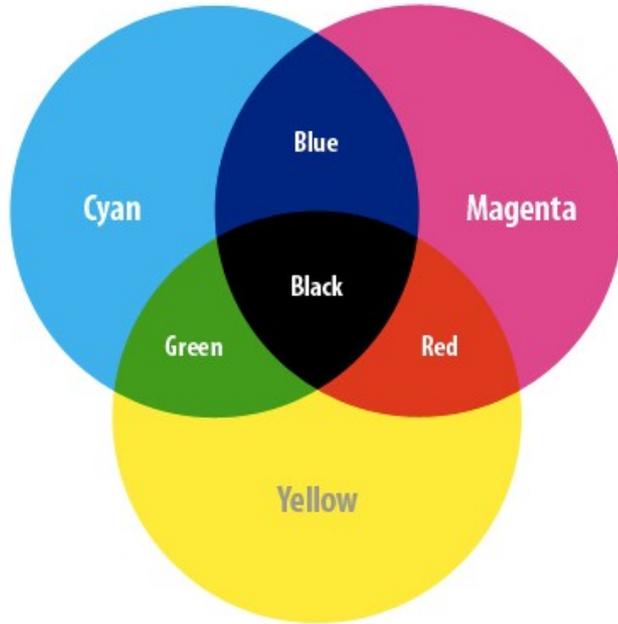
```
private var orange : Color = Color(0.8, 0.4, 0.0, 0.3);
```

```
renderer.material.color = orange;
```



