

 무료 전자 책

배우기

unity3d

Free unaffiliated eBook created from
Stack Overflow contributors.

#unity3d

Examples.....	13
GameObject	13
GameObject	13
.....	14
MonoBehaviour GameObject	14
GameObject	14
4: MonoBehaviour	15
Examples.....	15
(override)	15
5: ScriptableObject.....	16
.....	16
AssetBundles ScriptableObjects.....	16
Examples.....	16
.....	16
ScriptableObject	16
ScriptableObject	17
ScriptableObjects PlayMode	17
ScriptableObjects	17
6: Unity Profiler.....	19
.....	19
.....	19
.....	19
iOS.....	19
Examples.....	20
.....	20
.....	20
7: Unity Git	21
Examples.....	21
Unity Git Large File Storage (LFS)	21
.....	21
Git & Git-LFS	21

1 : Git GUI	21
2 : Git & Git-LFS	21
.....	21
Unity Git	22
Unity	22
Unity	22
.....	23
.....	23
8: (VR)	24
Examples.....	24
VR	24
SDK :.....	24
:.....	24
VR	24
.....	25
9:	26
Examples.....	26
.....	26
.....	28
.....	29
10:	31
.....	31
.....	31
Examples.....	31
C #	31
.....	31
11:	33
.....	33
Unity	33
Examples.....	33
,	33
.....

.....	34
.....	36
12:	38
Examples.....	38
.....	38
.....	38
.....	38
.....	39
.....	40
.....	40
.....	41
13:	43
.....	43
.....	43
Examples.....	43
RuntimeInitializeOnLoadMethodAttribute	43
Unity C # MonoBehaviour.....	44
Unity	45
.....	47
Unitys Entity-Component	48
MonoBehaviour ScriptableObject	49
14:	53
Examples.....	53
(MVC)	53
15:	57
.....	57
Examples.....	57
Raycast.....	57
Physics2D Raycast2D.....	57
.....	58
.....

16:	60
Examples	60
.....	60
.....	60
17:	62
Examples	62
.....	62
.....	62
18:	64
.....	64
Examples	64
.....	64
TouchPhase	64
19:	66
Examples	66
.....	66
.....	66
.....	66
Rigidbody	66
.....	66
.....	66
isKinematic	66
.....	66
.....	67
.....	67
20: 3	69
.....	69
.....	69
Examples	69
.....	69

Vector3.zero Vector3.one	69
.....	70
.....	71
Vector3	72
.....	72
Vector2 Vector4	72
.....	73
Lerp LerpUnclamped.....	73
MoveTowards.....	74
SmoothDamp.....	75
21:	78
.....	78
Examples.....	78
.....	78
.....	78
22: (UI)	81
Examples.....	81
.....	81
.....	81
23:	83
Examples.....	83
.....	83
()	83
()	83
Zip	84
.....	84
24:	87
.....	87
.....	87
SerializeField	87
Examples.....	88
.....	

.....	90
.....	90
.....	91
.....	93
25: ()	97
.....	97
.....	97
Examples	97
.....	97
.....	98
26:	101
.....	101
.....	101
Examples	101
.....	101
.....	103
.....	105
.....	109
1	109
2	110
.....	111
.....	111
.....	112
.....	113
?	113
EditorWindow	113
.....	113
.....	114
.....	116
SceneView	117
27:	120

.....	120
Examples.....	120
-	120
28:	121
Examples.....	121
.....	121
.....	122
2D	123
.....	125
29:	128
Examples.....	128
GetKey, GetKeyDown GetKeyUp	128
().....	129
().....	129
().....	130
(,)	131
30:	133
Examples.....	133
.....	133
.....	133
.....	133
.....	134
.....	134
31:	136
Examples.....	136
.....	136
.unity	136
32:	137
Examples.....	137
.....	137
101.....	137
.....	137

.....	138
.....	138
33:	139
.....	139
Examples.....	139
.....	139
.....	139
.....	140
.....	140
.....	140
.....	141
.....	141
34: (IMGUI)	145
.....	145
Examples.....	145
GUILayout.....	145
35:	146
.....	146
Examples.....	146
.....	146
/	146
.....	146
.....	146
.....	146
.....	146
.....	146
.....	147
.....	147
.....	147
.....	147

.....	147
.....	148
.....	148
.....	149
.....	150
.....	150
36:	151
Examples.....	151
Colliders.....	151
.....	151
.....	151
.....	151
.....	152
.....	152
.....	152
.....	152
.....	153
.....	153
.....	154
.....	154
.....	154
.....	154
.....	154
.....	155
.....	156
.....	156
Colliders.....	157
.....	157
Collider	157
.....	157
37: API	159

.....	159
Examples.....	159
.....	159
.....	160
.....	162
.....	162
38:	163
.....	163
.....	163
.....	163
YieldInstructions	163
Examples.....	163
.....	163
.....	164
.....	165
MonoBehaviour	166
coroutines	166
.....	168
39:	170
.....	170
Examples.....	170
.....	170
.....	170
40:	171
.....	171
Examples.....	171
.....	171
.....	171
.....	171
.....	172
GameObject :.....	172

GameObject172

GameObject173

.....173

.....175

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [unity3d](#)

It is an unofficial and free unity3d ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official unity3d.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: unity3d

Unity . C # / JavaScript UnityScript . . . <https://unity3d.com/get-unity/download/archive> .

Unity 2017.1.0	2017-07-10
5.6.2	2017-06-21
5.6.1	2017-05-11
5.6.0	2017-03-31
5.5.3	2017-03-31
5.5.2	2017-02-24
5.5.1	2017-01-24
5.5	2016-11-30
5.4.3	2016-11-17
5.4.2	2016-10-21
5.4.1	2016-09-08
5.4.0	2016-07-28
5.3.6	2016-07-20
5.3.5	2016-05-20
5.3.4	2016-03-15
5.3.3	2016-02-23
5.3.2	2016-01-28
5.3.1	2015-12-18
5.3.0	2015-12-08
5.2.5	2016-06-01
5.2.4	2015-12-16
5.2.3	2015-11-19

5.2.2	2015-10-21
5.2.1	2015-09-22
5.2.0	2015-09-08
5.1.5	2015-06-07
5.1.4	2015-10-06
5.1.3	2015-08-24
5.1.2	2015-07-16
5.1.1	2015-06-18
5.1.0	2015-06-09
5.0.4	2015-07-06
5.0.3	2015-06-09
5.0.2	2015-05-13
5.0.1	2015-04-01
5.0.0	2015-03-03
4.7.2	2016-05-31
4.7.1	2016-02-25
4.7.0	2015-12-17
4.6.9	2015-10-15
4.6.8	2015-08-26
4.6.7	2015-07-01
4.6.6	2015-06-08
4.6.5	2015-04-30
4.6.4	2015-03-26
4.6.3	2015-02-19
4.6.2	2015-01-29
4.6.1	2014-12-09

4.6.0	2014-11-25
4.5.5	2014-10-13
4.5.4	2014-09-11
4.5.3	2014-08-12
4.5.2	2014-07-10
4.5.1	2014-06-12
4.5.0	2014-05-27
4.3.4	2014-01-29
4.3.3	2014-01-13
4.3.2	2013-12-18
4.3.1	2013-11-28
4.3.0	2013-11-12
4.2.2	2013-10-10
4.2.1	2013-09-05
4.2.0	2013-07-22
4.1.5	2013-06-08
4.1.4	2013-06-06
4.1.3	2013-05-23
4.1.2	2013-03-26
4.1.0	2013-03-13
4.0.1	2013-01-12
4.0.0	2012-11-13
3.5.7	2012-12-14
3.5.6	2012-09-27
3.5.5	2012-08-08
3.5.4	2012-07-20

3.5.3	2012-06-30
3.5.2	2012-05-15
3.5.1	2012-04-12
3.5.0	2012-02-14
3.4.2	2011-10-26
3.4.1	2011-09-20
3.4.0	2011 7 26

Examples

Unity Windows Mac . .

Unity 4 .

1. - ()
2. - 35 ()
3. - 1 125 USD - 24 .
4. - [Unity](#) .

EULA , 10 **Unity Plus** () . 200,000 **Unity Pro** (Enterprise) .

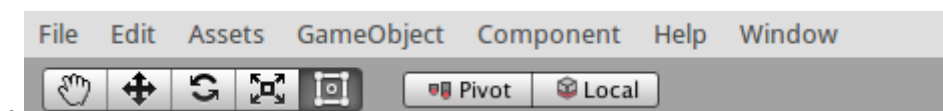
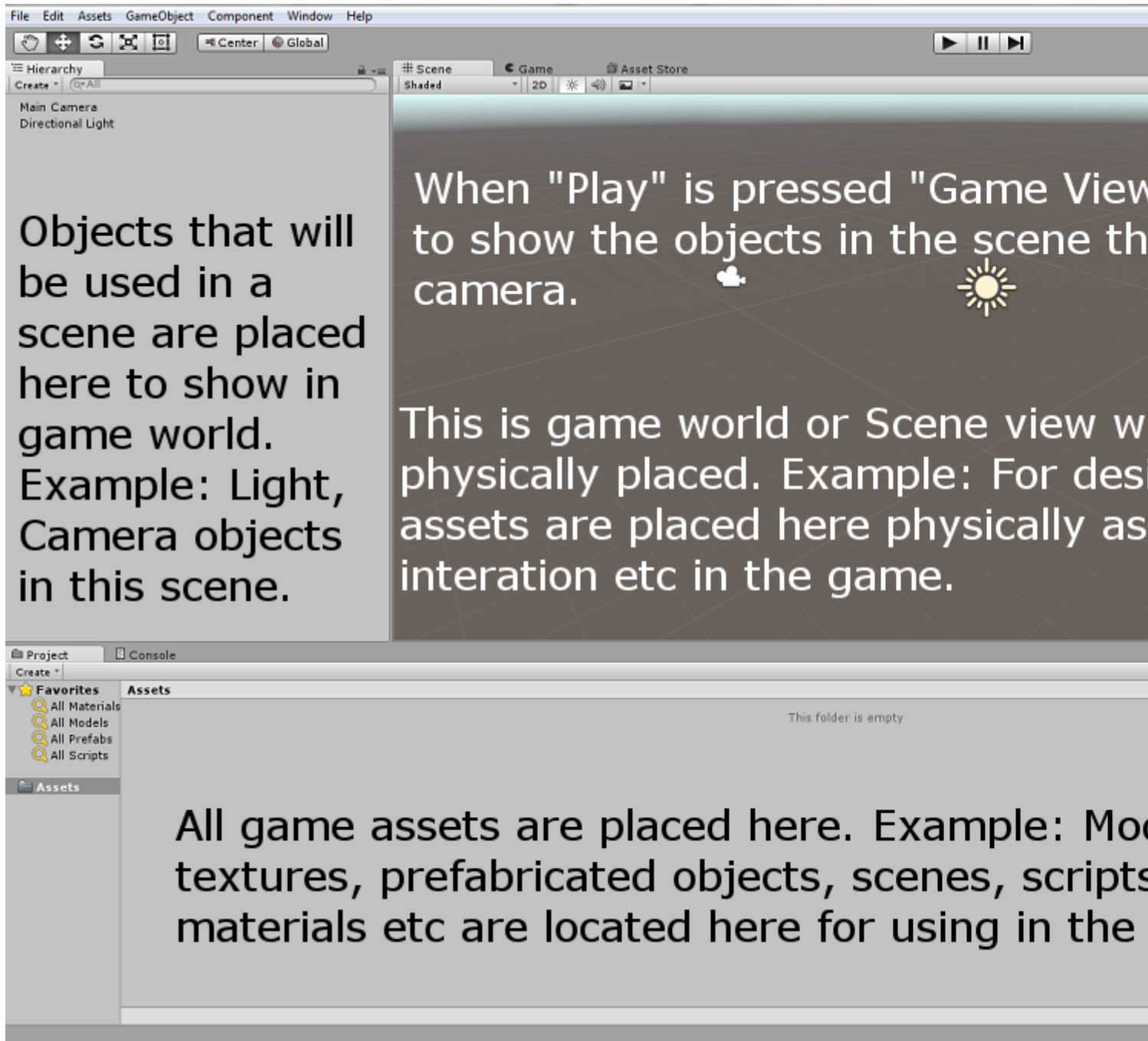
1. [Unity](#) .
2. , MonoDevelop IDE, .

Unity Unity [Unity 5.5.1](#) .

Unity

Unity . :

- Windows Unity 5.3.1f1 .
- Mac /Applications/Unity . (: /Applications/Unity5.3.1f1) .
- Unity Alt . () .



Hierarchy Create Empty GameObject Create Empty . Project Create > C# Script Create . .

Hierarchy GameObject Inspector . Hierarchy . MonoDevelop IDE .

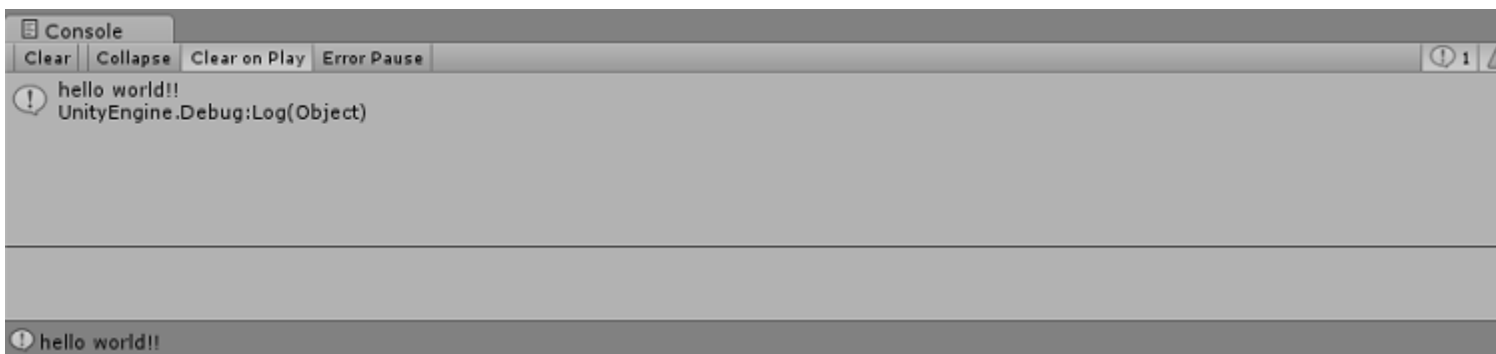
```
Debug.Log("hello world!!"); Debug.Log("hello world!!"); .
```

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

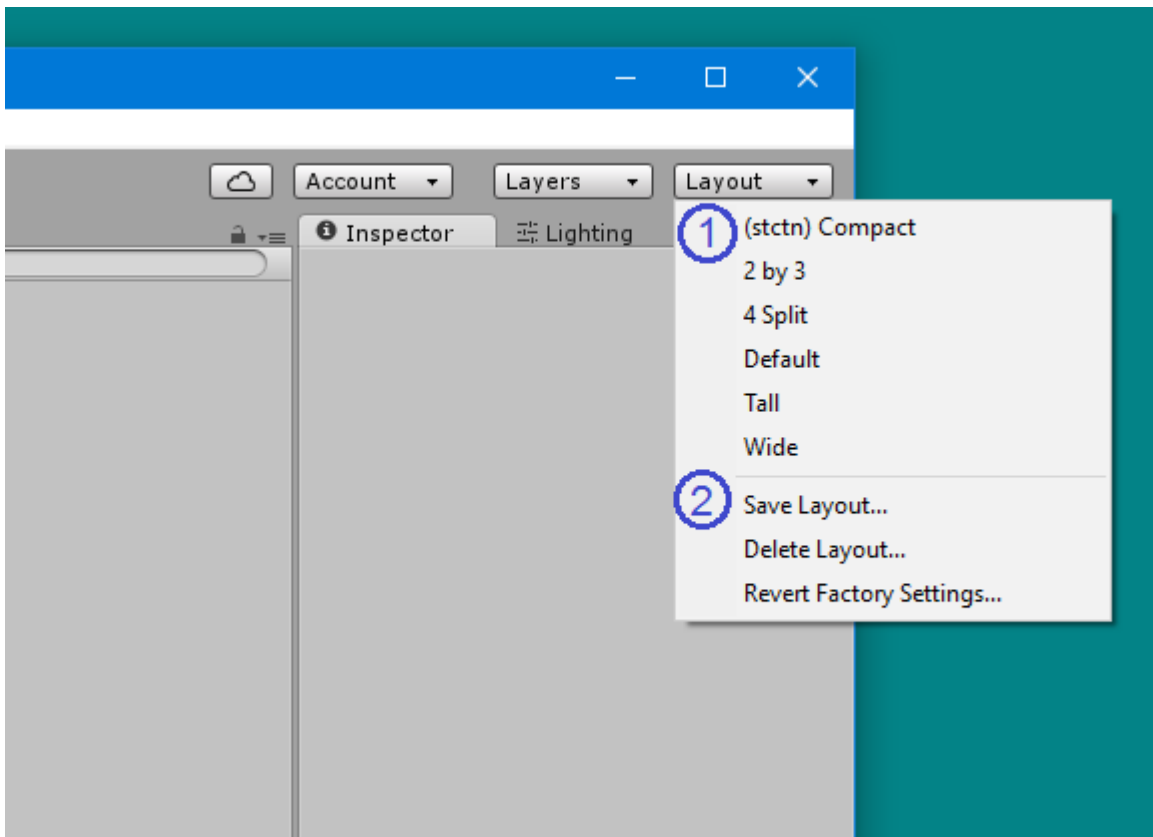
```
public class BasicCode : MonoBehaviour {  
  
    // Use this for initialization  
    void Start () {  
        Debug.Log("hello world!!");  
    }  
  
    // Update is called once per frame  
    void Update () {  
  
    }  
  
}
```

Debug.Log("hello world!!"); Debug.Log("hello world!!"); void Start() . . . **Play** .

Console .

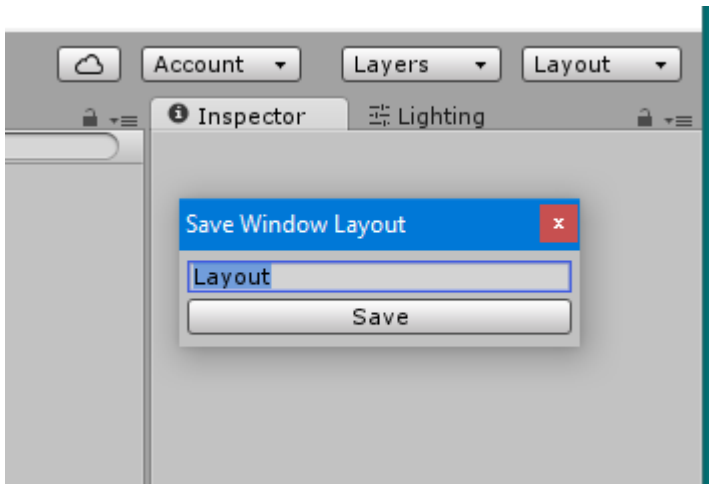


Unity Editor :

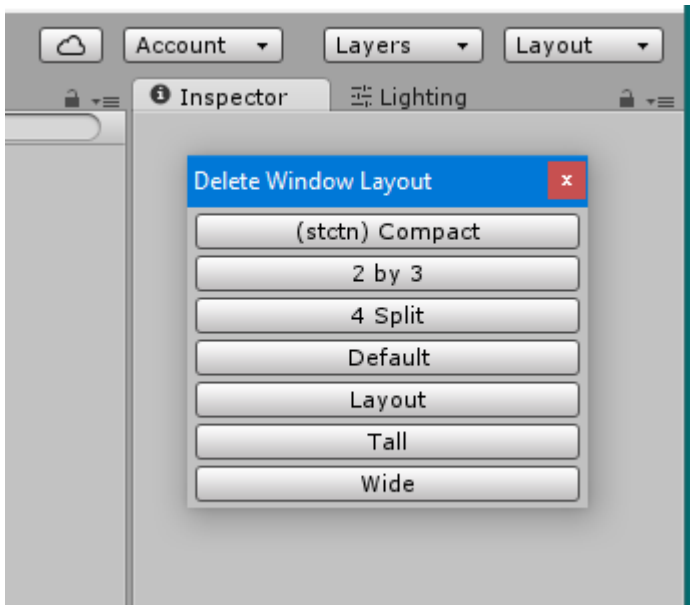


Unity 5 (2 x 3, 4 Split, Default, Tall, Wide) (1) .

(2) " ..."

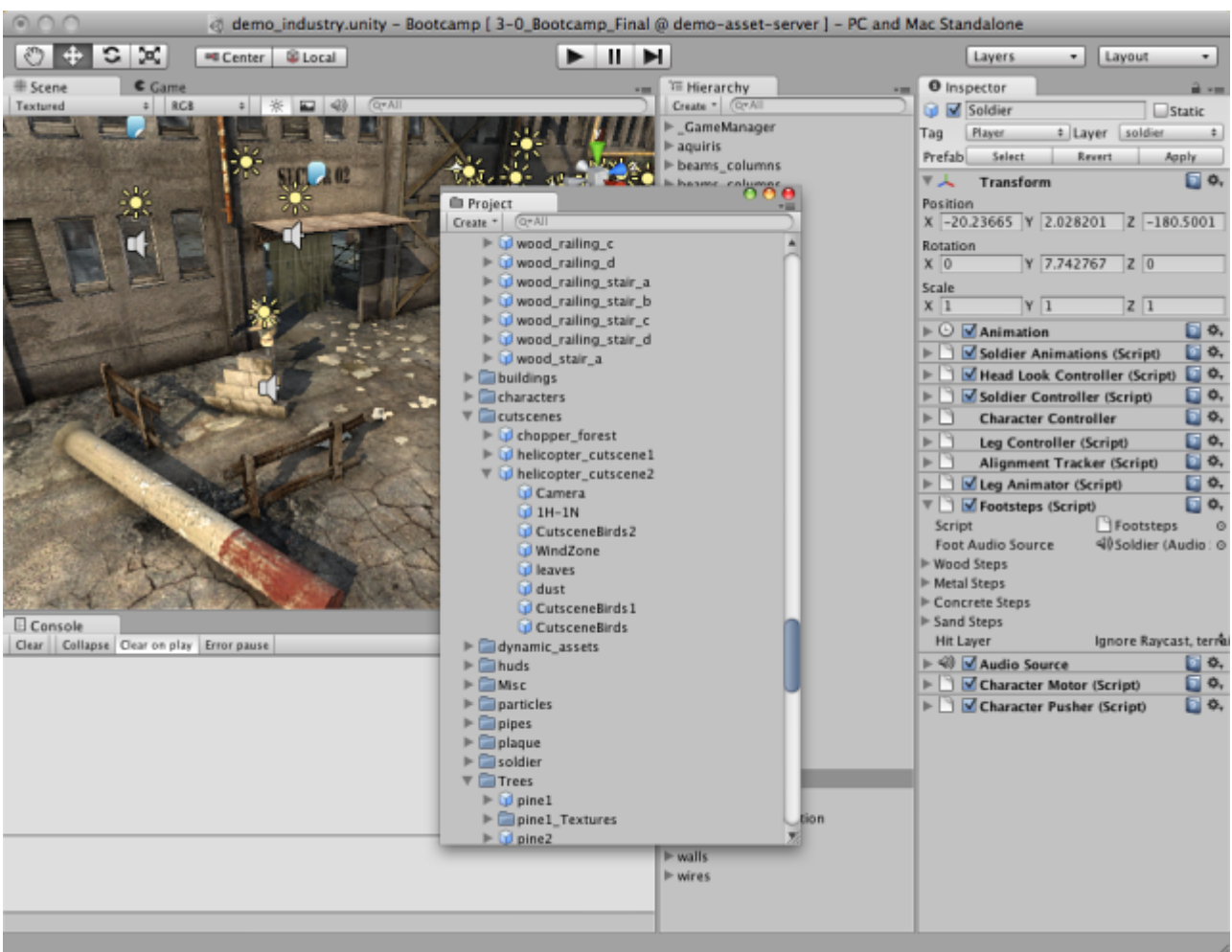
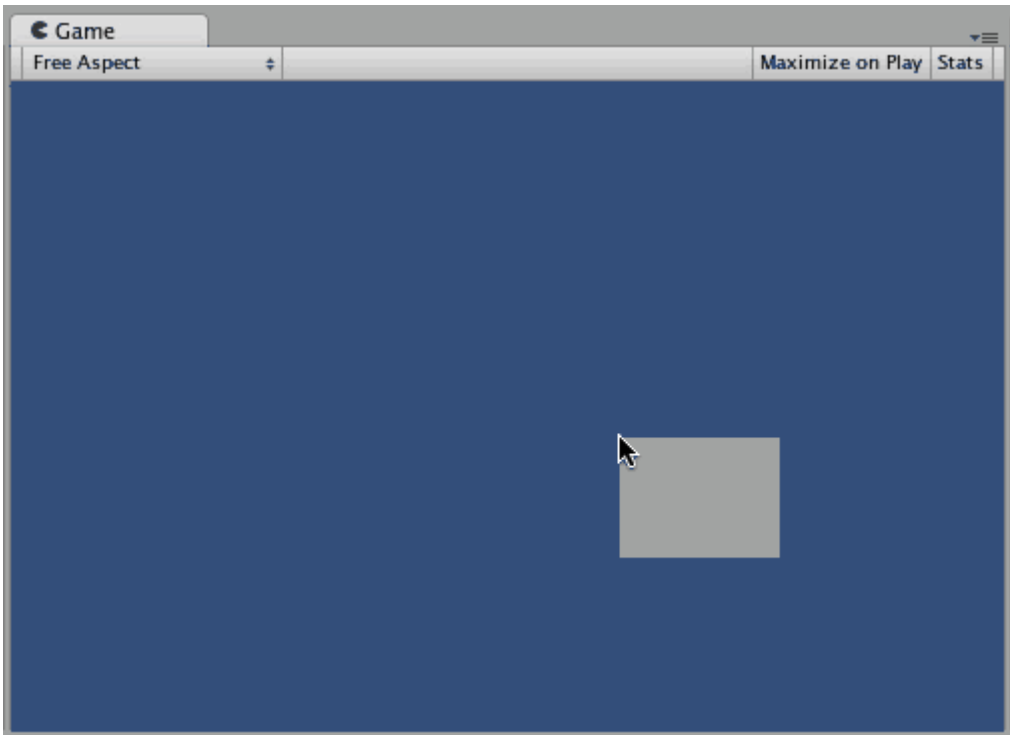


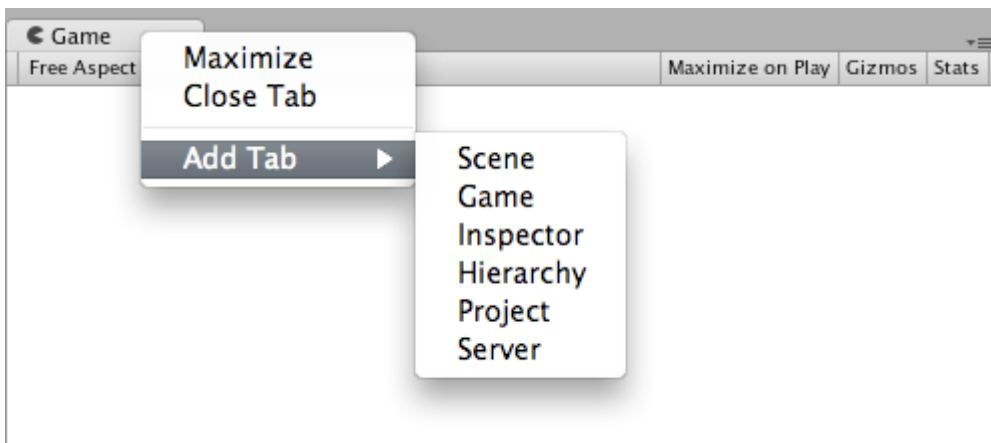
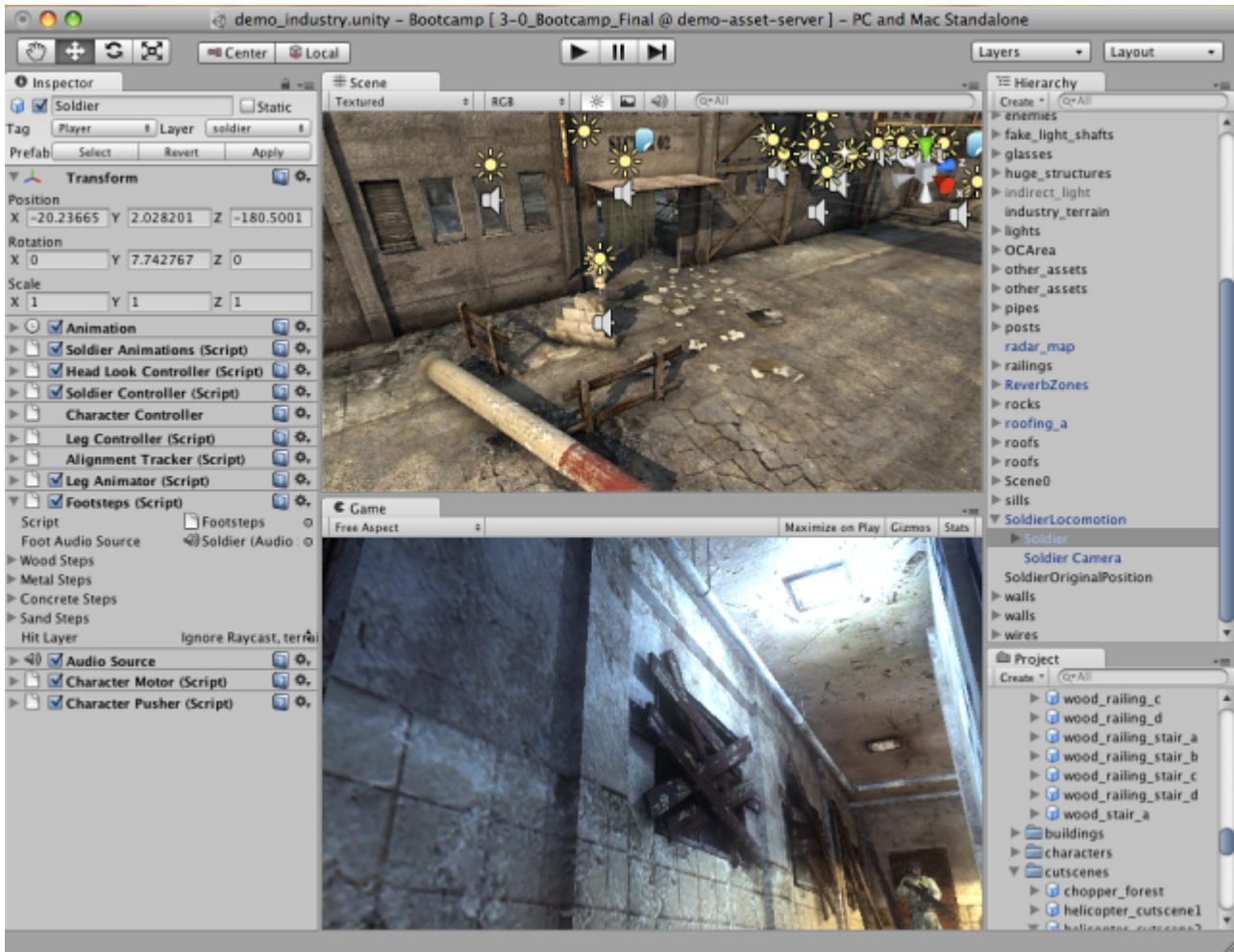
(2) " ..."



"Revert Factory Settings ..." (2) .

. Dock .





unity3d : <https://riptutorial.com/ko/unity3d/topic/846/unity3d->

2: Android Plugins 101 -

Unity Android . OS .

Android

Unity Android .

1. Java Android C# Java .
2. Android OS C#

Unity .

- [AndroidJavaObject](#) - . and . AndroidJavaObject
- [AndroidJavaClass](#) - [AndroidJavaObject](#) . .
- [Get / Set](#) [GetStatic / SetStatic](#)
- [Call / CallStatic](#)

-
1. [Android Studio](#) Java
 2. JAR / AAR ([JAR](#) [AAR](#))
 3. JAR / AAR **Assets / Plugins / Android** Unity .
 4. Unity (C#) .

!

JAR / AAR C# C# .

. 1 ., ?

, ?

1. - 1
2. Android ? - 2

""(, 1 2) ! .

Examples

UnityAndroidPlugin.cs

Unity C

```
using UnityEngine;
using System.Collections;

public static class UnityAndroidPlugin {

}
```

UnityAndroidNative.java

Android Studio Java .

```
package com.axs.unityandroidplugin;
import android.util.Log;
import android.widget.Toast;
import android.app.ActivityManager;
import android.content.Context;

public class UnityAndroidNative {

}
```

UnityAndroidPluginGUI.cs

Unity C # .

```
using UnityEngine;
using System.Collections;

public class UnityAndroidPluginGUI : MonoBehaviour {

    void OnGUI () {

    }

}
```

Android Plugins 101 - : <https://riptutorial.com/ko/unity3d/topic/10032/android-plugins-101--->

3: GameObjects

- `GameObject ()`;
- `GameObject FindGameObjectWithTag ()`;
- `GameObject [] FindGameObjectsWithTag ()`;
- `public static Object FindObjectOfType ()`;
- `public static Object [] FindObjectsOfType ()`;

?

GameObjects . : FindObjectOfType Update, FixedUpdate .

- `FindObjectOfType Find`
- `FindGameObjectWithTag . Unity .`
- `""GameObjects (: UI) GameObject`
- `GameObject`
- `GameObject Object Pooling`
- .

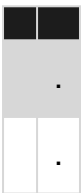
Unity .

- `FindObjectsOfType() static . .`
- `Dictionary .`

Examples

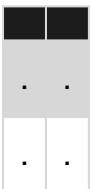
GameObject

```
var go = GameObject.Find("NameOfTheObject");
```



GameObject

```
var go = GameObject.FindGameObjectWithTag("Player");
```



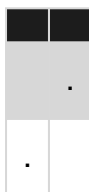
```
[SerializeField]
GameObject[] gameObjects;
```



MonoBehaviour GameObject

```
ExampleScript script = GameObject.FindObjectOfType<ExampleScript>();
GameObject go = script.gameObject;
```

```
FindObjectOfType() null .
```



GameObject

```
Transform tr = GetComponent<Transform>().Find("NameOfTheObject");
GameObject go = tr.gameObject;
```

```
Find null .
```



GameObjects : <https://riptutorial.com/ko/unity3d/topic/3793/gameobjects--->

4: MonoBehaviour

Examples

(override)

Awake , Start , Update .

. . .

. Awake , Start , OnEnable , OnDisable , Update , LateUpdate FixedUpdate . , (C++) (C#)

.

. . . Unity .

Unity : 10000 Update () IL2CPP . IL2CPP

MonoBehaviour : <https://riptutorial.com/ko/unity3d/topic/2304/monobehaviour-->

5: ScriptableObject

AssetBundles ScriptableObjects

ScriptableObject AssetBundle ScriptableObjects Unity .

AssetBundle GameObject ScriptableObject . [Dependency Injection](#) .

Examples

ScriptableObjects MonoBehaviours . . ScriptableObject MonoBehaviours
ScriptableObject .

.

