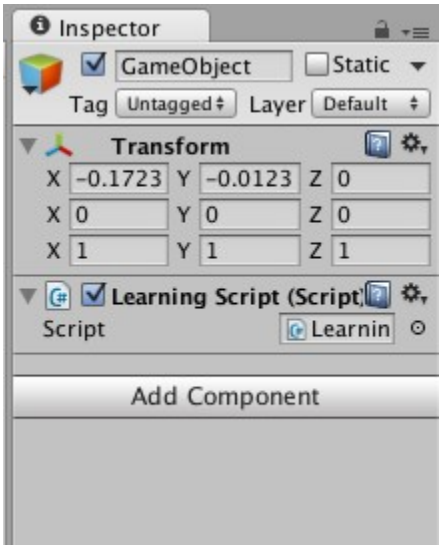
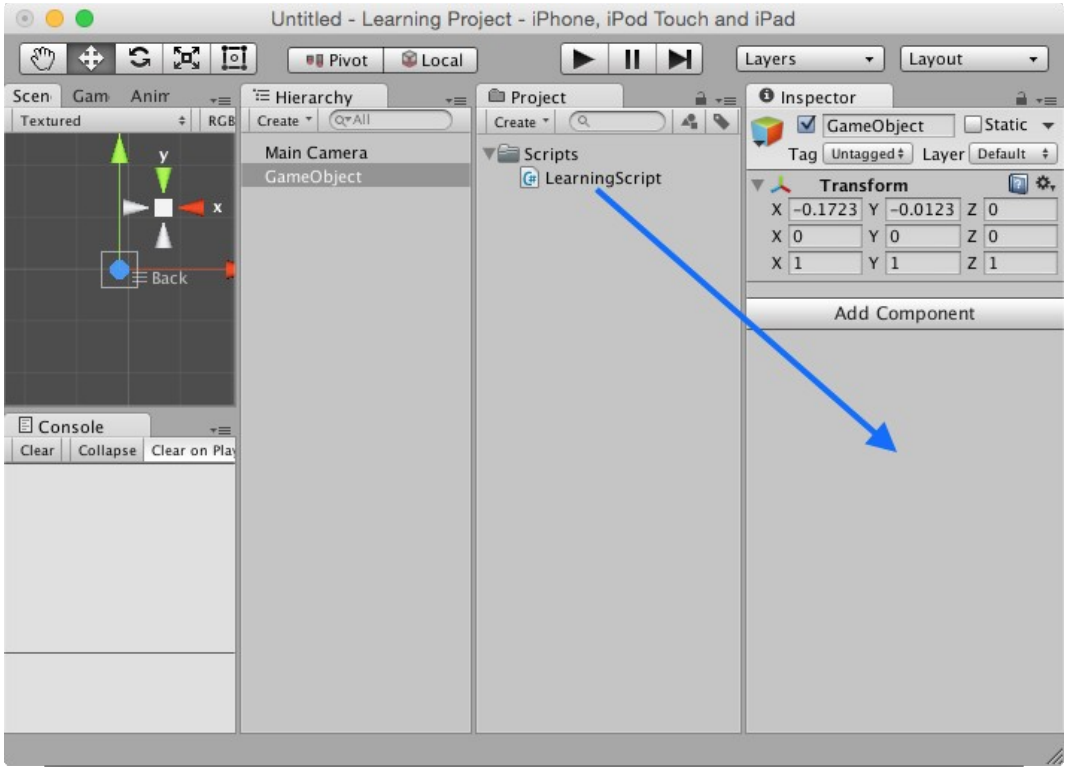
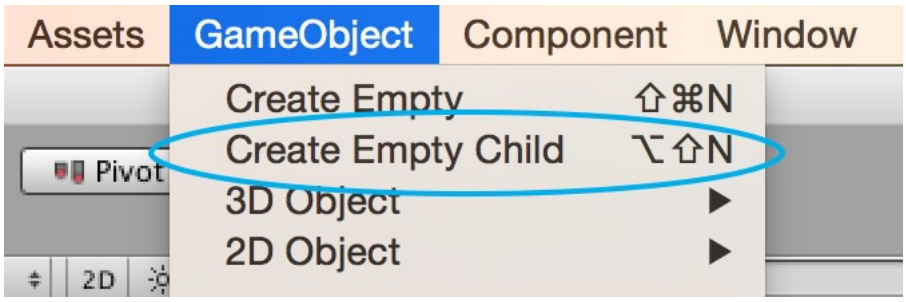
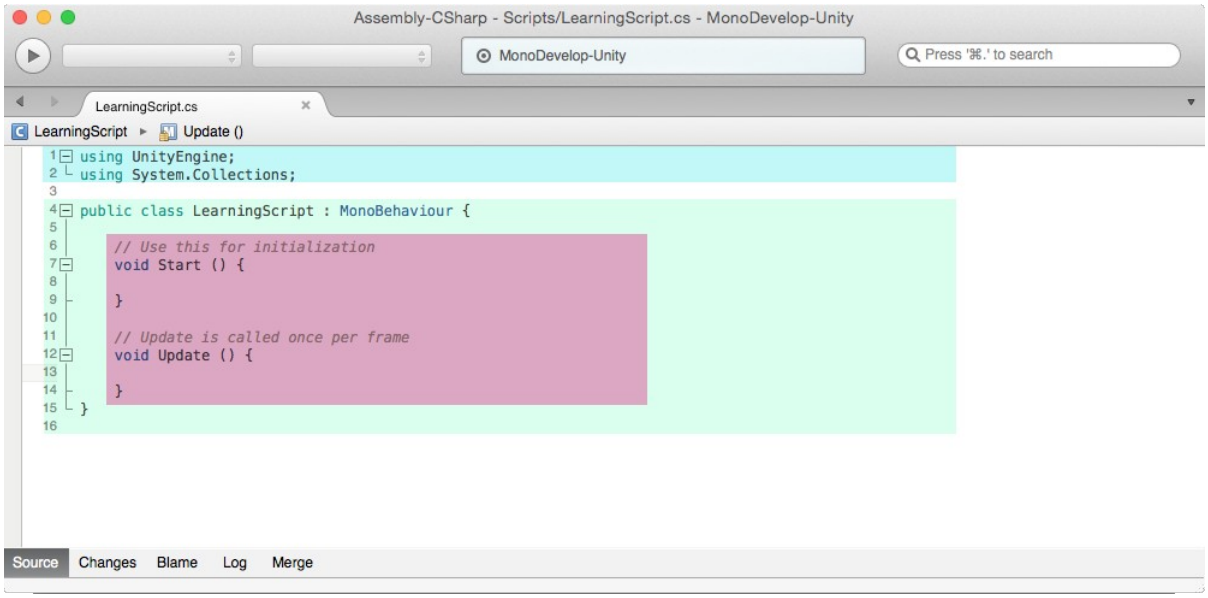
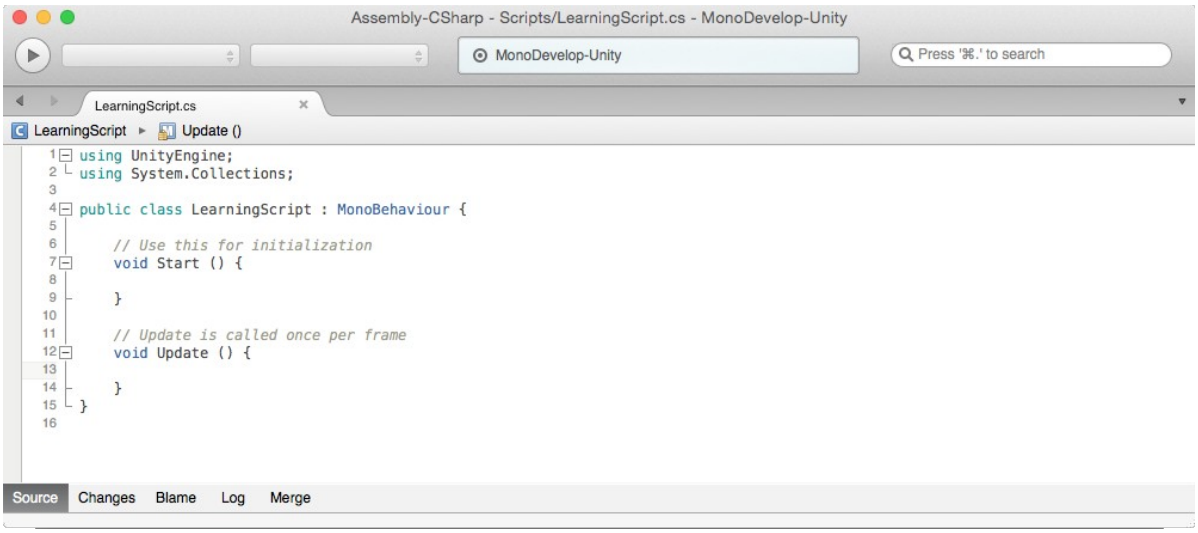
Chapter 1: Discovering Your Hidden Scripting Skills and Getting Your Environment Ready

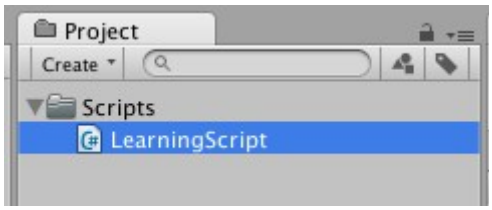


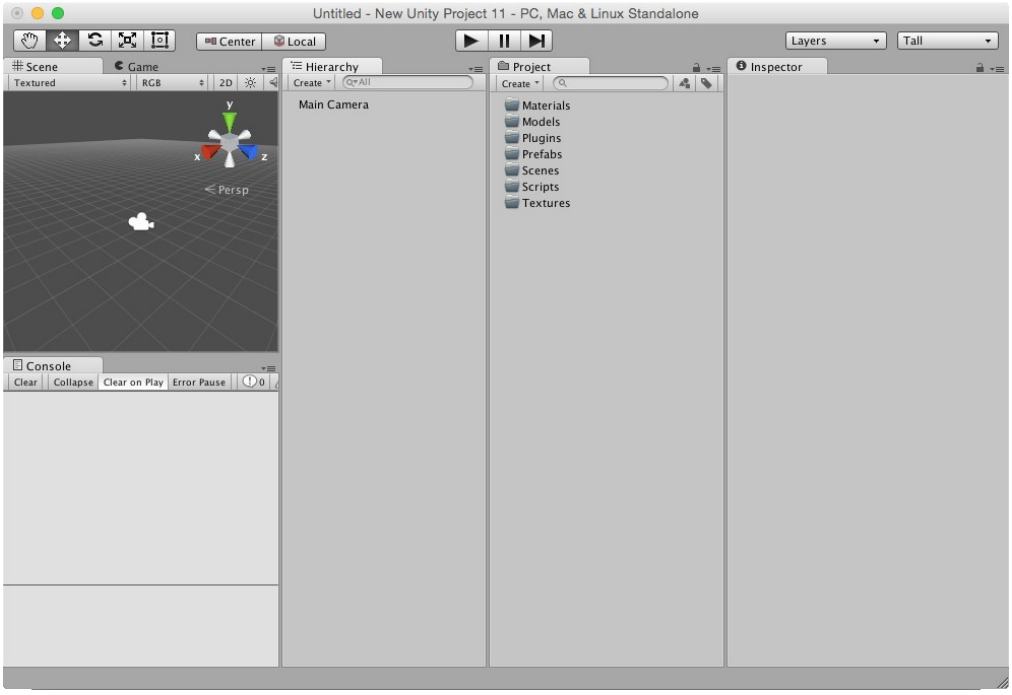


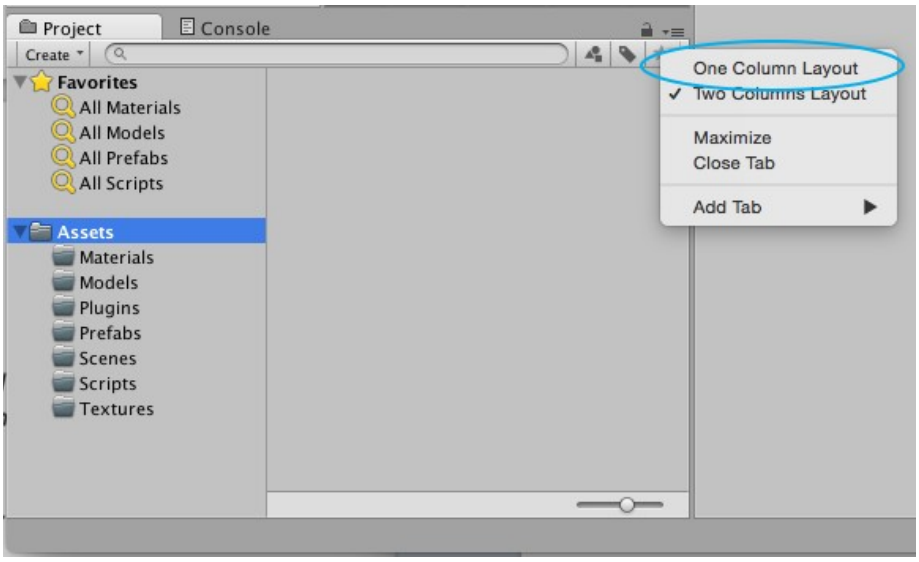

Assets
GameObject
Component
Window
Create Empty
Create Empty Child
3D Object
2D Object
Pivot
2D

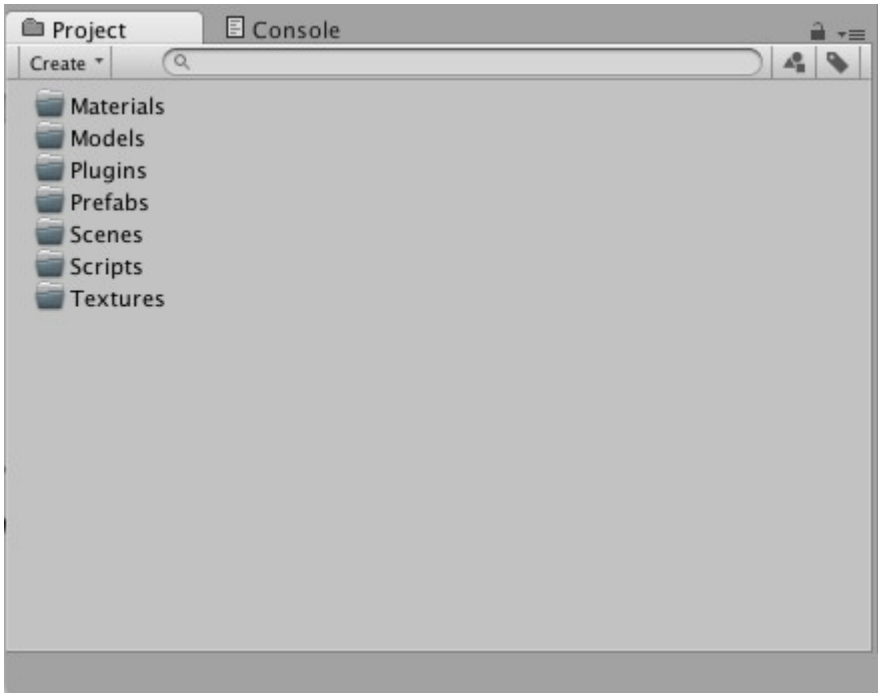

Assembly-CSharp - Scripts/LearningScript.cs - MonoDevelop-Unity
MonoDevelop-Unity
Press '⌘.' to search
LearningScript.cs
LearningScript
Update ()
using UnityEngine;
using System.Collections;
public class LearningScript : MonoBehaviour {
// Use this for initialization
void Start () {
}
// Update is called once per frame
void Update () {
}
}
Source
Changes
Blame
Log
Merge

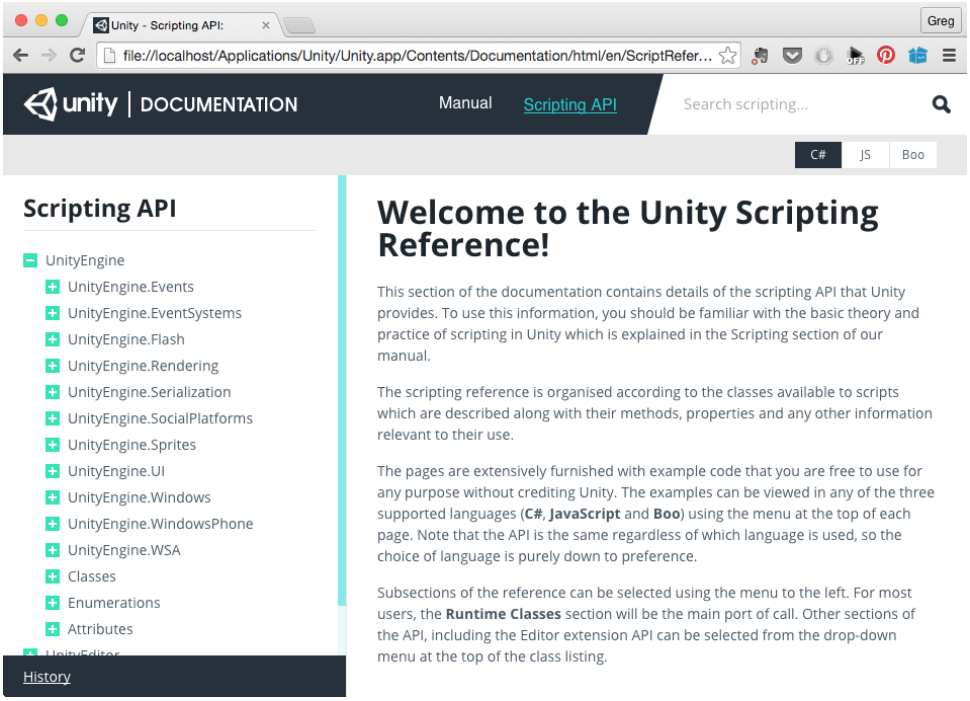

Assembly-CSharp - Scripts/LearningScript.cs - MonoDevelop-Unity
MonoDevelop-Unity
Press '⌘.' to search
LearningScript.cs
LearningScript
Update ()
using UnityEngine;
using System.Collections;
public class LearningScript : MonoBehaviour {
// Use this for initialization
void Start () {
}
// Update is called once per frame
void Update () {
}
}
Source
Changes
Blame
Log
Merge