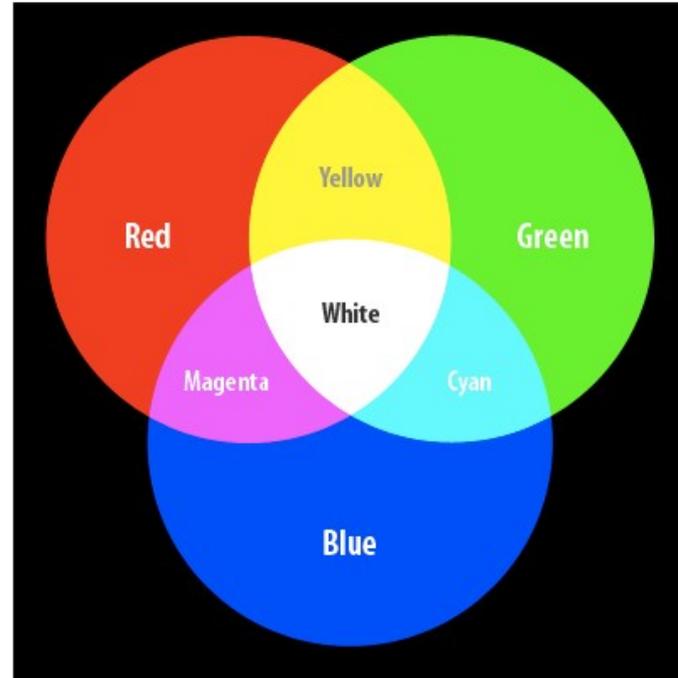
Color Systems

CMYK – The Subtractive System



CMYK Color Model

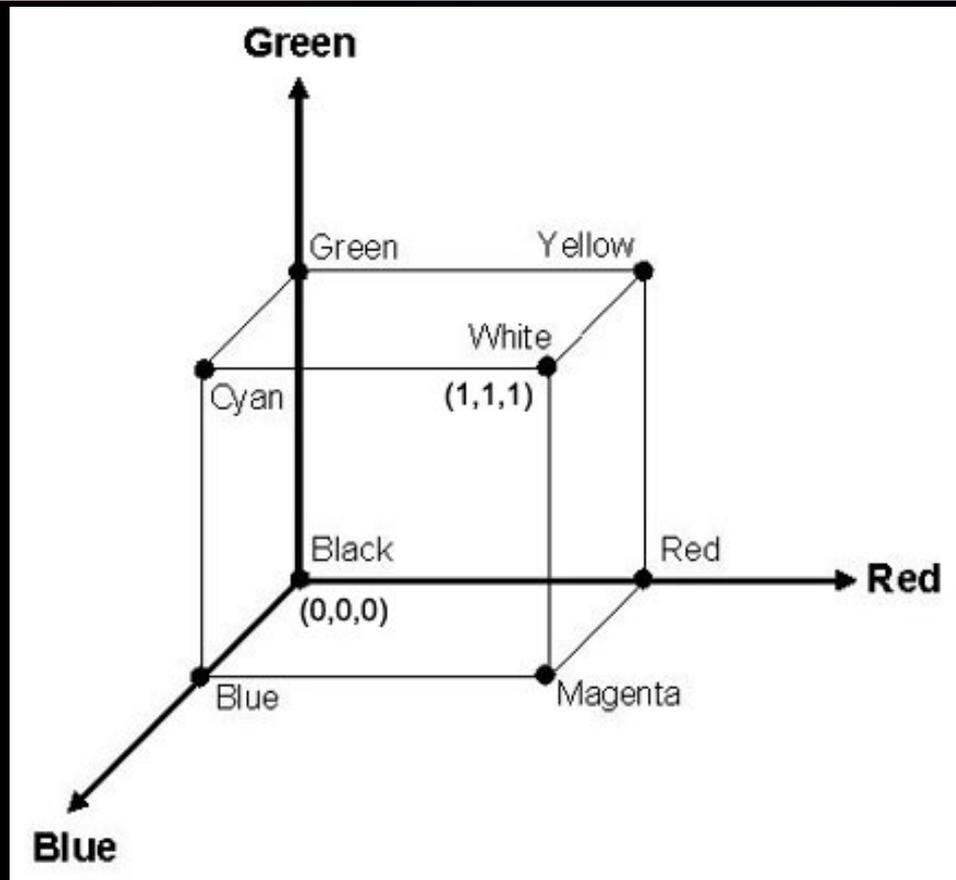
RGB – The Additive System



RGB Color Model



Color Cube



RGBA Color

Representation of RGBA colors

Values for red, green, blue and alpha are floating point values with a range from 0 to 1

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?



Center Local

Hierarchy

- create + Q: All
- First Person Controller
- Graphics
- Main Camera
- Main Camera
- GameObject
- Cube
- Directional light
- Plane

Scene

Textured RGB 2D

Game

Free Aspect Maximize on Play Stats Gizmos

Inspector

cubeMaterial

Shader Transparent/Diffuse

Main Color

Base (RGB) Trans (A)

Tiling	Offset
x 1	0
y 1	0

None (Texture) Select

Console

Project

Assets

- cubeMaterial
- rotateCube
- Standard Assets
- Transition Out
- triggerScript
- Variables
- varsandFunctions

Console

Clear Collapse Clear on Play Error Pause

15

UnityEngine.Debug:Log(Object) 4

There are 2 audio listeners in the scene. Please ensure there is always exactly one audio listener. 3849

Collided: 1 times! 4

UnityEngine.MonoBehaviour:print(Object) 1

Collided: 2 times! 1

UnityEngine.MonoBehaviour:print(Object) 1

Collided: 3 times! 1

UnityEngine.MonoBehaviour:print(Object) 1

Collided: 4 times! 1

UnityEngine.MonoBehaviour:print(Object) 1

cubeMaterial

Final Script

```
var target : Collider;
private var counter : int = 0;
var mySound : AudioClip;
private var orange : Color = Color(0.8, 0.4, 0.0, 0.3);
function OnTriggerEnter(cubeTrigger : Collider)
{
if (cubeTrigger == target)
{
GetComponent.<AudioSource>().PlayOneShot(mySound);
counter = counter + 1;
print("Collided: " + counter + " times!");
GetComponent.<Renderer>().material.color = orange;
}
}
```