ScriptableObject

ScriptableObject .

```
using UnityEngine;

[CreateAssetMenu(menuName = "StackOverflow/Examples/MyScriptableObject")]
public class MyScriptableObject : ScriptableObject
{
    [SerializeField]
    int mySerializedNumber;

    int helloWorldCount = 0;

    public void HelloWorld()
    {
        helloWorldCount++;
        Debug.LogFormat("Hello! My number is {0}.", mySerializedNumber);
        Debug.LogFormat("I have been called {0} times.", helloWorldCount);
    }
}
```

CreateAssetMenu Unity **Assets / Create** . **Assets / Create / StackOverflow / Examples** .

ScriptableObject Inspector ScriptableObject .

```
using UnityEngine;

public class SampleScript : MonoBehaviour {

    [SerializeField]
    MyScriptableObject myScriptableObject;

    void OnEnable ()
```

```

    {
        myScriptableObject.HelloWorld();
    }
}

```

ScriptableObject

ScriptableObject.CreateInstance<T>() **ScriptableObject** .

```
T obj = ScriptableObject.CreateInstance<T>();
```

T ScriptableObject .

```
, ScriptableObject .new ScriptableObject() .
```

ScriptableObject

ScriptableObjects PlayMode .

ScriptableObject .

public **SerializeField** SerializeField . MonoBehaviours

```

public class MyScriptableObject : ScriptableObject
{
    [SerializeField]
    int mySerializedValue;

    public int MySerializedValue
    {
        get { return mySerializedValue; }
    }
}

```

Publicable ScriptableObject .

```

public class MyScriptableObject : ScriptableObject
{
    // Private fields are not serialized and will reset to default on reset
    private int mySerializedValue;

    public int MySerializedValue
    {
        get { return mySerializedValue; }
        set { mySerializedValue = value; }
    }
}

```

ScriptableObjects

ScriptableObjects Resources.FindObjectsOfTypeAll() .

```
T[] instances = Resources.FindObjectsOfTypeAll<T>();
```

T **ScriptableObject** . .

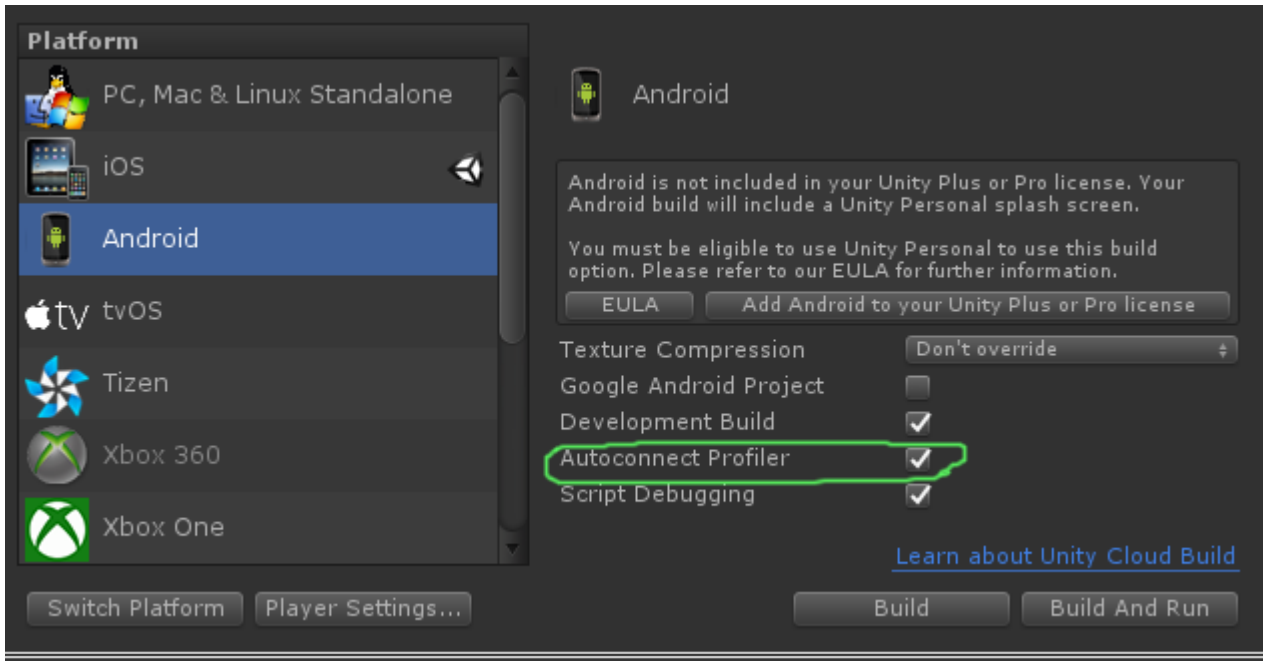
. **ScriptableObjects** .

: .OnEnable() OnEnable() .

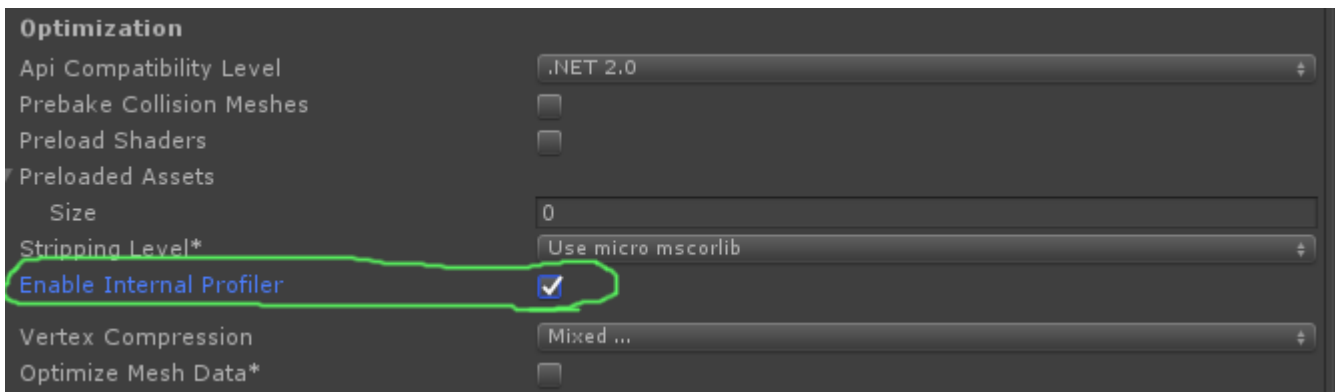
ScriptableObject : <https://riptutorial.com/ko/unity3d/topic/3434/scriptableObject>

6: Unity Profiler

Build **Autopconnect Profiler** Build Settings "Build and Run" .



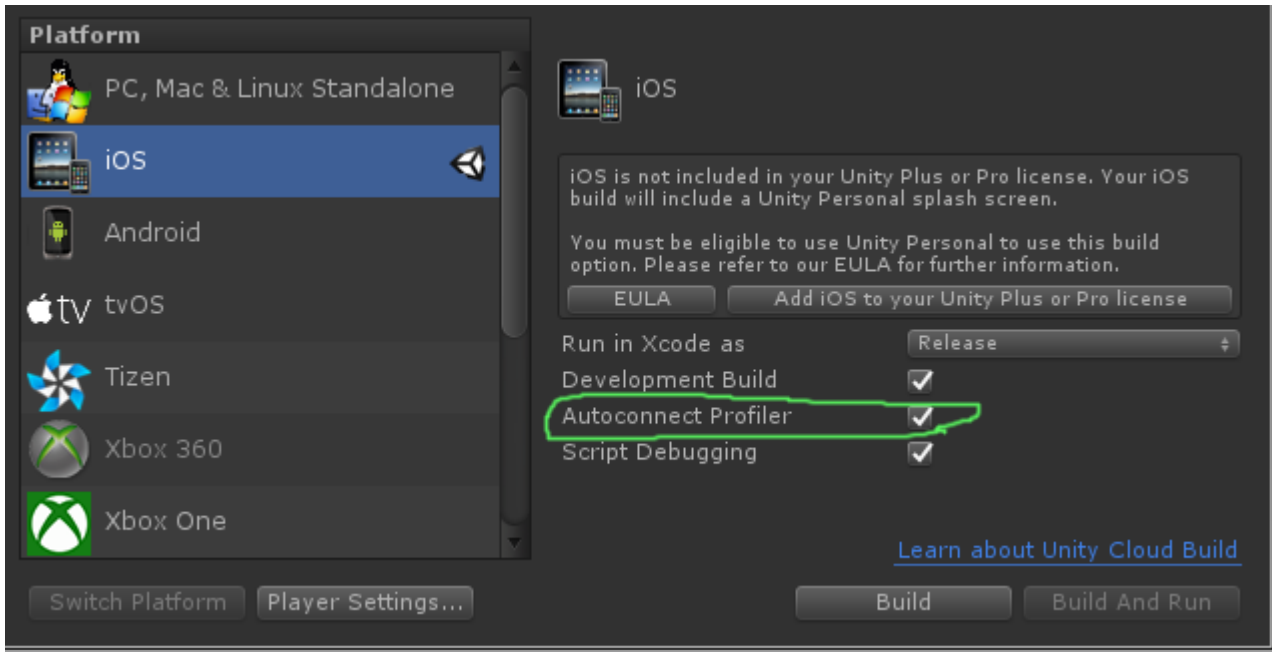
Other Settings [Android Player](#) LogCat .



" " LogCat "" Android .

iOS

" " .



iOS . . .

Examples

Profiler Window Profiler.BeginSample Profiler.EndSample .

ConditionalAttribute .

```
public class SomeClass : MonoBehaviour
{
    void SomeFunction()
    {
        Profiler.BeginSample("SomeClass.SomeFunction");
        // Various call made here
        Profiler.EndSample();
    }
}
```

"SomeClass.SomeFunction" .

Unity Profiler : <https://riptutorial.com/ko/unity3d/topic/6974/unity-profiler>

7: Unity Git

Examples

Unity Git Large File Storage (LFS)

Git . (> 5 MB) . Git .

2015 GitHub [Git LFS](#) . !

, Git . Git .

Git & Git-LFS

. Git Git GUI .

1 : Git GUI

Git GUI GUI

- [Sourcetree](#) ()
- [Github](#) ()
- [SmartGit](#) ()

google Git-LFS

2 : Git & Git-LFS

- [Install Git](#) .. [Git LFS](#) .

Git LFS Git LFS . Unity .gitattributes :

```
# Image formats:
*.tga filter=lfs diff=lfs merge=lfs -text
*.png filter=lfs diff=lfs merge=lfs -text
*.tif filter=lfs diff=lfs merge=lfs -text
*.jpg filter=lfs diff=lfs merge=lfs -text
*.gif filter=lfs diff=lfs merge=lfs -text
*.psd filter=lfs diff=lfs merge=lfs -text

# Audio formats:
*.mp3 filter=lfs diff=lfs merge=lfs -text
*.wav filter=lfs diff=lfs merge=lfs -text
*.aiff filter=lfs diff=lfs merge=lfs -text
```

```
# 3D model formats:
*.fbx filter=lfs diff=lfs merge=lfs -text
*.obj filter=lfs diff=lfs merge=lfs -text

# Unity formats:
*.sbsar filter=lfs diff=lfs merge=lfs -text
*.unity filter=lfs diff=lfs merge=lfs -text

# Other binary formats
*.dll filter=lfs diff=lfs merge=lfs -text
```

Unity Git

Unity Git .

Unity .

. .gitignore . [GitHub](#) [Unity.gitignore](#) [gitignore.io](#) .

```
# Unity Generated
[Tt]emp/
[Ll]ibrary/
[Oo]bj/

# Unity3D Generated File On Crash Reports
sysinfo.txt

# Visual Studio / MonoDevelop Generated
ExportedObj/
obj/
*.csproj
*.unityproj
*.sln
*.suo
*.tmp
*.user
*.userprefs
*.pidb
*.booproj
*.svd

# OS Generated
desktop.ini
.DS_Store
.DS_Store?
.Spotlight-V100
.Trashes
ehthumbs.db
Thumbs.db
```

.gitignore .

Unity

Unity .

1. (v4.5) Unity → Preferences → Packages → Repository External .
2. Edit → Project Settings → Editor → Version Control Mode Visible Meta Files Edit → Project Settings → Editor → Version Control Mode .
3. Edit → Project Settings → Editor → Asset Serialization Mode Force Text Edit → Project Settings → Editor → Asset Serialization Mode .
4. File .

Git Unity Git . Unity *.meta , Git .

Git post-merge Unity /.git/hooks/ . Git / , .

Unity 2 Unity Scene (*.unity) . . Unity SmartMerge . .git .gitconfig
(Windows : %USERPROFILE%\ .gitconfig , Linux / Mac OS X : ~/.gitconfig).

```
[merge]
tool = unityyamlmerge

[mergetool "unityyamlmerge"]
trustExitCode = false
cmd = '<path to UnityYAMLMerge>' merge -p "$BASE" "$REMOTE" "$LOCAL" "$MERGED"
```

Windows UnityYAMLMerge .

```
C:\Program Files\Unity\Editor\Data\Tools\UnityYAMLMerge.exe
```

```
C:\Program Files (x86)\Unity\Editor\Data\Tools\UnityYAMLMerge.exe
```

MacOSX :

```
/Applications/Unity/Unity.app/Contents/Tools/UnityYAMLMerge
```

/ .git mergetool UnityYAMLMerge .

Unity Git : <https://riptutorial.com/ko/unity3d/topic/2195/unity--git-->

8: (VR)

Examples

VR

VR . **Google Cardboard** , **Samsung GearVR** , **HTC Vive**, **Oculus**, **PS VR** PC .

Unity **Oculus Rift** , **Google Carboard** , **Steam VR** , **Playstation VR** , **Gear VR** **Microsoft Hololens** .

SDK . , sdk .

SDK :

- [Google Cardboard](#)
-
- [Samsung GearVR \(Unity 5.3 \)](#)
-
- [HTC Vive / Open VR](#)
- [Microsoft Hololens](#)

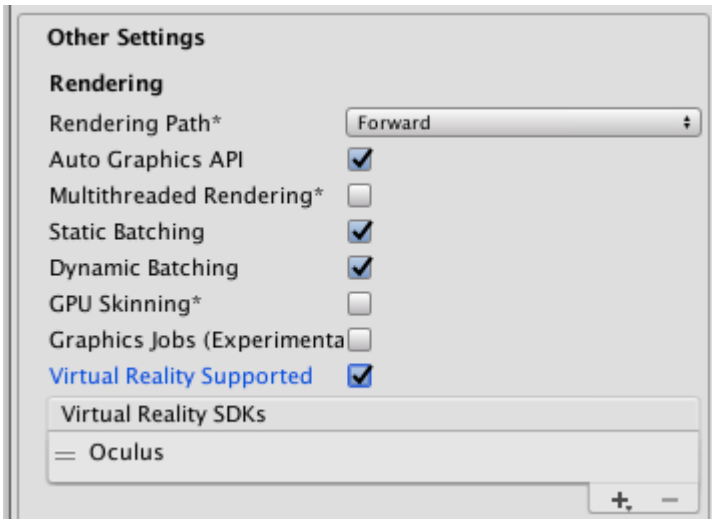
:

- [Google Cardboard / Daydream](#)
- [GearVR](#)
-
- [HTC Vive](#)
- [Microsoft Hololens](#)

VR

(> >) .

.



Virtual Reality SDK VR .

VR . .

1. 3 ()

2. 6 ()

3 DOF HMD (Head-Mounted Display) HMD 3 (,) 3 . yaw . HMD X, Y, Z ().

6 DOF , X, Y Z , 6 .

6 DOF Room-scale VR RAM 6ft x 6ft . 3DOF (200) .

.

- (6 DOF)
- [HTC Vive](#) (6 DOF)
- (3 DOF)
- [VR Oculus](#) (3 DOF)
- [Google Cardboard](#) (3 DOF)

(VR) : <https://riptutorial.com/ko/unity3d/topic/5787/---vr->

9:

Examples

. (prefab) (instantfab) (instantiate / destroy). .

. () .

.

```
public class ObjectPool : MonoBehaviour
{
    public GameObject prefab;
    public int amount = 0;
    public bool populateOnStart = true;
    public bool growOverAmount = true;

    private List<GameObject> pool = new List<GameObject>();

    void Start()
    {
        if (populateOnStart && prefab != null && amount > 0)
        {
            for (int i = 0; i < amount; i++)
            {
                var instance = Instantiate(Prefab);
                instance.SetActive(false);
                pool.Add(instance);
            }
        }
    }

    public GameObject Instantiate (Vector3 position, Quaternion rotation)
    {
        foreach (var item in pool)
        {
            if (!item.activeInHierarchy)
            {
                item.transform.position = position;
                item.transform.rotation = rotation;
                item.SetActive( true );
                return item;
            }
        }

        if (growOverAmount)
        {
            var instance = (GameObject)Instantiate(prefab, position, rotation);
            pool.Add(instance);
            return instance;
        }

        return null;
    }
}
```

```

public GameObject prefab;
public int amount = 0;
public bool populateOnStart = true;
public bool growOverAmount = true;

private List<GameObject> pool = new List<GameObject>();

```

- GameObject prefab: .
- int amount: .
- bool populateOnStart: . prefab Instantiate
- bool growOverAmount: true .
- List<GameObject> pool: / .

Start .

```

void Start()
{
    if (populateOnStart && prefab != null && amount > 0)
    {
        for (int i = 0; i < amount; i++)
        {
            var instance = Instantiate(Prefab);
            instance.SetActive(false);
            pool.Add(instance);
        }
    }
}

```

, prefab 0 ().

. . .

Instantiate . .

```

public GameObject Instantiate (Vector3 position, Quaternion rotation)
{
    foreach (var item in pool)
    {
        if (!item.activeInHierarchy)
        {
            item.transform.position = position;
            item.transform.rotation = rotation;
            item.SetActive(true);
            return item;
        }
    }

    if (growOverAmount)
    {
        var instance = (GameObject)Instantiate(prefab, position, rotation);
        pool.Add(instance);
        return instance;
    }
}

```



```

    return null;
}

```

Instantiate Instantiate , .

Instantiate . . ().

. : prefab . .

"" . NullReferenceExceptions null GameObject .

!

. .

. create function Func Action . .

```

public class ResourcePool<T> where T : class
{
    private readonly List<T> objectPool = new List<T>();
    private readonly Action<T> cleanUpAction;
    private readonly Func<T> createAction;

    public ResourcePool(Action<T> cleanUpAction, Func<T> createAction)
    {
        this.cleanUpAction = cleanUpAction;
        this.createAction = createAction;
    }

    public void Return(T resource)
    {
        this.objectPool.Add(resource);
    }

    private void PurgeSingleResource()
    {
        var resource = this.Rent();
        this.cleanUpAction(resource);
    }

    public void TrimResourcesBy(int count)
    {
        count = Math.Min(count, this.objectPool.Count);
        for (int i = 0; i < count; i++)
        {
            this.PurgeSingleResource();
        }
    }

    public T Rent()
    {
        int count = this.objectPool.Count;
        if (count == 0)
        {
            Debug.Log("Creating new object.");
            return this.createAction();
        }
        else
    }
}

```

```

        {
            Debug.Log("Retrieving existing object.");
            T resource = this.objectPool[count-1];
            this.objectPool.RemoveAt(count-1);
            return resource;
        }
    }
}

```

```

public class Test : MonoBehaviour
{
    private ResourcePool<GameObject> objectPool;

    [SerializeField]
    private GameObject enemyPrefab;

    void Start()
    {
        this.objectPool = new ResourcePool<GameObject>(Destroy, () =>
Instantiate(this.enemyPrefab) );
    }

    void Update()
    {
        // To get existing object or create new from pool
        var newEnemy = this.objectPool.Rent();
        // To return object to pool
        this.objectPool.Return(newEnemy);
        // In this example the message 'Creating new object' should only be seen on the frame
call
        // after that the same object in the pool will be returned.
    }
}

```

: .
.

```

public class Weapon : MonoBehaviour {

    // The Bullet prefab that the Weapon will create
    public Bullet bulletPrefab;

    // This List is our object pool, which starts out empty
    private List<Bullet> availableBullets = new List<Bullet>();

    // The Transform that will act as the Bullet starting position
    public Transform bulletInstantiationPoint;

    // To spawn a new Bullet, this method either grabs an available Bullet from the pool,
    // otherwise Instantiates a new Bullet
    public Bullet CreateBullet () {
        Bullet newBullet = null;

        // If a Bullet is available in the pool, take the first one and make it active
        if (availableBullets.Count > 0) {
            newBullet = availableBullets[availableBullets.Count - 1];

```

```

        // Remove the Bullet from the pool
        availableBullets.RemoveAt(availableBullets.Count - 1);

        // Set the Bullet's position and make its GameObject active
        newBullet.transform.position = bulletInstantiationPoint.position;
        newBullet.gameObject.SetActive(true);
    }
    // If no Bullets are available in the pool, Instantiate a new Bullet
    else {
        newBullet newObject = Instantiate(bulletPrefab, bulletInstantiationPoint.position,
Quaternion.identity);

        // Set the Bullet's Weapon so we know which pool to return to later on
        newBullet.weapon = this;
    }

    return newBullet;
}
}

public class Bullet : MonoBehaviour {

    public Weapon weapon;

    // When Bullet collides with something, rather than Destroying it, we return it to the
pool
    public void ReturnToPool () {
        // Add Bullet to the pool
        weapon.availableBullets.Add(this);

        // Disable the Bullet's GameObject so it's hidden from view
        gameObject.SetActive(false);
    }
}
}

```

: <https://riptutorial.com/ko/unity3d/topic/2276/>

10:

Unity Ads Google AdMob Unity .

Unity Ads .

Unity Ads .

. , .

Unity Ads .

Examples

C

```
using UnityEngine;
using UnityEngine.Advertisements;

public class Example : MonoBehaviour
{
    #if !UNITY_ADS // If the Ads service is not enabled
    public string gameId; // Set this value from the inspector
    public bool enableTestMode = true; // Enable this during development
    #endif

    void InitializeAds () // Example of how to initialize the Unity Ads service
    {
        #if !UNITY_ADS // If the Ads service is not enabled
        if (Advertisement.isSupported) { // If runtime platform is supported
            Advertisement.Initialize(gameId, enableTestMode); // Initialize
        }
        #endif
    }

    void ShowAd () // Example of how to show an ad
    {
        if (Advertisement.isInitialized || Advertisement.IsReady()) { // If the ads are ready
        to be shown
            Advertisement.Show(); // Show the default ad placement
        }
    }
}
```

```
#pragma strict
import UnityEngine.Advertisements;

#if !UNITY_ADS // If the Ads service is not enabled
public var gameId : String; // Set this value from the inspector
public var enableTestMode : boolean = true; // Enable this during development
#endif

function InitializeAds () // Example of how to initialize the Unity Ads service
```

```
{
    #if !UNITY_ADS // If the Ads service is not enabled
    if (Advertisement.isSupported) { // If runtime platform is supported
        Advertisement.Initialize(gameId, enableTestMode); // Initialize
    }
    #endif
}

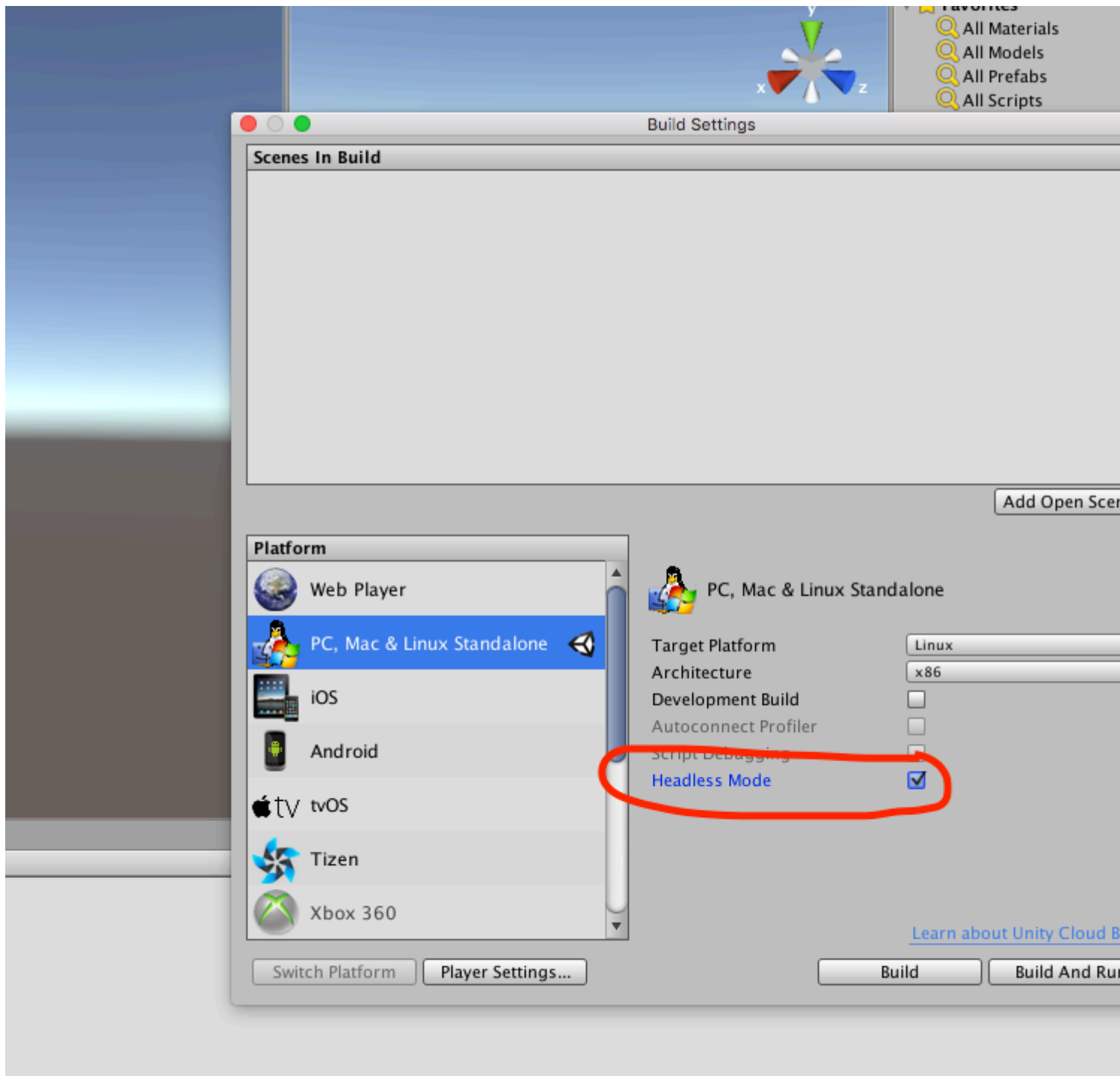
function ShowAd () // Example of how to show an ad
{
    if (Advertisement.isInitialized && Advertisement.IsReady()) { // If the ads are ready to
be shown
        Advertisement.Show(); // Show the default ad placement
    }
}
```

: [https://riptutorial.com/ko/unity3d/topic/9796/-](https://riptutorial.com/ko/unity3d/topic/9796/)

11:

Unity

Linux



Examples

,

Unity HLA (High Level API) .

.

HLA .

MessageBase . .

```
using System;
using UnityEngine.Networking;

public class MyNetworkMessage : MessageBase
{
    public string message;
}
```

9999 10 .

HLA ID . Unity Networking MsgType . ID 32 . . .

ID .

```
using UnityEngine;
using System.Collections;
using UnityEngine.Networking;

public class Server : MonoBehaviour {

    int port = 9999;
    int maxConnections = 10;

    // The id we use to identify our messages and register the handler
    short messageID = 1000;

    // Use this for initialization
    void Start () {
        // Usually the server doesn't need to draw anything on the screen
        Application.runInBackground = true;
        CreateServer();
    }

    void CreateServer() {
        // Register handlers for the types of messages we can receive
        RegisterHandlers ();

        var config = new ConnectionConfig ();
        // There are different types of channels you can use, check the official documentation
        config.AddChannel (QosType.ReliableFragmented);
        config.AddChannel (QosType.UnreliableFragmented);

        var ht = new HostTopology (config, maxConnections);

        if (!NetworkServer.Configure (ht)) {
            Debug.Log ("No server created, error on the configuration definition");
            return;
        }
    }
}
```

```

    } else {
        // Start listening on the defined port
        if(NetworkServer.Listen (port))
            Debug.Log ("Server created, listening on port: " + port);
        else
            Debug.Log ("No server created, could not listen to the port: " + port);
    }
}

void OnApplicationQuit() {
    NetworkServer.Shutdown ();
}

private void RegisterHandlers () {
    // Unity have different Messages types defined in MessageType
    NetworkServer.RegisterHandler (MessageType.Connect, OnClientConnected);
    NetworkServer.RegisterHandler (MessageType.Disconnect, OnClientDisconnected);

    // Our message use his own message type.
    NetworkServer.RegisterHandler (messageID, OnMessageReceived);
}

private void RegisterHandler(short t, NetworkMessageDelegate handler) {
    NetworkServer.RegisterHandler (t, handler);
}

void OnClientConnected(NetworkMessage netMessage)
{
    // Do stuff when a client connects to this server

    // Send a thank you message to the client that just connected
    MyNetworkMessage messageContainer = new MyNetworkMessage();
    messageContainer.message = "Thanks for joining!";

    // This sends a message to a specific client, using the connectionId
    NetworkServer.SendToClient(netMessage.conn.connectionId,messageID,messageContainer);

    // Send a message to all the clients connected
    messageContainer = new MyNetworkMessage();
    messageContainer.message = "A new player has conected to the server";

    // Broadcast a message a to everyone connected
    NetworkServer.SendToAll(messageID,messageContainer);
}

void OnClientDisconnected(NetworkMessage netMessage)
{
    // Do stuff when a client disssconnects
}

void OnMessageReceived(NetworkMessage netMessage)
{
    // You can send any object that inherence from MessageBase
    // The client and server can be on different projects, as long as the MyNetworkMessage
or the class you are using have the same implementation on both projects
    // The first thing we do is deserialize the message to our custom type
    var objectMessage = netMessage.ReadMessage<MyNetworkMessage>();
    Debug.Log("Message received: " + objectMessage.message);
}
}
}

```



```

using System;
using UnityEngine;
using UnityEngine.Networking;

public class Client : MonoBehaviour
{
    int port = 9999;
    string ip = "localhost";

    // The id we use to identify our messages and register the handler
    short messageID = 1000;

    // The network client
    NetworkClient client;

    public Client ()
    {
        CreateClient();
    }

    void CreateClient()
    {
        var config = new ConnectionConfig ();

        // Config the Channels we will use
        config.AddChannel (QosType.ReliableFragmented);
        config.AddChannel (QosType.UnreliableFragmented);

        // Create the client ant attach the configuration
        client = new NetworkClient ();
        client.Configure (config,1);

        // Register the handlers for the different network messages
        RegisterHandlers();

        // Connect to the server
        client.Connect (ip, port);
    }

    // Register the handlers for the different message types
    void RegisterHandlers () {

        // Unity have different Messages types defined in MsgType
        client.RegisterHandler (messageID, OnMessageReceived);
        client.RegisterHandler(MsgType.Connect, OnConnected);
        client.RegisterHandler(MsgType.Disconnect, OnDisconnected);
    }

    void OnConnected(NetworkMessage message) {
        // Do stuff when connected to the server

        MyNetworkMessage messageContainer = new MyNetworkMessage();
        messageContainer.message = "Hello server!";

        // Say hi to the server when connected
        client.Send(messageID,messageContainer);
    }
}

```

```
void OnDisconnected(NetworkMessage message) {
    // Do stuff when disconnected to the server
}

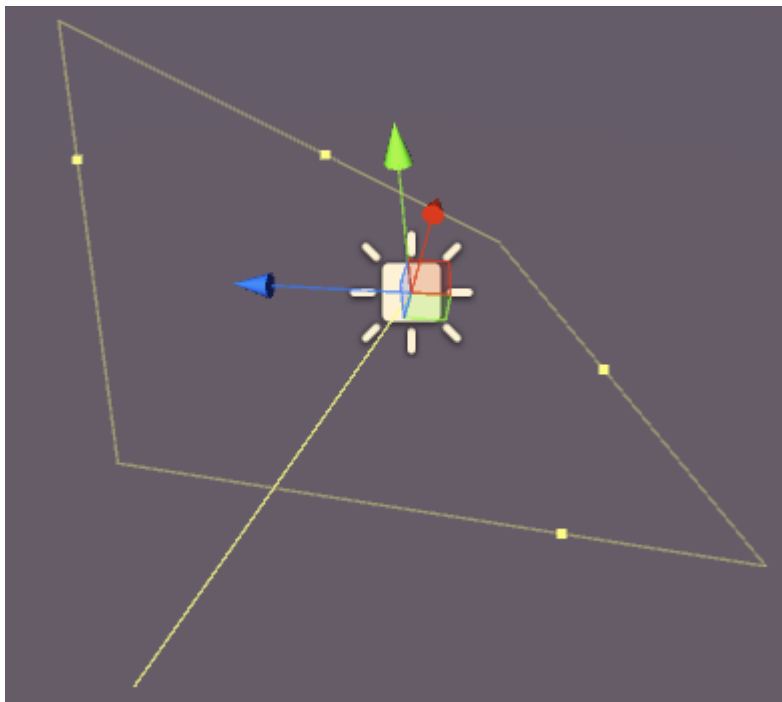
// Message received from the server
void OnMessageReceived(NetworkMessage netMessage)
{
    // You can send any object that inherence from MessageBase
    // The client and server can be on different projects, as long as the MyNetworkMessage
or the class you are using have the same implementation on both projects
    // The first thing we do is deserialize the message to our custom type
    var objectMessage = netMessage.ReadMessage<MyNetworkMessage>();

    Debug.Log("Message received: " + objectMessage.message);
}
}
```

: <https://riptutorial.com/ko/unity3d/topic/5671/>

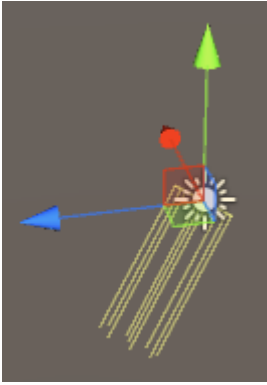
12:

Examples



- - .
- - .
- **Color** - .
- **Intensity** - 0 - 8.
- - 0 - 8 .
- **Draw Halo** - .
- **Flare ()** - .
- -, , .
- - .

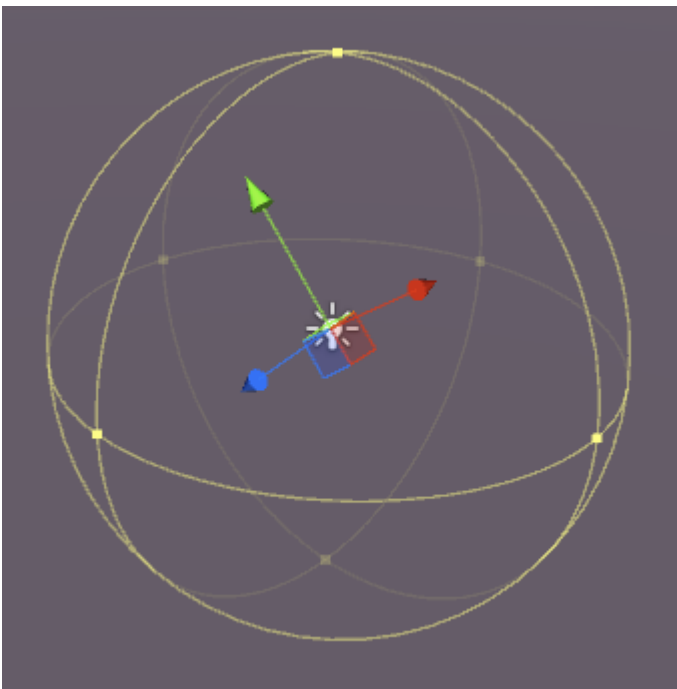
(.). "" GameObject . . .