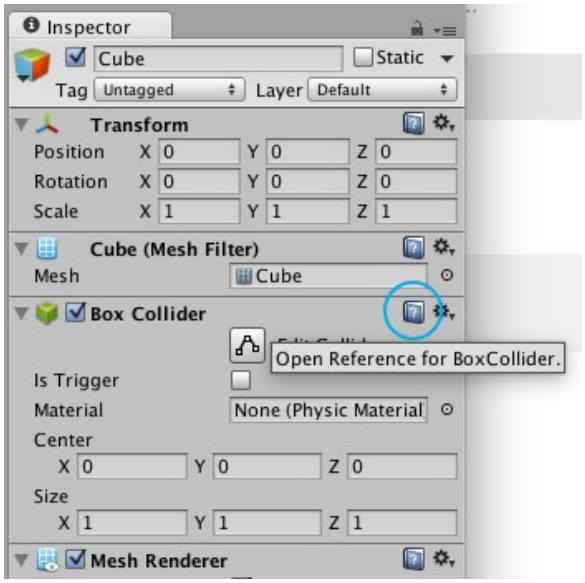

Project
Create
Scripts
LearningScript

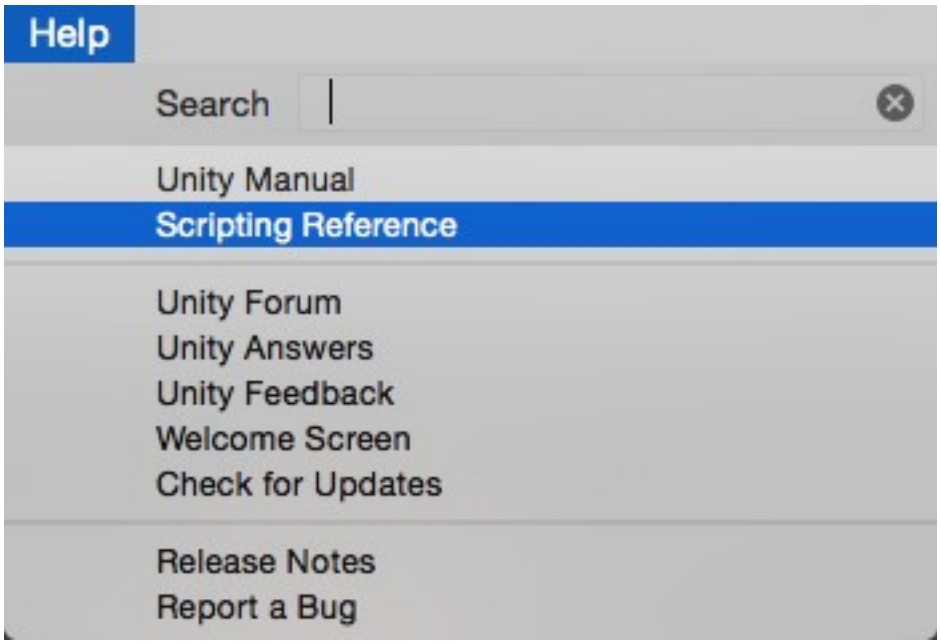

Untitled - New Unity Project 11 - PC, Mac & Linux Standalone
Center
Local
Layers
Tall
Scene
Game
Hierarchy
Project
Inspector
Textured
RGB
2D
Persp
Main Camera
Materials
Models
Plugins
Prefabs
Scenes
Scripts
Textures
Console
Clear
Collapse
Clear on Play
Error Pause

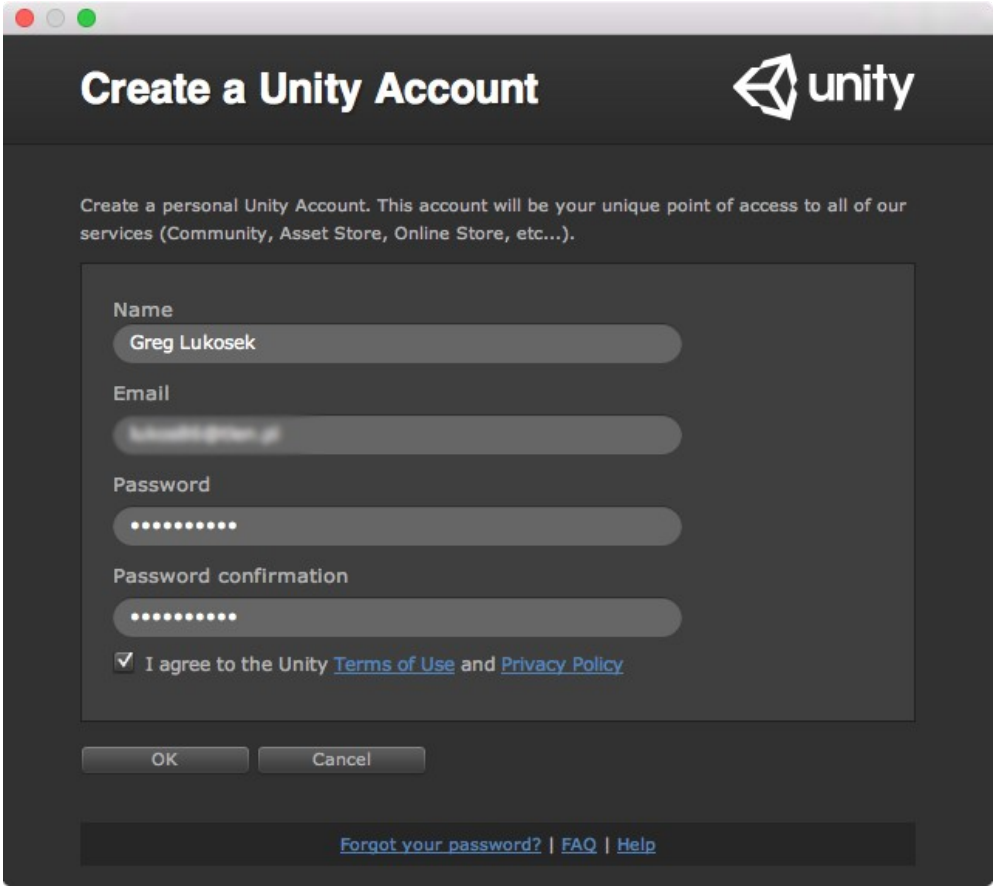

Project
Console
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
Materials
Models
Plugins
Prefabs
Scenes
Scripts
Textures
One Column Layout
Two Columns Layout
Maximize
Close Tab
Add Tab

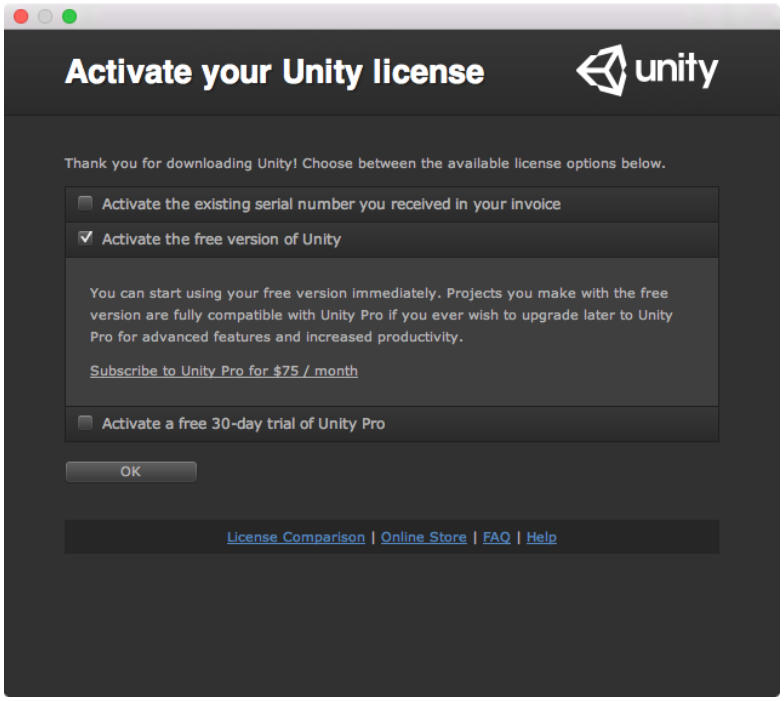

Project
Console
Create
Materials
Models
Plugins
Prefabs
Scenes
Scripts
Textures


Unity - Scripting API:
Greg
file://localhost/Applications/Unity/Unity.app/Contents/Documentation/html/en/ScriptRefer...
unity | DOCUMENTATION
Manual
Scripting API
Search scripting...
C#
JS
Boo
Scripting API
UnityEngine
UnityEngine.Events
UnityEngine.EventSystems
UnityEngine.Flash
UnityEngine.Rendering
UnityEngine.Serialization
UnityEngine.SocialPlatforms
UnityEngine.Sprites
UnityEngine.UI
UnityEngine.Windows
UnityEngine.WindowsPhone
UnityEngine.WSA
Classes
Enumerations
Attributes
History
Welcome to the Unity Scripting Reference!
This section of the documentation contains details of the scripting API that Unity provides. To use this information, you should be familiar with the basic theory and practice of scripting in Unity which is explained in the Scripting section of our manual.
The scripting reference is organised according to the classes available to scripts which are described along with their methods, properties and any other information relevant to their use.
The pages are extensively furnished with example code that you are free to use for any purpose without crediting Unity. The examples can be viewed in any of the three supported languages (C#, JavaScript and Boo) using the menu at the top of each page. Note that the API is the same regardless of which language is used, so the choice of language is purely down to preference.
Subsections of the reference can be selected using the menu to the left. For most users, the Runtime Classes section will be the main port of call. Other sections of the API, including the Editor extension API can be selected from the drop-down menu at the top of the class listing.


Inspector
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
Box Collider
Open Reference for BoxCollider.
Is Trigger
Material None (Physic Material
Center
X 0 Y 0 Z 0
Size
X 1 Y 1 Z 1
Mesh Renderer


Help
Search
Unity Manual
Scripting Reference
Unity Forum
Unity Answers
Unity Feedback
Welcome Screen
Check for Updates
Release Notes
Report a Bug


Create a Unity Account
unity
Create a personal Unity Account. This account will be your unique point of access to all of our services (Community, Asset Store, Online Store, etc...).
Name
Greg Lukosek
Email
Password
Password confirmation
I agree to the Unity Terms of Use and Privacy Policy
OK
Cancel
Forgot your password? | FAQ | Help


Activate your Unity license
unity
Thank you for downloading Unity! Choose between the available license options below.
Activate the existing serial number you received in your invoice
Activate the free version of Unity
You can start using your free version immediately. Projects you make with the free version are fully compatible with Unity Pro if you ever wish to upgrade later to Unity Pro for advanced features and increased productivity.
Subscribe to Unity Pro for $75 / month
Activate a free 30-day trial of Unity Pro
OK
License Comparison | Online Store | FAQ | Help

Chapter 2: Introducing the Building Blocks for Unity Scripts

Input.GetKeyUp

public static bool **GetKeyUp**(string **name**);

Parameters

Description

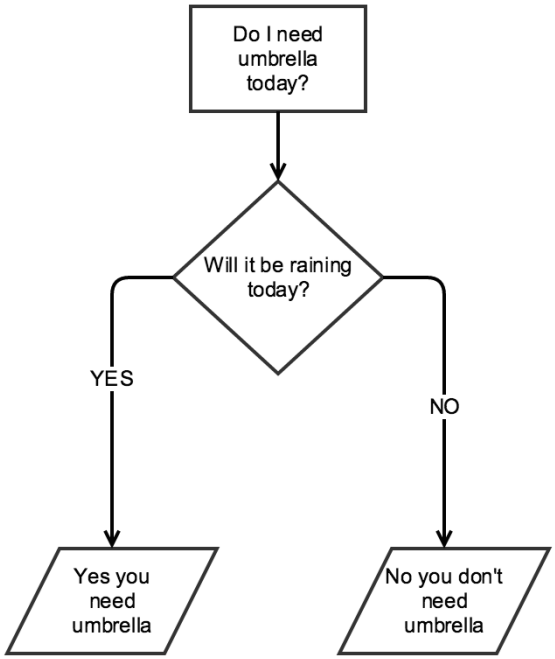
Returns true during the frame the user releases the key identified by `name`.

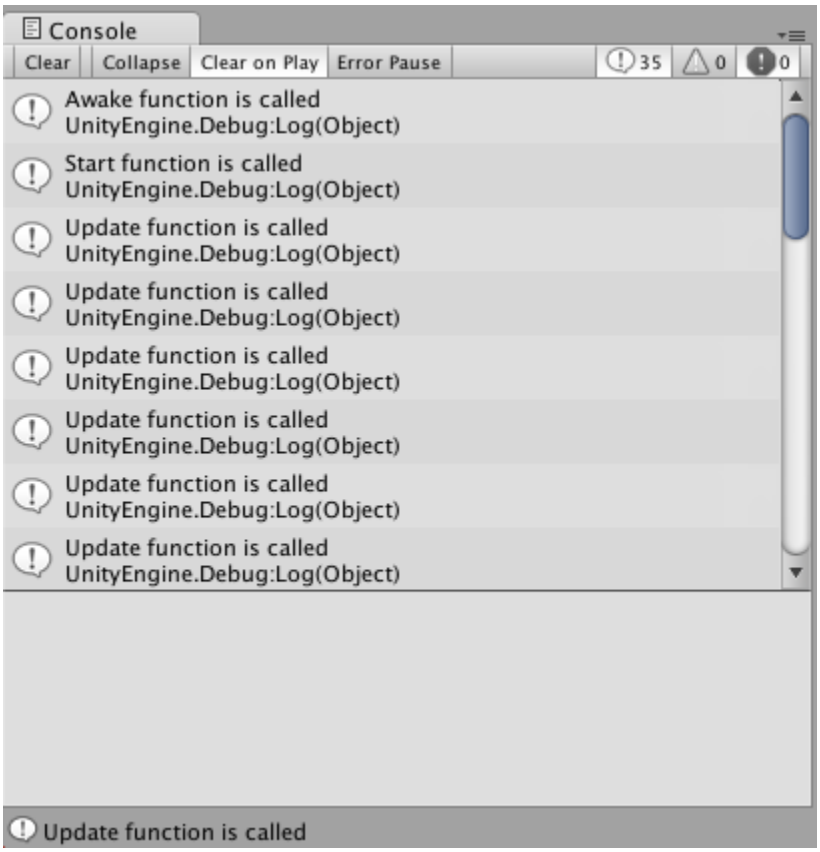
```
7   void Start () {
8
9       int speedLimit = 60;
10
11      if (speedLimit == 70) {
12          Debug.Log("I can drive at maximum speed");
13      }
14      else if (speedLimit < 70 && speedLimit >= 30) {
15          Debug.Log("Speed limit is less than 70 and more or equals to 30");
16      }
17      else if (speedLimit < 30) {
18          Debug.Log("I better be driving slowly, 30 mph or less");
19      }
20  }
21
22
```

```
16
17      public bool imHugry = false;
18      public bool areKidsHungry = true;
19
20      void Start () {
21
22          if (imHugry || areKidsHungry) {
23              Debug.Log("I should cook some food");
24          }
25      }
26
```

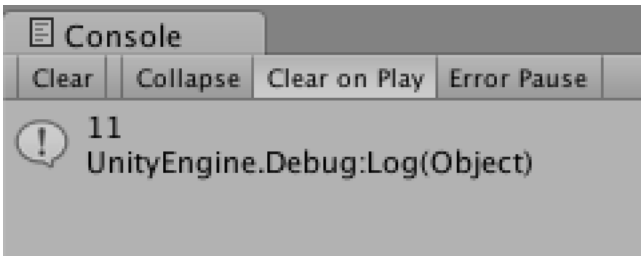
```
6       public bool imLateForMeeting = true;
7       public bool roadConditionsArePerfect = true;
8
9       void Start () {
10
11          if (imLateForMeeting && roadConditionsArePerfect) {
12              Debug.Log("I need to drive fast");
13          }
14      }
```

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class LearningStatements : MonoBehaviour {
5
6      public bool willItBeRainingToday = true;
7
8      void Start () {
9
10         if (willItBeRainingToday) {
11             Debug.Log("Yes you need umbrella");
12         } else {
13             Debug.Log("No, you dont need umbrella");
14         }
15     }
16 }
17
```


Do I need umbrella today?
Will it be raining today?
YES
NO
Yes you need umbrella
No you don't need umbrella

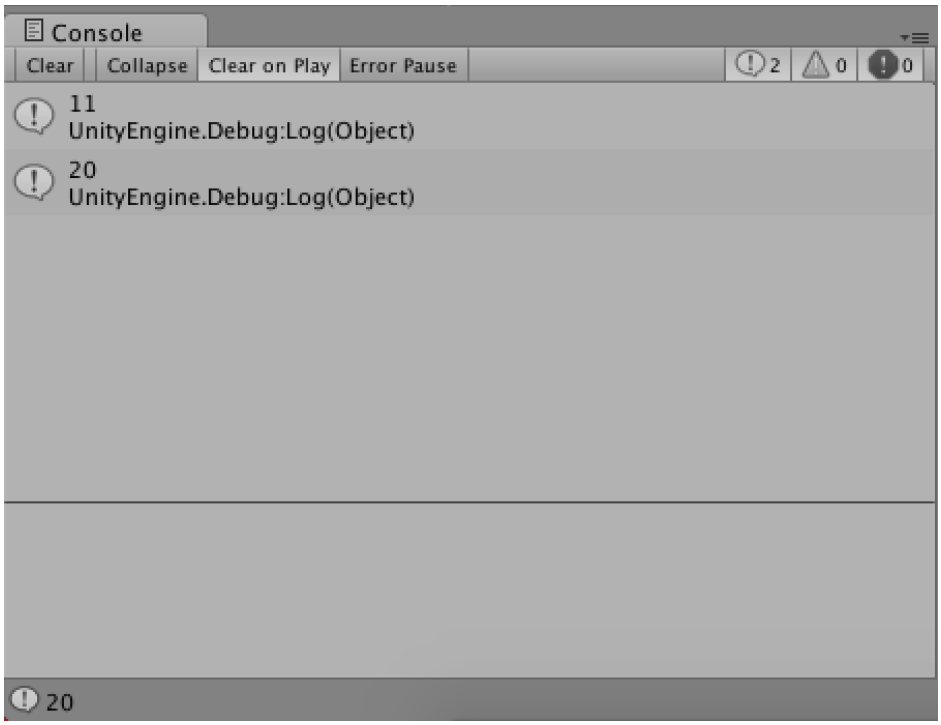

Console
Clear
Collapse
Clear on Play
Error Pause
35
0
0
Awake function is called
UnityEngine.Debug:Log(Object)
Start function is called
UnityEngine.Debug:Log(Object)
Update function is called
UnityEngine.Debug:Log(Object)
Update function is called
UnityEngine.Debug:Log(Object)
Update function is called
UnityEngine.Debug:Log(Object)
Update function is called
UnityEngine.Debug:Log(Object)
Update function is called
UnityEngine.Debug:Log(Object)
Update function is called
UnityEngine.Debug:Log(Object)
Update function is called

```
1   using UnityEngine;
2   using System.Collections;
3
4   public class LearningMethods : MonoBehaviour {
5
6
7       void Awake() {
8           Debug.Log("Awake function is called");
9       }
10
11      // Use this for initialization
12      void Start () {
13          Debug.Log("Start function is called");
14      }
15
16      // Update is called once per frame
17      void Update () {
18          Debug.Log("Update function is called");
19      }
20  }
21
22
23
```



```
1   using UnityEngine;
2   using System.Collections;
3
4   public class LearningScript : MonoBehaviour {
5
6       public int number1 = 2;
7       public int number2 = 9;
8
9       // Use this for initialization
10      void Start () {
11
12      }
13
14      // Update is called once per frame
15      void Update () {
16          if (Input.GetKeyUp(KeyCode.Return)) AddTwoNumbers();
17      }
18
19
20      void AddTwoNumbers() {
21
22          Debug.Log( number1 + number2);
23      }
24
25  }
26
```

```
11          Debug.Log(2 + 9);
12
13          Debug.Log(11 + myNumber);
```




```
using UnityEngine;
using System.Collections;

public class LearningScript : MonoBehaviour {

    public int myNumber = 9;

    // Use this for initialization
    void Start () {

        Debug.Log(2 + 9);

        Debug.Log(11 + myNumber);

    }

    // Update is called once per frame
    void Update () {

    }
}
```

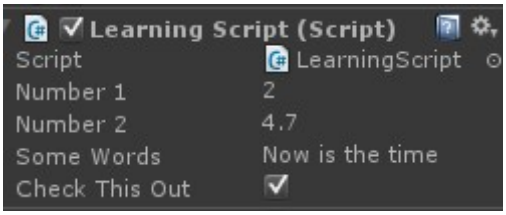
Chapter 3: Getting into the Details of Variables

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class LearningScript : MonoBehaviour
5  {
6      string block1 = "Block 1 text";
7
8      void Start ()
9      {
10         Debug.Log(block1);
11         string block2 = "Block 2 text";
12         Debug.Log(block2);
13         {
14             Debug.Log(block1);
15             Debug.Log(block2);
16             string block3 = "Block 3 text";
17             Debug.Log(block3);
18         }
19     }
20 }
21
```

Code Block 1

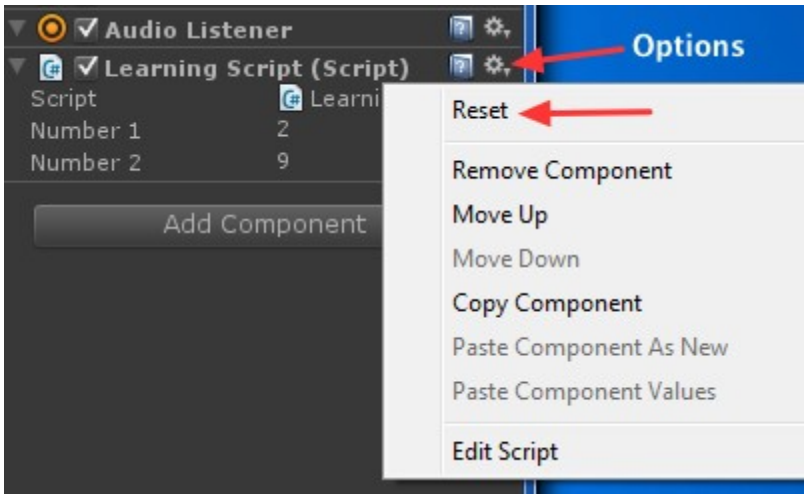
Code Block 2

Code Block 3



```
1  using UnityEngine;
2  using System.Collections;
3
4  public class LearningScript : MonoBehaviour
5  {
6      public int number1 = 2;
7      public float number2 = 4.7f;
8      public string someWords = "Now is the time";
9      public bool checkThisOut = true;
10
11     void Start ()
12     {
13
14     }
15
16     void Update ()
17     {
18
19     }
```

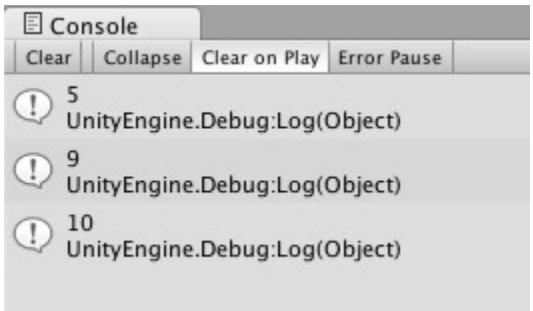
Type	Contents of the variable
int	A simple integer, such as the number 3
float	A number with a decimal, such as the number 3.14
string	Characters in double quotes, such as, “Watch me go now”
bool	A boolean, either **true** or **false**



Chapter 4: Getting into the Details of Methods

```
1   using UnityEngine;
2   using System.Collections;
3
4   public class LearningReusableMethodsWithReturn : MonoBehaviour {
5
6
7       public int number1 = 2;
8       public int number2 = 3;
9
10
11
12      void Start () {
13
14          int sumResult = AddTwoNumbers(number1, number2);
15
16          DisplayResult(sumResult);
17
18      }
19
20
21
22      int AddTwoNumbers (int firstNumber, int secondNumber) {
23
24          int result = firstNumber + secondNumber;
25          return result;
26
27      }
28
29
30      void DisplayResult (int total) {
31
32          Debug.Log("The grand total is: " + total);
33
34      }
35
36  }
37
38
39
40
```

```
int AddTwoNumbers (int firstNumber, int secondNumber) {

    int result = firstNumber + secondNumber;
    return result;

}
```



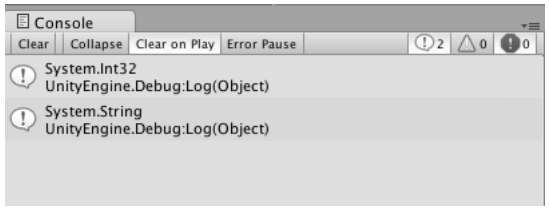

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class LearningReusableMethods : MonoBehaviour {
5
6
7      public int number1 = 2;
8      public int number2 = 3;
9      public int number3 = 7;
10
11
12
13     void Start () {
14
15         AddAndPrintTwoNumbers(number1, number2);
16         AddAndPrintTwoNumbers(number1, number3);
17         AddAndPrintTwoNumbers(number2, number3);
18
19     }
20
21
22
23     void AddAndPrintTwoNumbers(int firstNumber, int secondNumber) {
24
25         int result = firstNumber + secondNumber;
26         Debug.Log(result);
27
28     }
29
30
31
32 }
33
34
35
36
37
38
```

```
void AddAndPrintTwoNumbers(int number1, int number2) {

    int result = number1 + number2;
    Debug.Log(result);

}
```

Chapter 5: Lists, Arrays, and Dictionaries

```
18
19          if (personalDetails.Contains("firstName")) {
20              Debug.Log((string)personalDetails["firstName"]);
21          }
22          else {
23              Debug.Log("First name isnt stored in the hashtable");
24          }
25
```

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class LearningDictionaries : MonoBehaviour {
5
6      public Hashtable personalDetails = new Hashtable();
7
8
9      void Start() {
10
11         personalDetails.Add("firstName", "Greg");
12         personalDetails.Add("lastName", "Lukosek");
13         personalDetails.Add("gender", "male");
14         personalDetails.Add("isMarried", true);
15         personalDetails.Add("age", 29);
16
17     }
18
19 }
20
21
```



```
1  using UnityEngine;
2  using System.Collections;
3
4  public class LearningArrayList : MonoBehaviour {
5
6
7      public ArrayList inventory = new ArrayList();
8
9
10     void Start() {
11
12         inventory.Add(10);
13         inventory.Add(20);
14         inventory.Add("Adam");
15         inventory.Add(GameObject.Find("Player"));
16
17
18         Debug.Log(inventory[1].GetType());
19         Debug.Log(inventory[2].GetType());
20
21     }
22
23
24 }
```

```
1  using UnityEngine;
2  using System.Collections;
3  using System.Collections.Generic;
4
5
6  public class LearningLists : MonoBehaviour {
7
8
9      public List<string> familyMembers = new List<string>();
10
11
12     void Start() {
13
14         familyMembers.Add("Greg");
15         familyMembers.Add("Kate");
16         familyMembers.Add("Adam");
17         familyMembers.Add("Mia");
18
19
20         string thirdFamilyMember = familyMembers[2];
21         Debug.Log(thirdFamilyMember);
22
23     }
24
25 }
26
27
```

```
1  using UnityEngine;
2  using System.Collections;
3  using System.Collections.Generic;
4
5
6  public class LearningLists : MonoBehaviour {
7
8
9      public List<string> familyMembers = new List<string>();
10
11
12     void Start() {
13
14         familyMembers.Add("Greg");
15         familyMembers.Add("Kate");
16         familyMembers.Add("Adam");
17         familyMembers.Add("Mia");
18
19     }
20
21 }
22
```

Manual Scripting API Search scripting...

C# JS

GameObject.FindGameObjectsWithTag

SWITCH TO MANUAL

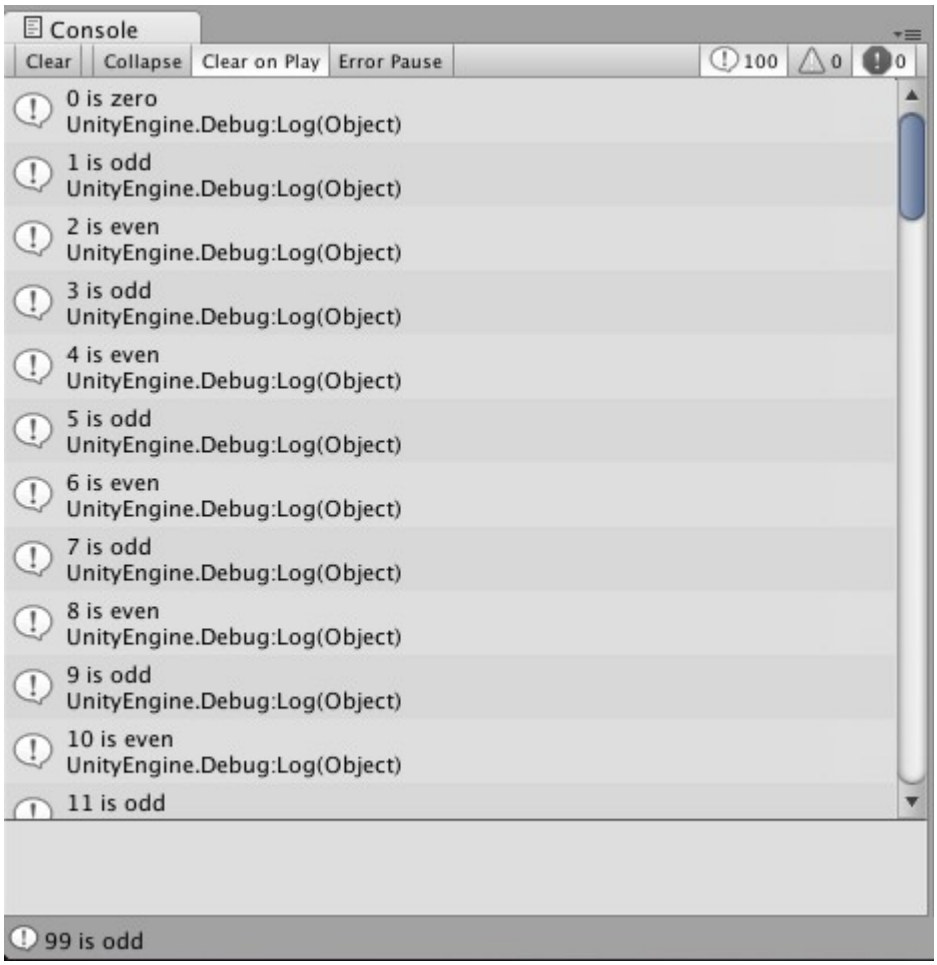
public static GameObject[] **FindGameObjectsWithTag**(string **tag**);

Parameters

tag	The name of the tag to search `GameObjects` for.

Chapter 6: Loops

```
1   using UnityEngine;
2   using System.Collections;
3   using System.Collections.Generic;
4
5   public class LearningLoopsSearching : MonoBehaviour {
6
7       public List<string> familyMembers = new List<string>();
8
9
10      void Start() {
11
12          familyMembers.Add("Greg");
13          familyMembers.Add("Kate");
14          familyMembers.Add("Adam");
15          familyMembers.Add("Mia");
16
17
18          int adamsIndex = -1;
19
20          for( int i = 0; i < familyMembers.Count; i++) {
21
22              if (familyMembers[i] == "Adam") {
23                  adamsIndex = i;
24                  break;
25              }
26          }
27
28
29          if (adamsIndex == -1) {
30              Debug.Log("Adam value is not stored in the list");
31          }
32          else {
33              Debug.Log("Adam value found at index " + adamsIndex);
34          }
35
36      }
37
38  }
```


Console
Clear
Collapse
Clear on Play
Error Pause
100
0
0
0 is zero
UnityEngine.Debug:Log(Object)
1 is odd
UnityEngine.Debug:Log(Object)
2 is even
UnityEngine.Debug:Log(Object)
3 is odd
UnityEngine.Debug:Log(Object)
4 is even
UnityEngine.Debug:Log(Object)
5 is odd
UnityEngine.Debug:Log(Object)
6 is even
UnityEngine.Debug:Log(Object)
7 is odd
UnityEngine.Debug:Log(Object)
8 is even
UnityEngine.Debug:Log(Object)
9 is odd
UnityEngine.Debug:Log(Object)
10 is even
UnityEngine.Debug:Log(Object)
11 is odd
99 is odd

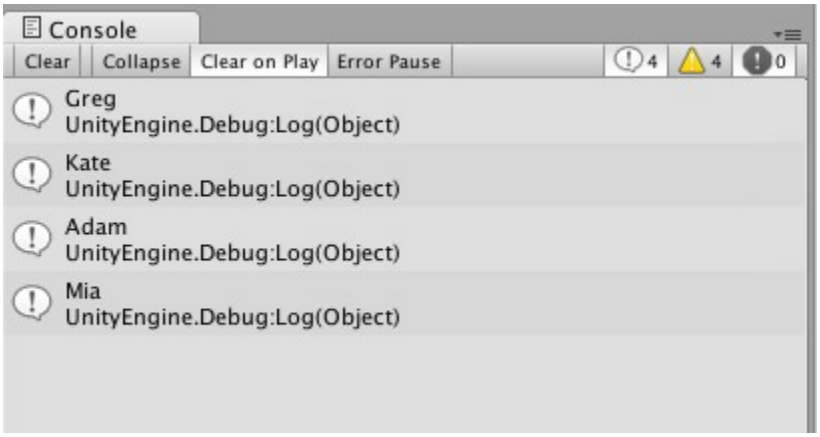
```
1   using UnityEngine;
2   using System.Collections;
3
4   public class LearningLoopsWithStatements : MonoBehaviour {
5
6
7       void Start () {
8
9           for( int i = 0; i < 100; i++) {
10
11              if (i == 0) {
12                  Debug.Log(i + " is zero");
13              }
14              else if (IsNumberEven(i)) {
15                  Debug.Log(i + " is even");
16              }
17              else {
18                  Debug.Log(i + " is odd");
19              }
20          }
21      }
22
23
24      public bool IsNumberEven(int number) {
25
26          if (number % 2 == 0) {
27              return true;
28          }
29          else {
30              return false;
31          }
32      }
33
34
35  }
36
37
38
```

```
19
20          int i = 0;
21
22          while (i<10) {
23
24              //loop block
25              Debug.Log(i);
26              i++;
27          }
28
29
```

```
while (condition) {

    //loop block

}
```

```
1  using UnityEngine;
2  using System.Collections;
3  using System.Collections.Generic;
4
5
6  public class LearningLoopsFor : MonoBehaviour {
7
8
9      public List<string> familyMembers = new List<string>();
10
11
12     void Start() {
13
14         familyMembers.Add("Greg");
15         familyMembers.Add("Kate");
16         familyMembers.Add("Adam");
17         familyMembers.Add("Mia");
18
19
20         for( int i = 0; i < familyMembers.Count; i++) {
21
22             //loop block
23             Debug.Log(familyMembers[i]);
24         }
25
26
27     }
28
29
30 }
31
```

```
for( int i = 0; i < 10; i++) {
    //loop block
}
```



```
1  using UnityEngine;
2  using System.Collections;
3  using System.Collections.Generic;
4
5  public class LearningLoopsForeach : MonoBehaviour {
6
7
8      public List<string> familyMembers = new List<string>();
9
10
11     void Start() {
12
13         familyMembers.Add("Greg");
14         familyMembers.Add("Kate");
15         familyMembers.Add("Adam");
16         familyMembers.Add("Mia");
17
18
19
20         foreach (string familyMember in familyMembers) {
21
22             Debug.Log(familyMember);
23
24         }
25
26     }
27
28 }
```

```
foreach (Type elementName in myCollectionVariable) {

    //loop block

}
```

Chapter 7: Object, a Container with Variables and Methods

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class LearningObjectsWithConstructors : MonoBehaviour {
5
6
7      public Person man;
8      public Person woman;
9
10
11
12     void Start() {
13
14         man = new Person("Greg", "Lukosek");
15         woman = new Person("Kate", "Lukosek");
16
17         man.spouse = woman;
18         woman.spouse = man;
19
20         if (man.IsMarriedWith(woman)) {
21             Debug.Log(man.firstName + " is married to " + woman.firstName);
22         }
23         else {
24             Debug.Log(man.firstName + " and " + woman.firstName + " are not married");
25         }
26     }
27 }
28
```

```
void Start() {

    new Person(
                ▲ 2 of 2 ▼
                public Person (
                  string pFirstName,
                  string pLastName
                )
```

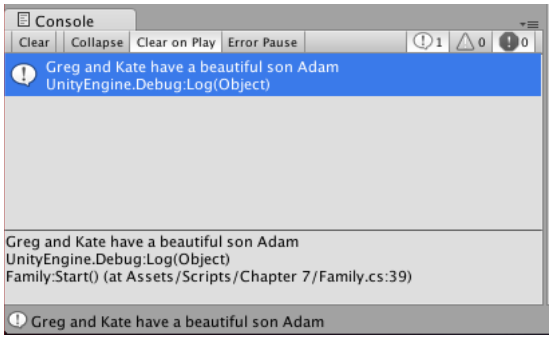
```
void Start() {

    new Person(
                public Person ()        ▲ 1 of 2 ▼
```

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class Person {
5
6      public string firstName = "";
7      public string lastName = "";
8      public Person spouse;
9
10
11     public Person () {
12
13     }
14
15
16     public Person (string pFirstName, string pLastName) {
17         this.firstName = pFirstName;
18         this.lastName = pLastName;
19     }
20
21
22     public bool IsMarriedWith (Person otherPerson) {
23
24         if (spouse != null) {
25             //Person object is stored in spouse variable
26             if (otherPerson == this.spouse) {
27                 //otherPerson object is the same as stored spouse
28                 return true;
29             }
30             else {
31                 //not married
32                 return false;
33             }
34         }
35         else {
36             //spouse variable is not assigned so this
37             //Person is not married at all
38             return false;
39         }
40     }
41
42 }
43
```

```
public Person (string pFirstName, string pLastName) {
    this.firstName = pFirstName;
    this.lastName = pLastName;
}
```