Directional Light .

- - , .
- **Color** - .
- **Intensity** - 0 - 8.
- - 0 - 8 .
- - , .
- - .
- - .
- **Draw Halo** - .
- **Flare ()** - .
- - , , .
- - .

Point Light . . .

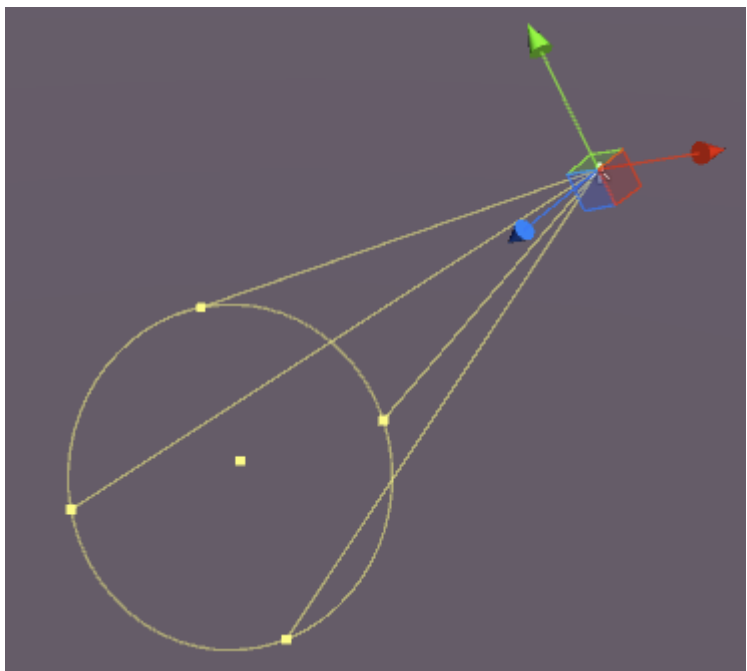


:

- - , .
-

- .
- **Color** - .
- **Intensity** - 0 - 8.
- - 0 - 8 .
- -, .
- - .
- **Draw Halo** - .
- **Flare ()** - .
- -, , .
- - .

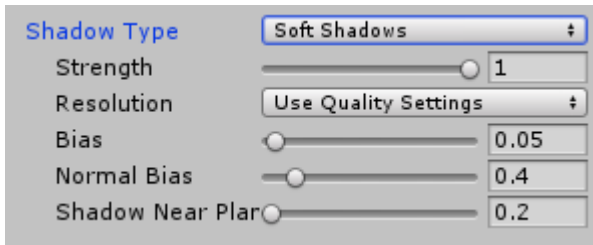
.. "" , .



:

- -, .
- - .
- - .
- **Color** - .
- **Intensity** - 0 - 8.
- - 0 - 8 .
- -, .
- - .
- **Draw Halo** - .
- **Flare ()** - .
- -, , .
- - .

- **Strength** - 0 - 1.
- - .
- - .
- **Normal Bias ()** - .
- - 0.1 - 10.

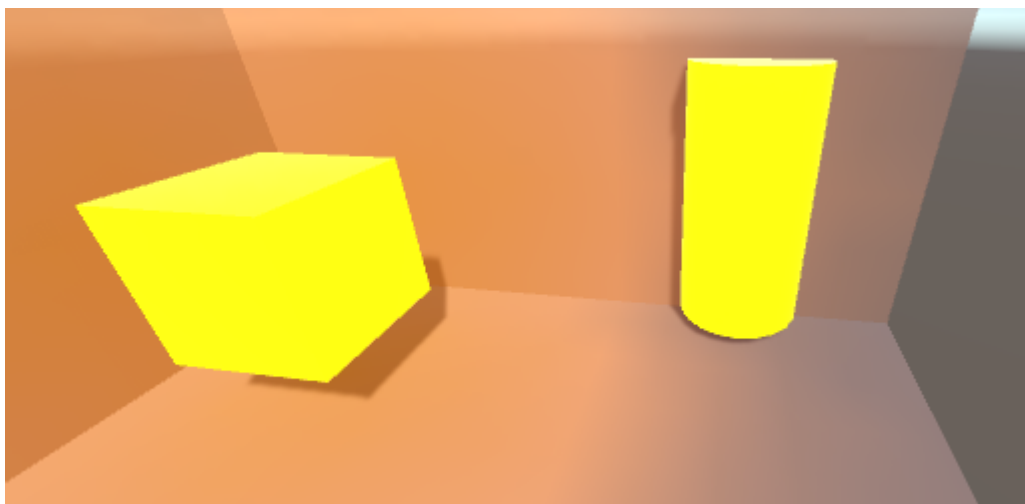


() . .



0 . .

- - .
- - .
- - .



""

```

Renderer renderer = GetComponent<Renderer>();
Material mat = renderer.material;
mat.SetColor("_EmissionColor", Color.yellow);

```

2 .

: <https://riptutorial.com/ko/unity3d/topic/7884/>-

13:

Singletons ([: gameprogrammingpatterns.com Singleton](http://gameprogrammingpatterns.com/Singleton)), Unity GameObject (:) .

GameObject () .

[ScriptableObject \(UnityDoc\)](#) [Singletons](#) . [Unity](#) . [Inversion of Control](#) .

```
// MyAudioManager.cs
using UnityEngine;

[CreateAssetMenu] // Remember to create the instance in editor
public class MyAudioManager : ScriptableObject {
    public void PlaySound() {}
}
```

```
// MyGameObject.cs
using UnityEngine;

public class MyGameObject : MonoBehaviour
{
    [SerializeField]
    MyAudioManager audioManager; //Insert through Inspector

    void OnEnable()
    {
        audioManager.PlaySound();
    }
}
```

- [C#](#)

Examples

RuntimeInitializeOnLoadMethodAttribute

Unity 5.2.5 [RuntimeInitializeOnLoadMethodAttribute](#) [MonoBehaviour](#) . . .

```
using UnityEngine;

sealed class GameDirector : MonoBehaviour
{
    // Because of using RuntimeInitializeOnLoadMethod attribute to find/create and
    // initialize the instance, this property is accessible and
    // usable even in Awake() methods.
    public static GameDirector Instance
    {
        get; private set;
    }
}
```



```

}

// Thanks to the attribute, this method is executed before any other MonoBehaviour
// logic in the game.
[RuntimeInitializeOnLoadMethod(RuntimeInitializeLoadType.BeforeSceneLoad)]
static void OnRuntimeMethodLoad()
{
    var instance = FindObjectOfType<GameDirector>();

    if (instance == null)
        instance = new GameObject("Game Director").AddComponent<GameDirector>();

    DontDestroyOnLoad(instance);

    Instance = instance;
}

// This Awake() will be called immediately after AddComponent() execution
// in the OnRuntimeMethodLoad(). In other words, before any other MonoBehaviour's
// in the scene will begin to initialize.
private void Awake()
{
    // Initialize non-MonBehaviour logic, etc.
    Debug.Log("GameDirector.Awake()", this);
}
}

```

:

1. GameDirector.OnRuntimeMethodLoad() ...
2. GameDirector.Awake()
3. GameDirector.OnRuntimeMethodLoad() .
4. OtherMonoBehaviour1.Awake()
5. OtherMonoBehaviour2.Awake()

Unity C# MonoBehaviour

private .

if Singleton ().

```

using UnityEngine;

public class SingletonExample : MonoBehaviour {

    private static SingletonExample _instance;

    void Awake(){

        if (_instance == null){

            _instance = this;
            DontDestroyOnLoad(this.gameObject);

            //Rest of your Awake code

        } else {

```

```

        Destroy(this);
    }
}

//Rest of your class code
}

```

Unity

MonoBehaviour

Unity 5 . . . public class MySingleton : Singleton<MySingleton> {} . (override) AwakeSingleton
 Awake . . .

1. **DisallowMultipleComponent** MonoBehaviour .
2. . Awake AwakeSingleton AwakeSingleton .
3. .
4. . instance null check instantiated == . ()
5. Unity .
6. :

- FindInactive : .
- Persist : .
- DestroyOthers : .
- Lazy : "on the fly"(Awake) "on demand"(getter) .

```

using UnityEngine;

[DisallowMultipleComponent]
public abstract class Singleton<T> : MonoBehaviour where T : Singleton<T>
{
    private static volatile T instance;
    // thread safety
    private static object _lock = new object();
    public static bool FindInactive = true;
    // Whether or not this object should persist when loading new scenes. Should be set in
    Init().
    public static bool Persist;
    // Whether or not destroy other singleton instances if any. Should be set in Init().
    public static bool DestroyOthers = true;
    // instead of heavy comparision (instance != null)
    // http://blogs.unity3d.com/2014/05/16/custom-operator-should-we-keep-it/
    private static bool instantiated;

    private static bool applicationIsQuitting;

    public static bool Lazy;

    public static T Instance
    {
        get
        {
            if (applicationIsQuitting)

```

```

    {
        Debug.LogWarningFormat("[Singleton] Instance '{0}' already destroyed on
application quit. Won't create again - returning null.", typeof(T));
        return null;
    }
    lock (_lock)
    {
        if (!instantiated)
        {
            Object[] objects;
            if (FindInactive) { objects = Resources.FindObjectsOfTypeAll(typeof(T)); }
            else { objects = FindObjectsOfType(typeof(T)); }
            if (objects == null || objects.Length < 1)
            {
                GameObject singleton = new GameObject();
                singleton.name = string.Format("{0} [Singleton]", typeof(T));
                Instance = singleton.AddComponent<T>();
                Debug.LogWarningFormat("[Singleton] An Instance of '{0}' is needed in
the scene, so '{1}' was created{2}", typeof(T), singleton.name, Persist ? " with
DontDestoryOnLoad." : ".");
            }
            else if (objects.Length >= 1)
            {
                Instance = objects[0] as T;
                if (objects.Length > 1)
                {
                    Debug.LogWarningFormat("[Singleton] {0} instances of '{1}'!",
objects.Length, typeof(T));
                    if (DestroyOthers)
                    {
                        for (int i = 1; i < objects.Length; i++)
                        {
                            Debug.LogWarningFormat("[Singleton] Deleting extra '{0}'
instance attached to '{1}'", typeof(T), objects[i].name);
                            Destroy(objects[i]);
                        }
                    }
                    return instance;
                }
            }
            return instance;
        }
    }
}
protected set
{
    instance = value;
    instantiated = true;
    instance.AwakeSingleton();
    if (Persist) { DontDestroyOnLoad(instance.gameObject); }
}
}

// if Lazy = false and gameObject is active this will set instance
// unless instance was called by another Awake method
private void Awake()
{
    if (Lazy) { return; }
    lock (_lock)
    {
        if (!instantiated)

```

```

        {
            Instance = this as T;
        }
        else if (DestroyOthers && Instance.GetInstanceID() != GetInstanceID())
        {
            Debug.LogWarningFormat("[Singleton] Deleting extra '{0}' instance attached to
'{1}'", typeof(T), name);
            Destroy(this);
        }
    }
}

// this might be called for inactive singletons before Awake if FindInactive = true
protected virtual void AwakeSingleton() {}

protected virtual void OnDestroy()
{
    applicationIsQuitting = true;
    instantiated = false;
}
}

```

() .

```

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

public abstract class MonoBehaviourSingleton<T> : MonoBehaviour {

    private static Dictionary<Type, object> _singletons
        = new Dictionary<Type, object>();

    public static T Instance {
        get {
            return (T)_singletons[typeof(T)];
        }
    }

    void OnEnable() {
        if (_singletons.ContainsKey(GetType())) {
            Destroy(this);
        } else {
            _singletons.Add(GetType(), this);
            DontDestroyOnLoad(this);
        }
    }
}

```

MonoBehaviour MonoBehaviourSingleton . .

```

using UnityEngine;
using System.Collections;

public class SingletonImplementation : MonoBehaviourSingleton<SingletonImplementation> {

    public string Text= "String Instance";
}

```

```
// Use this for initialisation
IEnumerator Start () {
    var demonstration = "SingletonImplementation.Start()\n" +
        "Note that the this text logs only once and\n"
        "only one class instance is allowed to exist.";
    Debug.Log(demonstration);
    yield return new WaitForSeconds(2f);
    var secondInstance = new GameObject();
    secondInstance.AddComponent<SingletonImplementation>();
}
}
```

```
// Logs: String Instance
Debug.Log(SingletonImplementation.Instance.Text);
```

DRY (), .

Unitys Entity-Component

GameObject .

- .
- Entity-Component Unity .
- (:)
- .
- Singletons MonoBehaviours / Components .
- (Singletons GameObject), .
- mock (mock).
- Unity ([Unity](#) [Singleton](#))

Test.cs ():

```
using UnityEngine;
using UnityEngine.Assertions;

public class Test : MonoBehaviour {
    void Start() {
        ExampleSingleton singleton = ExampleSingleton.instance;
        Assert.IsNotNull(singleton); // automatic initialization on first usage
        Assert.AreEqual("abc", singleton.myVar1);
        singleton.myVar1 = "123";
        // multiple calls to instance() return the same object:
        Assert.AreEqual(singleton, ExampleSingleton.instance);
        Assert.AreEqual("123", ExampleSingleton.instance.myVar1);
    }
}
```

ExampleSingleton.cs ():

```
using UnityEngine;
```

```

using UnityEngine.Assertions;

public class ExampleSingleton : MonoBehaviour {
    public static ExampleSingleton instance { get { return Singleton.get<ExampleSingleton>(); } }
    public string myVar1 = "abc";
    public void Start() { Assert.AreEqual(this, instance, "Singleton more than once in scene"); }
}

/// <summary> Helper that turns any MonoBehaviour or other Component into a Singleton
</summary>
public static class Singleton {
    public static T get<T>() where T : Component {
        return GetOrAddGo("Singletons").GetOrAddChild("" + typeof(T)).GetOrAddComponent<T>();
    }
    private static GameObject GetOrAddGo(string goName) {
        var go = GameObject.Find(goName);
        if (go == null) { return new GameObject(goName); }
        return go;
    }
}

public static class GameObjectExtensionMethods {
    public static GameObject GetOrAddChild(this GameObject parentGo, string childName) {
        var childGo = parentGo.transform.FindChild(childName);
        if (childGo != null) { return childGo.gameObject; } // child found, return it
        var newChild = new GameObject(childName); // no child found, create it
        newChild.transform.SetParent(parentGo.transform, false); // add it to parent
        return newChild;
    }

    public static T GetOrAddComponent<T>(this GameObject parentGo) where T : Component {
        var comp = parentGo.GetComponent<T>();
        if (comp == null) { return parentGo.AddComponent<T>(); }
        return comp;
    }
}

```

GameObject Singleton .

MonoBehaviour ScriptableObject

MonoBehaviour . Singleton . .

- .
- Singleton .
- Unity UI . "Proxy Component" "GameManager.Instance.SomeGlobalMethod ()" 1 .

ScriptableObjects MonoBehaviour . ScriptableObject MonoBehaviour .

- Unity .
- Unity .
- ().
- Unity Singleton .
- "SingletonClassName.Instance" .

- Update, Awake, Start, FixedUpdate, StartCoroutine MonoBehaviour

```

/*****
 * Better Singleton by David Darias
 * Use as you like - credit where due would be appreciated :D
 * Licence: WTFPL V2, Dec 2014
 * Tested on Unity v5.6.0 (should work on earlier versions)
 * 03/02/2017 - v1.1
 * *****/

using System;
using UnityEngine;
using SingletonScriptableObjectNamespace;

public class SingletonScriptableObject<T> :
SingletonScriptableObjectNamespace.BehaviourScriptableObject where T :
SingletonScriptableObjectNamespace.BehaviourScriptableObject
{
    //Private reference to the scriptable object
    private static T _instance;
    private static bool _instantiated;
    public static T Instance
    {
        get
        {
            if (_instantiated) return _instance;
            var singletonName = typeof(T).Name;
            //Look for the singleton on the resources folder
            var assets = Resources.LoadAll<T>("");
            if (assets.Length > 1) Debug.LogError("Found multiple " + singletonName + "s on
the resources folder. It is a Singleton ScriptableObject, there should only be one.");
            if (assets.Length == 0)
            {
                _instance = CreateInstance<T>();
                Debug.LogError("Could not find a " + singletonName + " on the resources
folder. It was created at runtime, therefore it will not be visible on the assets folder and
it will not persist.");
            }
            else _instance = assets[0];
            _instantiated = true;
            //Create a new game object to use as proxy for all the MonoBehaviour methods
            var baseObject = new GameObject(singletonName);
            //Deactivate it before adding the proxy component. This avoids the execution of
the Awake method when the the proxy component is added.
            baseObject.SetActive(false);
            //Add the proxy, set the instance as the parent and move to DontDestroyOnLoad
scene
            SingletonScriptableObjectNamespace.BehaviourProxy proxy =
baseObject.AddComponent<SingletonScriptableObjectNamespace.BehaviourProxy>();
            proxy.Parent = _instance;
            Behaviour = proxy;
            DontDestroyOnLoad(Behaviour.gameObject);
            //Activate the proxy. This will trigger the MonoBehaviourAwake.
            proxy.gameObject.SetActive(true);
            return _instance;
        }
    }
    //Use this reference to call MonoBehaviour specific methods (for example StartCoroutine)
    protected static MonoBehaviour Behaviour;
    public static void BuildSingletonInstance() {
SingletonScriptableObjectNamespace.BehaviourScriptableObject i = Instance; }

```

```

    private void OnDestroy(){ _instantiated = false; }
}

// Helper classes for the SingletonScriptableObject
namespace SingletonScriptableObjectNamespace
{
    #if UNITY_EDITOR
    //Empty custom editor to have cleaner UI on the editor.
    using UnityEditor;
    [CustomEditor(typeof(BehaviourProxy))]
    public class BehaviourProxyEditor : Editor
    {
        public override void OnInspectorGUI(){ }
    }

    #endif

    public class BehaviourProxy : MonoBehaviour
    {
        public IBehaviour Parent;

        public void Awake() { if (Parent != null) Parent.MonoBehaviourAwake(); }
        public void Start() { if (Parent != null) Parent.Start(); }
        public void Update() { if (Parent != null) Parent.Update(); }
        public void FixedUpdate() { if (Parent != null) Parent.FixedUpdate(); }
    }

    public interface IBehaviour
    {
        void MonoBehaviourAwake();
        void Start();
        void Update();
        void FixedUpdate();
    }

    public class BehaviourScriptableObject : ScriptableObject, IBehaviour
    {
        public void Awake() { ScriptableObjectAwake(); }
        public virtual void ScriptableObjectAwake() { }
        public virtual void MonoBehaviourAwake() { }
        public virtual void Start() { }
        public virtual void Update() { }
        public virtual void FixedUpdate() { }
    }
}

```

SingletonScriptableObject GameManager .

```

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

//this attribute is optional but recommended. It will allow the creation of the singleton via
the asset menu.
//the singleton asset should be on the Resources folder.
[CreateAssetMenu(fileName = "GameManager", menuName = "Game Manager", order = 0)]
public class GameManager : SingletonScriptableObject<GameManager> {

    //any properties as usual

```



```

public int Lives;
public int Points;

//optional (but recommended)
//this method will run before the first scene is loaded. Initializing the singleton here
//will allow it to be ready before any other GameObjects on every scene and will
//will prevent the "initialization on first usage".
[RuntimeInitializeOnLoadMethod(RuntimeInitializeLoadType.BeforeSceneLoad)]
public static void BeforeSceneLoad() { BuildSingletonInstance(); }

//optional,
//will run when the Singleton Scriptable Object is first created on the assets.
//Usually this happens on edit mode, not runtime. (the override keyword is mandatory for
this to work)
public override void ScriptableObjectAwake(){
    Debug.Log(GetType().Name + " created." );
}

//optional,
//will run when the associated MonoBehaviour awakes. (the override keyword is mandatory
for this to work)
public override void MonoBehaviourAwake(){
    Debug.Log(GetType().Name + " behaviour awake." );

    //A coroutine example:
    //Singleton Objects do not have coroutines.
    //if you need to use coroutines use the attached MonoBehaviour
    Behaviour.StartCoroutine(SimpleCoroutine());
}

//any methods as usual
private IEnumerator SimpleCoroutine(){
    while(true){
        Debug.Log(GetType().Name + " coroutine step." );
        yield return new WaitForSeconds(3);
    }
}

//optional,
//Classic runtime Update method (the override keyword is mandatory for this to work).
public override void Update(){

}

//optional,
//Classic runtime FixedUpdate method (the override keyword is mandatory for this to work).
public override void FixedUpdate(){

}

}

/*
* Notes:
* - Remember that you have to create the singleton asset on edit mode before using it. You
have to put it on the Resources folder and of course it should be only one.
* - Like other Unity Singleton this one is accessible anywhere in your code using the
"Instance" property i.e: GameManager.Instance
*/

```

: <https://riptutorial.com/ko/unity3d/topic/2137/-->

14:

Examples

(MVC)

. (Unity) .

MVC , .

: . , . Unity .

.

- MonoBehaviour .
- Unity .
- Unity API Model ()

Player.cs

```
using System;

public class Player
{
    public delegate void PositionEvent(Vector3 position);
    public event PositionEvent OnPositionChanged;

    public Vector3 position
    {
        get
        {
            return _position;
        }
        set
        {
            if (_position != value) {
                _position = value;
                if (OnPositionChanged != null) {
                    OnPositionChanged(value);
                }
            }
        }
    }
    private Vector3 _position;
}
```

Vector3.cs

Vector3 .

```
using System;
```

```

public class Vector3
{
    public float x;
    public float y;
    public float z;

    public Vector3(float x, float y, float z)
    {
        this.x = x;
        this.y = y;
        this.z = z;
    }
}

```

: . MonoBehaviour . OnCollisionEnter , Start , Update Unity API .

- MonoBehaviour .
-

PlayerView.cs

```

using UnityEngine;

public class PlayerView : MonoBehaviour
{
    public void SetPosition(Vector3 position)
    {
        transform.position = position;
    }
}

```

:

- .
- .
- ()
- MonoBehaviour .

PlayerController.cs

```

using System;

public class PlayerController
{
    public Player model { get; private set; }
    public PlayerView view { get; private set; }

    public PlayerController(Player model, PlayerView view)
    {
        this.model = model;
        this.view = view;

        this.model.OnPositionChanged += OnPositionChanged;
    }
}

```

```

private void OnPositionChanged(Vector3 position)
{
    // Sync
    Vector3 pos = this.model.position;

    // Unity call required here! (we lost portability)
    this.view.SetPosition(new UnityEngine.Vector3(pos.x, pos.y, pos.z));
}

// Calling this will fire the OnPositionChanged event
private void SetPosition(Vector3 position)
{
    this.model.position = position;
}
}

```

PlayerFactory.cs

```

using System;

public class PlayerFactory
{
    public PlayerController controller { get; private set; }
    public Player model { get; private set; }
    public PlayerView view { get; private set; }

    public void Load()
    {
        // Put the Player prefab inside the 'Resources' folder
        // Make sure it has the 'PlayerView' Component attached
        GameObject prefab = Resources.Load<GameObject>("Player");
        GameObject instance = GameObject.Instantiate<GameObject>(prefab);
        this.model = new Player();
        this.view = instance.GetComponent<PlayerView>();
        this.controller = new PlayerController(model, view);
    }
}

```

...

Manager.cs

```

using UnityEngine;

public class Manager : MonoBehaviour
{
    [ContextMenu("Load Player")]
    private void LoadPlayer()
    {
        new PlayerFactory().Load();
    }
}

```

GameObject Manager "Load Player" .

.

: [https://riptutorial.com/ko/unity3d/topic/10842/-](https://riptutorial.com/ko/unity3d/topic/10842/)

15:

	Colliders .
queryTriggerInteraction .	

Examples

Raycast

maxDistance maxDistance direction origin .

origin direction maxDistance **GameObject** .

```
Physics.Raycast(origin, direction, maxDistance);
```

, **GameObject** **10** Hello World :

```
using UnityEngine;

public class TestPhysicsRaycast: MonoBehaviour
{
    void FixedUpdate()
    {
        Vector3 fwd = transform.TransformDirection(Vector3.forward);

        if (Physics.Raycast(transform.position, fwd, 10))
            print("Hello World");
    }
}
```

Physics2D Raycast2D

AI .

```
using UnityEngine;

public class Physics2dRaycast: MonoBehaviour
{
    public LayerMask LineOfSightMask;
    void FixedUpdate()
    {
        RaycastHit2D hit = Physics2D.Raycast(raycastRightPart, Vector2.down, 0.6f *
heightCharacter, LineOfSightMask);
        if(hit.collider != null)
        {
```

```

        //code when the ai can walk
    }
    else
    {
        //code when the ai cannot walk
    }
}
}

```

`. raycastRightPart . 0.6 . Layermask . .`

`RaycastHit2D null ., RaycastHit2D .`

`Raycast LayerMask . .`

`Raycast RaycastHit .`

[SoC](#) ,

`LayerMask ScriptableObject .`

```

// RaycastService.cs
using UnityEngine;

[CreateAssetMenu(menuName = "StackOverflow")]
public class RaycastService : ScriptableObject
{
    [SerializeField]
    LayerMask layerMask;

    public RaycastHit2D Raycast2D(Vector2 origin, Vector2 direction, float distance)
    {
        return Physics2D.Raycast(origin, direction, distance, layerMask.value);
    }

    // Add more methods as needed
}

```

```

// MyScript.cs
using UnityEngine;

public class MyScript : MonoBehaviour
{
    [SerializeField]
    RaycastService raycastService;

    void FixedUpdate()
    {
        RaycastHit2D hit = raycastService.Raycast2D(Vector2.zero, Vector2.down, 1f);
    }
}

```

`LayerMask . .`

`LayerMask RaycastService .`

•
•

: [https://riptutorial.com/ko/unity3d/topic/2826/-](https://riptutorial.com/ko/unity3d/topic/2826/)

16:

Examples

Physics . [Unity Documentation - Physics](#)

```
using UnityEngine;
class LayerExample {

    public int layer;

    void Start()
    {
        Collider[] colliders = Physics.OverlapSphere(transform.position, 5f, layer);
    }
}
```

Collider .

LayerMask .

```
using UnityEngine;
class LayerMaskExample{

    public LayerMask mask;
    public Vector3 direction;

    void Start()
    {
        if(Physics.Raycast(transform.position, direction, 35f, mask))
        {
            Debug.Log("Raycast hit");
        }
    }
}
```

```
using UnityEngine;
class NameToLayerExample{

    void Start()
    {
        int layerindex = LayerMask.NameToLayer("Obstacle");
    }
}
```

```
public static bool IsInLayerMask(this GameObject @object, LayerMask layerMask)
```

```
{  
    bool result = (1 << @object.layer & layerMask) == 0;  
    return result;  
}
```

() layermask .

: <https://riptutorial.com/ko/unity3d/topic/4762/>

17:

Examples

- Android Apple Google Play .
- (Windows Windows , Linux Linux Penguin).
- .
- ...

```
void Update(){  
  
#if UNITY_IPHONE  
    //code here is only called when running on iPhone  
#endif  
  
#if UNITY_STANDALONE_WIN && !UNITY_EDITOR  
    //code here is only ran in a unity game running on windows outside of the editor  
#endif  
  
//other code that will be ran regardless of platform  
  
}
```

Unity .

partial . "" .

```
// ExampleClass.cs  
using UnityEngine;  
  
public partial class ExampleClass : MonoBehaviour  
{  
    partial void PlatformSpecificMethod();  
  
    void OnEnable()  
    {  
        PlatformSpecificMethod();  
    }  
}
```

. (ref) void .

```
// ExampleClass.Iphone.cs  
  
#if UNITY_IPHONE  
using UnityEngine;  
  
public partial class ExampleClass
```

```
{
    partial void PlatformSpecificMethod()
    {
        Debug.Log("I am an iPhone");
    }
}
#endif
```

```
// ExampleClass.Android.cs

#if UNITY_ANDROID
using UnityEngine;

public partial class ExampleClass
{
    partial void PlatformSpecificMethod()
    {
        Debug.Log("I am an Android");
    }
}
#endif
```

·
:
·
: <https://riptutorial.com/ko/unity3d/topic/4816/-->

18:

- public static int Input.touchCount
- Input.GetTouch (int)

Examples

Unity Input.GetTouch() Input.GetTouch() .

```
using UnityEngine;
using System.Collections;

public class TouchExample : MonoBehaviour {
    void Update() {
        if (Input.touchCount > 0 && Input.GetTouch(0).phase == TouchPhase.Began)
        {
            //Do Stuff
        }
    }
}
```

```
using UnityEngine;
using System.Collections;

public class TouchExample : MonoBehaviour {
    void Update() {
        for(int i = 0; i < Input.touchCount; i++)
        {
            if (Input.GetTouch(i).phase == TouchPhase.Began)
            {
                //Do Stuff
            }
        }
    }
}
```

TouchPhase

TouchPhase TouchPhase 5 .

- -
- - .
- - .
- - .
- -

```
public class TouchMoveExample : MonoBehaviour
{
    public float speed = 0.1f;

    void Update () {
        if(Input.touchCount > 0 && Input.GetTouch(0).phase == TouchPhase.Moved)
        {
            Vector2 touchDeltaPosition = Input.GetTouch(0).deltaPosition;
            transform.Translate(-touchDeltaPosition.x * speed, -touchDeltaPosition.y * speed,
0);
        }
    }
}
```

: [https://riptutorial.com/ko/unity3d/topic/6285/-](https://riptutorial.com/ko/unity3d/topic/6285/)

19:

Examples

GameObject . GameObject .

Component> Physics> Rigidbody .

Rigidbody

Rigidbody GameObject . AddForce() AddTorque() .

```
// Add a force to the order of myForce in the forward direction of the Transform.
GetComponent<Rigidbody>().AddForce(transform.forward * myForce);

// Add torque about the Y axis to the order of myTurn.
GetComponent<Rigidbody>().AddTorque(transform.up * torque * myTurn);
```

Rigidbody GameObject Rigidbody . GameObject GameObjects . . :

```
GetComponent<Rigidbody>().mass = 1000;
```

. . .

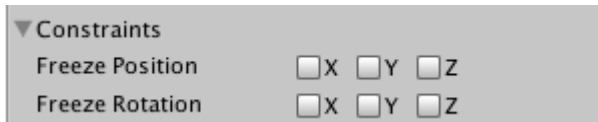
```
GetComponent<Rigidbody>().drag = 10;
```

isKinematic

Rigidbody Kinematic GameObject . .

```
GetComponent<Rigidbody>().isKinematic = true;
```

Rigidbody . RigidbodyConstraints.None .



:

```
// Freeze rotation on all axes.  
GetComponent<Rigidbody>().constraints = RigidbodyConstraints.FreezeRotation  
  
// Freeze position on all axes.  
GetComponent<Rigidbody>().constraints = RigidbodyConstraints.FreezePosition  
  
// Freeze rotation and motion on all axes.  
GetComponent<Rigidbody>().constraints = RigidbodyConstraints.FreezeAll
```

OR | .

```
// Allow rotation on X and Y axes and motion on Y and Z axes.  
GetComponent<Rigidbody>().constraints = RigidbodyConstraints.FreezePositionZ |  
RigidbodyConstraints.FreezeRotationX;
```

GameObject . .

-
-
-
-
-

GameObject .

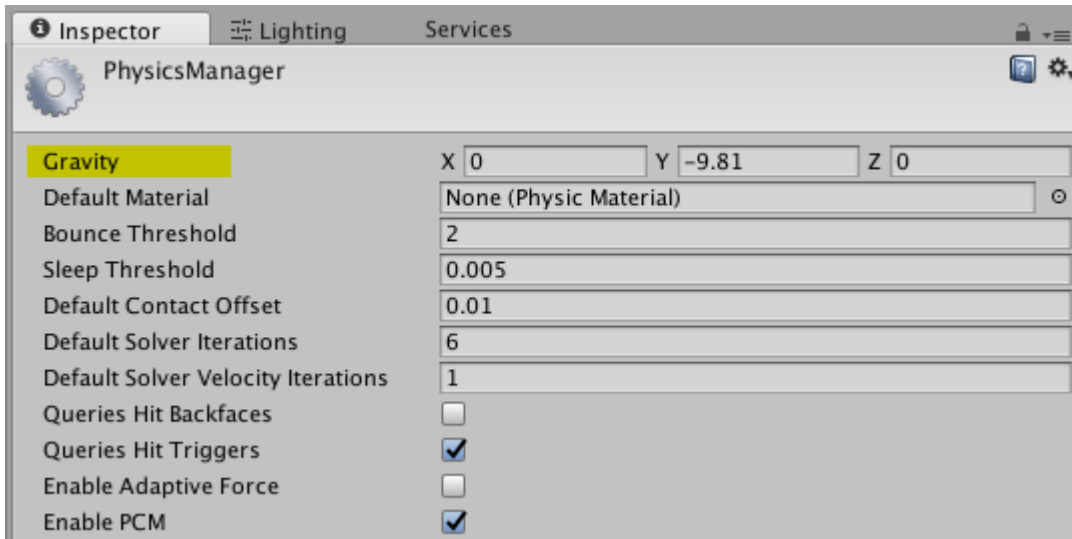
OnTriggerEnter() , OnTriggerStay() OnTriggerExit() . , GameObject . GameObject . ,
, .

Rigidbody useGravity . false Rigidbody ().

```
GetComponent<Rigidbody>().useGravity = false;
```

Rigidbody .

Rigidbody . Physics Settings .



3 . , useGravity True RigidBody 9.81 (y) .

: <https://riptutorial.com/ko/unity3d/topic/3680/>

20: 3

Vector3 3D UnityEngine . Vector3 Transform . Vector3 . [Unity API Vector3](#) .

- public Vector3 ();
- public Vector3 (float x, float y);
- public Vector3 (float x, float y, float z);
- Vector3.Lerp (Vector3 startPosition, Vector3 targetPosition, float movementFraction);
- Vector3.LerpUnclamped (Vector3 startPosition, Vector3 targetPosition, float movementFraction);
- Vector3.MoveTowards (Vector3 startPosition, Vector3 targetPosition, float distance);

Examples

Vector3 Vector3 . .

Vector3.zero **Vector3.one**

Vector3.zero Vector3.one Vector3 . , x , y z 1 Vector3 . Vector3.zero Vector3.one .