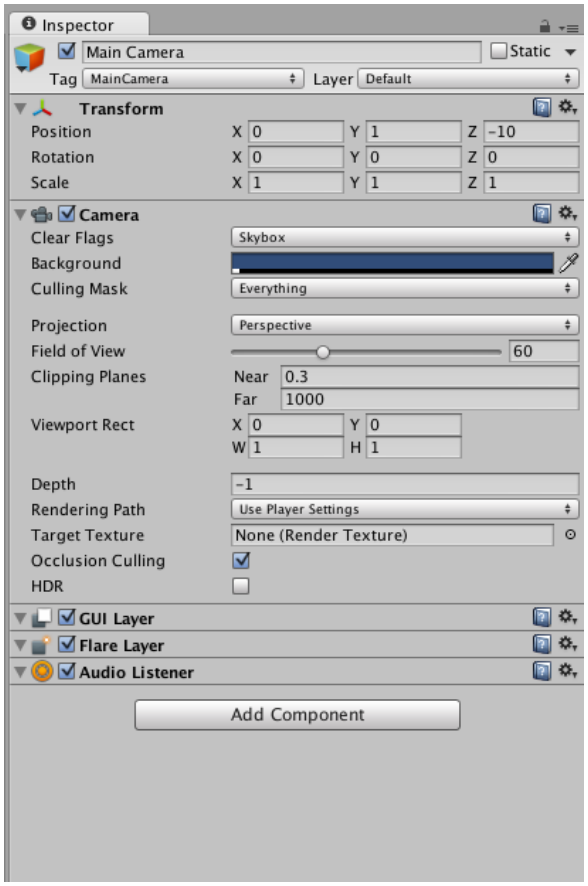
```
1  using UnityEngine;
2  using System.Collections;
3
4  public class LearningObjects : MonoBehaviour {
5
6
7      public Person man;
8      public Person woman;
9
10
11
12     void Start() {
13
14         man = new Person();
15         man.firstName = "Greg";
16         man.lastName = "Lukosek";
17
18         woman = new Person();
19         woman.firstName = "Kate";
20         woman.lastName = "Lukosek";
21
22         man.spouse = woman;
23         woman.spouse = man;
24
25
26         if (man.IsMarriedWith(woman)) {
27             Debug.Log(man.firstName + " is married to " + woman.firstName);
28         }
29         else {
30             Debug.Log(man.firstName + " and " + woman.firstName + " are not married");
31         }
32     }
33
34 }
35
36
```

```
1   using UnityEngine;
2   using System.Collections;
3
4   public class Person {
5
6       public string firstName = "";
7       public string lastName = "";
8       public Person spouse;
9
10
11      public bool IsMarriedWith (Person otherPerson) {
12
13          if (spouse != null) {
14              //Person object is stored in spouse variable
15              if (otherPerson == this.spouse) {
16                  //otherPerson object is the same as stored spouse
17                  return true;
18              }
19              else {
20                  //not married
21                  return false;
22              }
23          }
24          else {
25              //spouse variable is not assigned so this
26              //Person is not married at all
27              return false;
28          }
29      }
30
31  }
32
```

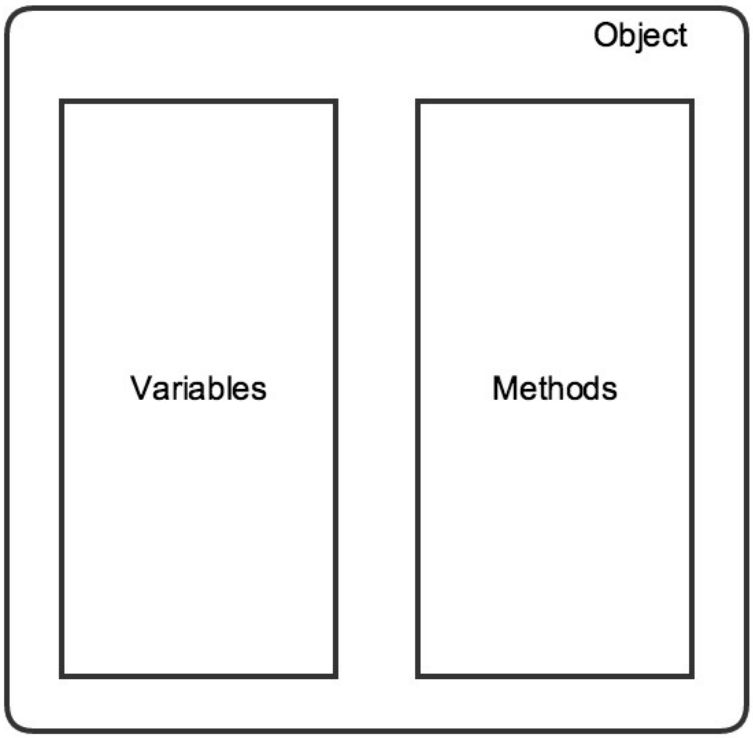


```
using UnityEngine;
using System.Collections;

public class Family : MonoBehaviour {


    public Person father;
    public Person mother;
    public Person son;


    void Start() {

        father = new Person();
        father.firstName = "Greg";
        father.lastName = "Lukosek";
        father.age = 29;
        father.isMale = true;
        father.isMarried = true;


        mother = new Person();
        mother.firstName = "Kate";
        mother.lastName = "Lukosek";
        mother.age = 28;
        mother.isMale = false;
        mother.isMarried = true;


        son = new Person();
        son.firstName = "Adam";
        son.lastName = "Lukosek";
        son.age = 3;
        son.isMale = true;
        son.isMarried = false;



        Debug.Log(father.firstName + " and " + mother.firstName +
                  " have a beautiful son " + son.firstName);


    }
}
```

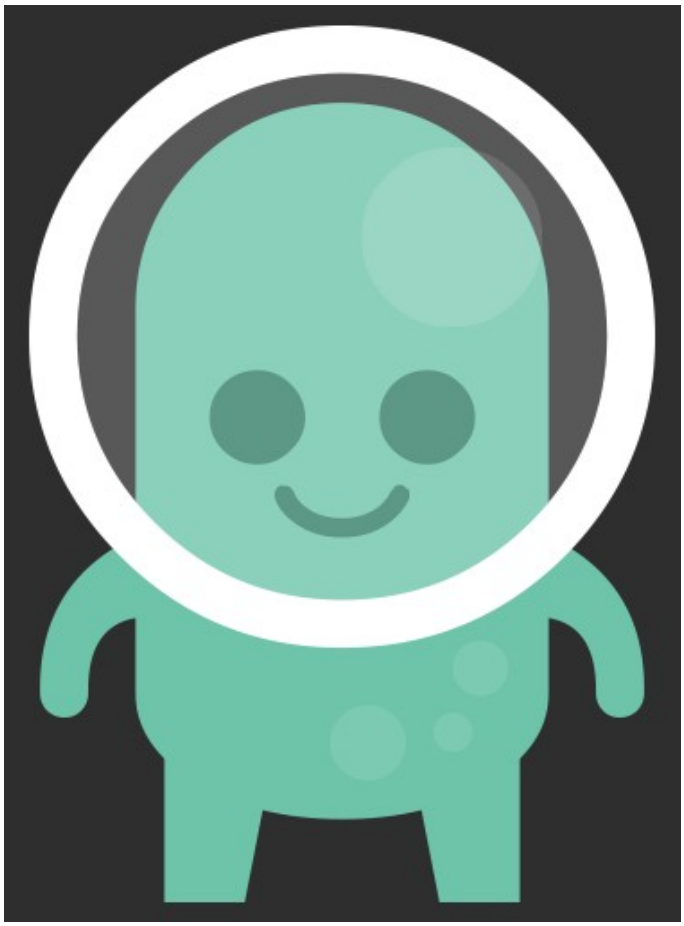
```
using UnityEngine;
using System.Collections;

public class Person {


    public string firstName = "";
    public string lastName = "";
    public int age = 0;
    public string address = "";
    public bool isMale = false;
    public bool isMarried = false;


}
```

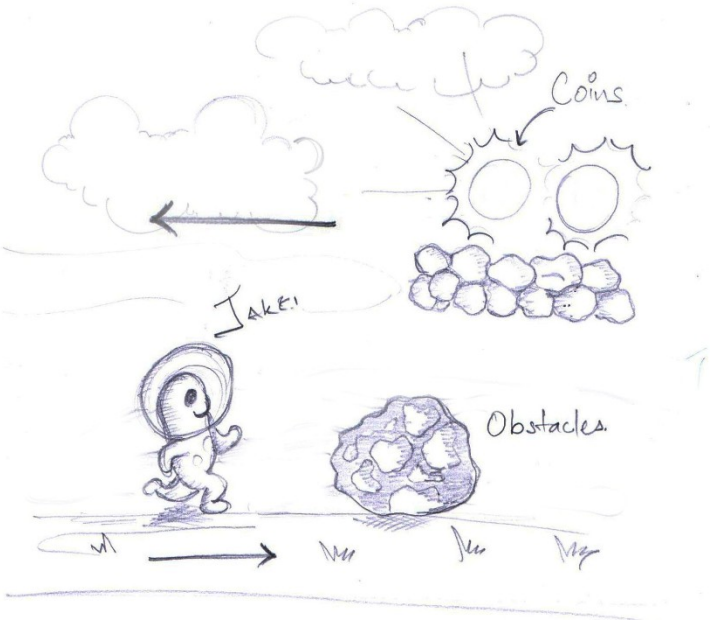


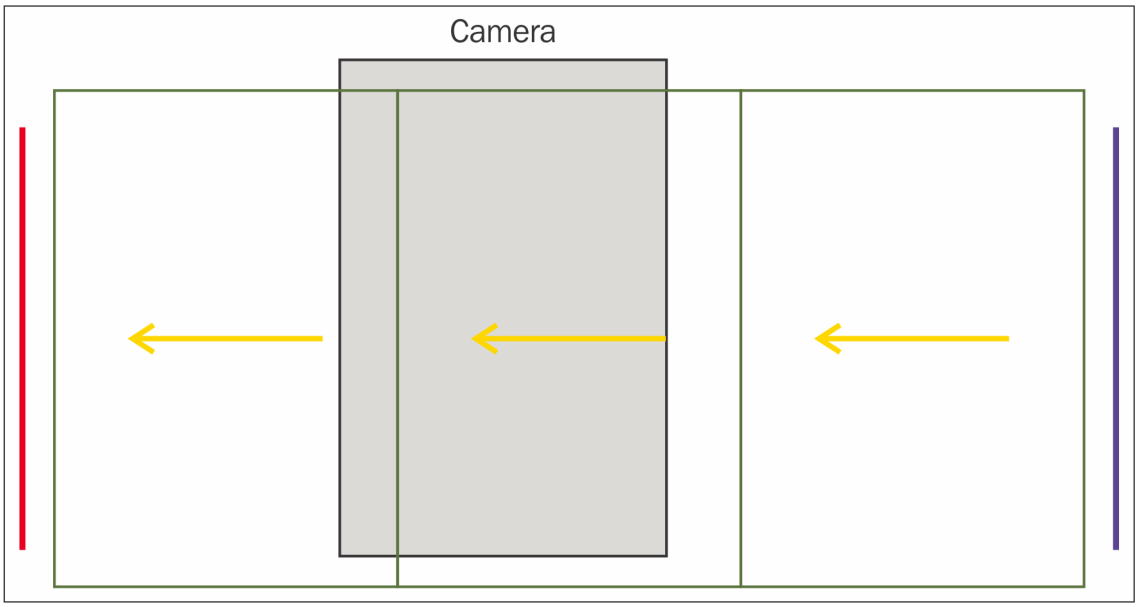

```
using UnityEngine;
using System.Collections;

public class LearningScript : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }
}
```

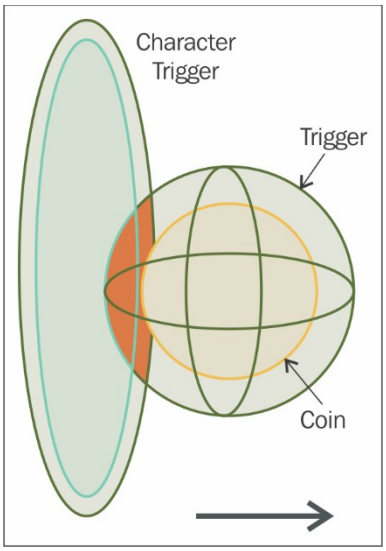

Object
Variables
Methods

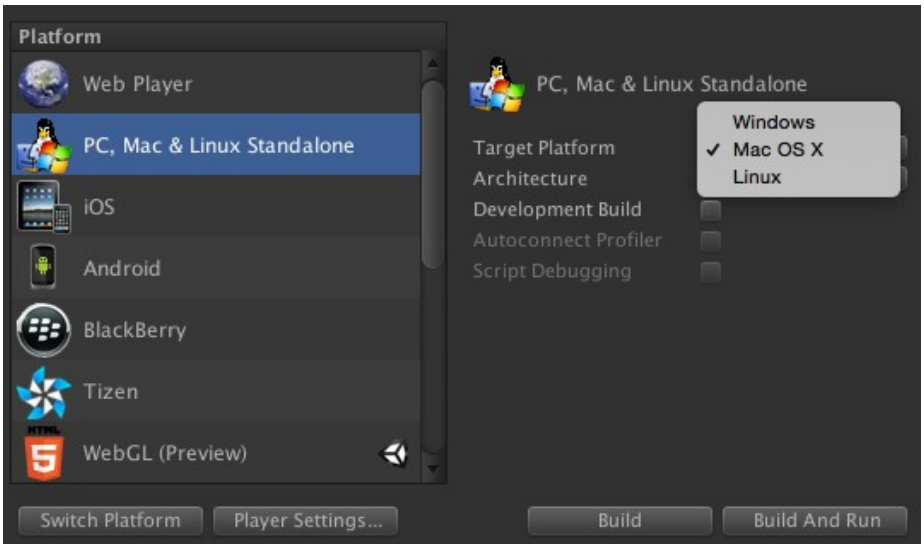
Chapter 8: Let's Make a Game! – From Idea to Development

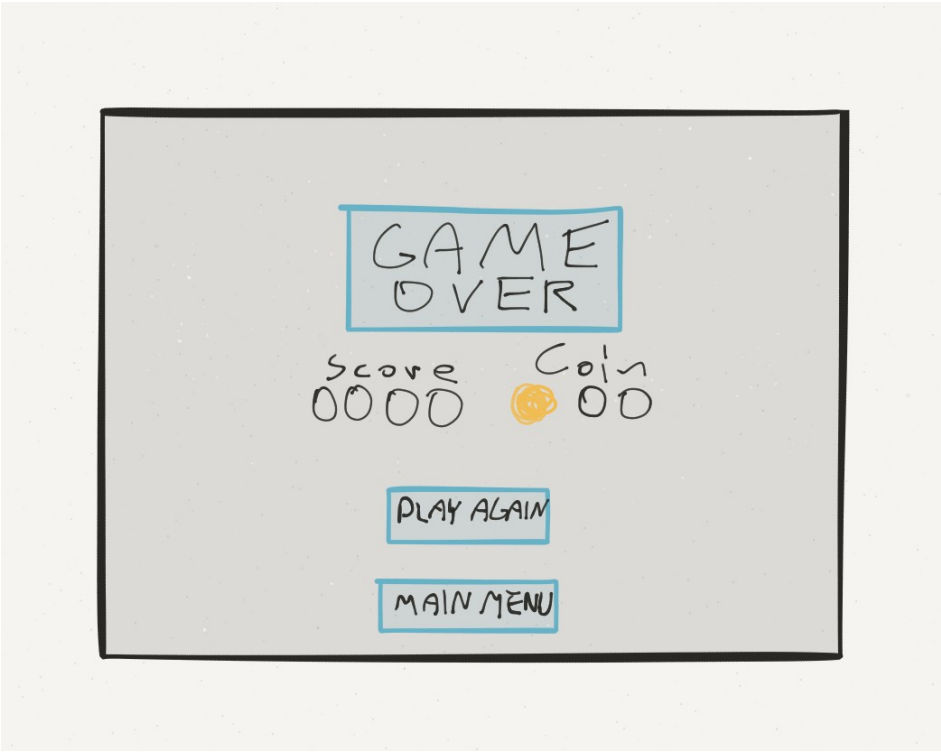



Coins
Jake!
Obstacles

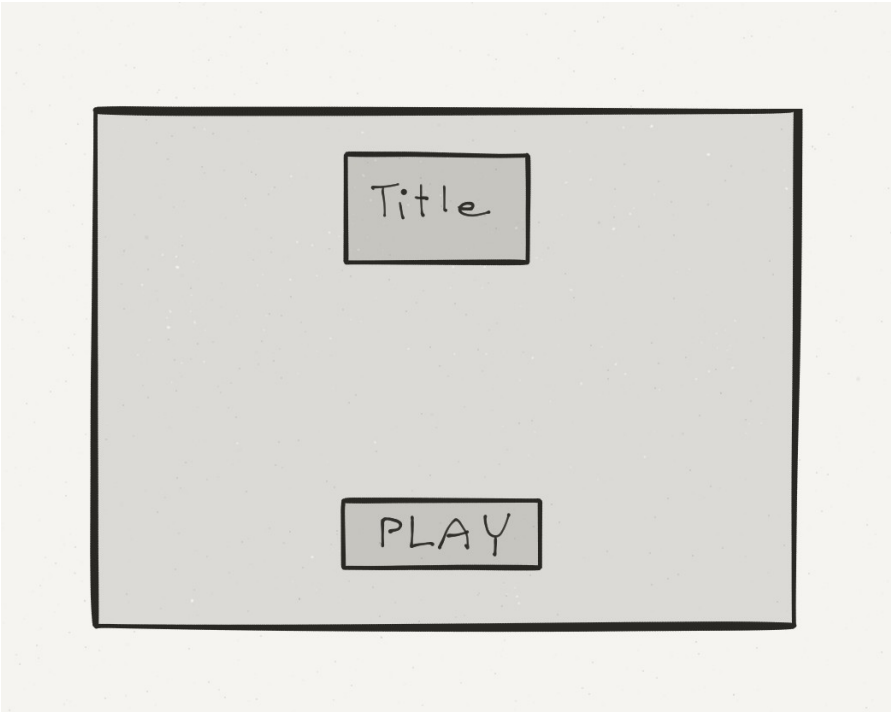

Camera


Character
Trigger
Trigger
Coin


Platform
Web Player
PC, Mac & Linux Standalone
iOS
Android
BlackBerry
Tizen
WebGL (Preview)
PC, Mac & Linux Standalone
Target Platform
Architecture
Development Build
Autoconnect Profiler
Script Debugging
Windows
Mac OS X
Linux
Switch Platform
Player Settings...
Build
Build And Run

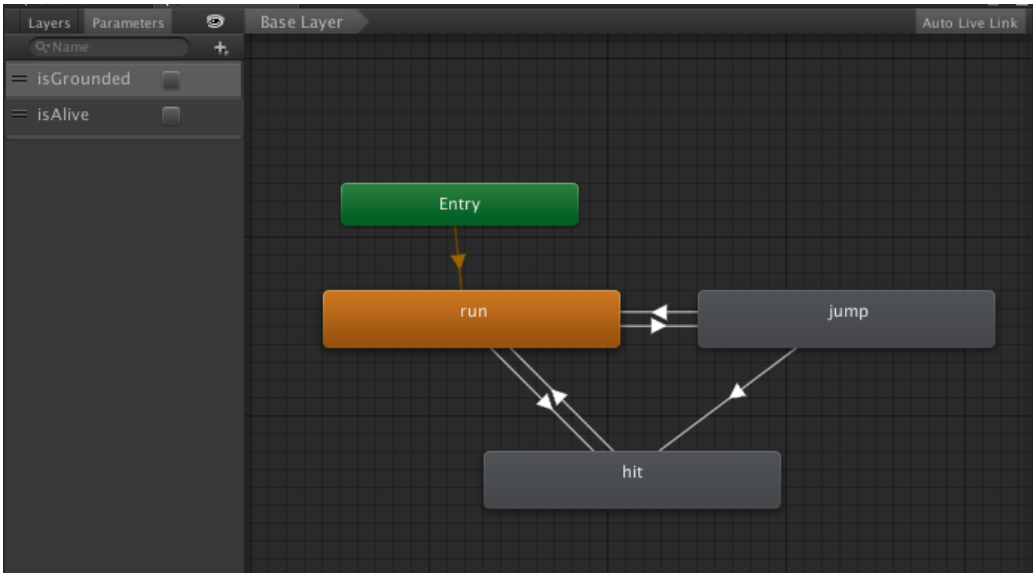

GAME OVER
Score
0000
Coin
00
PLAY AGAIN
MAIN MENU


Score
00000
Coin
counter
High score
000060

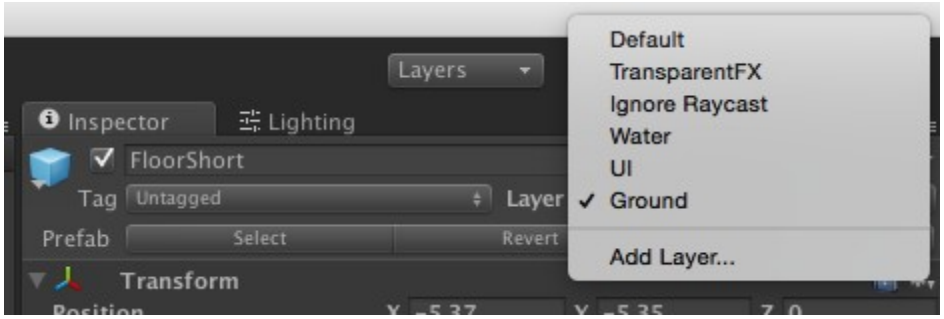

Title
PLAY

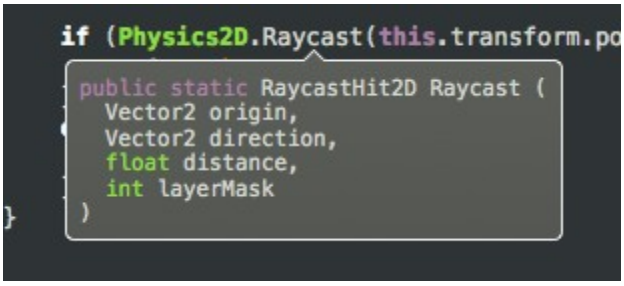
Chapter 9: Starting Your First Game

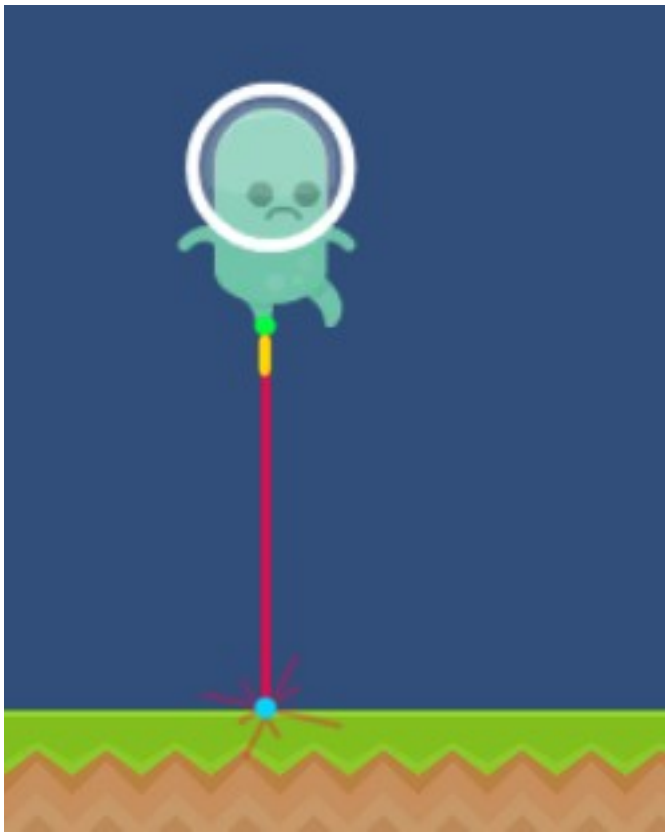
```
31      void FixedUpdate() {
32
33          if (rigidBody.velocity.x < runningSpeed) {
34              rigidBody.velocity = new Vector2(runningSpeed, rigidBody.velocity.y);
35          }
36      }
37
```




Player Controller (Script)
Script
PlayerController
Jump Force
25
Ground Layer
Ground
Nothing
Everything
Default
TransparentFX
Ignore Raycast
Water
UI
Ground


Layers
Inspector
Lighting
FloorShort
Tag
Untagged
Layer
Prefab
Select
Revert
Transform
Default
TransparentFX
Ignore Raycast
Water
UI
Ground
Add Layer...


if (Physics2D.Raycast(this.transform.po
public static RaycastHit2D Raycast (
Vector2 origin,
Vector2 direction,
float distance,
int layerMask
)



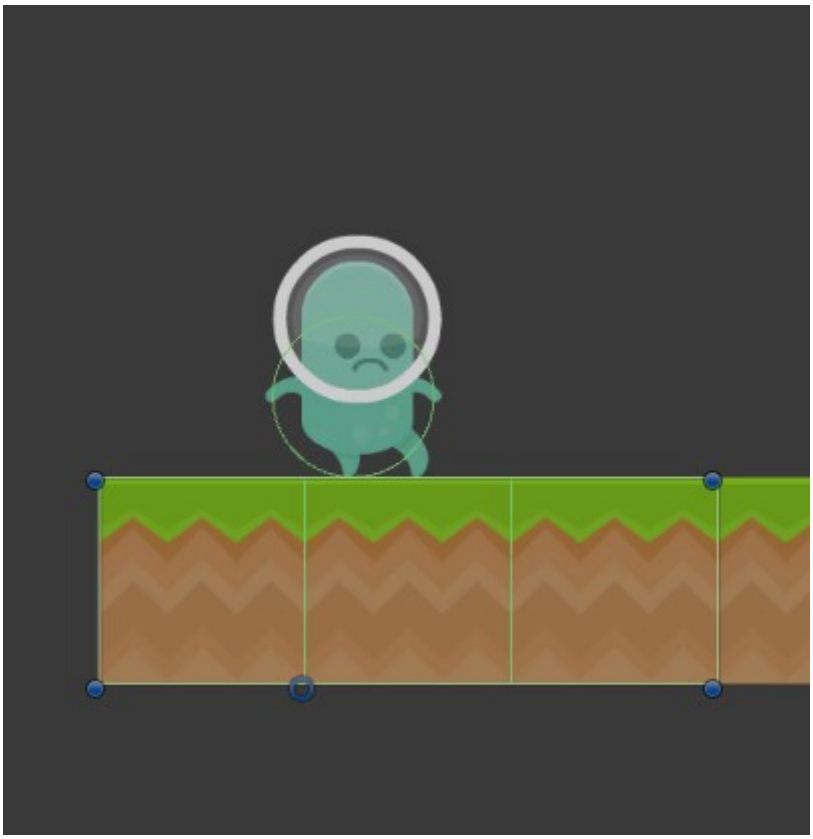
```
void Jump() {
    if (IsGrounded()) {
        rigidBody.AddForce(Vector2.up * jumpForce, ForceMode2D.Impulse);
    }
}

public LayerMask groundLayer;

bool IsGrounded() {

    if (Physics2D.Raycast(this.transform.position, Vector2.down, 0.2f, groundLayer.value)) {
        return true;
    }
    else {
        return false;
    }
}
```

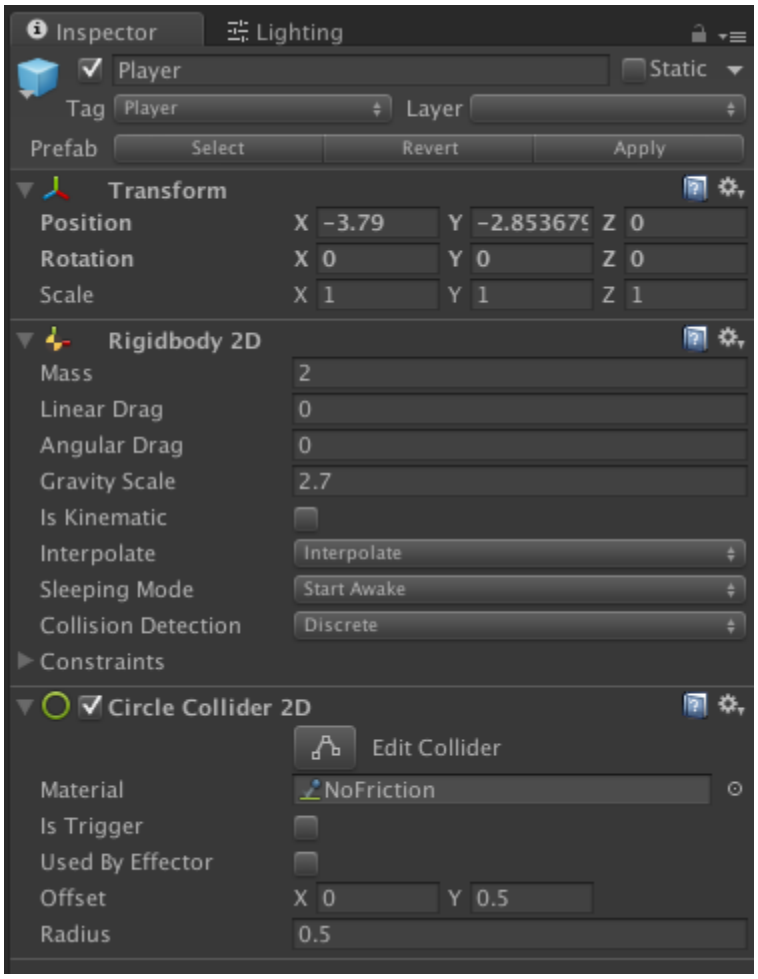
```
using UnityEngine;
using System.Collections;

public class PlayerController : MonoBehaviour {

    public float jumpForce = 6f;
    private Rigidbody2D rigidBody;

    void Awake() {
        rigidBody = GetComponent<Rigidbody2D>();
    }

    // Update is called once per frame
    void Update () {

        if (Input.GetMouseButtonDown(0)) {
            Jump();
        }
    }

    void Jump() {
        rigidBody.AddForce(Vector2.up * jumpForce, ForceMode2D.Impulse);
    }

}
```

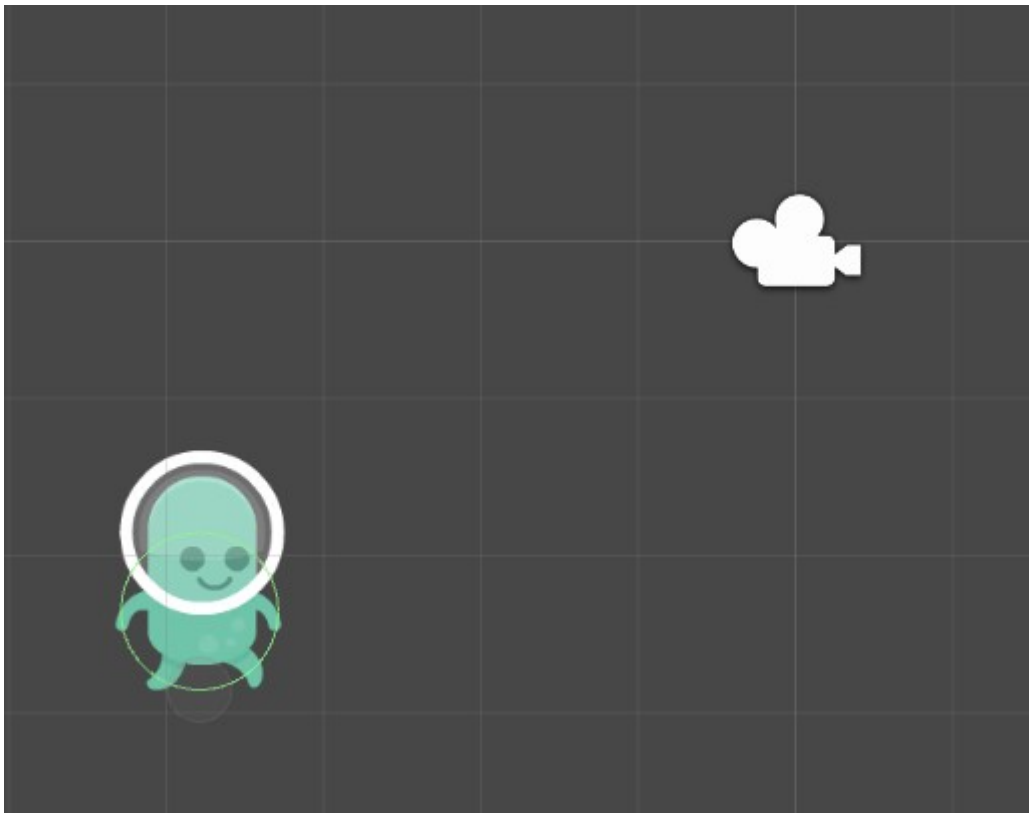
```
using UnityEngine;
using System.Collections;

public class PlayerController : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

        if (Input.GetMouseButtonDown(0)) {
            Debug.Log("Left mouse button clicked!");
        }

    }
}
```

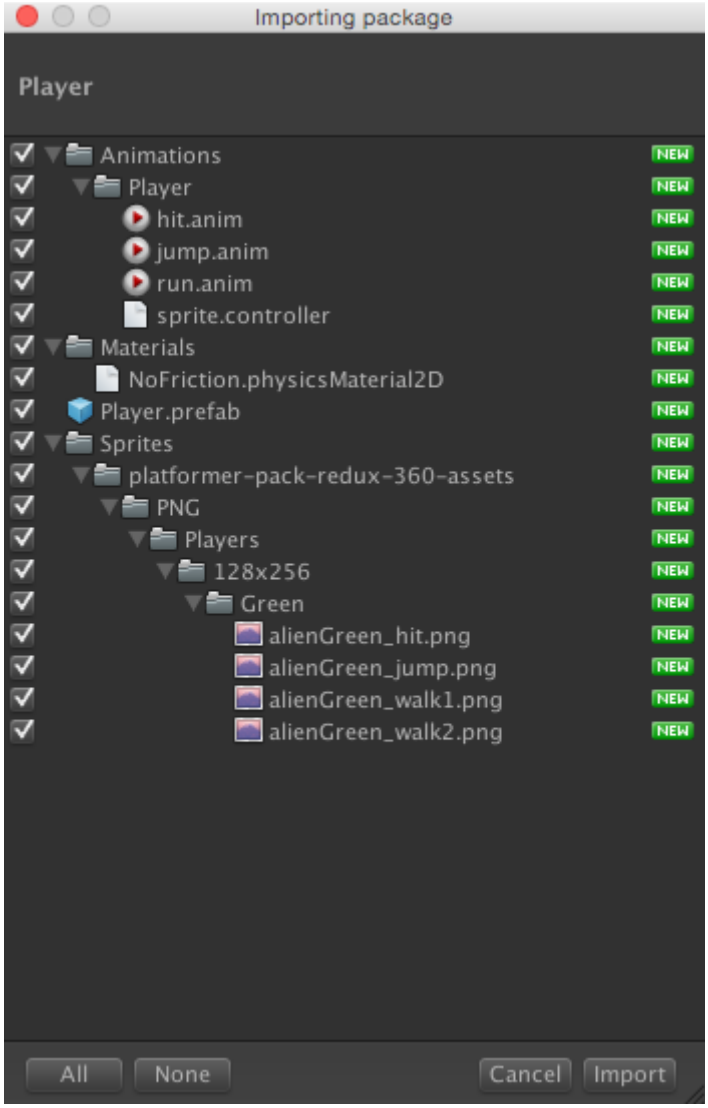
Input.GetMouseButtonDown

public static bool **GetMouseButtonDown**(int **button**);

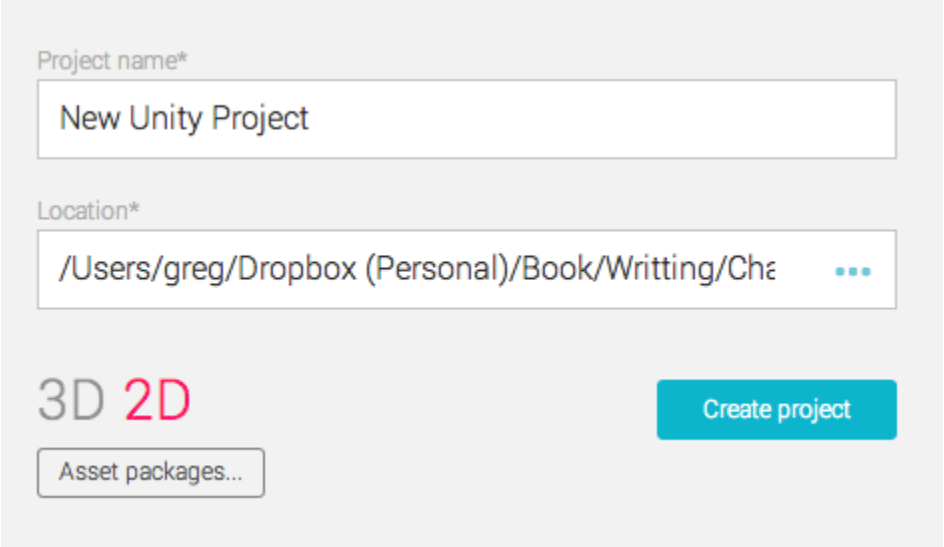



Inspector
Lighting
Player
Static
Tag
Player
Layer
Prefab
Select
Revert
Apply
Transform
Position
X -3.79
Y -2.853679
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Rigidbody 2D
Mass
2
Linear Drag
0
Angular Drag
0
Gravity Scale
2.7
Is Kinematic
Interpolate
Interpolate
Sleeping Mode
Start Awake
Collision Detection
Discrete
Constraints
Circle Collider 2D
Edit Collider
Material
NoFriction
Is Trigger
Used By Effector
Offset
X 0
Y 0.5
Radius
0.5




Importing package
Player
Animations
Player
hit.anim
jump.anim
run.anim
sprite.controller
Materials
NoFriction.physicsMaterial2D
Player.prefab
Sprites
platformer-pack-redux-360-assets
PNG
Players
128x256
Green
alienGreen_hit.png
alienGreen_jump.png
alienGreen_walk1.png
alienGreen_walk2.png
NEW
All
None
Cancel
Import


Project
Create
Animations
Animators
CameraFollow
Coin
Fonts
LeaveTrigger
LevelGenerator
MainScene
Materials
Player
PlayerAnimator
Scripts
skyGradient
Sprites
Project
Create
Animations
Enemies
LevelProps
Coin
CoinSpin
Player
hit
jump
run
sprite
Animators
Fonts
Materials
Prefabs
Scenes
MainScene
Scripts
Bee
CameraFollow
Collectable
GameManager
KillTrigger
LeaveTrigger
LevelGenerator
LevelPiece
PlayerController
UI
ViewGameOver
ViewInGame
Sprites

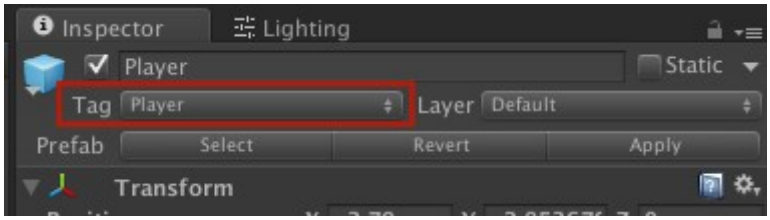

Project name*
New Unity Project
Location*
/Users/greg/Dropbox (Personal)/Book/Writting/Cha
3D 2D
Create project
Asset packages...