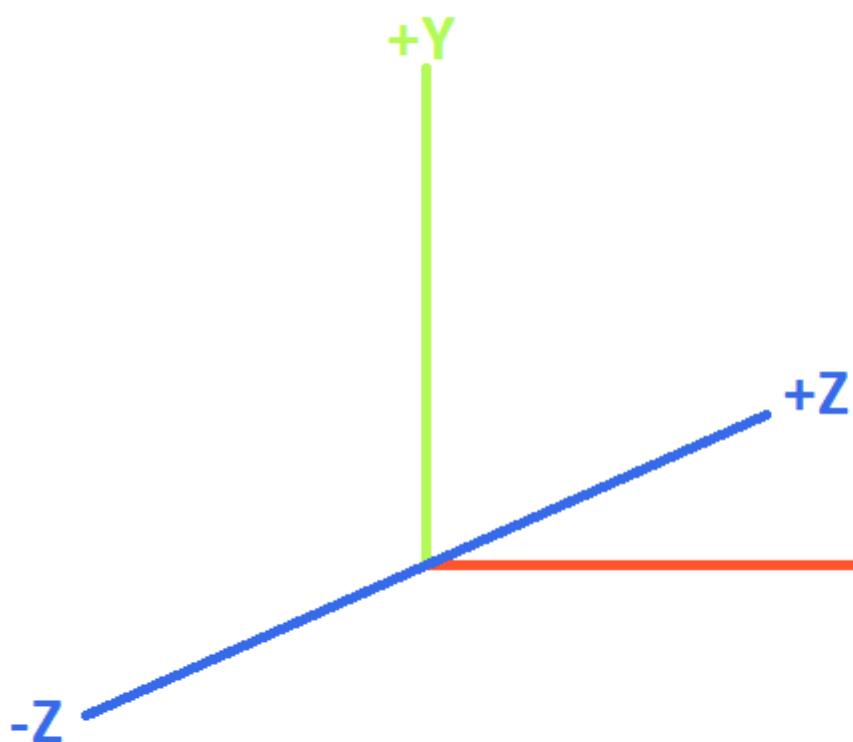
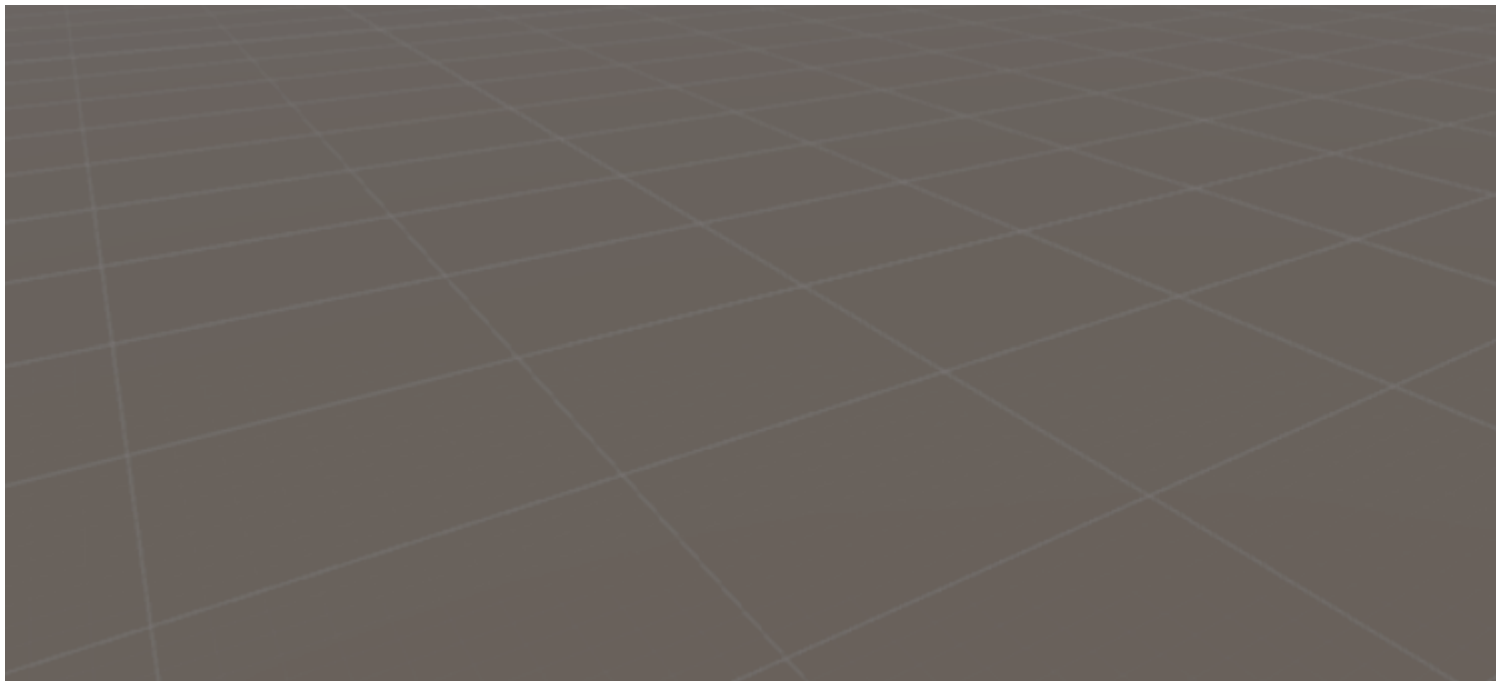
Vector3.zero .

Vector3.zero Vector3.one Vector3.one .

```
using UnityEngine;

public class Inflater : MonoBehaviour
{
    <summary>A sphere set up to inflate and deflate between two values.</summary>
    public ScaleBetween sphere;

    ///<summary>On start, set the sphere GameObject up to inflate
    /// and deflate to the corresponding values.</summary>
    void Start()
    {
        // Vector3.zero = Vector3(0, 0, 0); Vector3.one = Vector3(1, 1, 1);
        sphere.SetScale(Vector3.zero, Vector3.one);
    }
}
```



LEFT-HANDED COORDINATE SYSTEM

Vector3 .

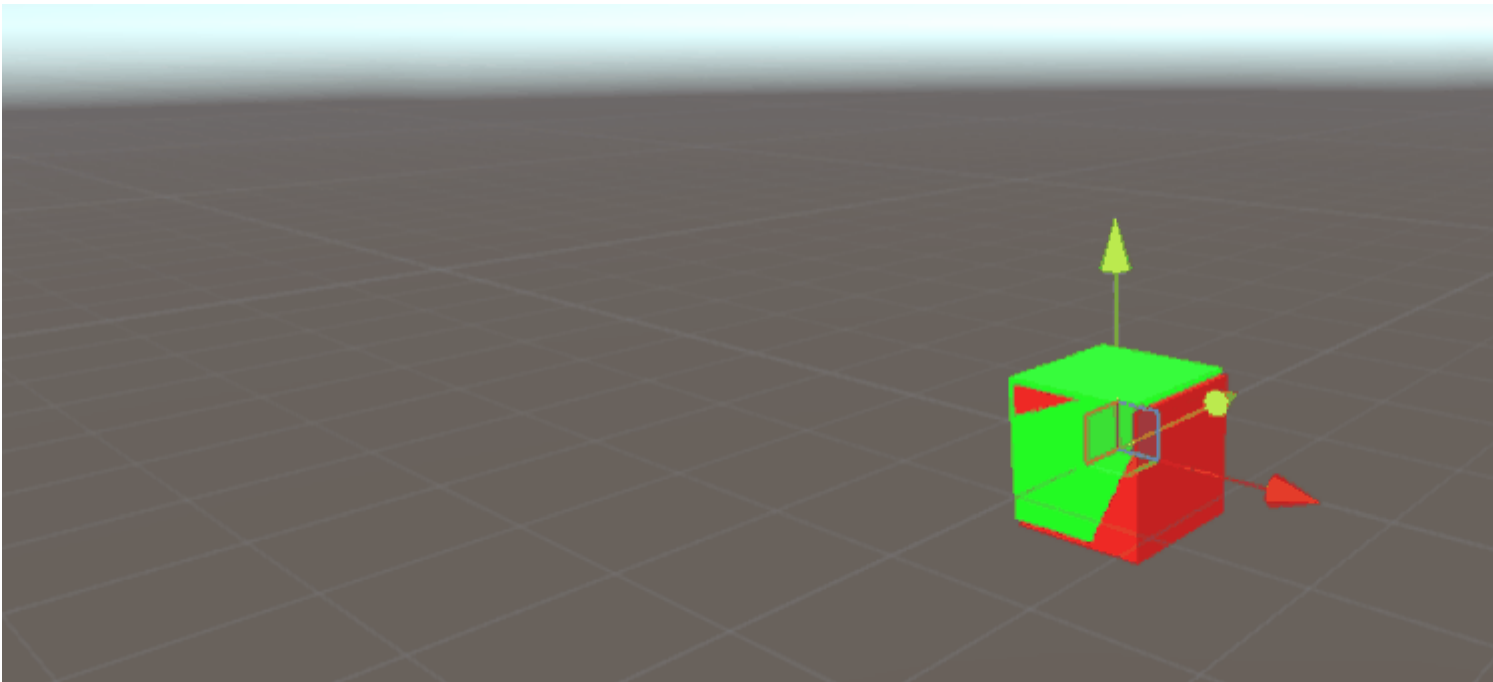
```
using UnityEngine;

public class StaticMover : MonoBehaviour
{
    <summary>GameObjects set up to move back and forth between two directions.</summary>
    public MoveBetween xMovement, yMovement, zMovement;

    ///<summary>On start, set each MoveBetween GameObject up to move
    /// in the corresponding direction(s).</summary>
    void Start()
    {
        // Vector3.left = Vector3(-1, 0, 0); Vector3.right = Vector3(1, 0, 0);
        xMovement.SetDirections(Vector3.left, Vector3.right);

        // Vector3.down = Vector3(0, -1, 0); Vector3.up = Vector3(0, 0, 1);
        yMovement.SetDirections(Vector3.down, Vector3.up);

        // Vector3.back = Vector3(0, 0, -1); Vector3.forward = Vector3(0, 0, 1);
        zMovement.SetDirections(Vector3.back, Vector3.forward);
    }
}
```



				<code>new Vector3()</code>
<code>Vector3.zero</code>	<code>0</code>	<code>0</code>	<code>0</code>	<code>new Vector3(0, 0, 0)</code>
<code>Vector3.one</code>	<code>1</code>	<code>1</code>	<code>1</code>	<code>new Vector3(1, 1, 1)</code>
<code>Vector3.left</code>	<code>-1</code>	<code>0</code>	<code>0</code>	<code>new Vector3(-1, 0, 0)</code>

				<code>new Vector3()</code>
<code>Vector3.right</code>	<code>1</code>	<code>0</code>	<code>0</code>	<code>new Vector3(1, 0, 0)</code>
<code>Vector3.down</code>	<code>0</code>	<code>-1</code>	<code>0</code>	<code>new Vector3(0, -1, 0)</code>
<code>Vector3.up</code>	<code>0</code>	<code>1</code>	<code>0</code>	<code>new Vector3(0, 1, 0)</code>
<code>Vector3.back</code>	<code>0</code>	<code>0</code>	<code>-1</code>	<code>new Vector3(0, 0, -1)</code>
<code>Vector3.forward</code>	<code>0</code>	<code>0</code>	<code>1</code>	<code>new Vector3(0, 0, 1)</code>

Vector3

`Vector3` `.Vector3` `.`

`Vector3` `.`

<code>new Vector3()</code>	<code>(0, 0, 0)</code> <code>Vector3</code> <code>.</code>
<code>new Vector3(float x, float y)</code>	<code>x y Vector3 . z 0.</code>
<code>new Vector3(float x, float y, float z)</code>	<code>x ,y z Vector3 .</code>

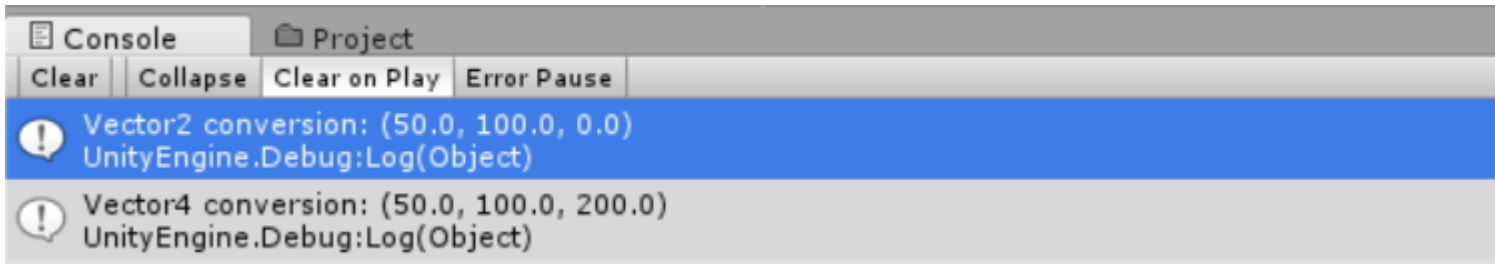
Vector2 Vector4

`Vector2` `Vector4` `Vector3` `.` , `Vector2` `Vector4` `Vector3` `.` `Vector2` `x y` `Vector4` `w` `.`

```
void VectorConversionTest ()
{
    Vector2 vector2 = new Vector2(50, 100);
    Vector4 vector4 = new Vector4(50, 100, 200, 400);

    Vector3 fromVector2 = vector2;
    Vector3 fromVector4 = vector4;

    Debug.Log("Vector2 conversion: " + fromVector2);
    Debug.Log("Vector4 conversion: " + fromVector4);
}
```



Vector3 Vector3 .

Lerp LerpUnclamped

Lerp .Lerp LerpUnclamped .

float . 0.5 , Vector3 . 0 1 Vector3 , (Vector3) (Vector3) . . .

Lerp 0 1 . .LerpUnclamped , .

Lerp LerpUnclamped .

```
using UnityEngine;

public class Lerpung : MonoBehaviour
{
    /// <summary>The red box will use Lerp to move. We will link
    /// this object in via the inspector.</summary>
    public GameObject lerpObject;
    /// <summary>The starting position for our red box.</summary>
    public Vector3 lerpStart = new Vector3(0, 0, 0);
    /// <summary>The end position for our red box.</summary>
    public Vector3 lerpTarget = new Vector3(5, 0, 0);

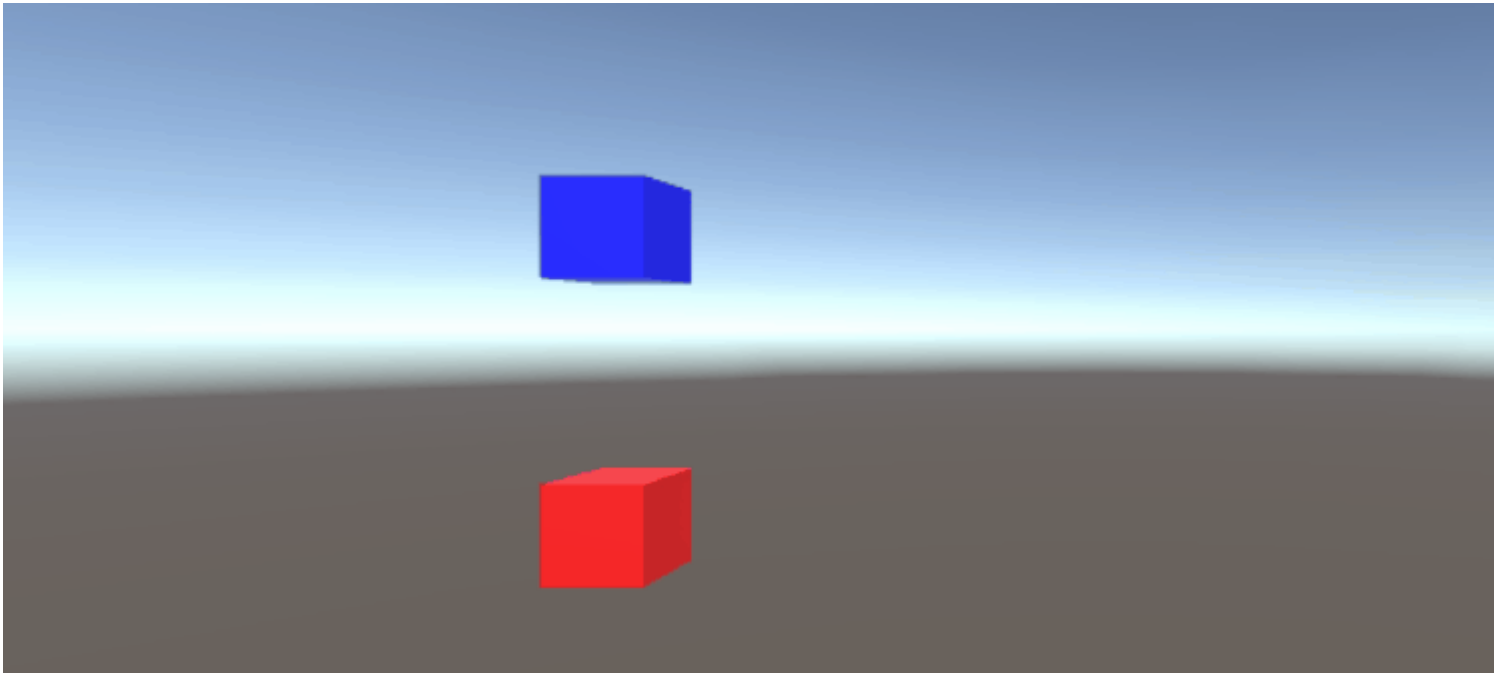
    /// <summary>The blue box will use LerpUnclamped to move. We will
    /// link this object in via the inspector.</summary>
    public GameObject lerpUnclampedObject;
    /// <summary>The starting position for our blue box.</summary>
    public Vector3 lerpUnclampedStart = new Vector3(0, 3, 0);
    /// <summary>The end position for our blue box.</summary>
    public Vector3 lerpUnclampedTarget = new Vector3(5, 3, 0);

    /// <summary>The current fraction to increment our lerp functions by.</summary>
    public float lerpFraction = 0;

    private void Update()
    {
        // First, I increment the lerp fraction.
        // delaTime * 0.25 should give me a value of +1 every second.
        lerpFraction += (Time.deltaTime * 0.25f);

        // Next, we apply the new lerp values to the target transform position.
        lerpObject.transform.position
            = Vector3.Lerp(lerpStart, lerpTarget, lerpFraction);
        lerpUnclampedObject.transform.position
            = Vector3.LerpUnclamped(lerpUnclampedStart, lerpUnclampedTarget, lerpFraction);
    }
}
```

```
}
```



MoveTowards

```
MoveTowards Lerp . MoveTowards Vector3 .  
LerpUnclamped LerpUnclamped Vector3 . Vector3 Vector3 . Vector3 " " . Vector3  
Vector3 .
```

```
MoveTowards .
```

```
using UnityEngine;  
  
public class MoveTowardsExample : MonoBehaviour  
{  
    /// <summary>The red cube will move up, the blue cube will move down,  
    /// the green cube will move left and the yellow cube will move right.  
    /// These objects will be linked via the inspector.</summary>  
    public GameObject upCube, downCube, leftCube, rightCube;  
    /// <summary>The cubes should move at 1 unit per second.</summary>  
    float speed = 1f;  
  
    void Update()  
    {  
        // We determine our distance by applying a deltaTime scale to our speed.  
        float distance = speed * Time.deltaTime;  
  
        // The up cube will move upwards, until it reaches the  
        // position of (Vector3.up * 2), or (0, 2, 0).  
        upCube.transform.position  
            = Vector3.MoveTowards(upCube.transform.position, (Vector3.up * 2f), distance);  
  
        // The down cube will move downwards, as it enforces a negative distance..
```

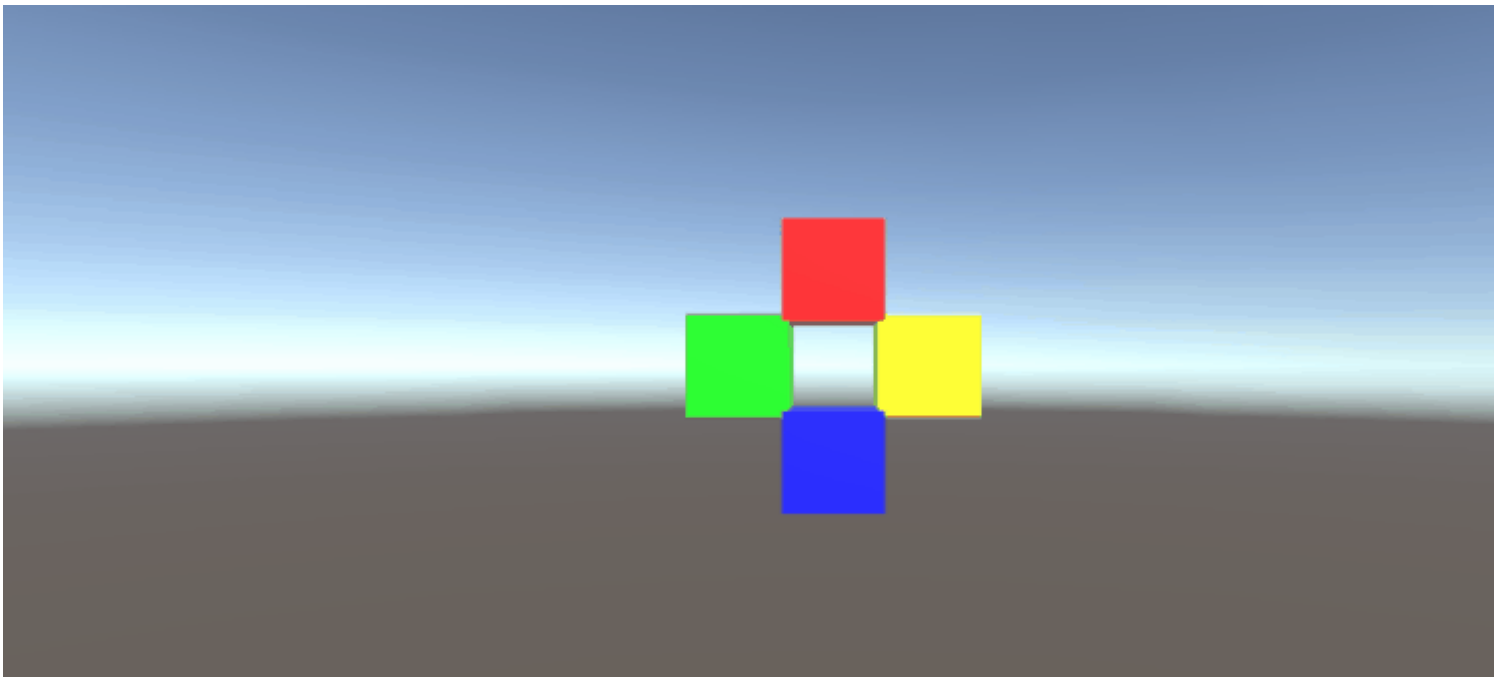
```

downCube.transform.position
    = Vector3.MoveTowards(downCube.transform.position, Vector3.up * 2f, -distance);

// The right cube will move to the right, indefinitely, as it is constantly updating
// its target position with a direction based off the current position.
rightCube.transform.position = Vector3.MoveTowards(rightCube.transform.position,
    rightCube.transform.position + Vector3.right, distance);

// The left cube does not need to account for updating its target position,
// as it is moving away from the target position, and will never reach it.
leftCube.transform.position
    = Vector3.MoveTowards(leftCube.transform.position, Vector3.right, -distance);
}
}

```



SmoothDamp

```

MoveTowards SmoothDamp . . .
Vector3 Vector3 float . , . .
float SmoothDamp float . . Time.deltaTime . MonoBehaviour.Update() .

```

```

using UnityEngine;

public class SmoothDampMovement : MonoBehaviour
{
    /// <summary>The red cube will imitate the default SmoothDamp function.
    /// The blue cube will move faster by manipulating the "time gap", while
    /// the green cube will have an enforced maximum speed. Note that these
    /// objects have been linked via the inspector.</summary>
    public GameObject smoothObject, fastSmoothObject, cappedSmoothObject;
}

```



```

/// <summary>We must instantiate the velocities, externally, so they may
/// be manipulated from within the function. Note that by making these
/// vectors public, they will be automatically instantiated as Vector3.Zero
/// through the inspector. This also allows us to view the velocities,
/// from the inspector, to observe how they change.</summary>
public Vector3 regularVelocity, fastVelocity, cappedVelocity;

/// <summary>Each object should move 10 units along the X-axis.</summary>
Vector3 regularTarget = new Vector3(10f, 0f);
Vector3 fastTarget = new Vector3(10f, 1.5f);
Vector3 cappedTarget = new Vector3(10f, 3f);

/// <summary>We will give a target time of 5 seconds.</summary>
float targetTime = 5f;

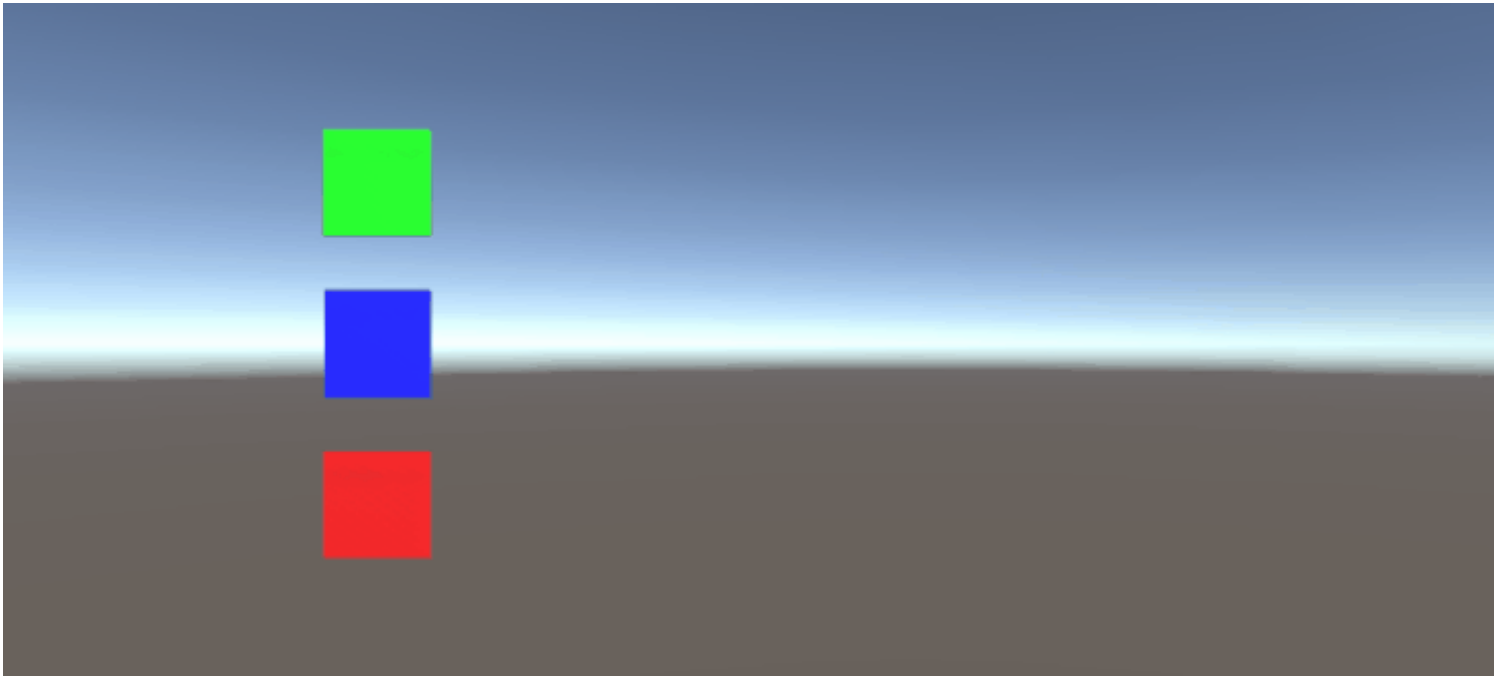
void Update()
{
    // The default SmoothDamp function will give us a general smooth movement.
    smoothObject.transform.position = Vector3.SmoothDamp(smoothObject.transform.position,
        regularTarget, ref regularVelocity, targetTime);

    // Note that a "maxSpeed" outside of reasonable limitations should not have any
    // effect, while providing a "deltaTime" of 0 tells the function that no time has
    // passed since the last SmoothDamp call, resulting in no movement, the second time.
    smoothObject.transform.position = Vector3.SmoothDamp(smoothObject.transform.position,
        regularTarget, ref regularVelocity, targetTime, 10f, 0f);

    // Note that "deltaTime" defaults to Time.deltaTime due to an assumption that this
    // function will be called once per update function. We can call the function
    // multiple times during an update function, but the function will assume that enough
    // time has passed to continue the same approximate movement. As a result,
    // this object should reach the target, quicker.
    fastSmoothObject.transform.position = Vector3.SmoothDamp(
        fastSmoothObject.transform.position, fastTarget, ref fastVelocity, targetTime);
    fastSmoothObject.transform.position = Vector3.SmoothDamp(
        fastSmoothObject.transform.position, fastTarget, ref fastVelocity, targetTime);

    // Lastly, note that a "maxSpeed" becomes irrelevant, if the object does not
    // realistically reach such speeds. Linear speed can be determined as
    // (Distance / Time), but given the simple fact that we start and end slow, we can
    // infer that speed will actually be higher, during the middle. As such, we can
    // infer that a value of (Distance / Time) or (10/5) will affect the
    // function. We will half the "maxSpeed", again, to make it more noticeable.
    cappedSmoothObject.transform.position = Vector3.SmoothDamp(
        cappedSmoothObject.transform.position,
        cappedTarget, ref cappedVelocity, targetTime, 1f);
}
}

```



3 : <https://riptutorial.com/ko/unity3d/topic/7827/-3>

21:

- void Transform.Translate (Vector3 translation, Space relativeTo = Space.Self)
- void transform.translate (float x, float y, float z, Space relativeTo = Space.Self)
- void Transform.Rotate (Vector3 eulerAngles, Space relativeTo = Space.Self)
- void Transform.Rotate (float xAngle, yAngle , zAngle , Space relativeTo = Space.Self)
- void Transform.Rotate (Vector3 , , Space relativeTo = Space.Self)
- void Transform.RotateAround (, ,)
- void Transform.LookAt (Transform , Vector3 worldUp = Vector3.up)
- void LookAt (Vector3 worldPosition, Vector3 worldUp = Vector3.up) .

Examples

(), (), , . . GameObject .

()

```
// Move an object 10 units in the positive x direction
transform.Translate(10, 0, 0);
```

```
// translating with a vector3
vector3 distanceToMove = new Vector3(5, 2, 0);
transform.Translate(distanceToMove);
```

```
// Rotate an object 45 degrees about the Y axis
transform.Rotate(0, 45, 0);
```

```
// Rotates an object about the axis passing through point (in world coordinates) by angle in degrees
transform.RotateAround(point, axis, angle);
// Rotates on it's place, on the Y axis, with 90 degrees per second
transform.RotateAround(Vector3.zero, Vector3.up, 90 * Time.deltaTime);
```

```
// Rotates an object to make it's forward vector point towards the other object
transform.LookAt(otherTransform);
// Rotates an object to make it's forward vector point towards the given position (in world coordinates)
transform.LookAt(new Vector3(10, 5, 0));
```

Unity .

isKinematic == true . [AddForce](#) .

Unity . .

```
var other = GetOtherGameObject();
other.transform.SetParent( transform );
other.transform.SetParent( transform, worldPositionStays );
```

```
.worldPositionStays false .
```

```
, ,
```

```
other.transform.IsChildOf( transform );
```

```
. .
```

```
transform.Find( "other" );  
transform.FindChild( "other" );
```

: FindChild

```
."/"
```

```
transform.Find( "other/another" );  
transform.FindChild( "other/another" );
```

GetChild

```
transform.GetChild( index );
```

GetChild .

```
int count = transform.childCount;
```

GameObject . (Z).

```
other.transform.SetSiblingIndex( index );
```

```
.
```

```
other.transform.SetAsFirstSibling();  
other.transform.SetAsLastSibling();
```

```
.
```

```
foreach(Transform child in transform)  
{  
    child.parent = null;  
}
```

Unity .

```
transform.DetachChildren();
```

DetachChildren() null ., .

(:)

: <https://riptutorial.com/ko/unity3d/topic/2190/>

22: (UI)

Examples

inspector . . .

```
using UnityEngine;
using UnityEngine.UI;

[RequireComponent(typeof(Button))]
public class AutomaticClickHandler : MonoBehaviour
{
    private void Awake()
    {
        var button = this.GetComponent<Button>();
        button.onClick.AddListener(HandleClick);
    }

    private void HandleClick()
    {
        Debug.Log("AutomaticClickHandler.HandleClick()", this);
    }
}
```

UI . . .

- Button : [onClick](#)
- : [onValueChanged](#)
- InputField : [onEndEdit](#) , [onValidateInput](#) , [onValueChanged](#)
- : [onValueChanged](#)
- ScrollRect : [onValueChanged](#)
- : [onValueChanged](#)
- : [onValueChanged](#)

, . EventTrigger . . .

```
using UnityEngine;
using UnityEngine.EventSystems;

[RequireComponent(typeof(EventTrigger))]
public class CustomListenersExample : MonoBehaviour
{
    void Start()
    {
        EventTrigger eventTrigger = GetComponent<EventTrigger>( );
        EventTrigger.Entry entry = new EventTrigger.Entry( );
        entry.eventID = EventTriggerType.PointerDown;
        entry.callback.AddListener( ( data ) => { OnPointerDownDelegate(
(PointerEventData)data ); } );
        eventTrigger.triggers.Add( entry );
    }

    public void OnPointerDownDelegate( PointerEventData data )
```

```
{  
    Debug.Log( "OnPointerDownDelegate called." );  
}  
}
```

eventID :

-
- PointerExit
- PointerDown
-
- PointerClick
-
-
-
- UpdateSelected
-
-
-
- InitializePotentialDrag
- BeginDrag
- EndDrag
-
-

(UI) : <https://riptutorial.com/ko/unity3d/topic/2296/----ui->

23:

Examples

. new WWW("https://urlexample.com"); URL Get .

```
using UnityEngine;
using System.Collections;

public class ExampleClass : MonoBehaviour
{
    public string url = "http://google.com";

    IEnumerator Start()
    {
        WWW www = new WWW(url); // One get.
        yield return www;
        Debug.Log(www.text); // The data of the url.
    }
}
```

()

WWW .

ID .

```
void Login(string id, string pwd)
{
    WWWForm dataParameters = new WWWForm(); // Create a new form.
    dataParameters.AddField("username", id);
    dataParameters.AddField("password", pwd); // Add fields.
    WWW www = new WWW(url+"/account/login",dataParameters);
    StartCoroutine("PostdataEnumerator", www);
}

IEnumerator PostdataEnumerator(WWW www)
{
    yield return www;
    if (!string.IsNullOrEmpty(www.error))
    {
        Debug.Log(www.error);
    }
    else
    {
        Debug.Log("Data Submitted");
    }
}
```

()

. WWW :

Zip

```
string mainUrl = "http://server/upload/";
string saveLocation;

void Start()
{
    saveLocation = "ftp://home/xxx/x.zip"; // The file path.
    StartCoroutine(PrepareFile());
}

// Prepare The File.
IEnumerator PrepareFile()
{
    Debug.Log("saveLoacation = " + saveLocation);

    // Read the zip file.
    WWW loadTheZip = new WWW(saveLocation);

    yield return loadTheZip;

    PrepareStepTwo(loadTheZip);
}

void PrepareStepTwo(WWW post)
{
    StartCoroutine(UploadTheZip(post));
}

// Upload.
IEnumerator UploadTheZip(WWW post)
{
    // Create a form.
    WWWForm form = new WWWForm();

    // Add the file.
    form.AddBinaryData("myTestFile.zip", post.bytes, "myFile.zip", "application/zip");

    // Send POST request.
    string url = mainUrl;
    WWW POSTZIP = new WWW(url, form);

    Debug.Log("Sending zip...");
    yield return POSTZIP;
    Debug.Log("Zip sent!");
}
```

coroutine . Unity coroutines [Coroutines](#) .

Unity (). . .

. . .

#:

```
using System.Net;
using System.Text;
```

```

public class TestCommunicationWithServer
{
    public string SendDataToServer(string url, string username, string password)
    {
        WebClient client = new WebClient();

        // This specialized key-value pair will store the form data we're sending to the
server
        var loginData = new System.Collections.Specialized.NameValueCollection();
        loginData.Add("Username", username);
        loginData.Add("Password", password);

        // Upload client data and receive a response
        byte[] opBytes = client.UploadValues(ServerIpAddress, "POST", loginData);

        // Encode the response bytes into a proper string
        string opResponse = Encoding.UTF8.GetString(opBytes);

        return opResponse;
    }
}

```

WebClient NameValueCollection using .