Chapter 10: Writing GameManager

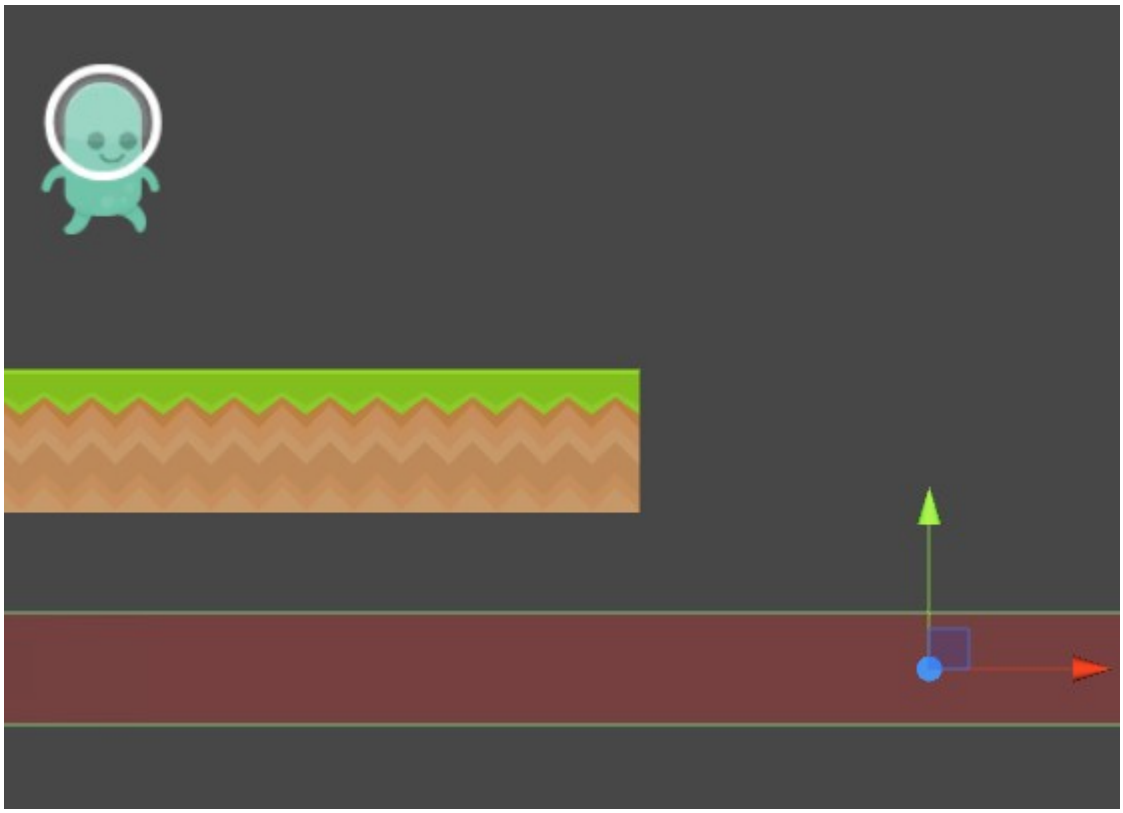
```
23
24      //called to start the game
25      public void StartGame() {
26          PlayerController.instance.StartGame();
27          SetGameState(GameState.inGame);
28      }
29
```

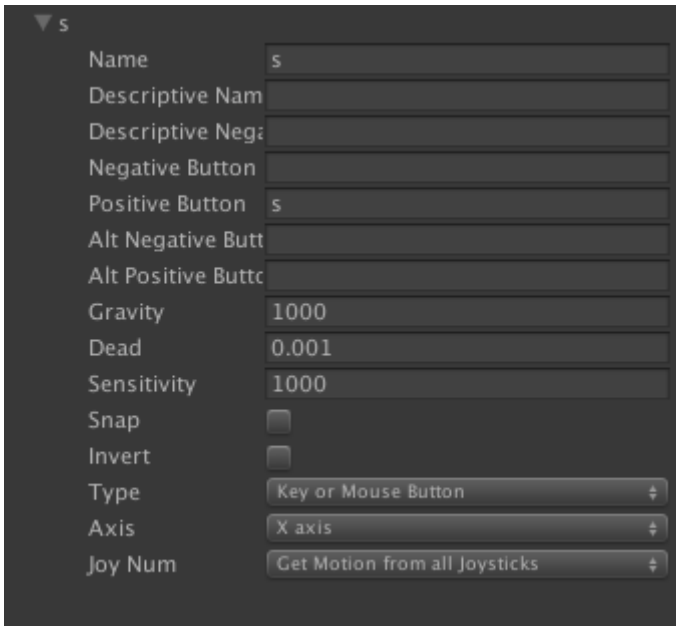
```
4   public class PlayerController : MonoBehaviour {
5
6       public static PlayerController instance;
7
8       public float jumpForce = 6f;
9       public float runningSpeed = 1.5f;
10      public Animator animator;
11
12      private Vector3 startingPosition;
13      private Rigidbody2D rigidBody;
14
15      void Awake() {
16          instance = this;
17          rigidBody = GetComponent<Rigidbody2D>();
18          startingPosition = this.transform.position;
19      }
20
21
22      public void StartGame() {
23          animator.SetBool("isAlive", true);
24          this.transform.position = startingPosition;
25      }
```

```
using UnityEngine;
using System.Collections;

public class KillTrigger : MonoBehaviour {

    void OnTriggerEnter2D(Collider2D other) {

        if (other.tag == "Player") {
            PlayerController.instance.Kill();
        }
    }
}
```




```
using UnityEngine;
using System.Collections;

public class KillTrigger : MonoBehaviour {

    void OnTriggerEnter2D(Collider2D other) {

        if (other.tag == "Player") {
            Debug.Log("Player collider entered the trigger");
        }
    }

}
```




s
Name
s
Descriptive Nam
Descriptive Nega
Negative Button
Positive Button
s
Alt Negative Butt
Alt Positive Butto
Gravity
1000
Dead
0.001
Sensitivity
1000
Snap
Invert
Type
Key or Mouse Button
Axis
X axis
Joy Num
Get Motion from all Joysticks

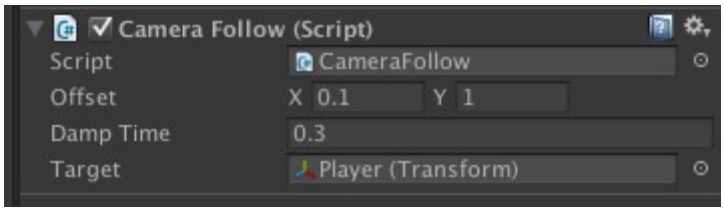
```
54
55      void Update() {
56
57          if (Input.GetButtonDown("s")) {
58              StartGame();
59          }
60      }
61
62
```

```
19
20      void Update () {
21
22          if (GameManager.instance.currentGameState == GameState.inGame)
23          {
24              if (Input.GetMouseButtonDown(0)) {
25                  Jump();
26              }
27              animator.SetBool("isGrounded", isGrounded());
28          }
29      }
30
31
32      void FixedUpdate() {
33
34          if (GameManager.instance.currentGameState == GameState.inGame)
35          {
36              if (rigidBody.velocity.x < runningSpeed) {
37                  rigidBody.velocity = new Vector2(runningSpeed, rigidBody.velocity.y);
38              }
39          }
40      }
```

```
1  using UnityEngine;
2  using System.Collections;
3
4  public enum GameState {
5      menu,
6      inGame,
7      gameOver
8  }
9
10 public class GameManager : MonoBehaviour {
11
12     public GameState currentGameState = GameState.menu;
13
14
15     void Start() {
16         StartGame();
17     }
18
19     //called to start the game
20     public void StartGame() {
21         SetGameState(GameState.inGame);
22     }
23
24     //called when player die
25     public void GameOver() {
26         SetGameState(GameState.gameOver);
27     }
28
29     //called when player decide to go back to the menu
30     public void BackToMenu() {
31         SetGameState(GameState.menu);
32     }
33
34     void SetGameState (GameState newGameState) {
35
36         if (newGameState == GameState.menu) {
37             //setup Unity scene for menu state
38         }
39         else if (newGameState == GameState.inGame) {
40             //setup Unity scene for inGame state
41         }
42         else if (newGameState == GameState.gameOver) {
43             //setup Unity scene for gameOver state
44         }
45
46         currentGameState = newGameState;
47     }
48 }
49
```

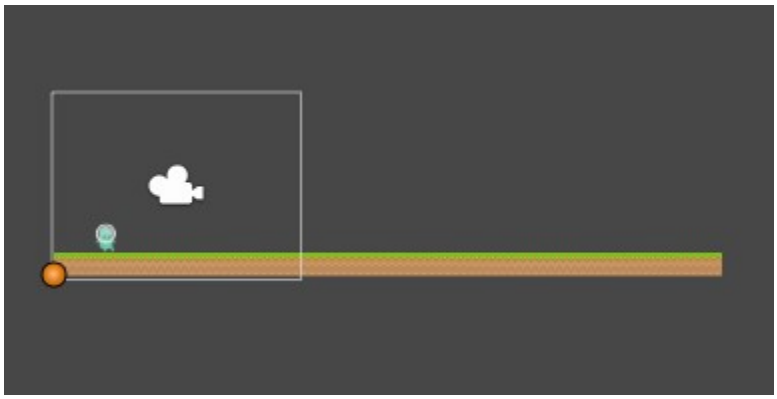
```
using UnityEngine;
using System.Collections;

public class GameManager : MonoBehaviour {


    //called to start the game
    public void StartGame() {

    }

    //called when player die
    public void GameOver() {

    }

    //called when player decide to go back to the menu
    public void BackToMenu() {

    }

}
```

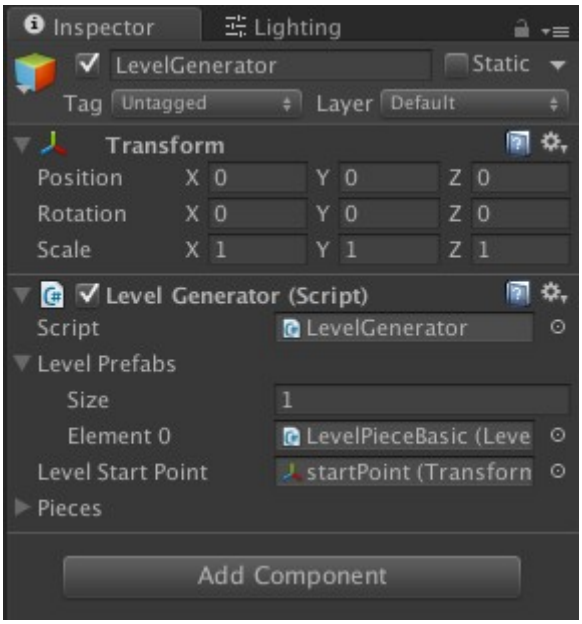
Chapter 11: The Game Level

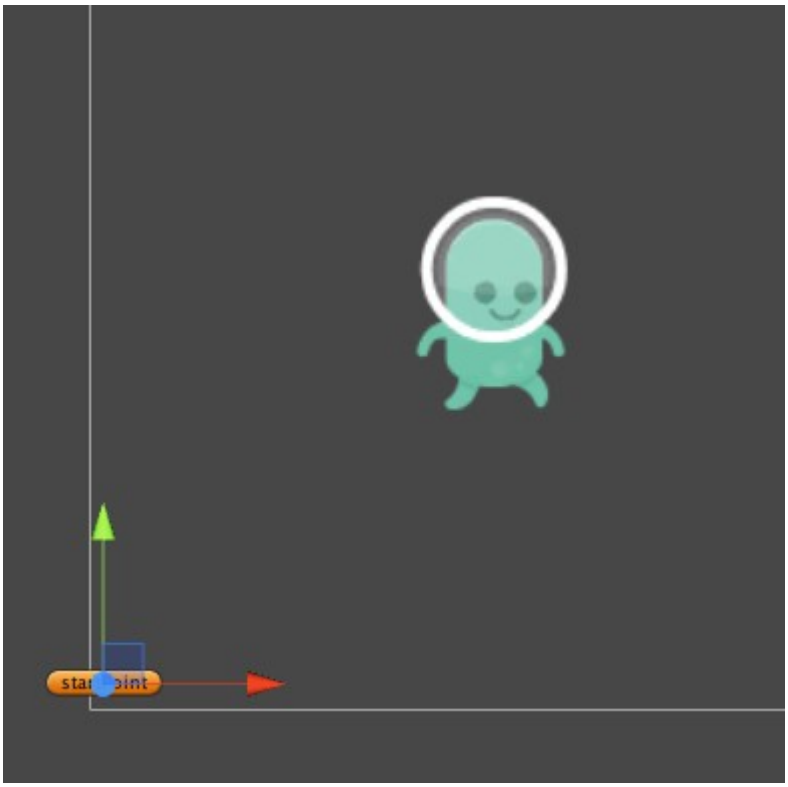



```
1  using UnityEngine;
2  using System.Collections;
3
4  public class LeaveTrigger : MonoBehaviour {
5
6      void OnTriggerEnter2D(Collider2D other) {
7
8          LevelGenerator.instance.AddPiece();
9          LevelGenerator.instance.RemoveOldestPiece();
10     }
11
12 }
```

```
58     public void RemoveOldestPiece() {
59
60         LevelPiece oldestPiece = pieces[0];
61
62         pieces.Remove(oldestPiece);
63         Destroy(oldestPiece.gameObject);
64     }
65
```




Inspector
Lighting
LevelGenerator
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
Level Generator (Script)
Script
LevelGenerator
Level Prefabs
Size
1
Element 0
LevelPieceBasic (Leve
Level Start Point
startPoint (Transforn
Pieces
Add Component



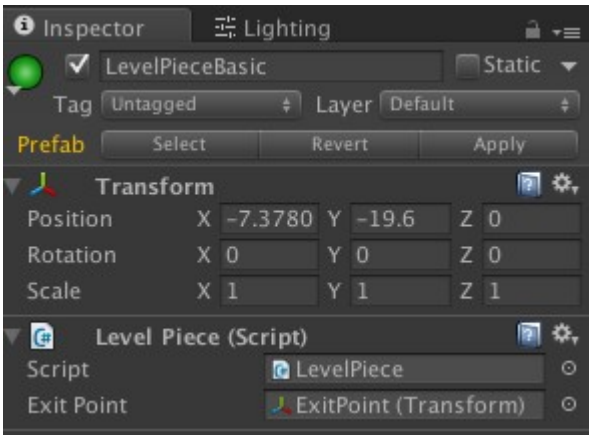
Random.Range

public static float **Range**(float **min**, float **max**);

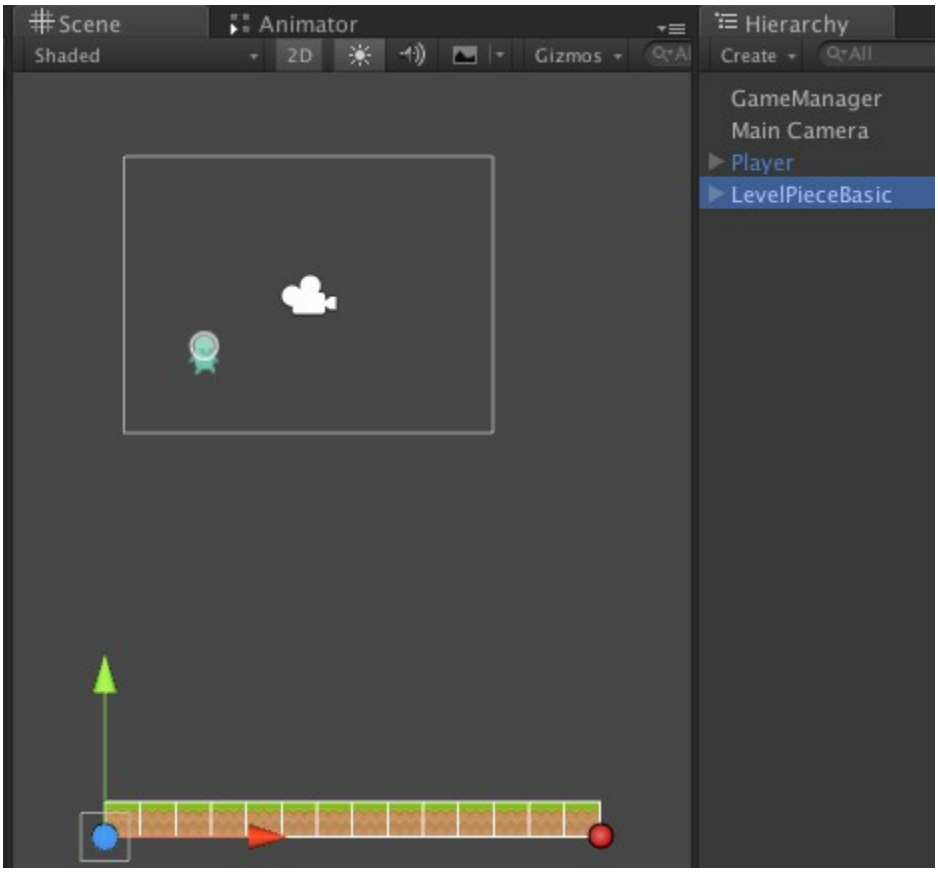
```
30      public void AddPiece() {
31
32          //pick the random number
33          int randomIndex = Random.Range(0, levelPrefabs.Count);
34
35          //Instantiate copy of random level prefab and store it in piece variable
36          LevelPiece piece = (LevelPiece)Instantiate(levelPrefabs[randomIndex]);
37          piece.transform.SetParent(this.transform, false);
38
39          Vector3 spawnPosition = Vector3.zero;
40
41          //position
42          if (pieces.Count == 0) {
43              //first piece
44              spawnPosition = levelStartPoint.position;
45          }
46          else {
47              //take exit point from last piece as a spawn point to new piece
48              spawnPosition = pieces[pieces.Count-1].exitPoint.position;
49          }
50
51          piece.transform.position = spawnPosition;
52          pieces.Add(piece);
53      }
```