SendDataToServer 3 () .

1. URL
- 2.
- 3.

WebClient . NameValueCollection .

UploadValues 3 .

1. IP
2. HTTP
3. ()

```

if(opResponse.Equals(ReturnMessage.Success))
{
    Debug.Log("Unity client has successfully sent and validated data on server.");
}

```

PHP . PHP . PHP Unity .

PHP :

```
// Check to see if the unity client send the form data
if(!isset($_REQUEST['Username']) || !isset($_REQUEST['Password']))
{
    echo "Empty";
}
else
{
    // Unity sent us the data - its here so do whatever you want

    echo "Success";
}
}
```

". () . () .

Unity . .

: [https://riptutorial.com/ko/unity3d/topic/5578/-](https://riptutorial.com/ko/unity3d/topic/5578/)

24:

- [AddComponentMenu (string menuName)]
- [AddComponentMenu (string menuName, int order)]
- [CanEditMultipleObjects]
- [ContextMenu (,)]
- [ContextMenu ()]
- [CustomEditor ()]
- [CustomEditor (, bool editorForChildClasses)]
- [CustomPropertyDrawer ()]
- [CustomPropertyDrawer (, useForChildren)]
- [DisallowMultipleComponent]
- [(GizmoType gizmo)]
- [DrawGizmo (GizmoType , drawnGizmoType)]
- [ExecuteInEditMode]
- [()]
- [HideInInspector]
- [InitializeOnLoad]
- [InitializeOnLoadMethod]
- [MenuItem (string itemName)]
- [MenuItem (string itemName, bool isValidFunction)]
- [MenuItem (itemName, bool isValidFunction, int)]
- [(int)]
- [PreferenceItem ()]
- [(float min, float max)]
- [RequireComponent ()]
- [RuntimeInitializeOnLoadMethod]
- [RuntimeInitializeOnLoadMethod (RuntimeInitializeLoadType loadType)]
- [SerializeField]
- [()]
- [TextArea (int minLines, int maxLines)]
- [()]

SerializeField

Unity .

- ()
- [SerializeField] nonstatic serialize .
- .
- .

SerializeField Unity serialize .

- UnityEngine.Object (: GameObject, Component, MonoBehaviour, Texture2D)

- int, string, float, bool
- Vector2 / 3 / 4, Quaternion, Matrix4x4, Color, Rect, LayerMask
-
-
-
-

Examples

```
[Header( "My variables" )]
public string MyString;

[HideInInspector]
public string MyHiddenString;

[Multiline( 5 )]
public string MyMultilineString;

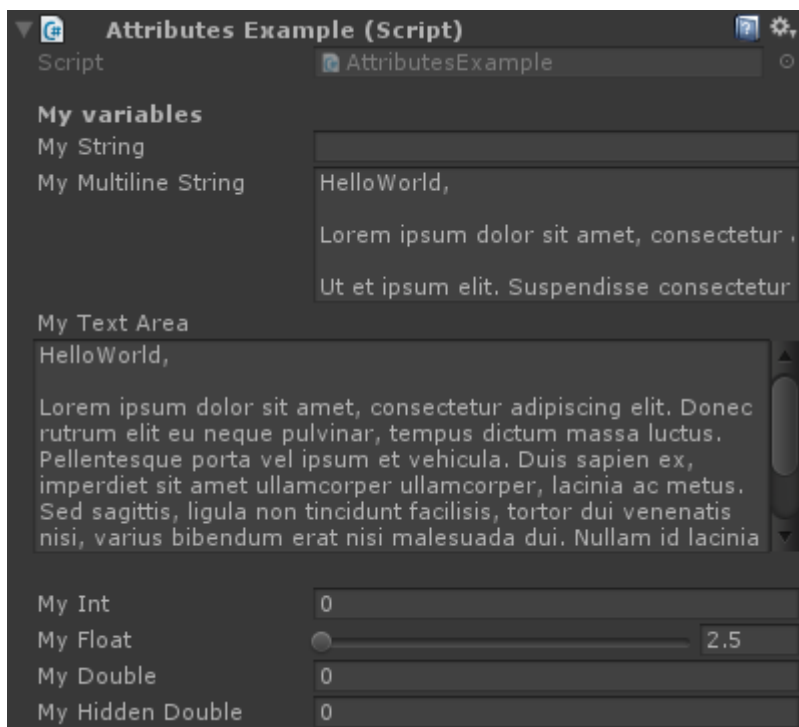
[TextArea( 2, 8 )]
public string MyTextArea;

[Space( 15 )]
public int MyInt;

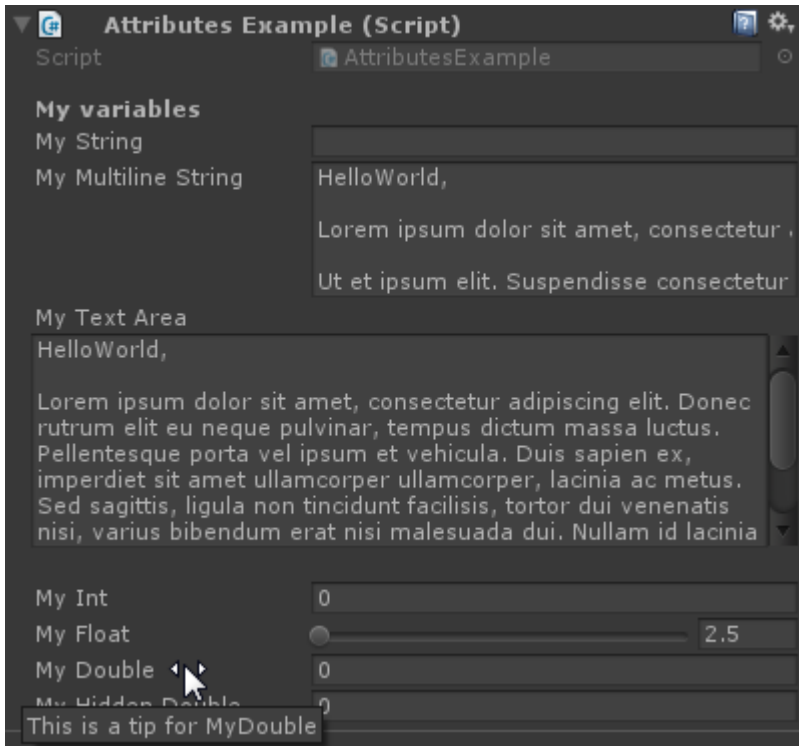
[Range( 2.5f, 12.5f )]
public float MyFloat;

[Tooltip( "This is a tip for MyDouble" )]
public double MyDouble;

[SerializeField]
private double myHiddenDouble;
```



:



```
[Header( "My variables" )]
public string MyString;
```

```
[HideInInspector]
public string MyHiddenString;
```

HideInInspector public inspector . . .

```
[Multiline( 5 )]
public string MyMultilineString;
```

```
[TextArea( 2, 8 )]
public string MyTextArea;
```

TextArea . . .

```
[Space( 15 )]
public int MyInt;
```

```
[Range( 2.5f, 12.5f )]
public float MyFloat;
```

Range . min max . . .

```
[Tooltip( "This is a tip for MyDouble" )]
public double MyDouble;
```

```
[SerializeField]
private double myHiddenDouble;
```

SerializeField Unity - .

```
[DisallowMultipleComponent]
[RequireComponent( typeof( Rigidbody ) )]
public class AttributesExample : MonoBehaviour
{
    [...]
}
```

```
[DisallowMultipleComponent]
```

DisallowMultipleComponent MonoBehaviour .

```
[RequireComponent( typeof( Rigidbody ) )]
```

RequireComponent MonoBehaviour () . MonoBehaviour () .

```
[ExecuteInEditMode]
public class AttributesExample : MonoBehaviour
{
    [RuntimeInitializeOnLoadMethod]
    private static void FooBar()
    {
        [...]
    }

    [RuntimeInitializeOnLoadMethod( RuntimeInitializeLoadType.BeforeSceneLoad )]
    private static void Foo()
    {
        [...]
    }

    [RuntimeInitializeOnLoadMethod( RuntimeInitializeLoadType.AfterSceneLoad )]
    private static void Bar()
    {
        [...]
    }

    void Update()
    {
        if ( Application.isEditor )
        {
            [...]
        }
        else

```

```

        {
            [...]
        }
    }
}

```

```

[ExecuteInEditMode]
public class AttributesExample : MonoBehaviour

```

ExecuteInEditMode Unity .

- .
- **OnGUI** .
- **OnRenderObject** .

```

[RuntimeInitializeOnLoadMethod]
private static void FooBar()

[RuntimeInitializeOnLoadMethod( RuntimeInitializeLoadType.BeforeSceneLoad )]
private static void Foo()

[RuntimeInitializeOnLoadMethod( RuntimeInitializeLoadType.AfterSceneLoad )]
private static void Bar()

```

RuntimeInitializeOnLoadMethod .

(). .

```

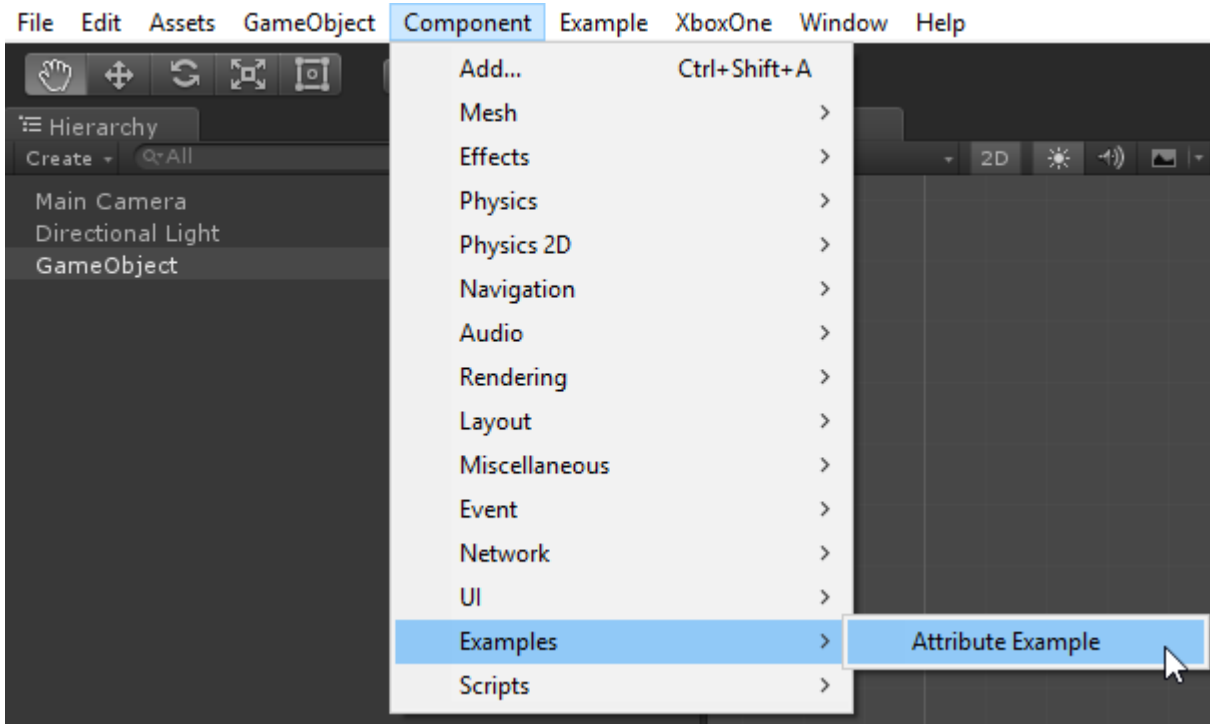
[AddComponentMenu( "Examples/Attribute Example" )]
public class AttributesExample : MonoBehaviour
{
    [ContextMenuItem( "My Field Action", "MyFieldContextAction" )]
    public string MyString;

    private void MyFieldContextAction()
    {
        [...]
    }

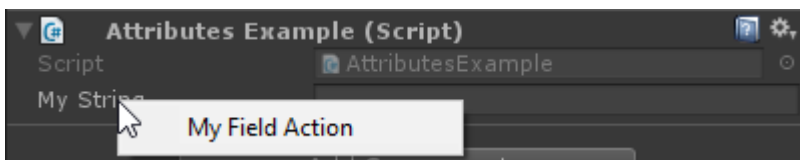
    [ContextMenu( "My Action" )]
    private void MyContextMenuAction()
    {
        [...]
    }
}

```

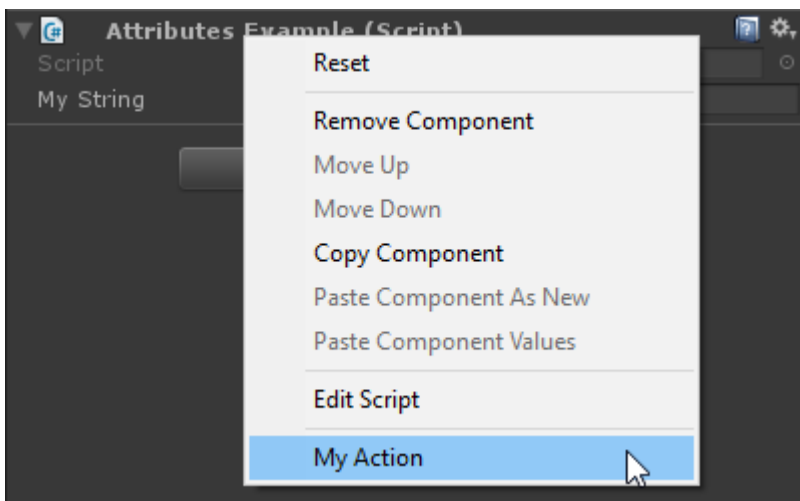
[AddComponentMenu]



[ContextMenuItem]



[ContextMenu]



```
[AddComponentMenu( "Examples/Attribute Example" )]
public class AttributesExample : MonoBehaviour
```

AddComponentMenu -> .

```
[ContextMenu( "My Field Action", "MyFieldContextAction" )]
public string MyString;
```

```
private void MyFieldContextAction()
{
    [...]
}
```

ContextMenuItem . . .

```
[ContextMenu( "My Action" )]
private void MyContextMenuAction()
{
    [...]
}
```

ContextMenu .

```
[InitializeOnLoad]
public class AttributesExample : MonoBehaviour
{
    static AttributesExample()
    {
        [...]
    }

    [InitializeOnLoadMethod]
    private static void Foo()
    {
        [...]
    }
}
```

```
[InitializeOnLoad]
public class AttributesExample : MonoBehaviour
{
    static AttributesExample()
    {
        [...]
    }
}
```

InitializeOnLoad . . .

```
[InitializeOnLoadMethod]
private static void Foo()
{
    [...]
}
```

InitializeOnLoad . . .

```
[CanEditMultipleObjects]
public class AttributesExample : MonoBehaviour
{
```

```

public int MyInt;

private static string prefsText = "";

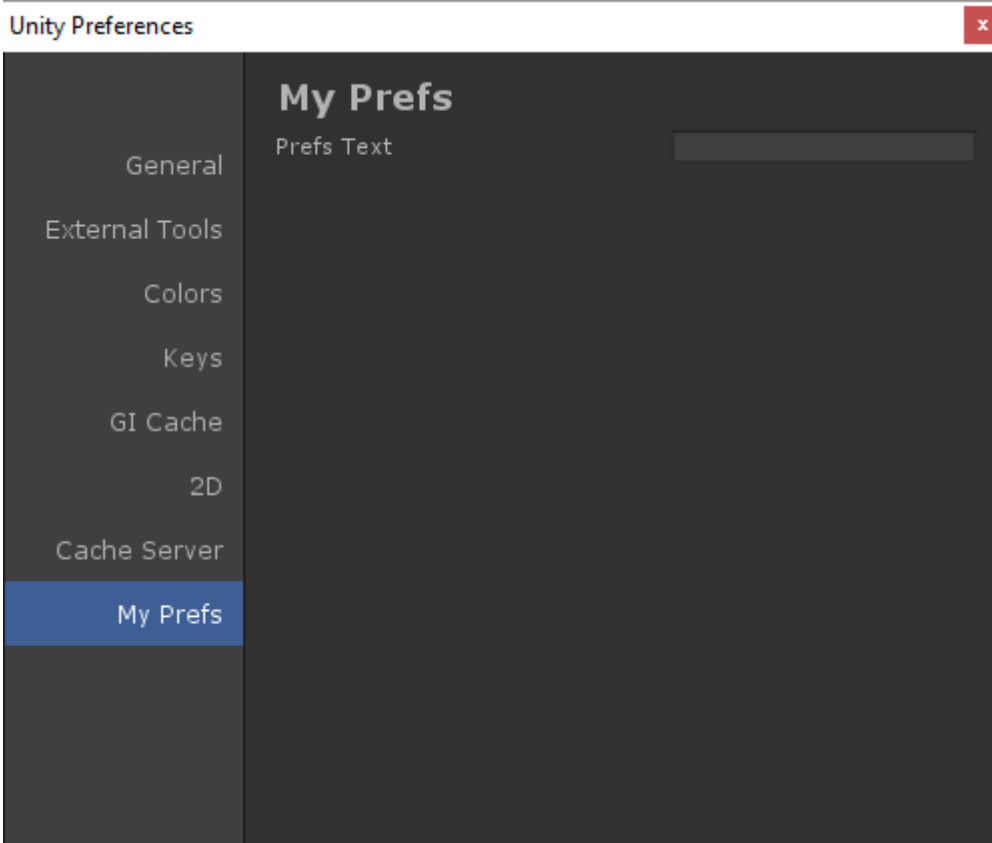
[PreferenceItem( "My Prefs" )]
public static void PreferencesGUI ()
{
    prefsText = EditorGUILayout.TextField( "Prefs Text", prefsText );
}

[MenuItem( "Attributes/Foo" )]
private static void Foo()
{
    [...]
}

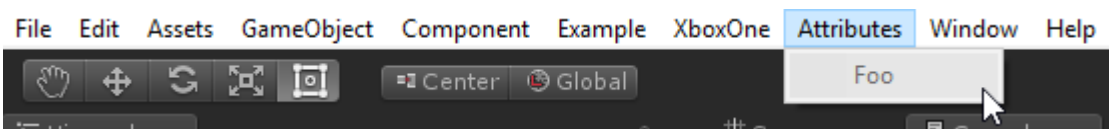
[MenuItem( "Attributes/Foo", true )]
private static bool FooValidate()
{
    return false;
}
}

```

[PreferenceItem]



[MenuItem]



```
[CanEditMultipleObjects]
public class AttributesExample : MonoBehaviour
```

CanEditMultipleObjects **GameObjects** . " " .

. .

```
[PreferenceItem( "My Prefs" )]
public static void PreferencesGUI()
```

PreferenceItem **Unity** . .

```
[MenuItem( "Attributes/Foo" )]
private static void Foo()
{
    [...]
}

[MenuItem( "Attributes/Foo", true )]
private static bool FooValidate()
{
    return false;
}
```

MenuItem . (false) validator .

```
[CustomEditor( typeof( MyComponent ) )]
public class AttributesExample : Editor
{
    [...]
}
```

CustomEditor . **Editor** .

```
[CustomPropertyDrawer( typeof( MyClass ) )]
public class AttributesExample : PropertyDrawer
{
    [...]
}
```

CustomPropertyDrawer . .

```
[DrawGizmo( GizmoType.Selected )]
private static void DoGizmo( AttributesExample obj, GizmoType type )
{
    [...]
}
```

DrawGizmo . . **DrawGizmo** **GizmoType** .

. .

: <https://riptutorial.com/ko/unity3d/topic/5535/>

25: ()

- AssetPostprocessor.OnPreprocessTexture ()

String.Contains() .

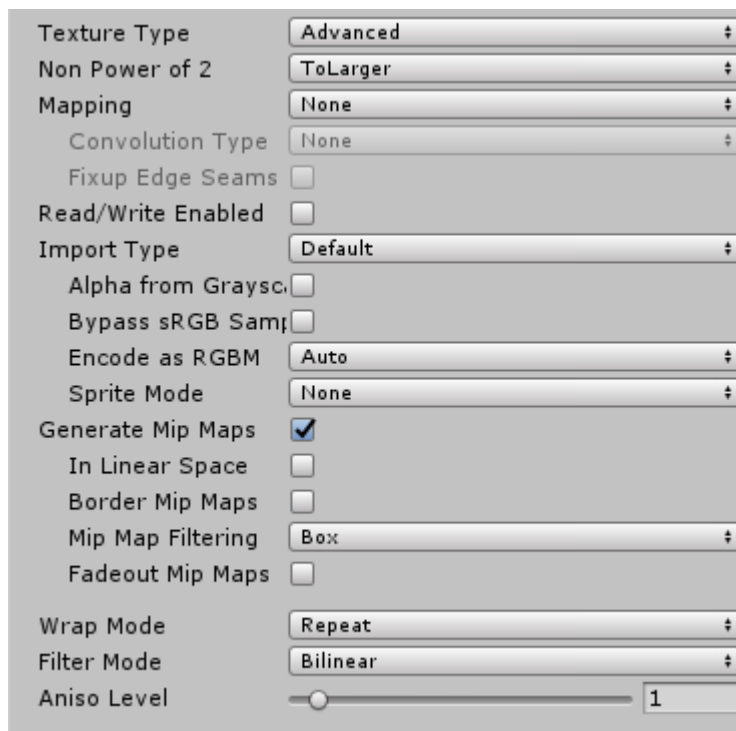
```
if (assetPath.Contains("ProcessThisFolder"))
{
    // Process asset
}
```

Examples

Assets TexturePostProcessor.cs .

```
using UnityEngine;
using UnityEditor;

public class TexturePostProcessor : AssetPostprocessor
{
    void OnPostprocessTexture(Texture2D texture)
    {
        TextureImporter importer = assetImporter as TextureImporter;
        importer.anisoLevel = 1;
        importer.filterMode = FilterMode.Bilinear;
        importer.mipmapEnabled = true;
        importer.npotScale = TextureImporterNPOTScale.ToLarger;
        importer.textureType = TextureImporterType.Advanced;
    }
}
```



.xls . JSON .

JSON

```
{
  "someValue": 123,
  "someOtherValue": 456.297,
  "someBoolValue": true,
  "someStringValue": "this is a string",
}
```

Example.test .

MonoBehaviour . JSON deserialize . TestData.cs .

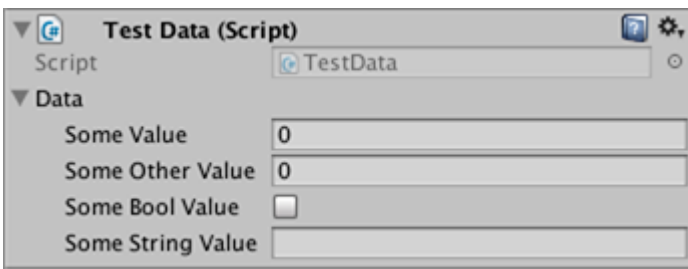
```
using UnityEngine;
using System.Collections;

public class TestData : MonoBehaviour {

    [System.Serializable]
    public class Data {
        public int someValue = 0;
        public float someOtherValue = 0.0f;
        public bool someBoolValue = false;
        public string someStringValue = "";
    }

    public Data data = new Data();
}
```

GameObject ,



Assets Editor . Editor TestDataAssetPostprocessor.cs TestDataAssetPostprocessor.cs .

```
using UnityEditor;
using UnityEngine;
using System.Collections;

public class TestDataAssetPostprocessor : AssetPostprocessor
{
    const string s_extension = ".test";

    // NOTE: Paths start with "Assets/"
}
```

```

static bool IsFileWeCareAbout(string path)
{
    return System.IO.Path.GetExtension(path).Equals(
        s_extension,
        System.StringComparison.Ordinal);
}

static void HandleAddedOrChangedFile(string path)
{
    string text = System.IO.File.ReadAllText(path);
    // should we check for error if the file can't be parsed?
    TestData.Data newData = JsonUtility.FromJson<TestData.Data>(text);

    string prefabPath = path + ".prefab";
    // Get the existing prefab
    GameObject existingPrefab =
        AssetDatabase.LoadAssetAtPath(prefabPath, typeof(Object)) as GameObject;
    if (!existingPrefab)
    {
        // If no prefab exists make one
        GameObject newGameObject = new GameObject();
        newGameObject.AddComponent<TestData>();
        PrefabUtility.CreatePrefab(prefabPath,
            newGameObject,
            ReplacePrefabOptions.Default);
        GameObject.DestroyImmediate(newGameObject);
        existingPrefab =
            AssetDatabase.LoadAssetAtPath(prefabPath, typeof(Object)) as GameObject;
    }

    TestData testData = existingPrefab.GetComponent<TestData>();
    if (testData != null)
    {
        testData.data = newData;
        EditorUtility.SetDirty(existingPrefab);
    }
}

static void HandleRemovedFile(string path)
{
    // Decide what you want to do here. If the source file is removed
    // do you want to delete the prefab? Maybe ask if you'd like to
    // remove the prefab?
    // NOTE: Because you might get many calls (like you deleted a
    // subfolder full of .test files you might want to get all the
    // filenames and ask all at once ("delete all these prefabs?").
}

static void OnPostprocessAllAssets (string[] importedAssets, string[] deletedAssets,
string[] movedAssets, string[] movedFromAssetPaths)
{
    foreach (var path in importedAssets)
    {
        if (IsFileWeCareAbout(path))
        {
            HandleAddedOrChangedFile(path);
        }
    }

    foreach (var path in deletedAssets)
    {

```



```

        if (IsFileWeCareAbout(path))
        {
            HandleRemovedFile(path);
        }
    }

    for (var ii = 0; ii < movedAssets.Length; ++ii)
    {
        string srcStr = movedFromAssetPaths[ii];
        string dstStr = movedAssets[ii];

        // the source was moved, let's move the corresponding prefab
        // NOTE: We don't handle the case if there already being
        // a prefab of the same name at the destination
        string srcPrefabPath = srcStr + ".prefab";
        string dstPrefabPath = dstStr + ".prefab";

        AssetDatabase.MoveAsset(srcPrefabPath, dstPrefabPath);
    }
}
}
}

```

Example.test **Unity Assets** . Example.test . Example.test . Example.test .
 Example.test .

: , Example.test Assets Example.test Example.test.prefab . Example.test AssetBundle .

() : <https://riptutorial.com/ko/unity3d/topic/5279/----->

26:

- [MenuItem (string itemName)]
- [MenuItem (string itemName, bool isValidFunction)]
- [MenuItem (itemName, bool isValidFunction, int)]
- [ContextMenu ()]
- [ContextMenu (,)]
- [(GizmoType gizmo)]
- [DrawGizmo (GizmoType , drawnGizmoType)]

MenuCommand	MenuCommand, MenuItem .
MenuCommand.context	
MenuCommand.userData	int.

Examples

Inspector . . .

```
using UnityEngine;
#if UNITY_EDITOR
using UnityEditor;
#endif

public class InspectorExample : MonoBehaviour {

    public int Level;
    public float BaseDamage;

    public float DamageBonus {
        get {
            return Level / 100f * 50;
        }
    }

    public float ActualDamage {
        get {
            return BaseDamage + DamageBonus;
        }
    }
}

#if UNITY_EDITOR
[CustomEditor( typeof( InspectorExample ) )]
public class CustomInspector : Editor {

    public override void OnInspectorGUI() {
        base.OnInspectorGUI();
    }
}
#endif
```

```

        var ie = (InspectorExample)target;

        EditorGUILayout.LabelField( "Damage Bonus", ie.DamageBonus.ToString() );
        EditorGUILayout.LabelField( "Actual Damage", ie.ActualDamage.ToString() );
    }
}
#endif

```

```

public class InspectorExample : MonoBehaviour {
    public int Level;
    public float BaseDamage;
}

```

Inspector ().

```

public float DamageBonus {
    get {
        return Level / 100f * 50;
    }
}

public float ActualDamage {
    get {
        return BaseDamage + DamageBonus;
    }
}

```

Unity . Inspector Custom Inspector .

custom inspector .

```

[CustomEditor( typeof( InspectorExample ) )]
public class CustomInspector : Editor {

```

Editor CustomEditor . .

OnInspectorGUI . .

```

public override void OnInspectorGUI() {
    base.OnInspectorGUI();
}

```

base.OnInspectorGUI () Unity . .

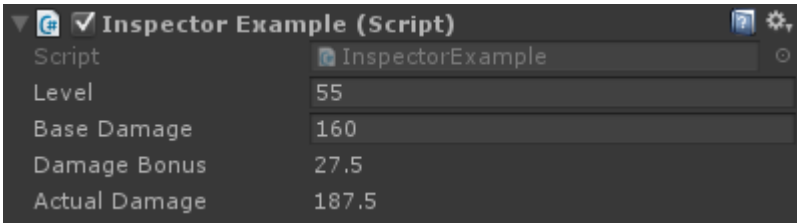
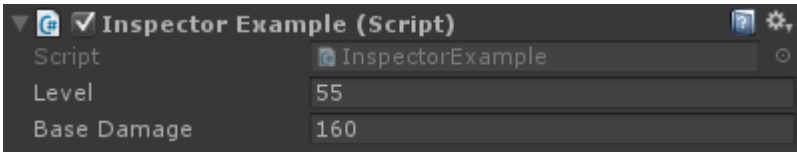
```

var ie = (InspectorExample)target;

EditorGUILayout.LabelField( "Damage Bonus", ie.DamageBonus.ToString() );
EditorGUILayout.LabelField( "Actual Damage", ie.ActualDamage.ToString() );

```

(Editor).



MonoBehaviour .

MonoBehaviour .

MonoBehaviour .

```
public enum Gender {
    Male,
    Female,
    Other
}

// Needs the Serializable attribute otherwise the CustomPropertyDrawer wont be used
[Serializable]
public class UserInfo {
    public string Name;
    public int Age;
    public Gender Gender;
}

// The class that you can attach to a GameObject
public class PropertyDrawerExample : MonoBehaviour {
    public UserInfo UInfo;
}

[CustomPropertyDrawer( typeof( UserInfo ) )]
public class UserInfoDrawer : PropertyDrawer {

    public override float GetPropertyHeight( SerializedProperty property, GUIContent label ) {
        // The 6 comes from extra spacing between the fields (2px each)
        return EditorGUIUtility.singleLineHeight * 4 + 6;
    }

    public override void OnGUI( Rect position, SerializedProperty property, GUIContent label )
    {
        EditorGUI.BeginProperty( position, label, property );

        EditorGUI.LabelField( position, label );

        var nameRect = new Rect( position.x, position.y + 18, position.width, 16 );
        var ageRect = new Rect( position.x, position.y + 36, position.width, 16 );
        var genderRect = new Rect( position.x, position.y + 54, position.width, 16 );

        EditorGUI.indentLevel++;
```

```

    EditorGUI.PropertyField( nameRect, property.FindPropertyRelative( "Name" ) );
    EditorGUI.PropertyField( ageRect, property.FindPropertyRelative( "Age" ) );
    EditorGUI.PropertyField( genderRect, property.FindPropertyRelative( "Gender" ) );

    EditorGUI.indentLevel--;

    EditorGUI.EndProperty();
}
}

```

. . PropertyDrawerExample GameObject .

```

public enum Gender {
    Male,
    Female,
    Other
}

[Serializable]
public class UserInfo {
    public string Name;
    public int Age;
    public Gender Gender;
}

public class PropertyDrawerExample : MonoBehaviour {
    public UserInfo UInfo;
}

```

Serializable . CustomPropertyDrawer .

CustomPropertyDrawer.

PropertyDrawer . CustomPropertyDrawer . .

```

[CustomPropertyDrawer( typeof( UserInfo ) )]
public class UserInfoDrawer : PropertyDrawer {

```

GetPropertyHeight . . , , . *EditorGUIUtility.singleLineHeight* * 4 6 .

```

public override float GetPropertyHeight( SerializedProperty property, GUIContent label ) {
    return EditorGUIUtility.singleLineHeight * 4 + 6;
}

```

OnGUI . *EditorGUI.BeginProperty ([..])* *EditorGUI.EndProperty ()* . .

```

public override void OnGUI( Rect position, SerializedProperty property, GUIContent label ) {
    EditorGUI.BeginProperty( position, label, property );

```

```

    EditorGUI.LabelField( position, label );

```

```

var nameRect = new Rect( position.x, position.y + 18, position.width, 16 );
var ageRect = new Rect( position.x, position.y + 36, position.width, 16 );
var genderRect = new Rect( position.x, position.y + 54, position.width, 16 );

```

16 + 2 * 16 (EditorGUIUtility.singleLineHeight).

UI , GUI , *EditorGUI.EndProperty* .

```

EditorGUI.indentLevel++;

EditorGUI.PropertyField( nameRect, property.FindPropertyRelative( "Name" ) );
EditorGUI.PropertyField( ageRect, property.FindPropertyRelative( "Age" ) );
EditorGUI.PropertyField( genderRect, property.FindPropertyRelative( "Gender" ) );

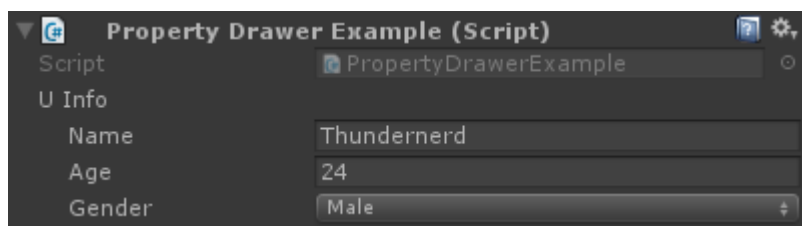
EditorGUI.indentLevel--;

EditorGUI.EndProperty();

```

EditorGUI.PropertyField SerializedProperty . OnGUI FindPropertyRelative ("...") .

property.FindPropertyRelative ("...") .



```

public class MenuItemsExample : MonoBehaviour {

    [MenuItem( "Example/DoSomething %#&d" )]
    private static void DoSomething() {
        // Execute some code
    }

    [MenuItem( "Example/DoAnotherThing", true )]
    private static bool DoAnotherThingValidator() {
        return Selection.gameObjects.Length > 0;
    }

    [MenuItem( "Example/DoAnotherThing _PGUP", false )]
    private static void DoAnotherThing() {
        // Execute some code
    }

    [MenuItem( "Example/DoOne %a", false, 1 )]
    private static void DoOne() {
        // Execute some code
    }
}

```

```

}

[MenuItem( "Example/DoTwo #b", false, 2 )]
private static void DoTwo() {
    // Execute some code
}

[MenuItem( "Example/DoFurther &c", false, 13 )]
private static void DoFurther() {
    // Execute some code
}

[MenuItem( "CONTEXT/Camera/DoCameraThing" )]
private static void DoCameraThing( MenuCommand cmd ) {
    // Execute some code
}

[ContextMenu( "ContextSomething" )]
private void ContentSomething() {
    // Execute some code
}

[ContextMenu( "Reset", "ResetDate" )]
[ContextMenu( "Set to Now", "SetDateToNow" )]
public string Date = "";

public void ResetDate() {
    Date = "";
}

public void SetDateToNow() {
    Date = DateTime.Now.ToString();
}
}

```

?

Example	Window	Help
DoOne		Ctrl+A
DoTwo		Shift+B
DoFurther		Alt+C
DoSomething		Ctrl+Shift+Alt+D
DoAnotherThing		PgUp

. *MenuItem* . / .

```

[MenuItem( "Example/DoSomething %#&d" )]
private static void DoSomething() {
    // Execute some code
}

```

. !

MenuItem .

- % - Windows Ctrl, OS X Cmd
- # - Shift
- & - Alt

% # & d Windows ctrl + shift + alt + D, OS X cmd + shift + alt + D .

(: 'D'), _ () .

- , , , -
- F1..F12 -
- HOME, END, PGUP, PGDN -

() . , GameObject , .

```
[MenuItem( "Example/DoAnotherThing", true )]
private static bool DoAnotherThingValidator() {
    return Selection.gameObjects.Length > 0;
}

[MenuItem( "Example/DoAnotherThing _PGUP", false )]
private static void DoAnotherThing() {
    // Execute some code
}
```

MenuItem () . .

10 .

```
[MenuItem( "Example/DoOne %a", false, 1 )]
private static void DoOne() {
    // Execute some code
}

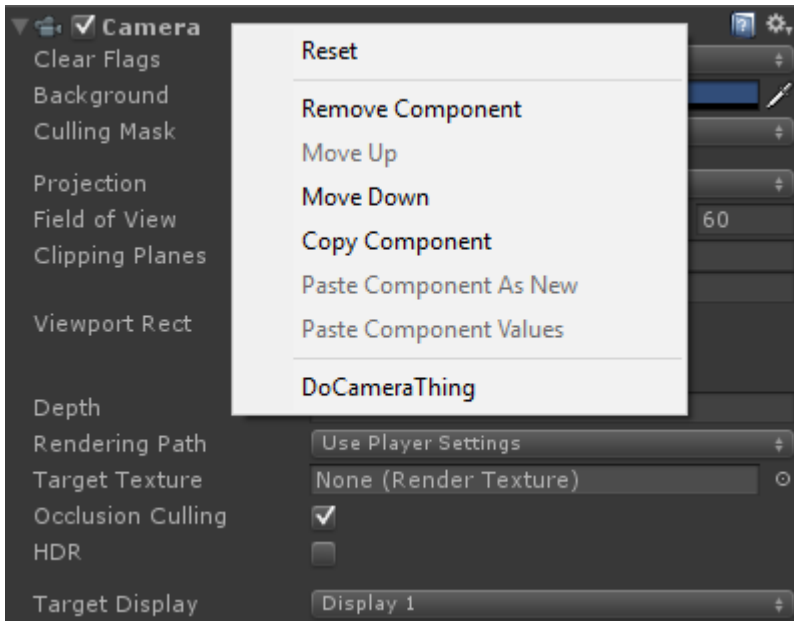
[MenuItem( "Example/DoTwo #b", false, 2 )]
private static void DoTwo() {
    // Execute some code
}

[MenuItem( "Example/DoFurther &c", false, 13 )]
private static void DoFurther() {
    // Execute some code
}
```

. CONTEXT () MenuItem MenuCommand .


```
[MenuItem( "CONTEXT/Camera/DoCameraThing" )]
private static void DoCameraThing( MenuCommand cmd ) {
    // Execute some code
}
```

?

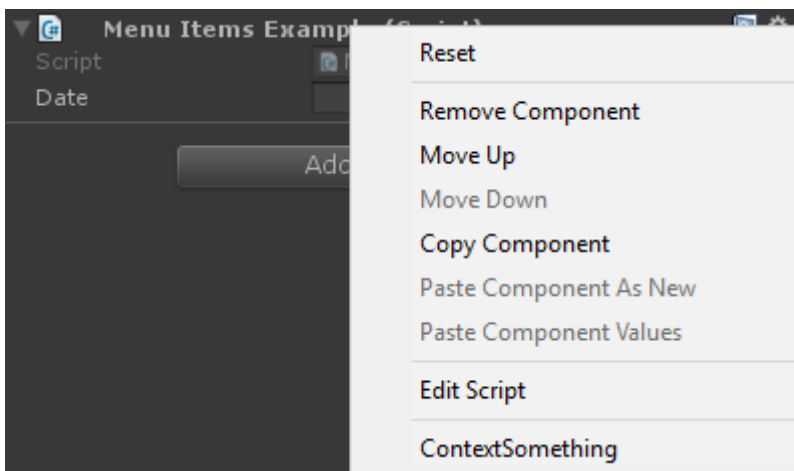


MenuCommand .

ContextMenu . . .

```
[ContextMenu( "ContextSomething" )]
private void ContentSomething() {
    // Execute some code
}
```

?



```
[ContextMenu( "Reset", "ResetDate" )]
[ContextMenu( "Set to Now", "SetDateToNow" )]
```

```

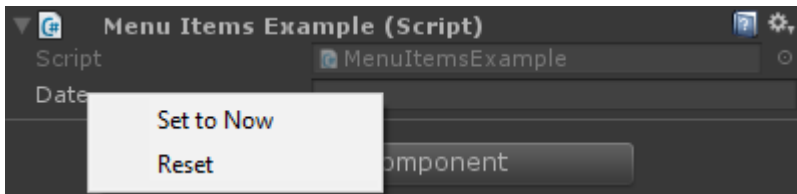
public string Date = "";

public void ResetDate() {
    Date = "";
}

public void SetDateToNow() {
    Date = DateTime.Now.ToString();
}

```

?



Gizmos . GameObjects (:) .

.

1

OnDrawGizmos *OnDrawGizmosSelected* (magic) .

```

public class GizmoExample : MonoBehaviour {

    public float GetDetectionRadius() {
        return 12.5f;
    }

    public float GetFOV() {
        return 25f;
    }

    public float GetMaxRange() {
        return 6.5f;
    }

    public float GetMinRange() {
        return 0;
    }

    public float GetAspect() {
        return 2.5f;
    }

    public void OnDrawGizmos() {
        var gizmoMatrix = Gizmos.matrix;
        var gizmoColor = Gizmos.color;

        Gizmos.matrix = Matrix4x4.TRS( transform.position, transform.rotation,
transform.lossyScale );
        Gizmos.color = Color.red;
    }
}

```

```

        Gizmos.DrawFrustum( Vector3.zero, GetFOV(), GetMaxRange(), GetMinRange(), GetAspect()
    );

    Gizmos.matrix = gizmoMatrix;
    Gizmos.color = gizmoColor;
}

public void OnDrawGizmosSelected() {
    Handles.DrawWireDisc( transform.position, Vector3.up, GetDetectionRadius() );
}
}

```

, (OnDrawGizmos) (OnDrawGizmosSelected) .

```

public void OnDrawGizmos() {
    var gizmoMatrix = Gizmos.matrix;
    var gizmoColor = Gizmos.color;

    Gizmos.matrix = Matrix4x4.TRS( transform.position, transform.rotation,
transform.lossyScale );
    Gizmos.color = Color.red;
    Gizmos.DrawFrustum( Vector3.zero, GetFOV(), GetMaxRange(), GetMinRange(), GetAspect() );

    Gizmos.matrix = gizmoMatrix;
    Gizmos.color = gizmoColor;
}

```

Gizmos , . Gizmos . *Gizmos.DrawFrustum* .

```

public void OnDrawGizmosSelected() {
    Handles.DrawWireDisc( transform.position, Vector3.up, GetDetectionRadius() );
}

```

GameObject . *Gizmos* *Handles* .

2

DrawGizmo .

```

public class GizmoDrawerExample {

    [DrawGizmo( GizmoType.Selected | GizmoType.NonSelected, typeof( GizmoExample ) )]
    public static void DrawGizmo( GizmoExample obj, GizmoType type ) {
        var gizmoMatrix = Gizmos.matrix;
        var gizmoColor = Gizmos.color;

        Gizmos.matrix = Matrix4x4.TRS( obj.transform.position, obj.transform.rotation,
obj.transform.lossyScale );
    }
}

```

```

    Gizmos.color = Color.red;
    Gizmos.DrawFrustum( Vector3.zero, obj.GetFOV(), obj.GetMaxRange(), obj.GetMinRange(),
obj.GetAspect() );

    Gizmos.matrix = gizmoMatrix;
    Gizmos.color = gizmoColor;

    if ( ( type & GizmoType.Selected ) == GizmoType.Selected ) {
        Handles.DrawWireDisc( obj.transform.position, Vector3.up, obj.GetDetectionRadius()
);
    }
}
}

```

. .

```

[DrawGizmo( GizmoType.Selected | GizmoType.NonSelected, typeof( GizmoExample ) )]
public static void DrawGizmo( GizmoExample obj, GizmoType type ) {

```

GizmoType ,Type DrawGizmo . .

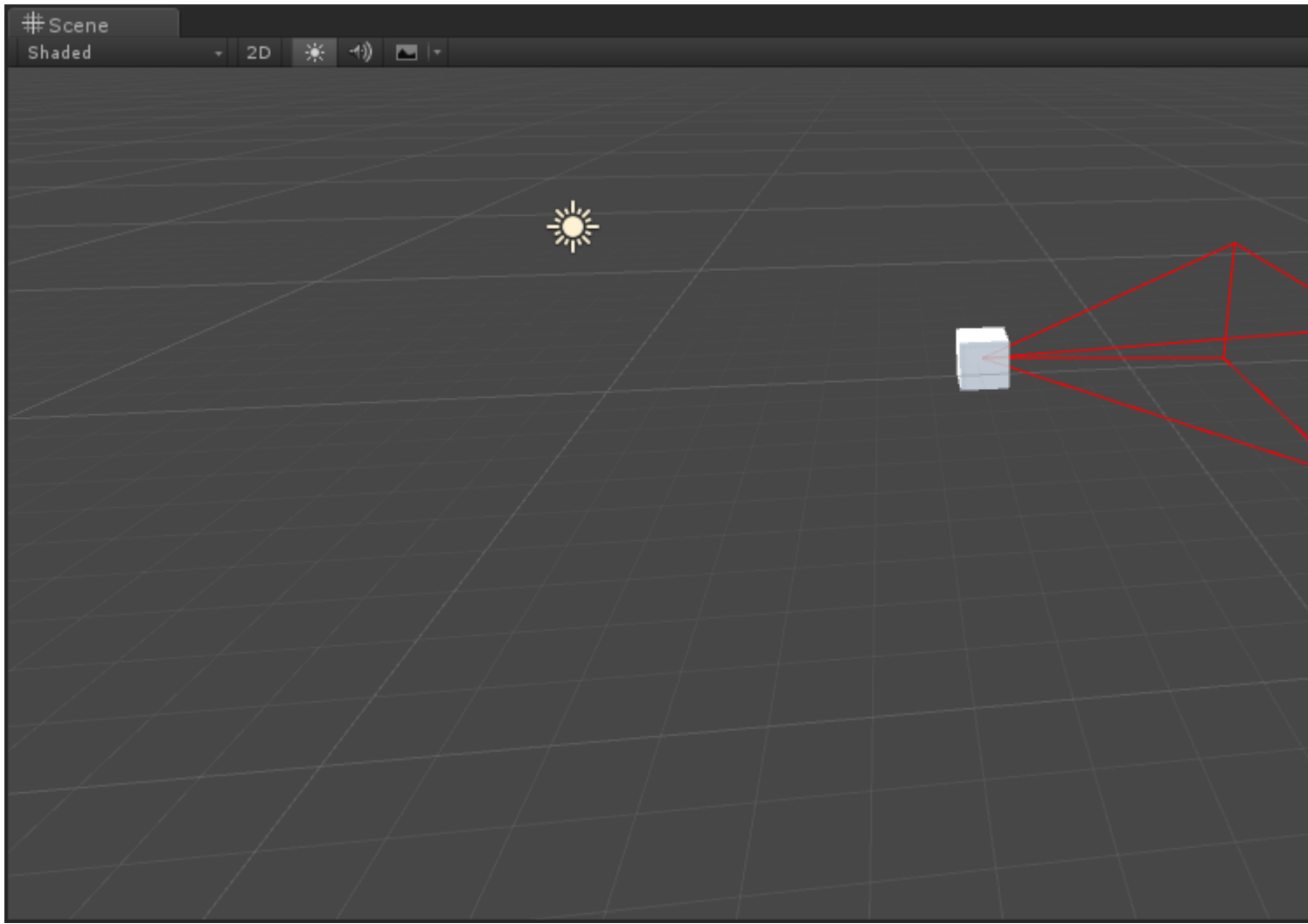
, . GizmoType.

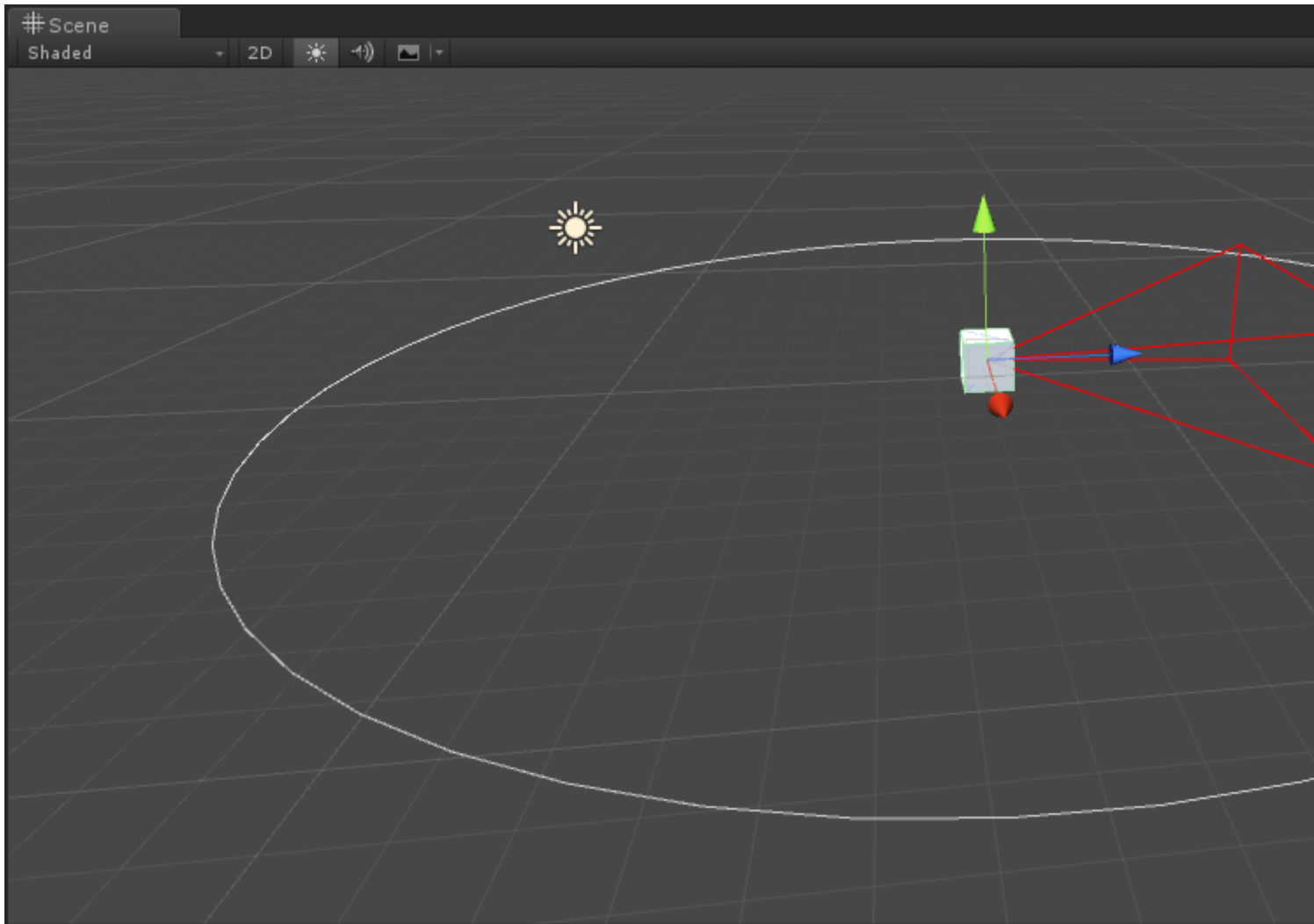
```

if ( ( type & GizmoType.Selected ) == GizmoType.Selected ) {
    Handles.DrawWireDisc( obj.transform.position, Vector3.up, obj.GetDetectionRadius() );
}

```

GizmoType AND .





?

(<http://www.riptutorial.com/unity3d/topic/2506/extend-the-editor> . [EditorWindow](#)
Unity UI Editor Windows () .

EditorWindow

. EditorWindow Init () OnGUI () . .

```
using UnityEngine;
using UnityEditor;

public class CustomWindow : EditorWindow
{
    // Add menu named "Custom Window" to the Window menu
    [MenuItem("Window/Custom Window")]
    static void Init()
    {
        // Get existing open window or if none, make a new one:
        CustomWindow window = (CustomWindow) EditorWindow.GetWindow(typeof(CustomWindow));
        window.Show();
    }
}
```

```

void OnGUI()
{
    GUILayout.Label("This is a custom Editor Window", EditorStyles.boldLabel);
}
}

```

3 .

1. EditorWindow .
2. Init () . [EditorWindow.GetWindow](#) CustomWindow
3. OnGUI ()

CustomWindow

This is a custom Editor Window

EditorWindow . [Selection \(\) SerializedObject SerializedProperty](#) .

```

using System.Linq;
using UnityEngine;
using UnityEditor;

public class CustomWindow : EditorWindow
{
    private AnimationClip _animationClip;
    private SerializedObject _serializedClip;
    private SerializedProperty _events;

    private string _text = "Hello World";

    // Add menu named "Custom Window" to the Window menu
    [MenuItem("Window/Custom Window")]
    static void Init()
    {
        // Get existing open window or if none, make a new one:
        CustomWindow window = (CustomWindow) EditorWindow.GetWindow(typeof(CustomWindow));
    }
}

```

```

        window.Show();
    }

    void OnGUI()
    {
        GUILayout.Label("This is a custom Editor Window", EditorStyles.boldLabel);

        // You can use EditorGUI, EditorGUILayout and GUILayout classes to display
anything you want
        // A TextField example
        _text = EditorGUILayout.TextField("Text Field", _text);

        // Note that you can modify an asset or a gameobject using an EditorWindow. Here
is a quick example with an AnimationClip asset
        // The _animationClip, _serializedClip and _events are set in OnSelectionChange()

        if (_animationClip == null || _serializedClip == null || _events == null) return;

        // We can modify our serializedClip like we would do in a Custom Inspector. For
example we can grab its events and display their information

        GUILayout.Label(_animationClip.name, EditorStyles.boldLabel);

        for (var i = 0; i < _events.arraySize; i++)
        {
            EditorGUILayout.BeginVertical();

            EditorGUILayout.LabelField(
                "Event : " +
_events.GetArrayElementAtIndex(i).FindPropertyRelative("functionName").stringValue,
                EditorStyles.boldLabel);

            EditorGUILayout.PropertyField(_events.GetArrayElementAtIndex(i).FindPropertyRelative("time"),
true,
                GUILayout.ExpandWidth(true));

            EditorGUILayout.PropertyField(_events.GetArrayElementAtIndex(i).FindPropertyRelative("functionName"),
true, GUILayout.ExpandWidth(true));

            EditorGUILayout.PropertyField(_events.GetArrayElementAtIndex(i).FindPropertyRelative("floatParameter"),
true, GUILayout.ExpandWidth(true));

            EditorGUILayout.PropertyField(_events.GetArrayElementAtIndex(i).FindPropertyRelative("intParameter"),
true, GUILayout.ExpandWidth(true));

            EditorGUILayout.PropertyField(
_events.GetArrayElementAtIndex(i).FindPropertyRelative("objectReferenceParameter"), true,
                GUILayout.ExpandWidth(true));

            EditorGUILayout.Separator();
            EditorGUILayout.EndVertical();
        }

        // Of course we need to Apply the modified properties. We don't our changes won't
be saved
        _serializedClip.ApplyModifiedProperties();
    }

```

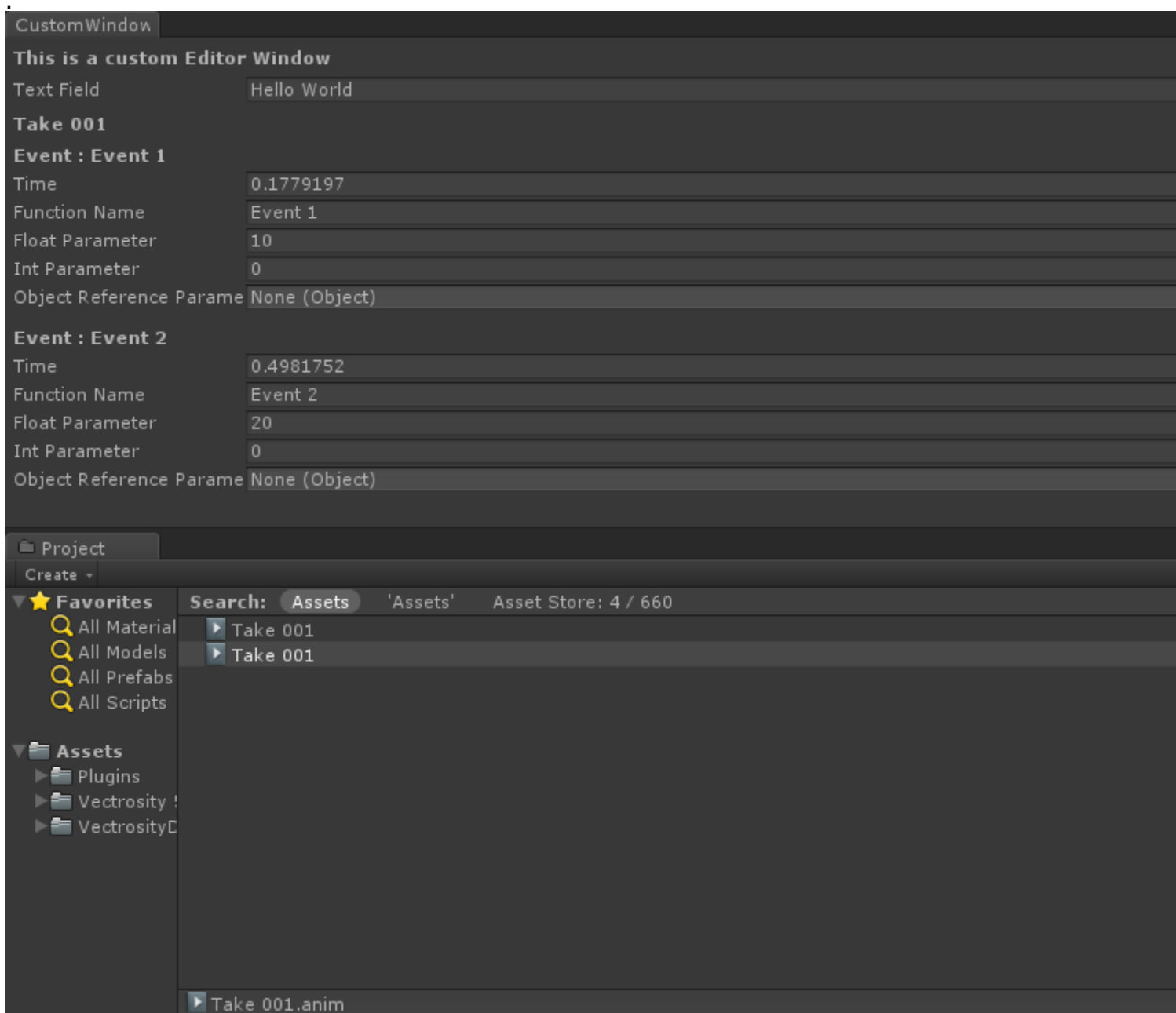


```

    /// This Message is triggered when the user selection in the editor changes. That's
    when we should tell our Window to Repaint() if the user selected another AnimationClip
    private void OnSelectionChange()
    {
        _animationClip =
            Selection.GetFiltered(typeof(AnimationClip),
            SelectionMode.Assets).FirstOrDefault() as AnimationClip;
        if (_animationClip == null) return;

        _serializedClip = new SerializedObject(_animationClip);
        _events = _serializedClip.FindProperty("m_Events");
        Repaint();
    }
}

```



EditorWindow . (NodeCanvas PlayMaker) EditorWindow .

SceneView

EditorWindow SceneView . EditorWindow SceneView / . . .

```
using UnityEngine;
using System;
using UnityEditor;

public class CustomWindow : EditorWindow {

    private enum Mode {
        View = 0,
        Paint = 1,
        Erase = 2
    }

    private Mode CurrentMode = Mode.View;

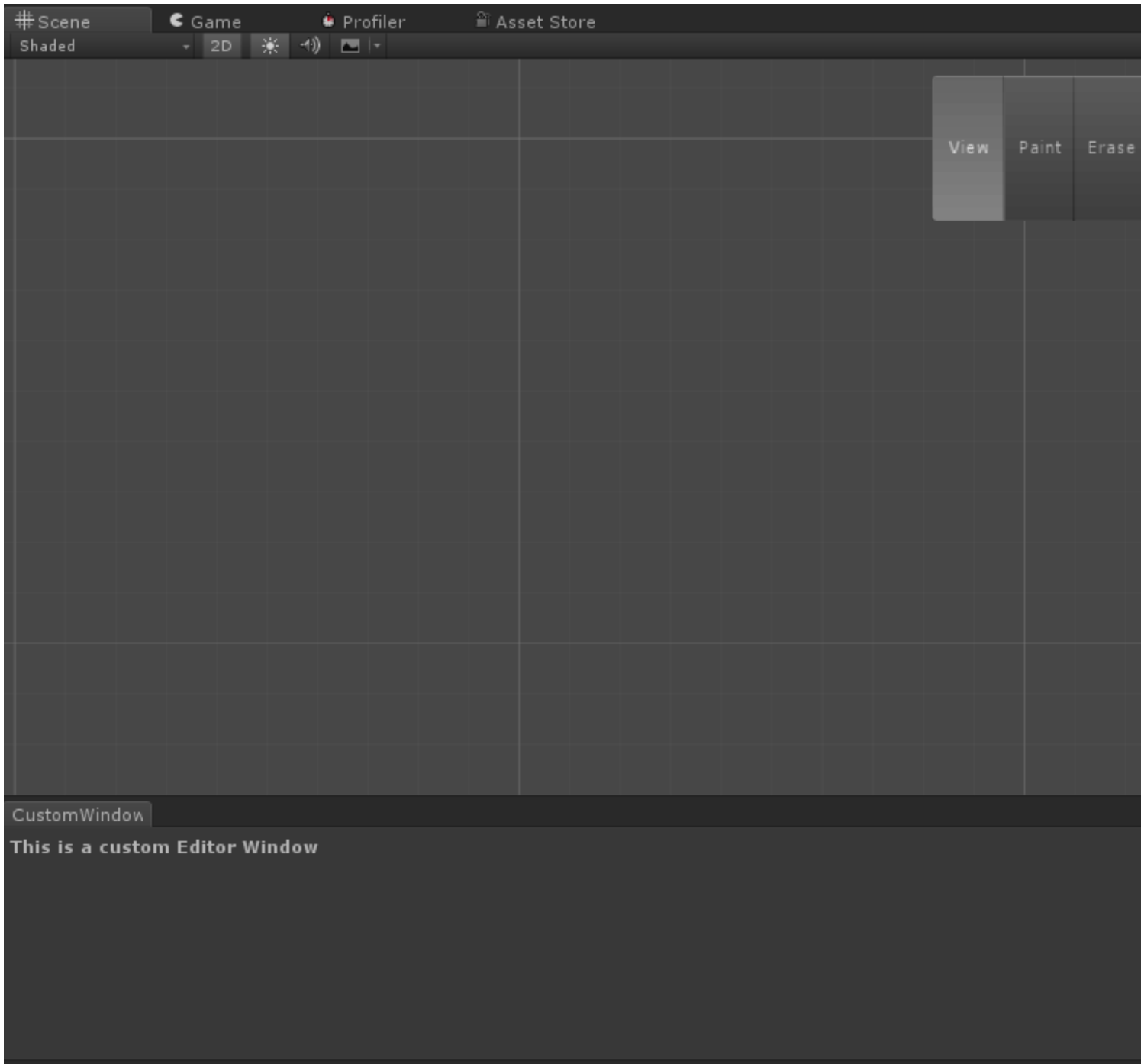
    [MenuItem ("Window/Custom Window")]
    static void Init () {
        // Get existing open window or if none, make a new one:
        CustomWindow window = (CustomWindow)EditorWindow.GetWindow (typeof (CustomWindow));
        window.Show ();
    }

    void OnGUI () {
        GUILayout.Label ("This is a custom Editor Window", EditorStyles.boldLabel);
    }

    void OnEnable () {
        SceneView.onSceneGUIDelegate = SceneViewGUI;
        if (SceneView.lastActiveSceneView) SceneView.lastActiveSceneView.Repaint ();
    }

    void SceneViewGUI(SceneView sceneView) {
        Handles.BeginGUI ();
        // We define the toolbars' rects here
        var ToolBarRect = new Rect ((SceneView.lastActiveSceneView.camera.pixelRect.width / 6),
10, (SceneView.lastActiveSceneView.camera.pixelRect.width * 4 / 6) ,
SceneView.lastActiveSceneView.camera.pixelRect.height / 5);
        GUILayout.BeginArea (ToolBarRect);
        GUILayout.BeginHorizontal ();
        GUILayout.FlexibleSpace ();
        CurrentMode = (Mode) GUILayout.Toolbar (
            (int) CurrentMode,
            Enum.GetNames (typeof (Mode)),
            GUILayout.Height (ToolBarRect.height));
        GUILayout.FlexibleSpace ();
        GUILayout.EndHorizontal ();
        GUILayout.EndArea ();
        Handles.EndGUI ();
    }
}
```

SceneView



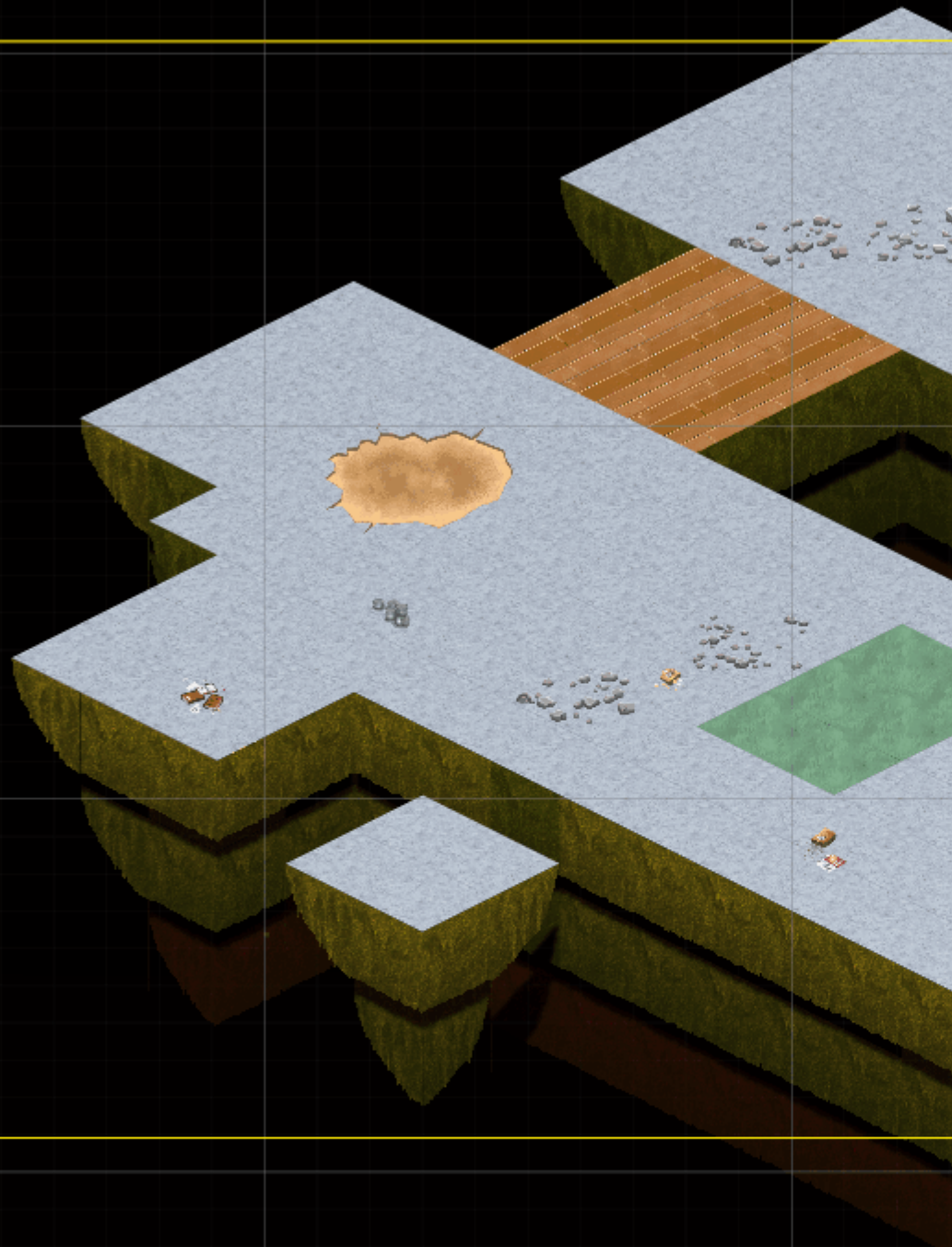
Position sceneView camera

- Camera Lock
- Draw Walkable Gizmo
- Draw Cover Gizmo
- Hide Map Hierarchy
- Show grid
- Draw GridCell Neighbors

Dimensions:

+ - + -

+ - + -



Map Editor

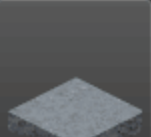




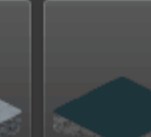
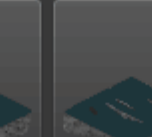





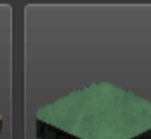
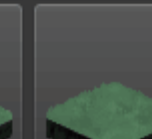


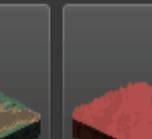

Project

Console Pro 3

Palette

Search Term :

[Grid]

 Aeroport_01	 Aeroport_02	 Aeroport_03	 Aeroport_04	 Aeroport_05	 petAeroport	 petAeroport	 petAeroport	 arpetBlue_
								

27:

Unity3D .

Examples

-

```
using UnityEngine;

public class Audio : MonoBehaviour {
    AudioSource audioSource;
    AudioClip audioClip;

    void Start() {
        audioClip = (AudioClip)Resources.Load("Audio/Soundtrack");
        audioSource.clip = audioClip;
        if (!audioSource.isPlaying) audioSource.Play();
    }
}
```

: [https://riptutorial.com/ko/unity3d/topic/8064/-](https://riptutorial.com/ko/unity3d/topic/8064/)

28:

Examples

Unity .

2 . Stand-In-Place . Animator . . .

Windows Animator . 2 Animator 2 . bool . "PerformRun", "PerformIdle" .
"PerformIdle" true .