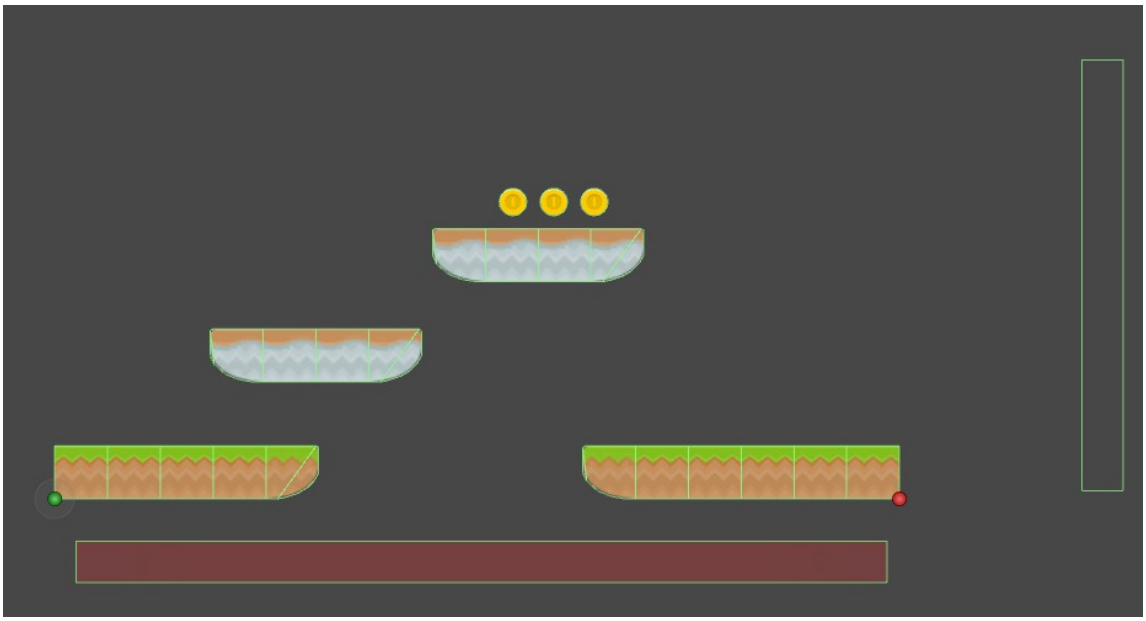
```
1  using UnityEngine;
2  using System.Collections;
3  using System.Collections.Generic;
4
5  public class LevelGenerator : MonoBehaviour {
6
7      public static LevelGenerator instance;
8      //all level pieces blueprints used to copy from
9      public List<LevelPiece> levelPrefabs = new List<LevelPiece>();
10     //starting point of the very first level piece
11     public Transform levelStartPoint;
12     //all level pieces that are currently in the level
13     public List<LevelPiece> pieces = new List<LevelPiece>();
14
15     void Awake() {
16         instance = this;
17     }
```

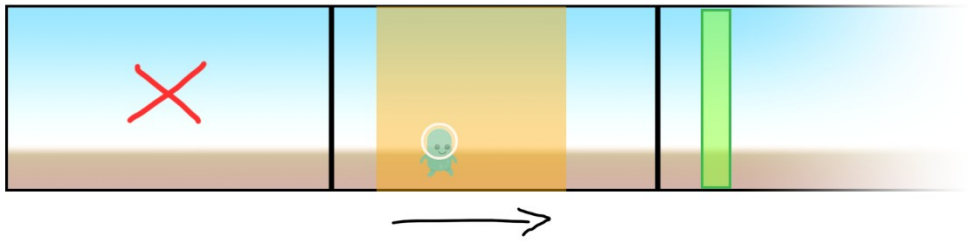
```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;

public class LevelGenerator : MonoBehaviour {

    public static LevelGenerator instance;
    public List<LevelPiece> levelPrefabs = new List<LevelPiece>();
    public Transform levelStartPoint;
    public List<LevelPiece> pieces = new List<LevelPiece>();

    void Awake() {
        instance = this;
    }

}

```



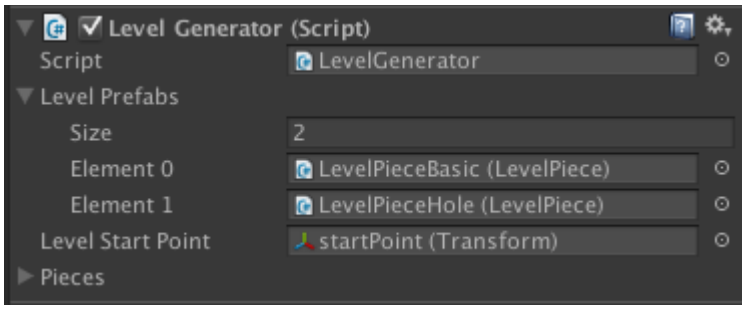

```
using UnityEngine;
using System.Collections;

public class LevelPiece : MonoBehaviour {

    public Transform exitPoint;


}

```

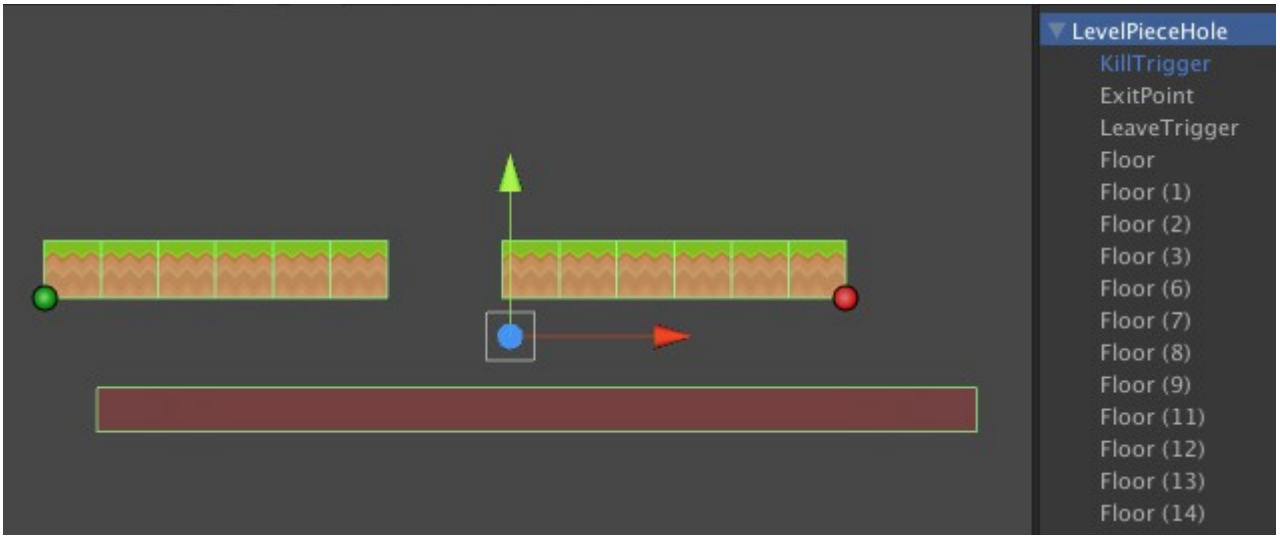

Scene
Animator
Hierarchy
Shaded
2D
Gizmos
Create
GameManager
Main Camera
Player
LevelPieceBasic





Chapter 12: The User Interface

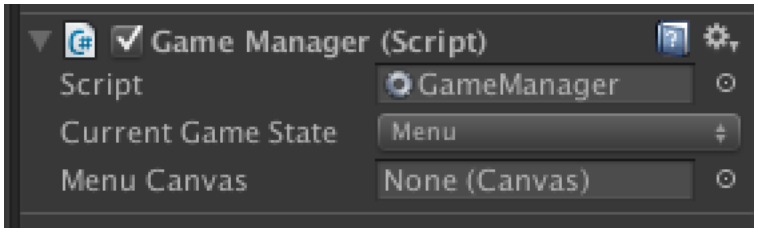




```
42      void SetGameState (GameState newGameState) {
43
44          if (newGameState == GameState.menu) {
45              //setup Unity scene for menu state
46              menuCanvas.enabled = true;
47              inGameCanvas.enabled = false;
48          }
49          else if (newGameState == GameState.inGame) {
50              //setup Unity scene for inGame state
51              menuCanvas.enabled = false;
52              inGameCanvas.enabled = true;
53          }
54          else if (newGameState == GameState.gameOver) {
55              //setup Unity scene for gameOver state
56              menuCanvas.enabled = false;
57              inGameCanvas.enabled = false;
58          }
59
60          currentGameState = newGameState;
61      }
62
```

JAKE

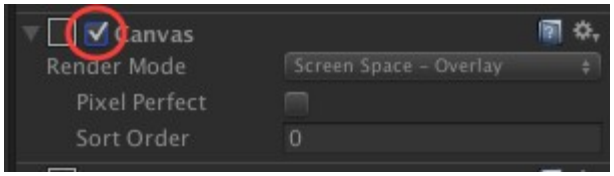
ON THE MYSTERIOUS PLANET

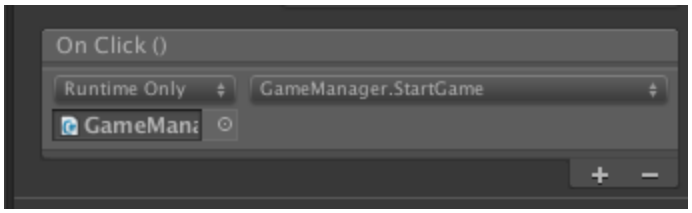


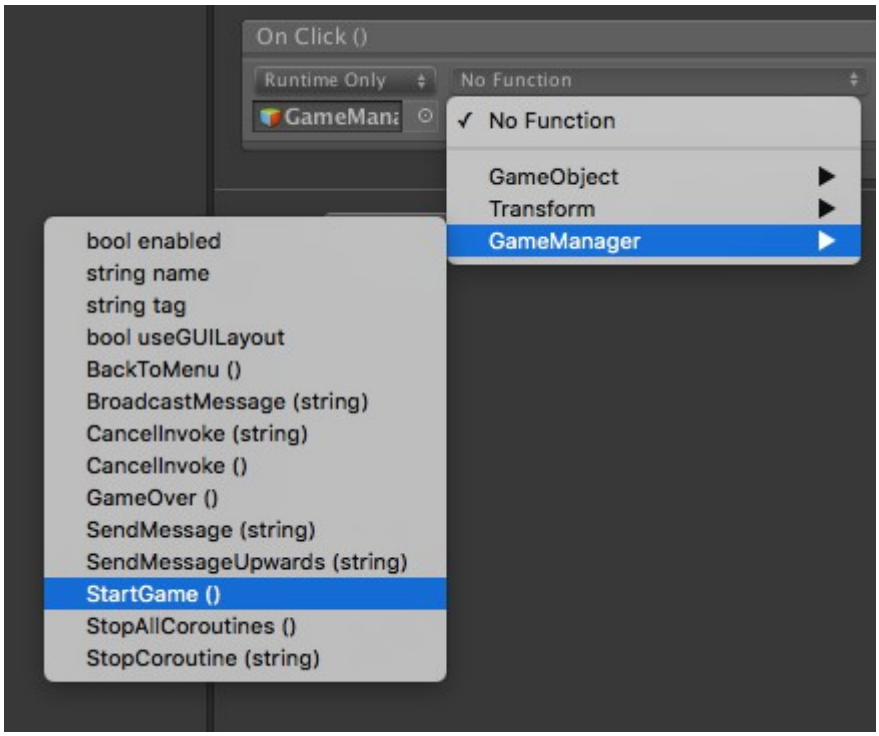
```
42      void SetGameState (GameState newGameState) {
43
44          if (newGameState == GameState.menu) {
45              //setup Unity scene for menu state
46              menuCanvas.enabled = true;
47          }
48          else if (newGameState == GameState.inGame) {
49              //setup Unity scene for inGame state
50              menuCanvas.enabled = false;
51          }
52          else if (newGameState == GameState.gameOver) {
53              //setup Unity scene for gameOver state
54              menuCanvas.enabled = false;
55          }
56
57          currentGameState = newGameState;
58      }
```

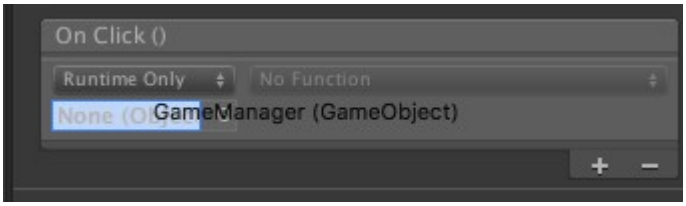
```
Console
Clear   Collapse   Clear on Play   Error Pause                    0   0   1

UnassignedReferenceException: The variable menuCanvas of GameManager has not been assigned.
You probably need to assign the menuCanvas variable of the GameManager script in the inspector.

UnassignedReferenceException: The variable menuCanvas of GameManager has not been assigned.
You probably need to assign the menuCanvas variable of the GameManager script in the inspector.
GameManager.SetGameState (GameState newGameState) (at Assets/Scripts/GameManager.cs:50)
GameManager.StartGame () (at Assets/Scripts/GameManager.cs:29)
UnityEngine.Events.InvokableCall.Invoke (System.Object[] args)
UnityEngine.Events.InvokableCallList.Invoke (System.Object[] parameters)
UnityEngine.Events.UnityEventBase.Invoke (System.Object[] parameters)
UnityEngine.Events.UnityEvent.Invoke ()
UnityEngine.UI.Button.Press () (at
/Users/builduser/buildslave/unity/build/Extensions/guisystem/UnityEngine.UI/UI/Core/Button.cs:3
5)
UnityEngine.UI.Button.OnPointerClick (UnityEngine.EventSystems.PointerEventData eventData) (at
/Users/builduser/buildslave/unity/build/Extensions/guisystem/UnityEngine.UI/UI/Core/Button.cs:4
4)
UnityEngine.EventSystems.ExecuteEvents.Execute (IPointerClickHandler handler,
UnityEngine.EventSystems.BaseEventData eventData) (at
```

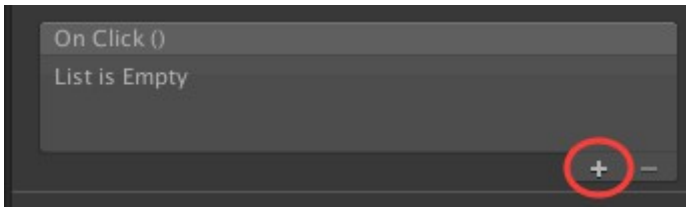
```
void SetGameState (GameState newGameState) {

    if (newGameState == GameState.menu) {
        //setup Unity scene for menu state
        menuCanvas.enabled = true;
    }
    else if (newGameState == GameState.inGame) {
        //setup Unity scene for inGame state
        menuCanvas.enabled = false;
    }
    else if (newGameState == GameState.gameOver) {
        //setup Unity scene for gameOver state
        menuCanvas.enabled = false;
    }

    currentGameState = newGameState;
}
```



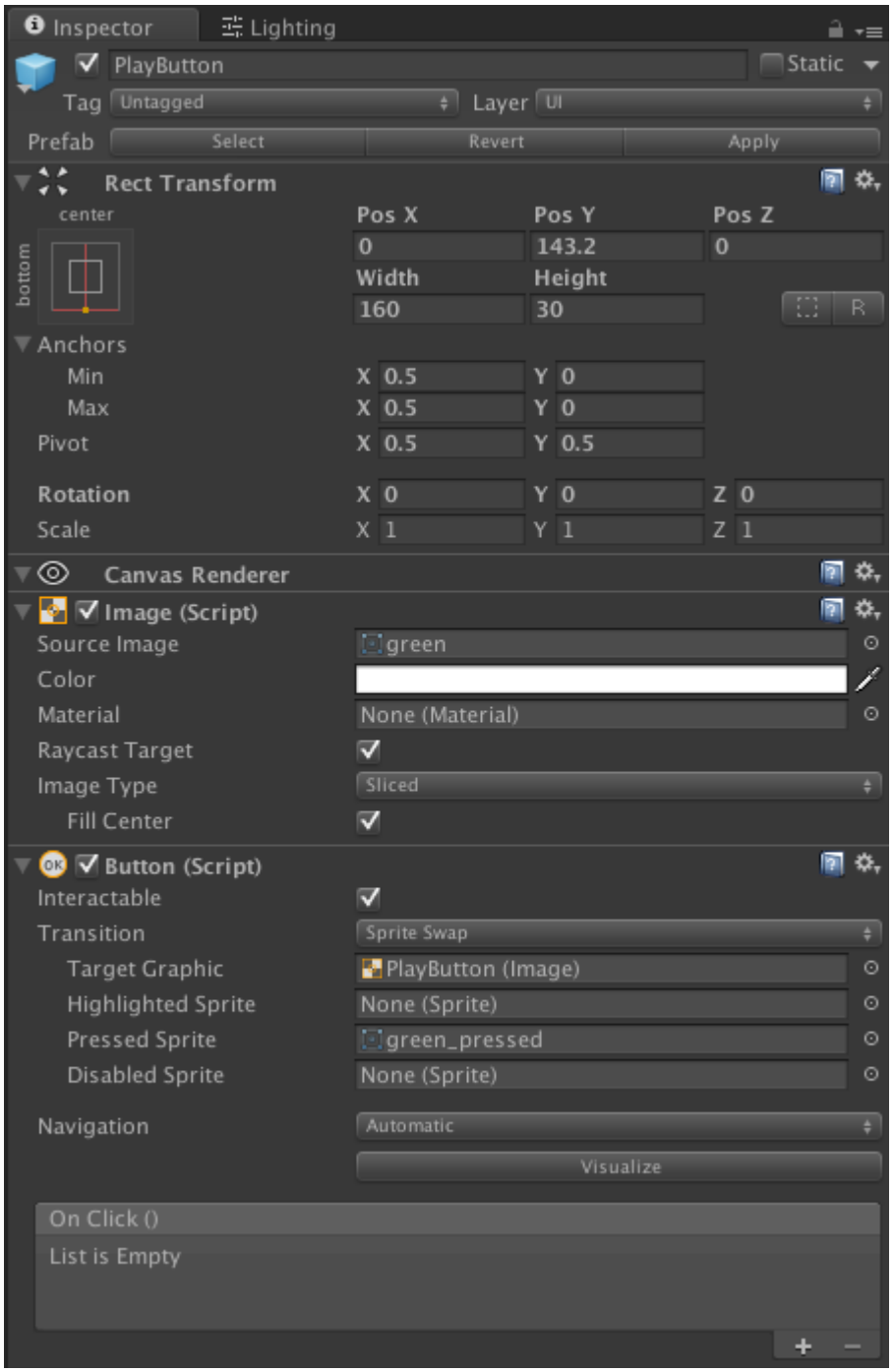



On Click ()
Runtime Only
No Function
GameManag
✓ No Function
GameObject
Transform
GameManager
bool enabled
string name
string tag
bool useGUILayout
BackToMenu ()
BroadcastMessage (string)
CancelInvoke (string)
CancelInvoke ()
GameOver ()
SendMessage (string)
SendMessageUpwards (string)
StartGame ()
StopAllCoroutines ()
StopCoroutine (string)


On Click ()
Runtime Only
No Function
GameManager (GameObject)