(.) .-> HasExit . . -> : PerformIdle. Run-> Idle PerformRun . C# . . .

```
using UnityEngine;
using System.Collections;

public class RootMotion : MonoBehaviour {

    //Public Variables
    [Header("Transform Variables")]
    public float RunSpeed = 0.1f;
    public float TurnSpeed = 6.0f;

    Animator animator;

    void Start()
    {
        /**
         * Initialize the animator that is attached on the current game object i.e. on which you
         will attach this script.
         */
        animator = GetComponent<Animator>();
    }

    void Update()
    {
        /**
         * The Update() function will get the bool parameters from the animator state machine and
         set the values provided by the user.
         * Here, I have only added animation for Run and Idle. When the Up key is pressed, Run
         animation is played. When we let go, Idle is played.
         */

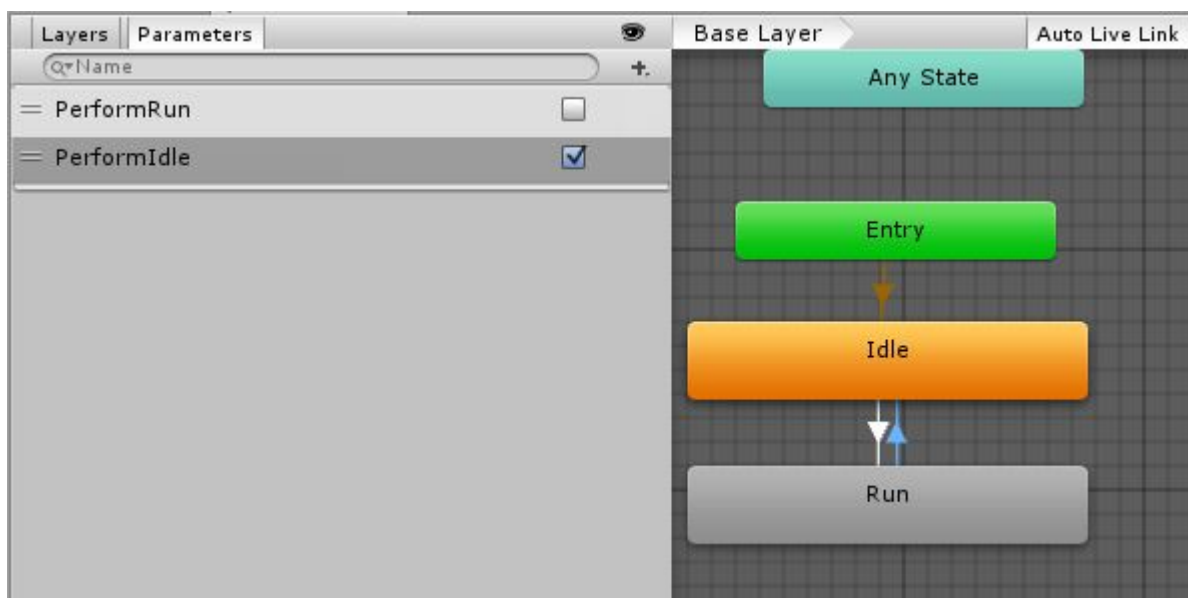
        if (Input.GetKey (KeyCode.UpArrow)) {
            animator.SetBool ("PerformRun", true);
            animator.SetBool ("PerformIdle", false);
        } else {
            animator.SetBool ("PerformRun", false);
            animator.SetBool ("PerformIdle", true);
        }
    }

    void OnAnimatorMove()
    {
        /**
```

```

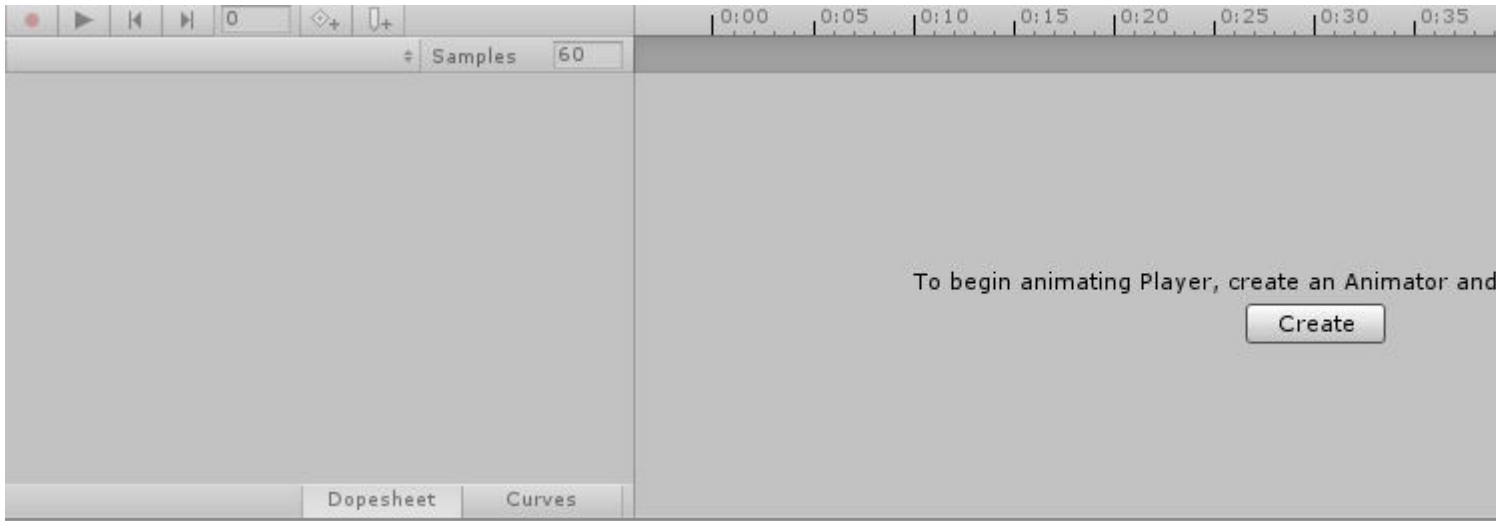
    * OnAnimatorMove() function will shadow the "Apply Root Motion" on the animator. Your
game objects position will now be determined
    * using this function.
    */
    if (Input.GetKey (KeyCode.UpArrow)){
        transform.Translate (Vector3.forward * RunSpeed);
        if (Input.GetKey (KeyCode.RightArrow)) {
            transform.Rotate (Vector3.up * Time.deltaTime * TurnSpeed);
        }
        else if (Input.GetKey (KeyCode.LeftArrow)) {
            transform.Rotate (-Vector3.up * Time.deltaTime * TurnSpeed);
        }
    }
}
}
}

```

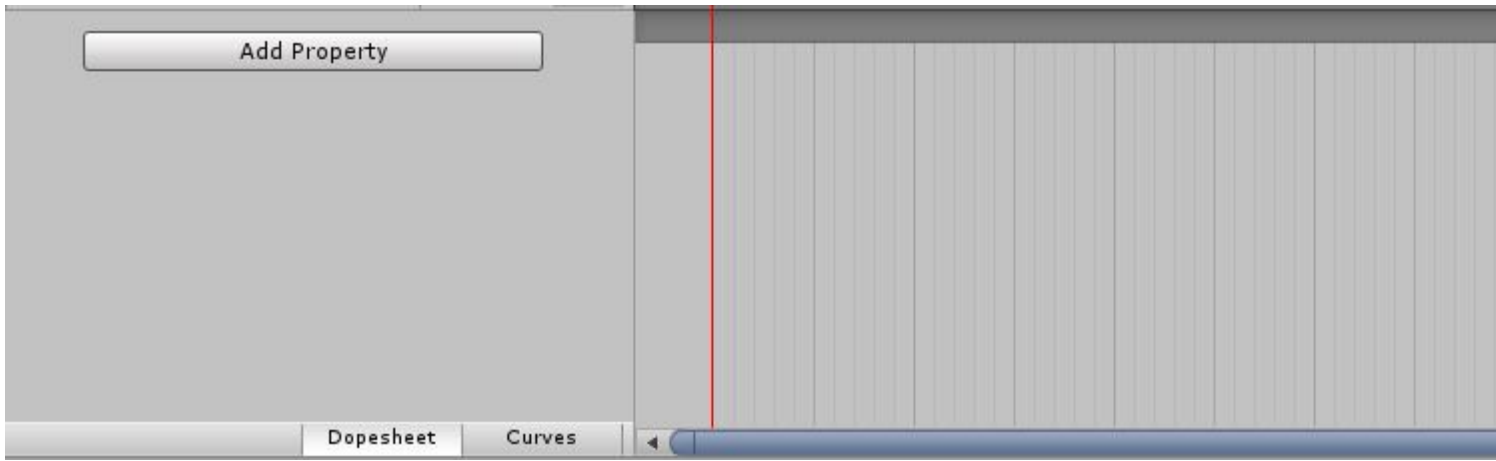


Unity Asset Store . . <https://www.assetstore.unity3d.com/en/#!/content/21874>

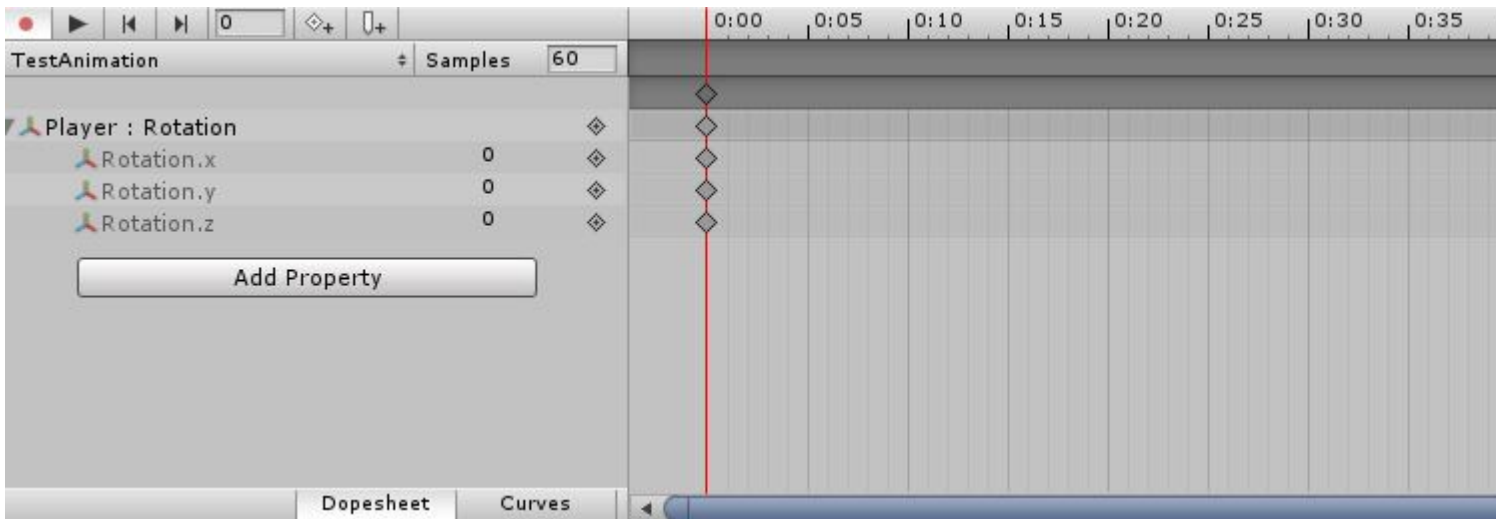
. Ctrl + 6 . . .



IdlePlayer, SprintPlayer, DyingPlayer . . . , Tranform anyother (: Collider, Mesh Renderer) .



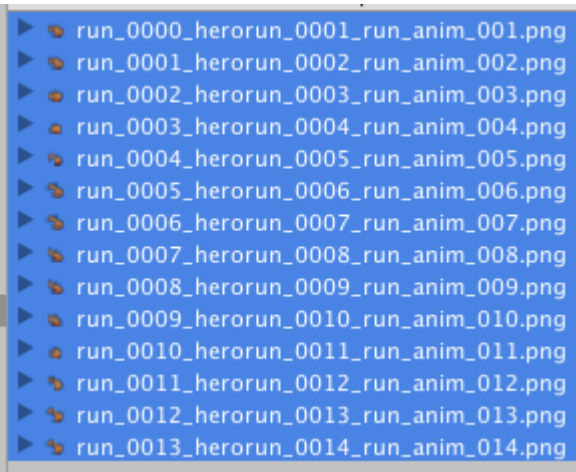
3D



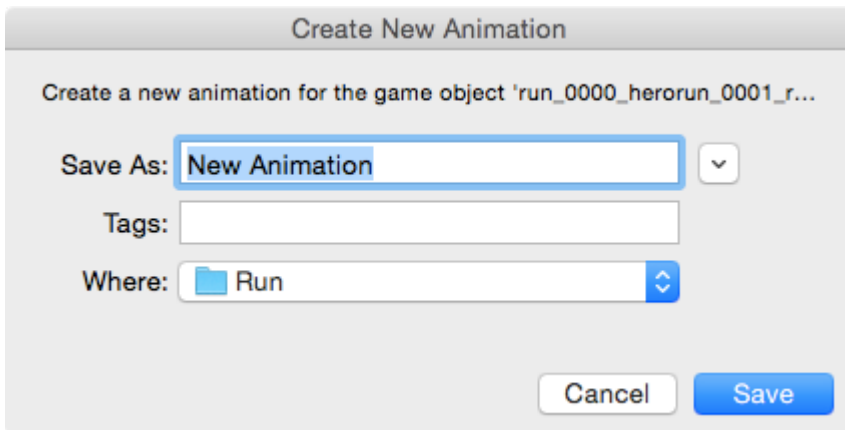
. X, Y Z . 1 0.0s 0.0 . 0.5s X 20.0 . 1.0s 0.0 . 1.0 . .

. . .

2D

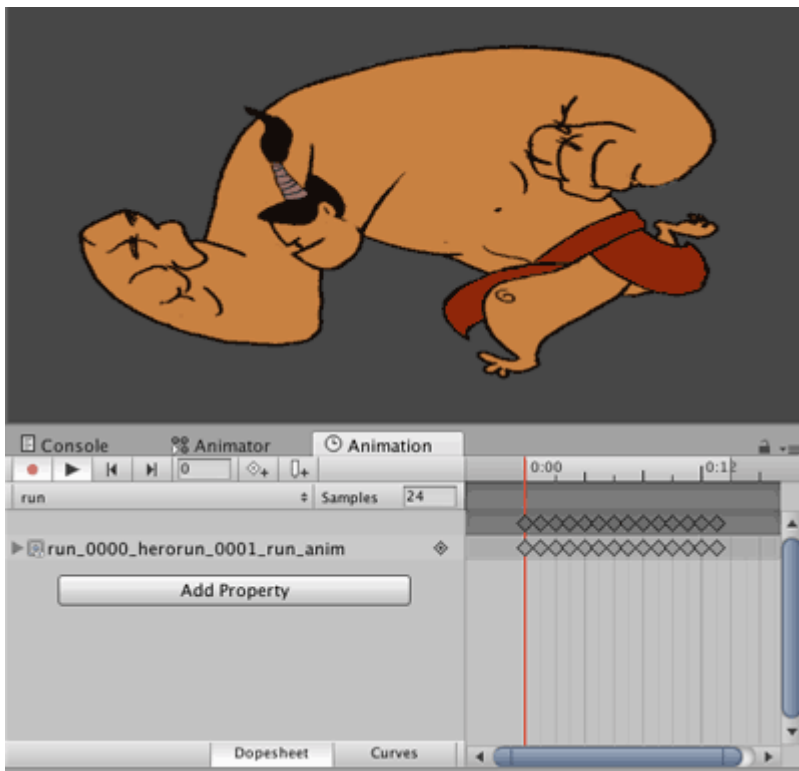
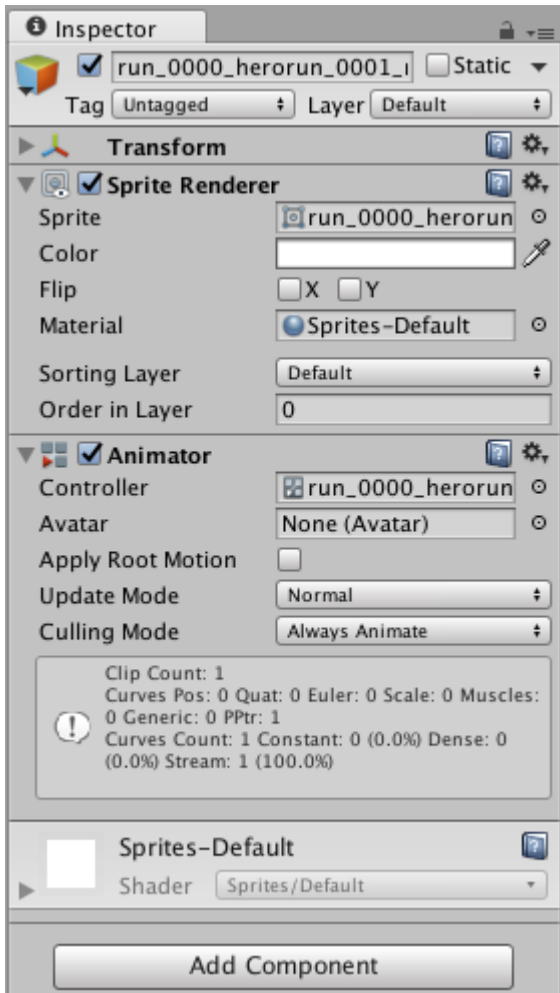


. Unity .



:

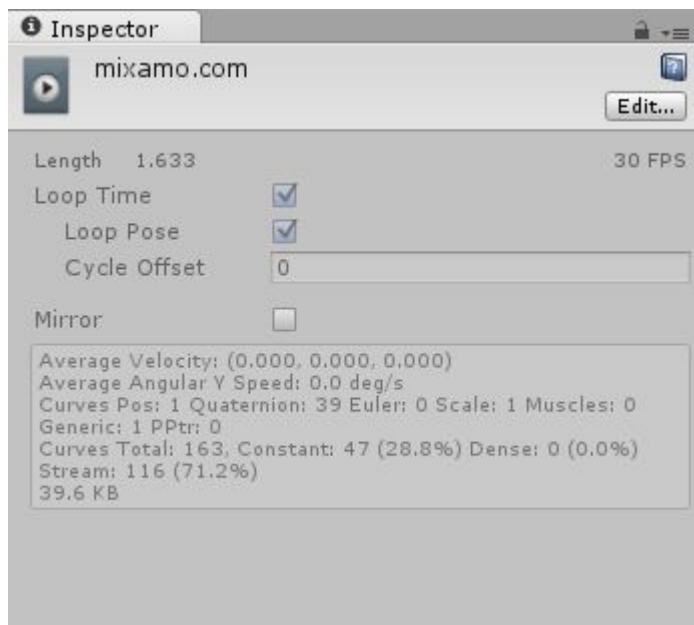
-
- ()
- (Animator)
-



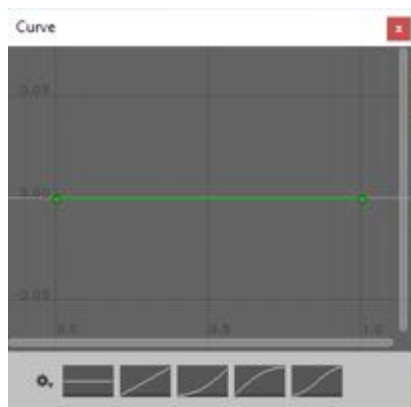
3D

. , 60 / X (= 0.0s, X = 0.0, = 30.0s; X = 1.0, = 60.0 , X = 0.0).

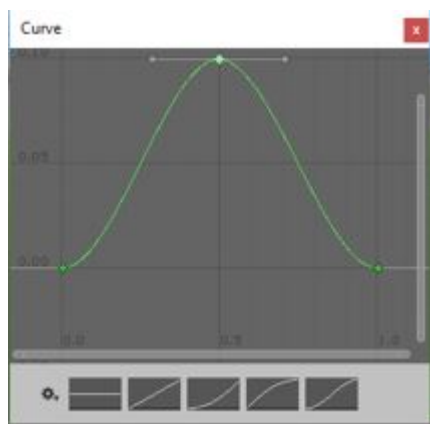
, ,
, . . .
. . .



. + . ForwardRunCurve . . .



. 2 . . .



Animator animator, ForwardRunCurve float curveValue;

float curveValue;

```
using UnityEngine;
using System.Collections;

public class RunAnimation : MonoBehaviour {

    Animator animator;
    float curveValue;

    void Start()
    {
        animator = GetComponent<Animator>();
    }

    void Update()
    {
        curveValue = animator.GetFloat("ForwardRunCurve");

        transform.Translate (Vector3.forward * curveValue);
    }
}
```

curveValue = animator.GetFloat("ForwardRunCurve");

: <https://riptutorial.com/ko/unity3d/topic/5448/>

29:

Examples

GetKey, GetKeyDown GetKeyUp

Update .

enum .

1. Input.GetKey :

Input.GetKey true true

```
public GameObject bulletPrefab;
public float shootForce = 50f;

void Update()
{
    if (Input.GetKey(KeyCode.Space))
    {
        Debug.Log("Shooting a bullet while SpaceBar is held down");

        //Instantiate bullet
        GameObject bullet = Instantiate(bulletPrefab, transform.position, transform.rotation)
as GameObject;

        //Get the Rigidbody from the bullet then add a force to the bullet
        bullet.GetComponent<Rigidbody>().AddForce(bullet.transform.forward * shootForce);
    }
}
```

2. Input.GetKeyDown Input.GetKeyDown .

Input.GetKeyDown true. Input.GetKey Input.GetKeyDown . UI / .

```
public Light flashLight;
bool enableFlashLight = false;

void Update()
{
    if (Input.GetKeyDown(KeyCode.Space))
    {
        //Toggle Light
        enableFlashLight = !enableFlashLight;
        if (enableFlashLight)
        {
            flashLight.enabled = true;
            Debug.Log("Light Enabled!");
        }
        else
        {
            flashLight.enabled = false;
            Debug.Log("Light Disabled!");
        }
    }
}
```

```

    }
}
}

```

3 .Reading Input.GetKeyUp :

Input.GetKeyDown . / .Input.GetKeyDown true true . , enable Input.GetKeyDown
Input.GetKeyUp .

```

public Light flashLight;
void Update()
{
    //Disable Light when Space Key is pressed
    if (Input.GetKeyDown(KeyCode.Space))
    {
        flashLight.enabled = true;
        Debug.Log("Light Enabled!");
    }

    //Disable Light when Space Key is released
    if (Input.GetKeyUp(KeyCode.Space))
    {
        flashLight.enabled = false;
        Debug.Log("Light Disabled!");
    }
}

```

()

Input.acceleration . 3D x, y z Vector3 .

```

void Update()
{
    Vector3 acclerometerValue = rawAccelValue();
    Debug.Log("X: " + acclerometerValue.x + " Y: " + acclerometerValue.y + " Z: " +
acclerometerValue.z);
}

Vector3 rawAccelValue()
{
    return Input.acceleration;
}

```

()

GameObject . . , . , Vector3.Lerp .

```

//The lower this value, the less smooth the value is and faster Accel is updated. 30 seems
fine for this
const float updateSpeed = 30.0f;

float AccelerometerUpdateInterval = 1.0f / updateSpeed;
float LowPassKernelWidthInSeconds = 1.0f;
float LowPassFilterFactor = 0;
Vector3 lowPassValue = Vector3.zero;

```

```

void Start()
{
    //Filter Accelerometer
    LowPassFilterFactor = AccelerometerUpdateInterval / LowPassKernelWidthInSeconds;
    lowPassValue = Input.acceleration;
}

void Update()
{
    //Get Raw Accelerometer values (pass in false to get raw Accelerometer values)
    Vector3 rawAccelValue = filterAccelValue(false);
    Debug.Log("RAW X: " + rawAccelValue.x + " Y: " + rawAccelValue.y + " Z: " +
rawAccelValue.z);

    //Get smoothed Accelerometer values (pass in true to get Filtered Accelerometer values)
    Vector3 filteredAccelValue = filterAccelValue(true);
    Debug.Log("FILTERED X: " + filteredAccelValue.x + " Y: " + filteredAccelValue.y + " Z: "
+ filteredAccelValue.z);
}

//Filter Accelerometer
Vector3 filterAccelValue(bool smooth)
{
    if (smooth)
        lowPassValue = Vector3.Lerp(lowPassValue, Input.acceleration, LowPassFilterFactor);
    else
        lowPassValue = Input.acceleration;

    return lowPassValue;
}

```

()

.

```

void Update()
{
    //Get Precise Accelerometer values
    Vector3 accelValue = preciseAccelValue();
    Debug.Log("PRECISE X: " + accelValue.x + " Y: " + accelValue.y + " Z: " + accelValue.z);
}

Vector3 preciseAccelValue()
{
    Vector3 accelResult = Vector3.zero;
    foreach (AccelerationEvent tempAccelEvent in Input.accelerationEvents)
    {
        accelResult = accelResult + (tempAccelEvent.acceleration * tempAccelEvent.deltaTime);
    }
    return accelResult;
}

```

.

```

void Update()
{
    //Get Precise Accelerometer values
    Vector3 accelValue = preciseAccelValue();
    Debug.Log("PRECISE X: " + accelValue.x + " Y: " + accelValue.y + " Z: " + accelValue.z);
}

Vector3 preciseAccelValue()
{
    Vector3 accelResult = Vector3.zero;
    for (int i = 0; i < Input.accelerationEventCount; ++i)
    {
        AccelerationEvent tempAccelEvent = Input.GetAccelerationEvent(i);
        accelResult = accelResult + (tempAccelEvent.acceleration * tempAccelEvent.deltaTime);
    }
    return accelResult;
}

```

. .

(, ,)

.

- Input.GetMouseButton(int button);
- Input.GetMouseButtonDown(int button);
- Input.GetMouseButtonUp(int button);

.

- 0 = .
- 1 = .
- 2 = .

GetMouseButton . true true .

```

void Update()
{
    if (Input.GetMouseButton(0))
    {
        Debug.Log("Left Mouse Button Down");
    }

    if (Input.GetMouseButton(1))
    {
        Debug.Log("Right Mouse Button Down");
    }

    if (Input.GetMouseButton(2))
    {
        Debug.Log("Middle Mouse Button Down");
    }
}

```

GetMouseButtonDown . true true . true .


```

void Update()
{
    if (Input.GetMouseButtonDown(0))
    {
        Debug.Log("Left Mouse Button Clicked");
    }

    if (Input.GetMouseButtonDown(1))
    {
        Debug.Log("Right Mouse Button Clicked");
    }

    if (Input.GetMouseButtonDown(2))
    {
        Debug.Log("Middle Mouse Button Clicked");
    }
}

```

GetMouseButtonUp . true true . .

```

void Update()
{
    if (Input.GetMouseButtonUp(0))
    {
        Debug.Log("Left Mouse Button Released");
    }

    if (Input.GetMouseButtonUp(1))
    {
        Debug.Log("Right Mouse Button Released");
    }

    if (Input.GetMouseButtonUp(2))
    {
        Debug.Log("Middle Mouse Button Released");
    }
}

```

: [https://riptutorial.com/ko/unity3d/topic/3413/-](https://riptutorial.com/ko/unity3d/topic/3413/)

30:

Examples

Unity Asset Store

- Unity Window → Asset Store Asset Store
- (Windows Ctrl + 9, Mac OS 9)
- : <https://www.assetstore.unity3d.com/>

Unity Asset Store

Buy Now

The image shows two parts: a screenshot of the Unity Ads app page on the App Store and a promotional banner for the game Crossy Road.

Unity Ads
Services/In-Game Advertising
Unity Technologies
★★★★ (1569)
Free
Download

Crossy Road earned over \$1 million in 45 days on iOS alone just with Unity Ads.

Unity Ads is the top performing video ad network, used by King and other developers in hit games like Hill Climb Racing, Crossy Road, and many more.

unity ADS

Crossy Road earned over **\$1 million** on iOS in **45 days** with delightful ads: watch a video, earn a virtual item

CROSSY ROAD™

Unity **Unity** . Unity .

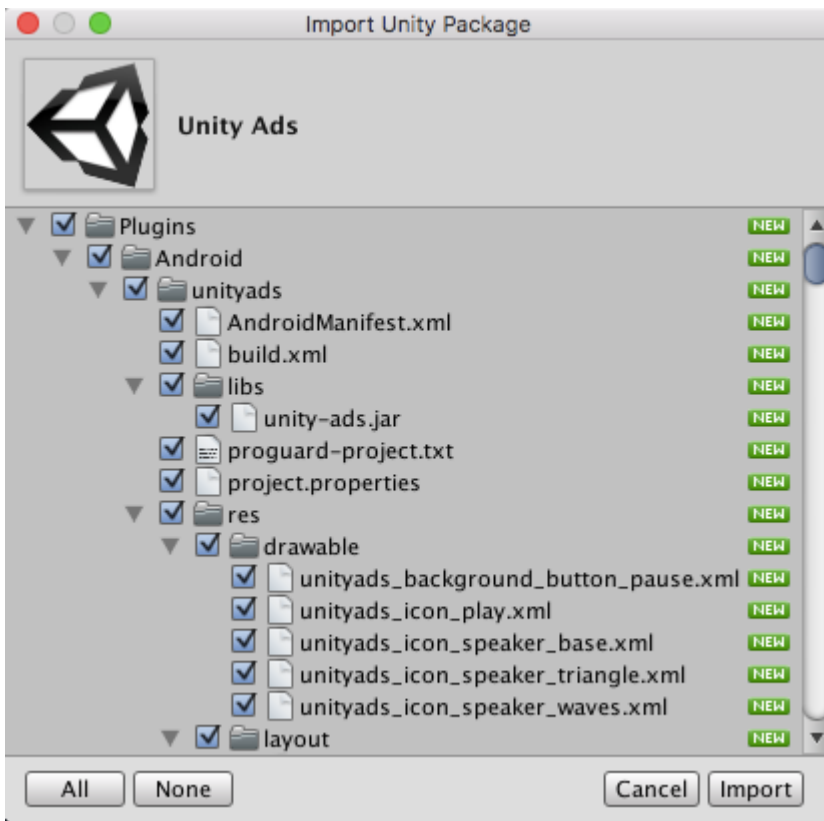
Unity Asset Store

Unity .

Unity .

.

.



- 1.
- 2.
3. ()
4. ' '>' ' ' .
5. .
- 6.
7. .

- ,

.

.

1. "Unity Asset Store purchase confirmation ..." . PDF .



UNITY3D.COM

Unity Technologies ApS

Vendersgade 28
1363 København K
Danmark

INVOICE

Invoice No.	[REDACTED]
Date	[REDACTED]
Due Date	[REDACTED]
Order No.	[REDACTED]

- <https://www.assetstore.unity3d.com/#!/account/transactions>

Credit Card / PayPal

Date	Action	Description
[REDACTED]	CREDIT CARD / PAYPAL	# [REDACTED] 30 Mesh Terrain Editor Pro

: [https://riptutorial.com/ko/unity3d/topic/5705/-](https://riptutorial.com/ko/unity3d/topic/5705/)

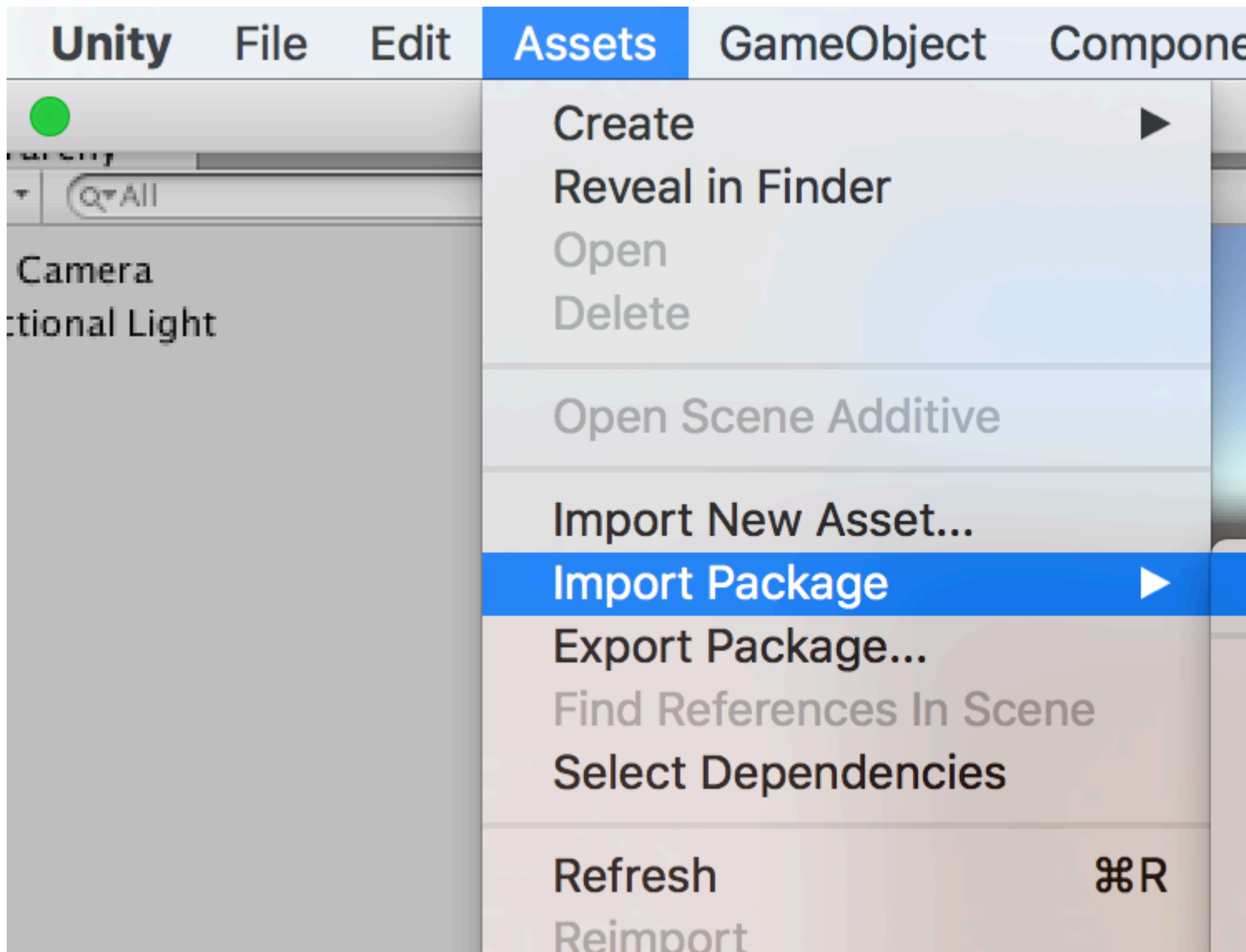
31:

Examples

(.unitypackage) Unity . SDK (: Oculus) .

.unity

Unity Assets > Import Package > Custom Package... .unitypackage Assets > Import Package > Custom Package... .



: <https://riptutorial.com/ko/unity3d/topic/4491/--->

32:

Examples

Resources . , , .

Assets **Resources** . . Resources .

. . .

```
//Example of how to load language specific audio from Resources

[RequireComponent(typeof(AudioSource))]
public class loadIntroAudio : MonoBehaviour {
    void Start () {
        string language = Application.systemLanguage.ToString();
        AudioClip ac = Resources.Load(language + "/intro") as AudioClip; //loading intro.mp3
        specific for user's language (note the file file extension should not be used)
        if (ac==null)
        {
            ac = Resources.Load("English/intro") as AudioClip; //fallback to the english
            version for any unsupported language
        }
        transform.GetComponent<AudioSource>().clip = ac;
        transform.GetComponent<AudioSource>().Play();
    }
}
```

101

' ' . " .

'Resources' Unity . [AssetBundles \(Unity Docs\)](#)

'Resources' Assets Resources . 'Resources' .