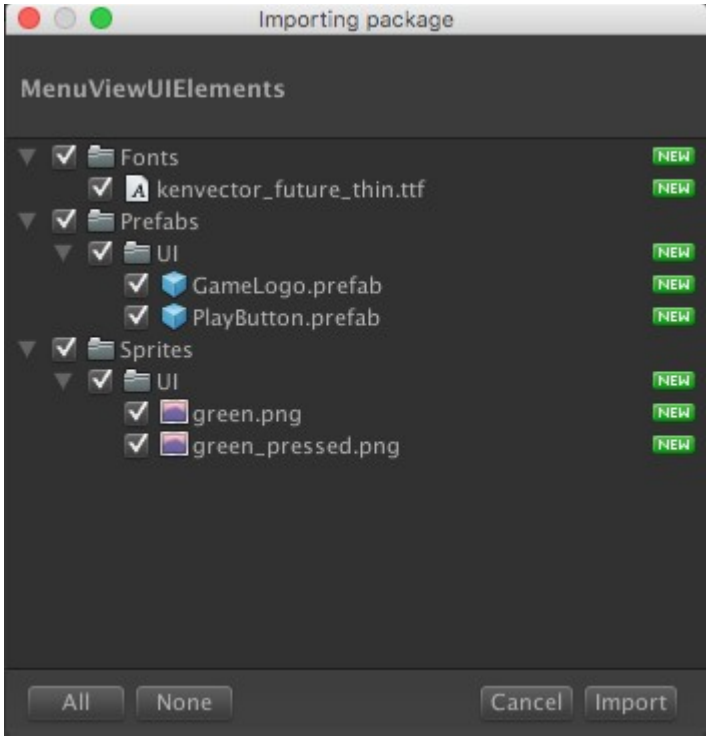

On Click ()
List is Empty

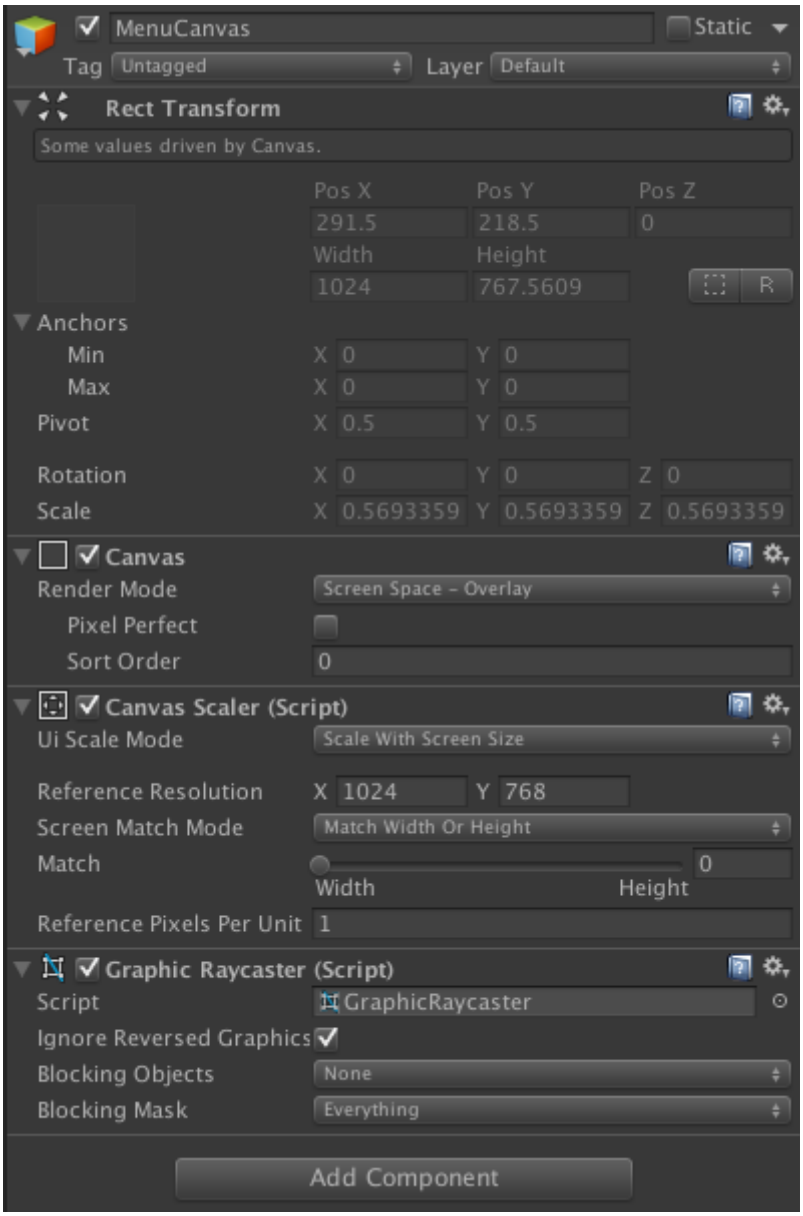

On Click ()
List is Empty

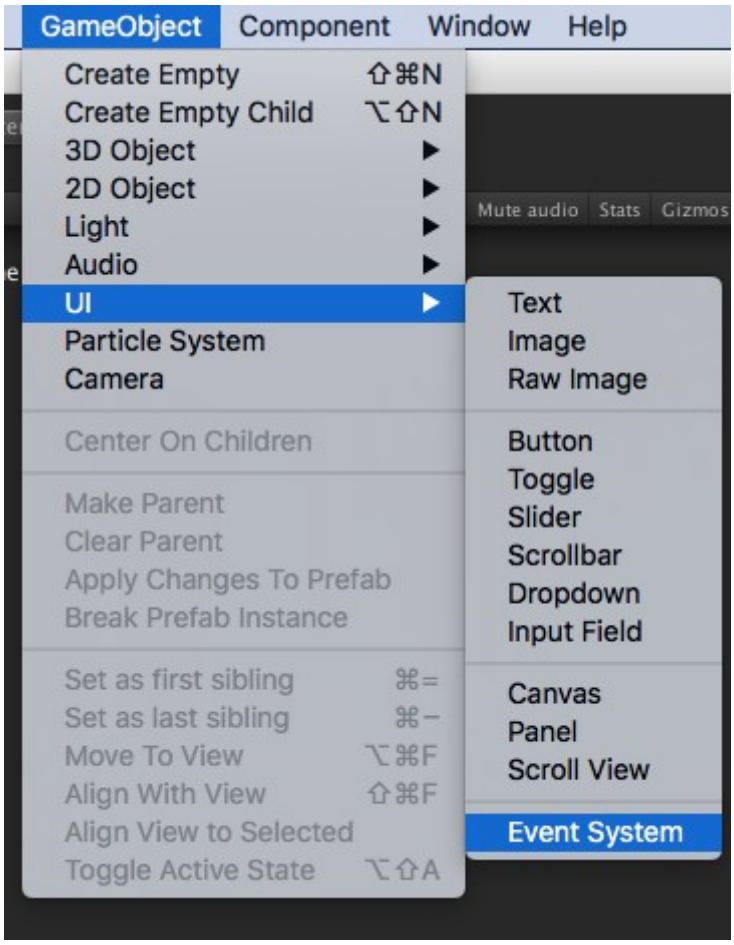

Inspector
Lighting
PlayButton
Static
Tag
Untagged
Layer
UI
Prefab
Select
Revert
Apply
Rect Transform
center
bottom
Pos X
Pos Y
Pos Z
0
143.2
0
Width
Height
160
30
R
Anchors
Min
X 0.5
Y 0
Max
X 0.5
Y 0
Pivot
X 0.5
Y 0.5
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
Canvas Renderer
Image (Script)
Source Image
green
Color
Material
None (Material)
Raycast Target
Image Type
Sliced
Fill Center
Button (Script)
Interactable
Transition
Sprite Swap
Target Graphic
PlayButton (Image)
Highlighted Sprite
None (Sprite)
Pressed Sprite
green_pressed
Disabled Sprite
None (Sprite)
Navigation
Automatic
Visualize
On Click ()
List is Empty

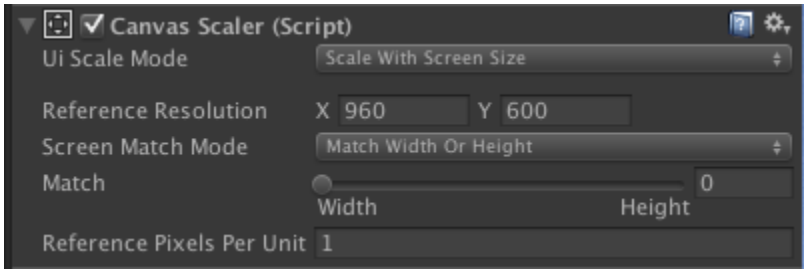

PLAY

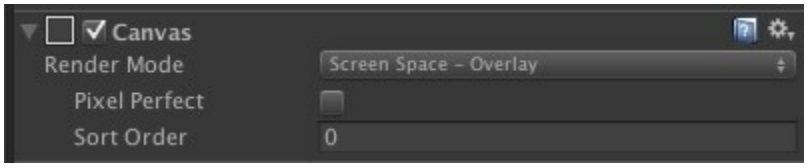

UI
MenuCanvas
PlayButton (GameObject)
EventSystem


Importing package
MenuViewUIElements
Fonts
kenvector_future_thin.ttf
Prefabs
UI
GameLogo.prefab
PlayButton.prefab
Sprites
UI
green.png
green_pressed.png
NEW
All
None
Cancel
Import


MenuCanvas
Static
Tag
Untagged
Layer
Default
Rect Transform
Some values driven by Canvas.
Pos X
Pos Y
Pos Z
291.5
218.5
0
Width
Height
1024
767.5609
R
Anchors
Min
X 0
Y 0
Max
X 0
Y 0
Pivot
X 0.5
Y 0.5
Rotation
X 0
Y 0
Z 0
Scale
X 0.5693359
Y 0.5693359
Z 0.5693359
Canvas
Render Mode
Screen Space - Overlay
Pixel Perfect
Sort Order
0
Canvas Scaler (Script)
Ui Scale Mode
Scale With Screen Size
Reference Resolution
X 1024
Y 768
Screen Match Mode
Match Width Or Height
Match
0
Width
Height
Reference Pixels Per Unit
1
Graphic Raycaster (Script)
Script
GraphicRaycaster
Ignore Reversed Graphics
Blocking Objects
None
Blocking Mask
Everything
Add Component


GameObject
Component
Window
Help
Create Empty
Create Empty Child
3D Object
2D Object
Light
Audio
UI
Particle System
Camera
Center On Children
Make Parent
Clear Parent
Apply Changes To Prefab
Break Prefab Instance
Set as first sibling
Set as last sibling
Move To View
Align With View
Align View to Selected
Toggle Active State
Mute audio
Stats
Gizmos
Text
Image
Raw Image
Button
Toggle
Slider
Scrollbar
Dropdown
Input Field
Canvas
Panel
Scroll View
Event System


Canvas Scaler (Script)
Ui Scale Mode
Scale With Screen Size
Reference Resolution
X 960
Y 600
Screen Match Mode
Match Width Or Height
Match
0
Width
Height
Reference Pixels Per Unit
1


Canvas
Render Mode
Screen Space - Overlay
Pixel Perfect
Sort Order
0


GUI Demo
Continue
New Game
Settings
Quit

Chapter 13: Collectables — What Next?

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class ViewInGame : MonoBehaviour {

    public Text scoreLabel;
    public Text coinLabel;
    public Text highscoreLabel;


    void Update() {
        if (GameManager.instance.currentGameState == GameState.inGame) {
            scoreLabel.text = PlayerController.instance.GetDistance().ToString("f0");
            coinLabel.text = GameManager.instance.collectedCoins.ToString();
            highscoreLabel.text = PlayerPrefs.GetFloat("highscore", 0).ToString("f0");
        }
    }
}

```

```
    public void Kill() {

        GameManager.instance.GameOver();
        animator.SetBool("isAlive", false);

        //check if highscore save if it is
        if (PlayerPrefs.GetFloat("highscore", 0) < this.GetDistance()) {
            //save new highscore
            PlayerPrefs.SetFloat("highscore", this.GetDistance());
        }
    }
```

Static Functions

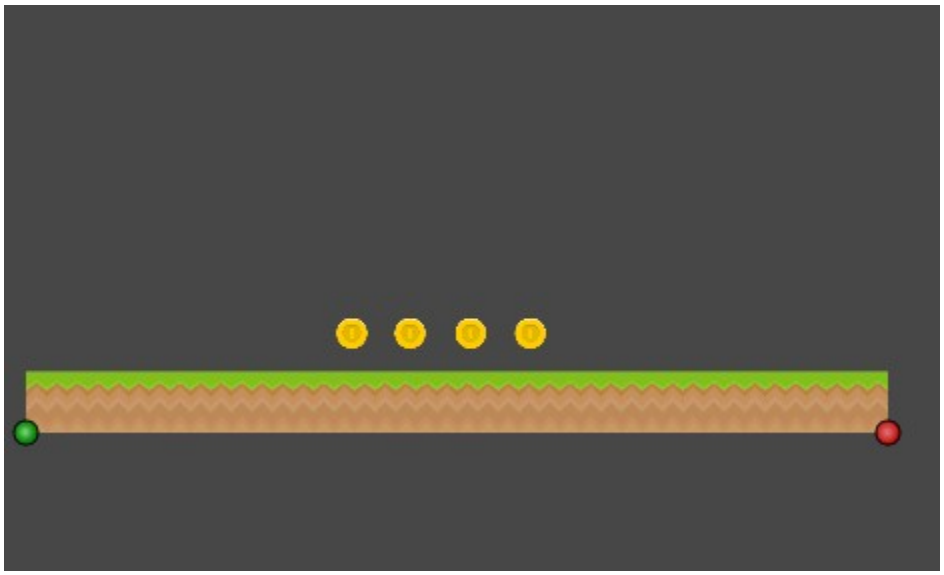
DeleteAll	Removes all keys and values from the preferences. Use with caution.
DeleteKey	Removes key and its corresponding value from the preferences.
GetFloat	Returns the value corresponding to key in the preference file if it exists.
GetInt	Returns the value corresponding to key in the preference file if it exists.
GetString	Returns the value corresponding to key in the preference file if it exists.
HasKey	Returns true if key exists in the preferences.
Save	Writes all modified preferences to disk.
SetFloat	Sets the value of the preference identified by key.
SetInt	Sets the value of the preference identified by key.
SetString	Sets the value of the preference identified by key.

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class ViewInGame : MonoBehaviour {

    public Text scoreLabel;
    public Text coinLabel;

    void Update() {
        if (GameManager.instance.currentGameState == GameState.inGame) {
            scoreLabel.text = PlayerController.instance.GetDistance().ToString("f0");
            coinLabel.text = GameManager.instance.collectedCoins.ToString();
        }
    }
}
```

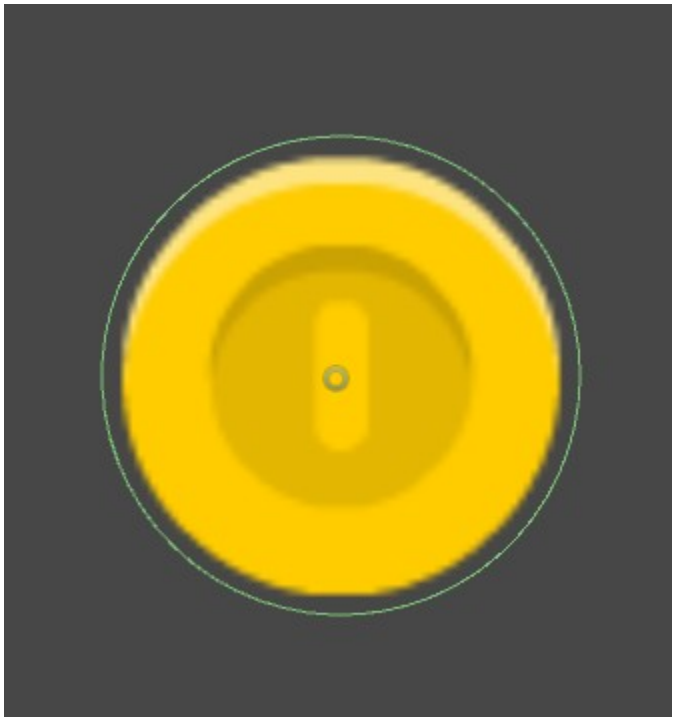
```
public float GetDistance() {

    float traveledDistance = Vector2.Distance(new Vector2(startingPosition.x, 0),
                                              new Vector2(this.transform.position.x, 0));
    return traveledDistance;
}
```

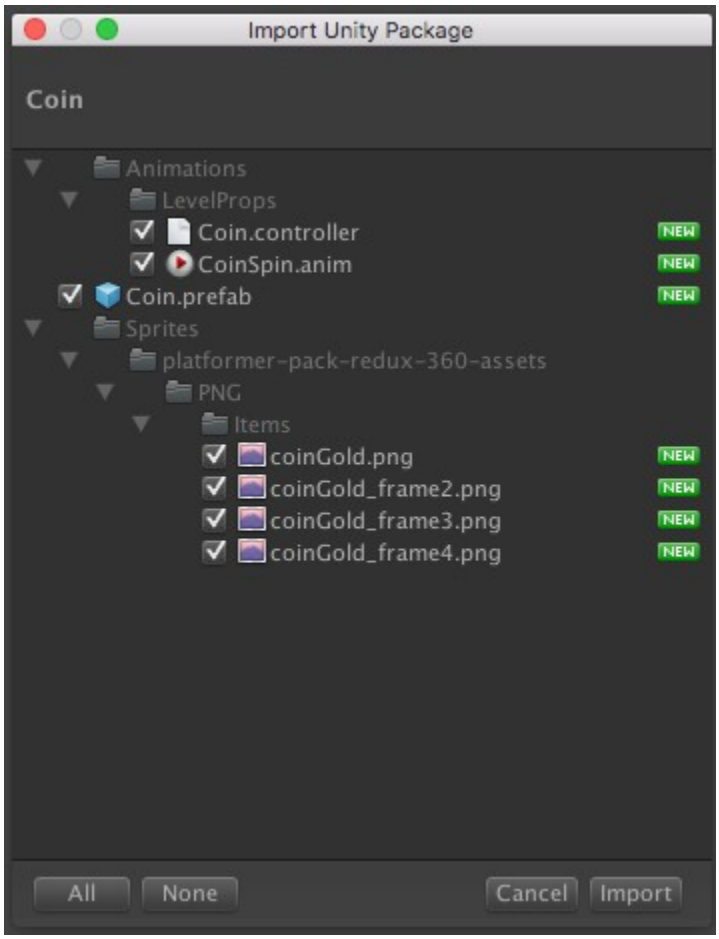

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class ViewInGame : MonoBehaviour {

    public Text coinLabel;

    void Update() {
        if (GameManager.instance.currentGameState == GameState.inGame) {
            coinLabel.text = GameManager.instance.collectedCoins.ToString();
        }
    }
}
```

```
    void Collect() {

        isCollected = true;
        Hide();
        GameManager.instance.CollectedCoin();
    }
```



```
using UnityEngine;
using System.Collections;

public class Collectable : MonoBehaviour {

    bool isCollected = false;

    void Show() {
        this.GetComponent<SpriteRenderer>().enabled = true;
        this.GetComponent<CircleCollider2D>().enabled = true;
        isCollected = false;
    }

    void Hide() {
        this.GetComponent<SpriteRenderer>().enabled = false;
        this.GetComponent<CircleCollider2D>().enabled = false;
    }


    void Collect() {

        isCollected = true;
        Hide();

    }

    void OnTriggerEnter2D(Collider2D other) {

        if (other.tag == "Player") {
            Collect();
        }
    }
}
```




Import Unity Package
Coin
Animations
LevelProps
Coin.controller
NEW
CoinSpin.anim
NEW
Coin.prefab
NEW
Sprites
platformer-pack-redux-360-assets
PNG
Items
coinGold.png
NEW
coinGold_frame2.png
NEW
coinGold_frame3.png
NEW
coinGold_frame4.png
NEW
All
None
Cancel
Import