[Resources.Load](#) . . .

```
public class ResourcesSample : MonoBehaviour {

    void Start () {
        //The following line will load a TextAsset named 'foobar' which was previously place
        under 'Assets/Resources/Stackoverflow/foobar.txt'
        //Note the absence of the '.txt' extension! This is important!

        var text = Resources.Load<TextAsset>("Stackoverflow/foobar").text;
        Debug.Log(string.Format("The text file had this in it :: {0}", text));
    }
}
```

. 3D .

```
//This example will load a multiple sprite texture from Resources named "A_Multiple_Sprite"
var sprites = Resources.LoadAll("A_Multiple_Sprite") as Sprite[];
```

. GameObject 'Resources / Sounds'

```
public class SoundManager : MonoBehaviour {

    void Start () {

        //An array of all sounds you want to load
        var filesToLoad = new string[] { "Foo", "Bar" };

        //Loop over the array, attach an Audio source for each sound clip and assign the
        //clip property.
        foreach(var file in filesToLoad) {
            var soundClip = Resources.Load<AudioClip>("Sounds/" + file);
            var audioSource = gameObject.AddComponent<AudioSource>();
            audioSource.clip = soundClip;
        }
    }
}
```

1.. (,) . Resources . . .

2. Resources.Load Resources.LoadAll [Resources.UnloadUnusedAssets](#)
[Resources.UnloadAsset](#) .

: <https://riptutorial.com/ko/unity3d/topic/4070/>

33:

- PrefabUtility.InstantiatePrefab ();
- AssetDatabase.LoadAssetAtPath (,);
- Object Object.Instantiate (Object original);
- Resources.Load ();

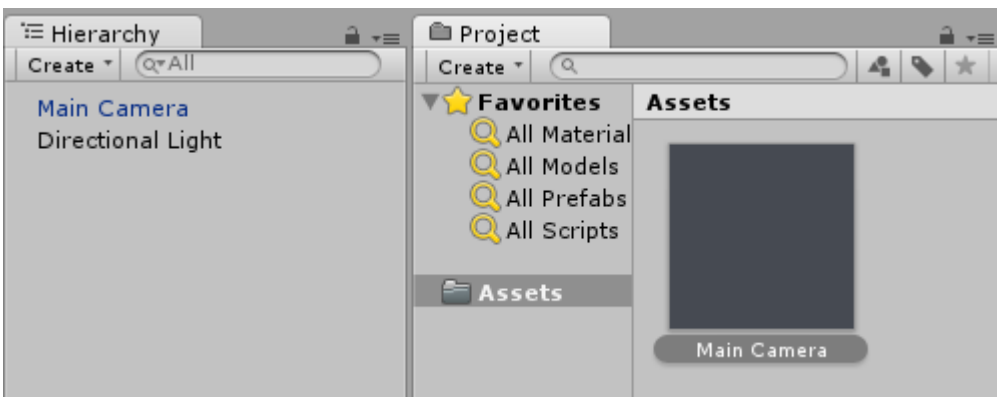
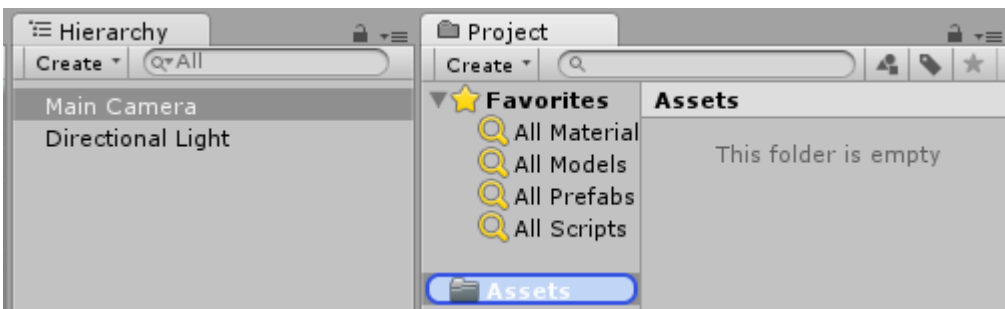
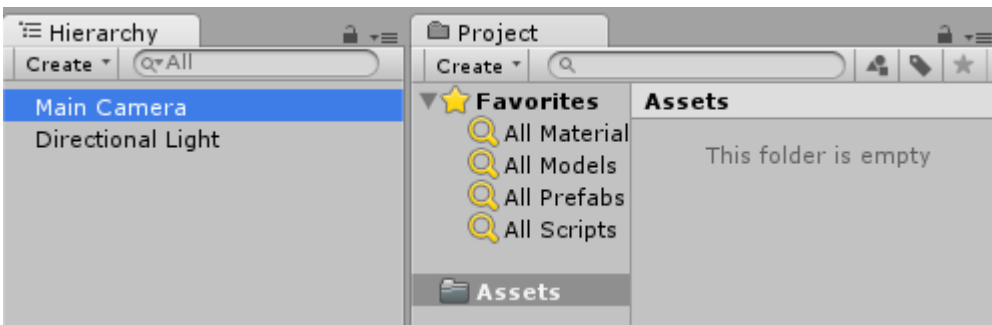
Examples

Prefab , , GameObject . . .

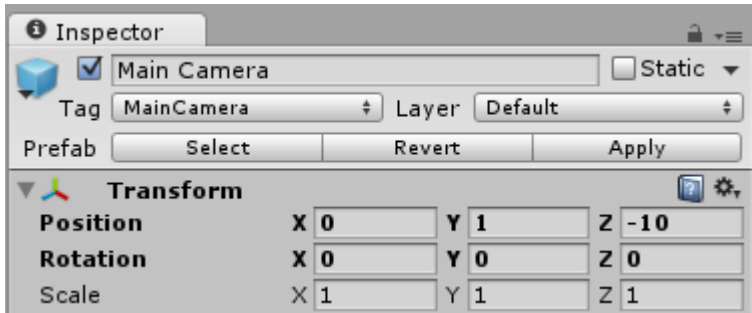
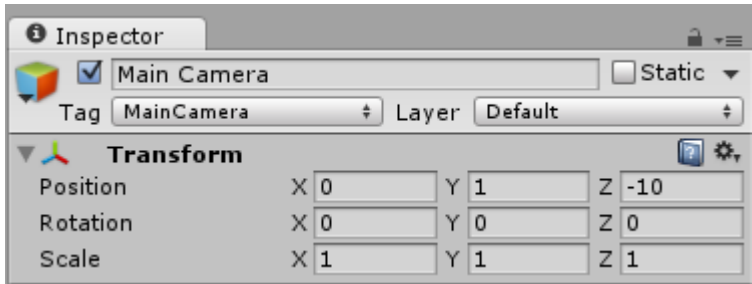
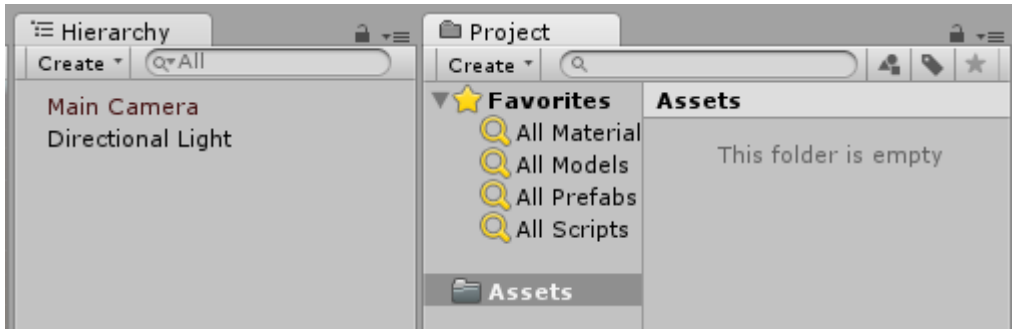
-
-
- /
-
- GameObjects

" " . GameObjects . . .

Assets .



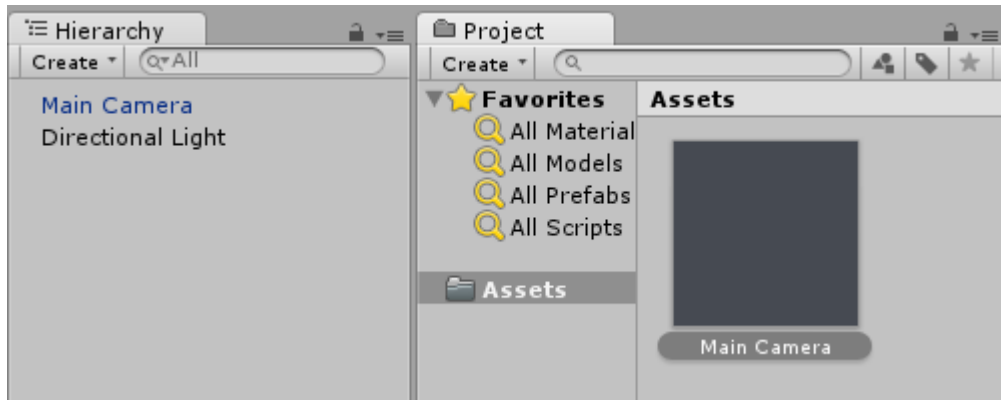
(prefab) .



Reflection to Prefab .

Select

(:).



- `PrefabUtility.InstantiatePrefab()` .

```
GameObject gameObject =
    (GameObject)PrefabUtility.InstantiatePrefab(AssetDatabase.LoadAssetAtPath("Assets/MainCamera.prefab",
    typeof(GameObject)));
```

(:5).

. MonoBehaviour public GameObject Unity Editor inspector :

```
public class SomeScript : MonoBehaviour {
    public GameObject prefab;
}
```

Resources.Load :

```
GameObject prefab = Resources.Load("Assets/Resources/MainCamera");
```

Instantiate Instantiate (:).

```
GameObject gameObject = Instantiate<GameObject>(prefab, new Vector3(0,0,0),
    Quaternion.identity);
```

: .

Unity . .

- . .

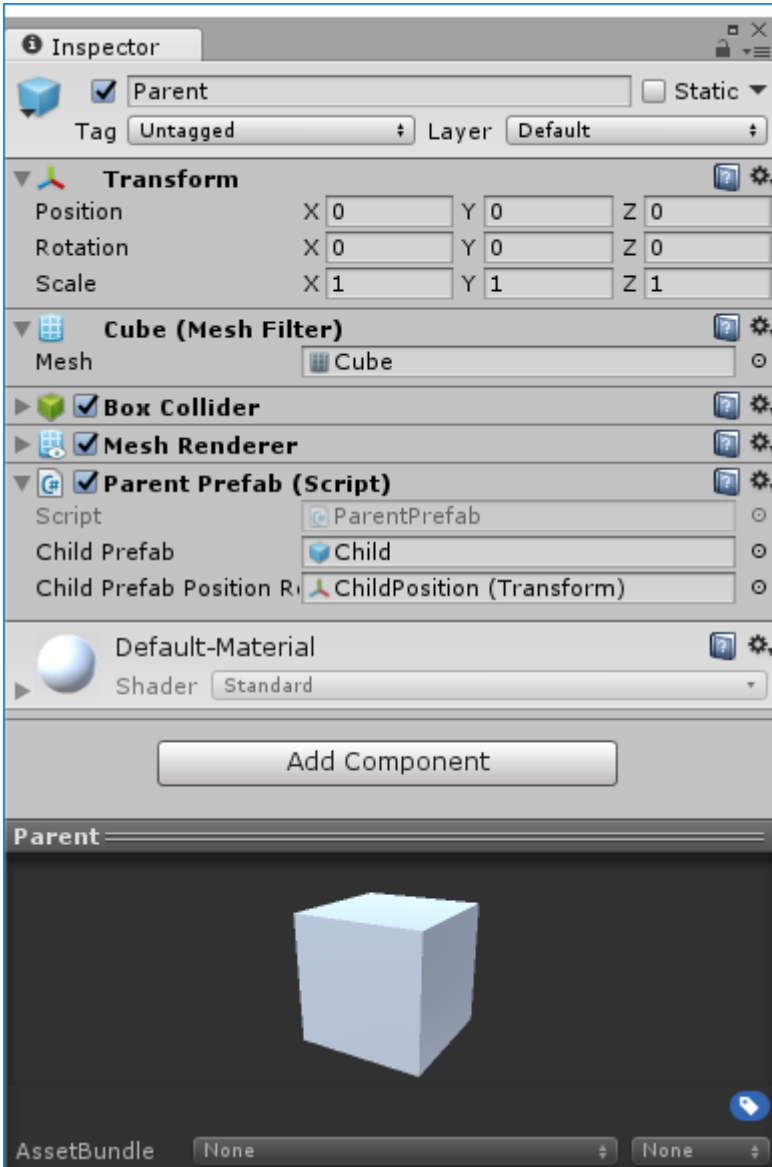
```
using UnityEngine;

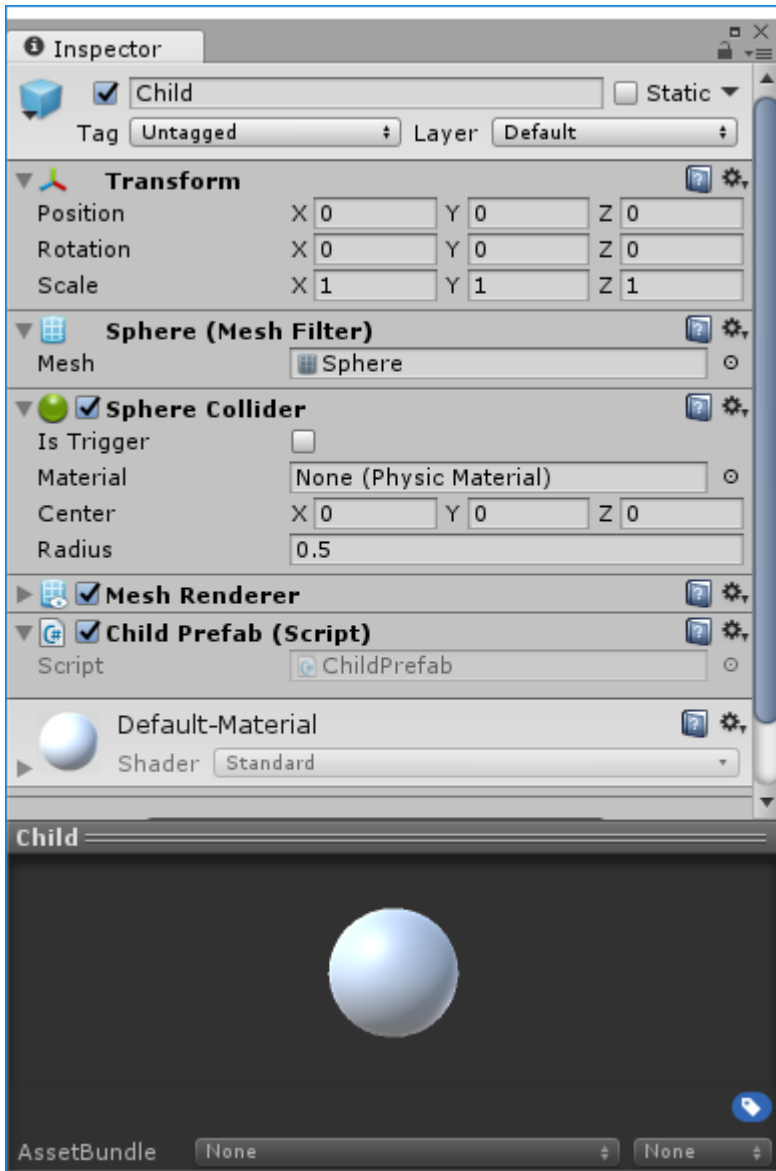
public class ParentPrefab : MonoBehaviour {

    [SerializeField] GameObject childPrefab;
    [SerializeField] Transform childPrefabPositionReference;

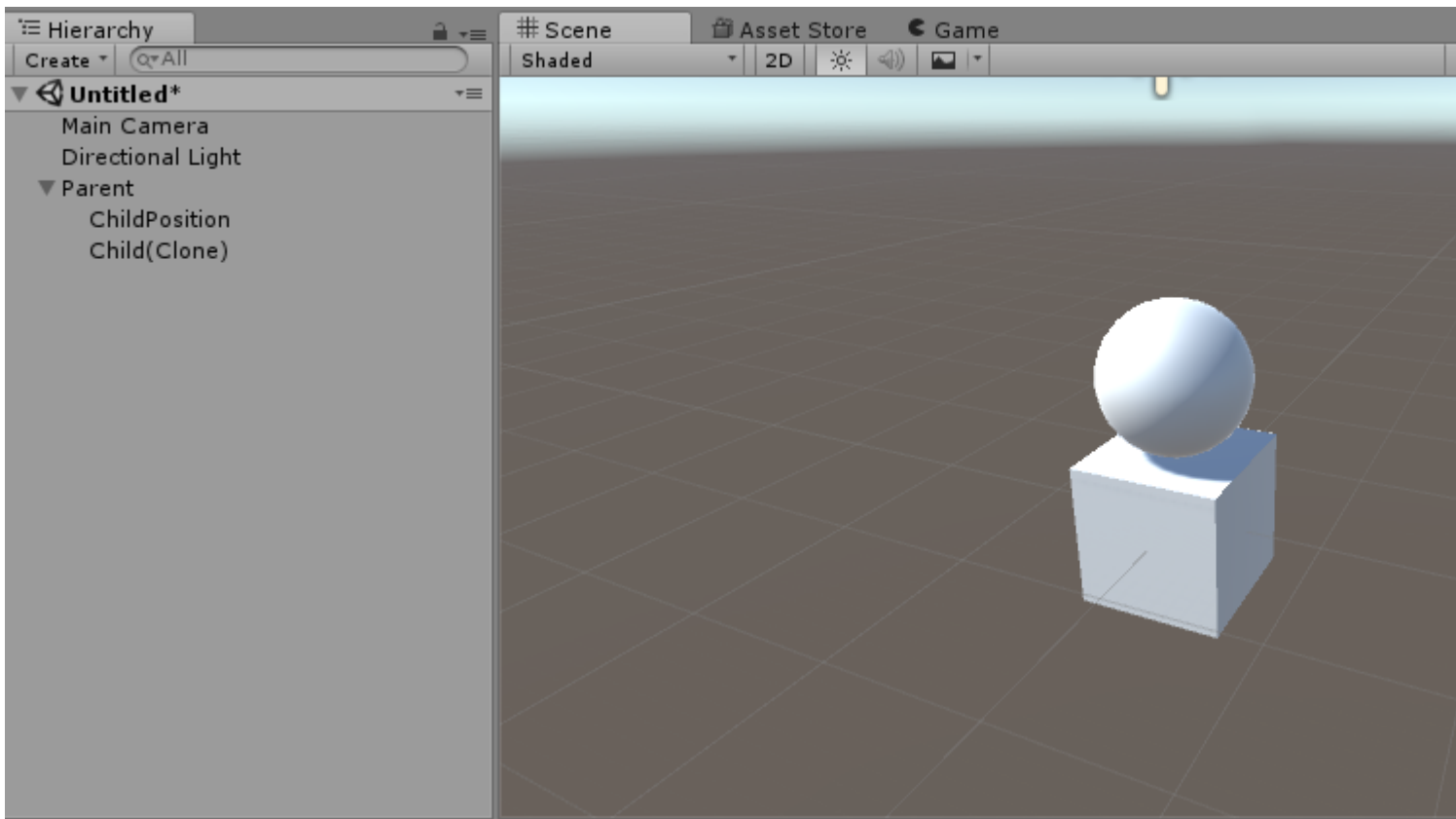
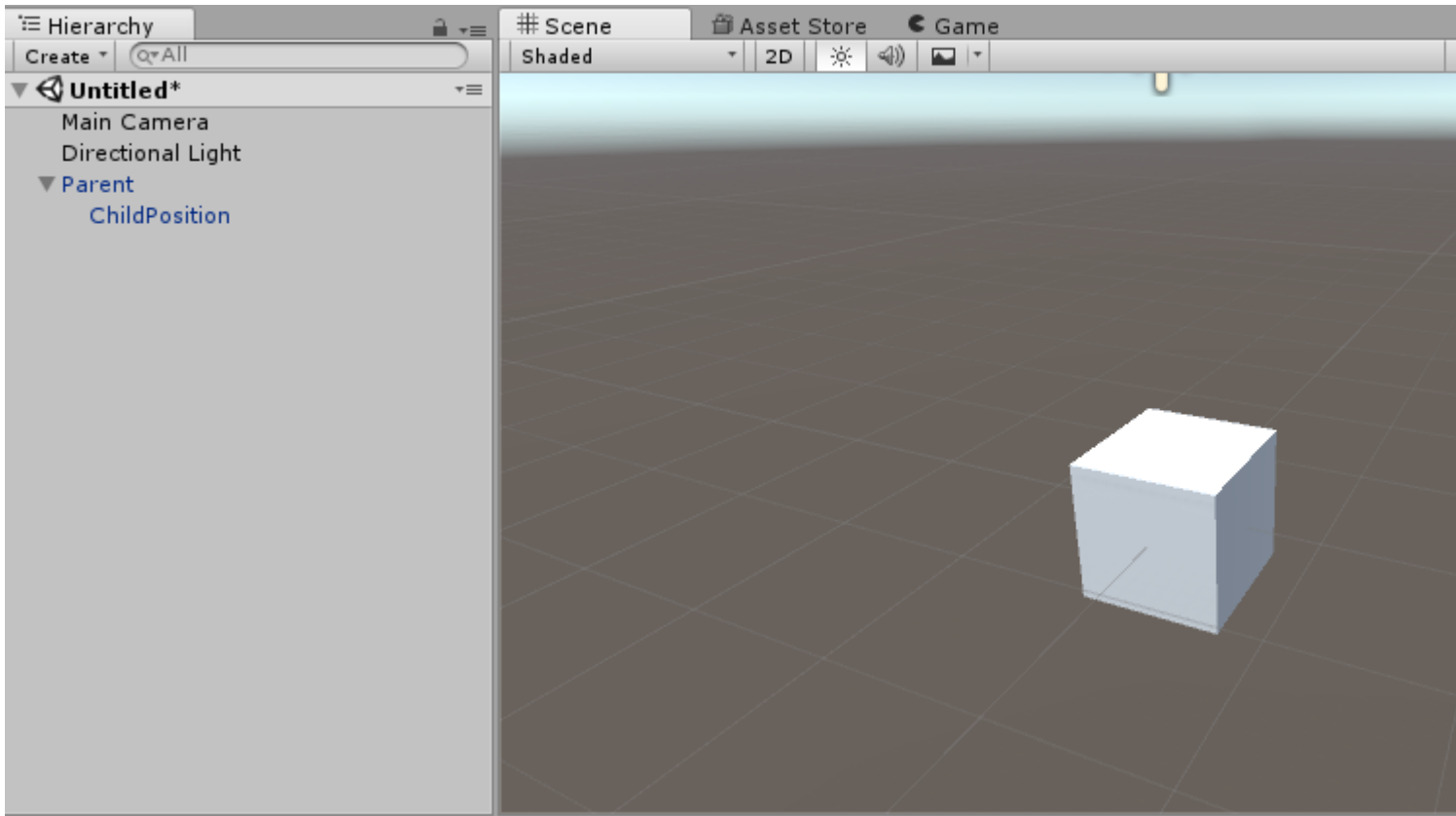
    // Use this for initialization
    void Start () {
        print("Hello, I'm a parent prefab!");
    }
}
```

```
Instantiate(  
    childPrefab,  
    childPrefabPositionReference.position,  
    childPrefabPositionReference.rotation,  
    gameObject.transform  
);  
}  
}
```





:



: <https://riptutorial.com/ko/unity3d/topic/2133/>

34: (IMGUI)

- public static void GUILayout.Label (, GUILayoutOption [])
- bool GUILayout.Button (, GUILayoutOption [])
- GUILayout.TextArea (, GUILayoutOption [])

Examples

GUILayout

UI .

```
void OnGUI ()
{
    GUILayout.Label ("I'm a simple label text displayed in game.");

    if ( GUILayout.Button("CLICK ME") )
    {
        GUILayout.TextArea ("This is a \n
                             multiline comment.")
    }
}
```

GUILayout OnGUI .

(IMGUI) : <https://riptutorial.com/ko/unity3d/topic/6947/-----imgui->

35:

1..

Examples

. Update .



magnitude sqrMagnitude . sqrt . sqrMagnitude .

```
if ((target.position - transform.position).sqrMagnitude < minDistance * minDistance))
```

Collider / Renderer . Bounds [Intersects](#) .

Bounds () ([Bounds.SqrDistance](#)).

Mesh.bounds . MeshRenderer.bounds .

Unity API [Coroutines](#) .

n .

Coroutines .

```
for (int y = 0; y < heightmap.Height; y++)
{
    for (int x = 0; x < heightmap.Width; x++)
    {
        // Generate pixel at (x, y)
        // Assign pixel at (x, y)

        // Process only 32768 pixels each frame
        if ((y * heightmap.Height + x) % 32 * 1024 == 0)
            yield return null; // Wait for next frame
    }
}
```

. yield return (2-3) for .

```
private void ProximityCheck()
{
    for (int i = 0; i < enemies.Length; i++)
    {
        if (Vector3.Distance(transform.position, enemies[i].transform.position) <
dangerDistance)
            return true;
    }
    return false;
}

private IEnumerator ProximityCheckCoroutine()
{
    while(true)
    {
        ProximityCheck();
        yield return new WaitForSeconds(.1f);
    }
}
```

CullingGroup API

. Coroutines yield return yield return. / , .

Unity , .

■ CPU .

```
string[] StringKeys = new string[] {
    "Key0",
    "Key1",
    "Key2"
};

void Update()
{
    for (var i = 0; i < 3; i++)
    {
```



```

        // Cached, no garbage generated
        Debug.Log(StringKeys[i]);
    }

    for (var i = 0; i < 3; i++)
    {
        // Not cached, garbage every cycle
        Debug.Log("Key" + i);
    }

    // The most memory-efficient way is to not create a cache at all and use literals or
    constants.
    // However, it is not necessarily the most readable or beautiful way.
    Debug.Log("Key0");
    Debug.Log("Key1");
    Debug.Log("Key2");
}

```

• , • •

• , •

• •

▪

, . Debug.Log("Object Name: " + obj.name) Debug.Log("Object Name: " + obj.name) . .

Conditional

```

using UnityEngine;
using System.Collections;

public class ConditionalDebugExample: MonoBehaviour
{
    IEnumerator Start()
    {
        while(true)
        {
            // This message will pop up in Editor but not in builds
            Log("Elapsed: " + Time.timeSinceLevelLoad);
            yield return new WaitForSeconds(1f);
        }
    }

    [System.Diagnostics.Conditional("UNITY_EDITOR")]
    void Log(string Message)
    {
        Debug.Log(Message);
    }
}

```

• •

```

// Faster string comparison
if (strA.Equals(strB, System.StringComparison.Ordinal)) {...}
// Compared to
if (strA == strB) {...}

// Less overhead
if (!string.IsNullOrEmpty(strA)) {...}
// Compared to
if (strA == "") {...}

// Faster lookups
Dictionary<string, int> myDic = new Dictionary<string, int>(System.StringComparer.Ordinal);
// Compared to
Dictionary<string, int> myDictionary = new Dictionary<string, int>();

```

update . null bool flat .

:

```

void Update()
{
    var renderer = GetComponent<Renderer>();
    renderer.material.SetColor("_Color", Color.green);
}

```

```

private Renderer myRenderer;
void Start()
{
    myRenderer = GetComponent<Renderer>();
}

void Update()
{
    myRenderer.material.SetColor("_Color", Color.green);
}

```

```

void Update()
{
    var enemy = GameObject.Find("enemy");
    enemy.transform.LookAt(new Vector3(0,0,0));
}

```

```

private Transform enemy;

void Start()
{
    this.enemy = GameObject.Find("enemy").transform;
}

void Update()
{
    enemy.LookAt(new Vector3(0, 0, 0));
}

```

Mathf

.

.

:

```
//Avoid StartCoroutine with method name
this.StartCoroutine("SampleCoroutine");

//Instead use the method directly
this.StartCoroutine(this.SampleCoroutine());

//Avoid send message
var enemy = GameObject.Find("enemy");
enemy.SendMessage("Die");

//Instead make direct call
var enemy = GameObject.Find("enemy") as Enemy;
enemy.Die();
```

.

.

,

.

```
void Update
{
}

void FixedUpdate
{
}
```

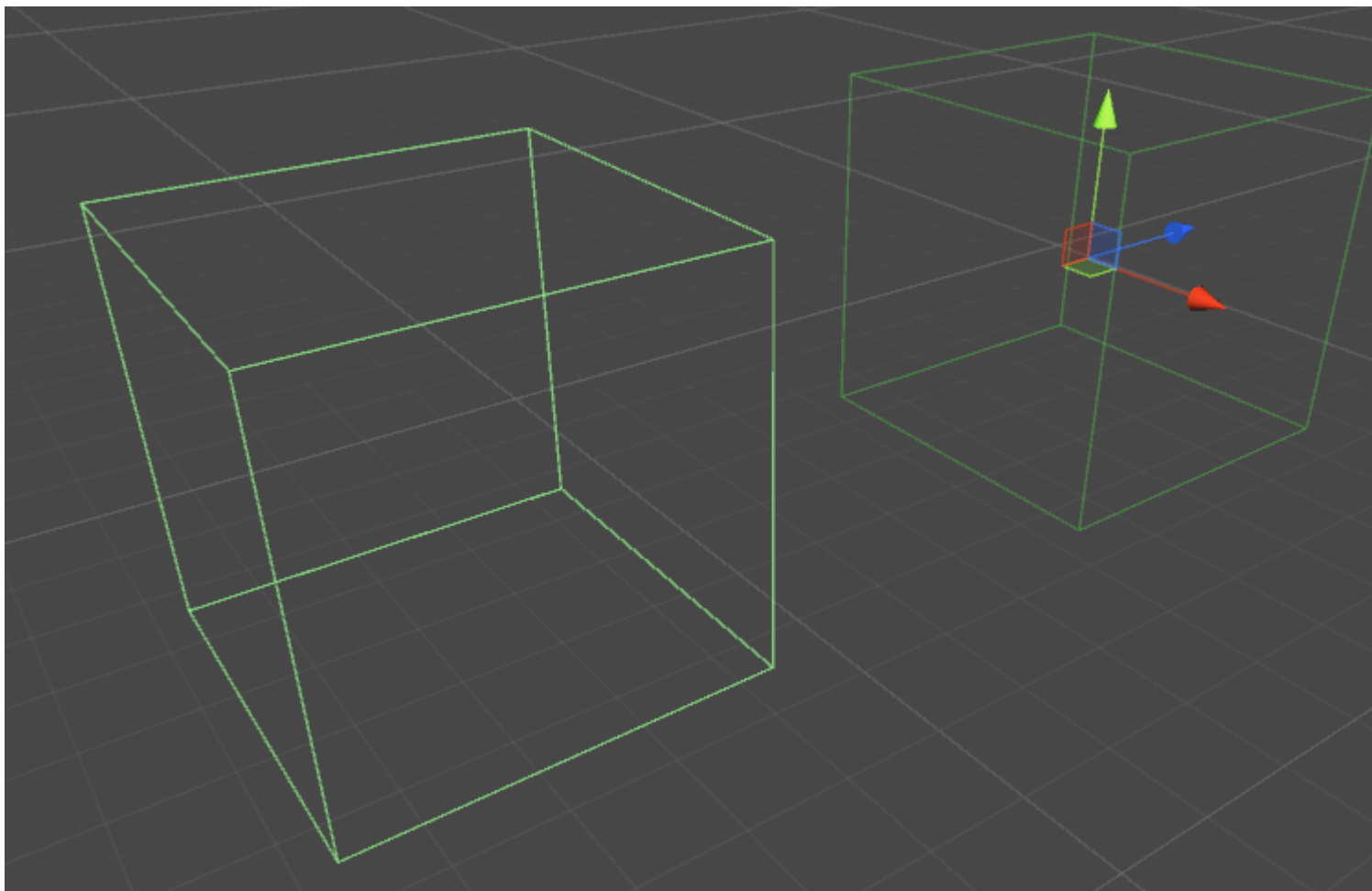
: <https://riptutorial.com/ko/unity3d/topic/3433/>

36:

Examples

Colliders

Collider.

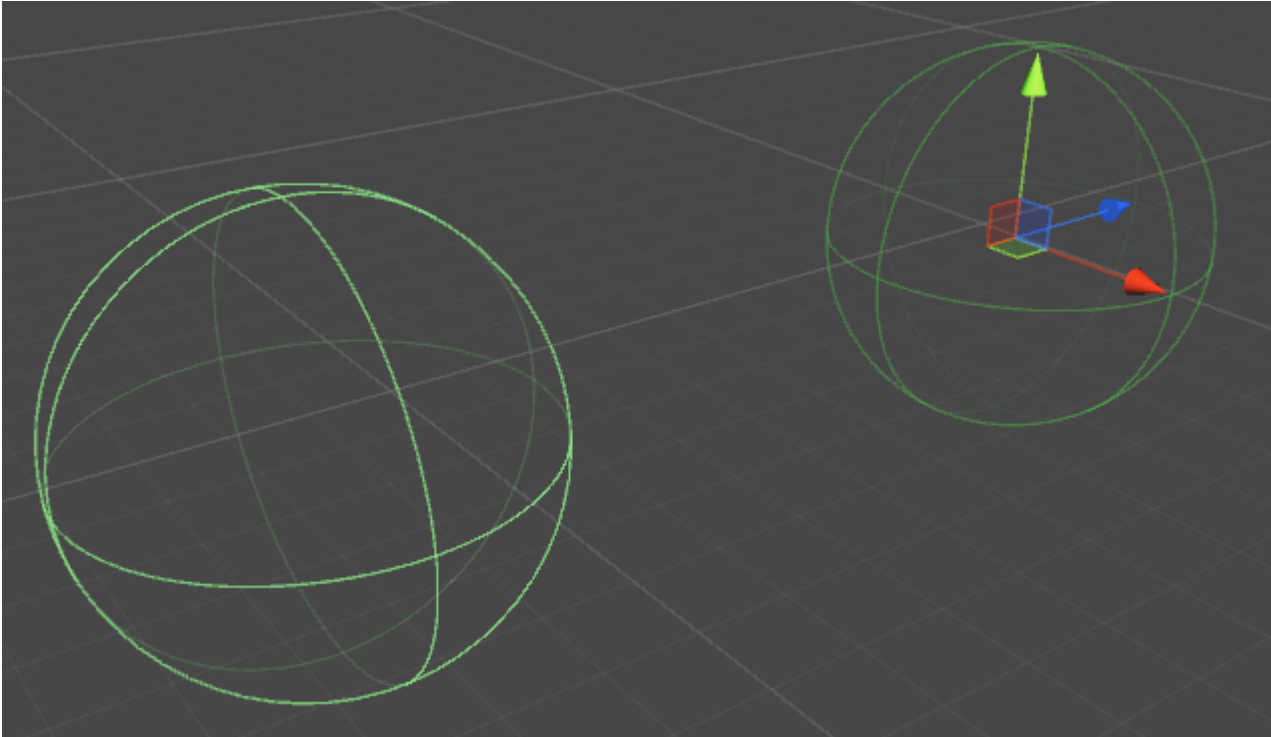


- **Is Trigger** - Box Collider Trigger Collider.
- **Material** - Box Collider .
- **Center** - Box Collider
- **Size** - Box Collider .

```
// Add a Box Collider to the current GameObject.  
BoxCollider myBC = BoxCollider(myGameObject.gameObject.AddComponent(typeof(BoxCollider)));  
  
// Make the Box Collider into a Trigger Collider.  
myBC.isTrigger= true;
```

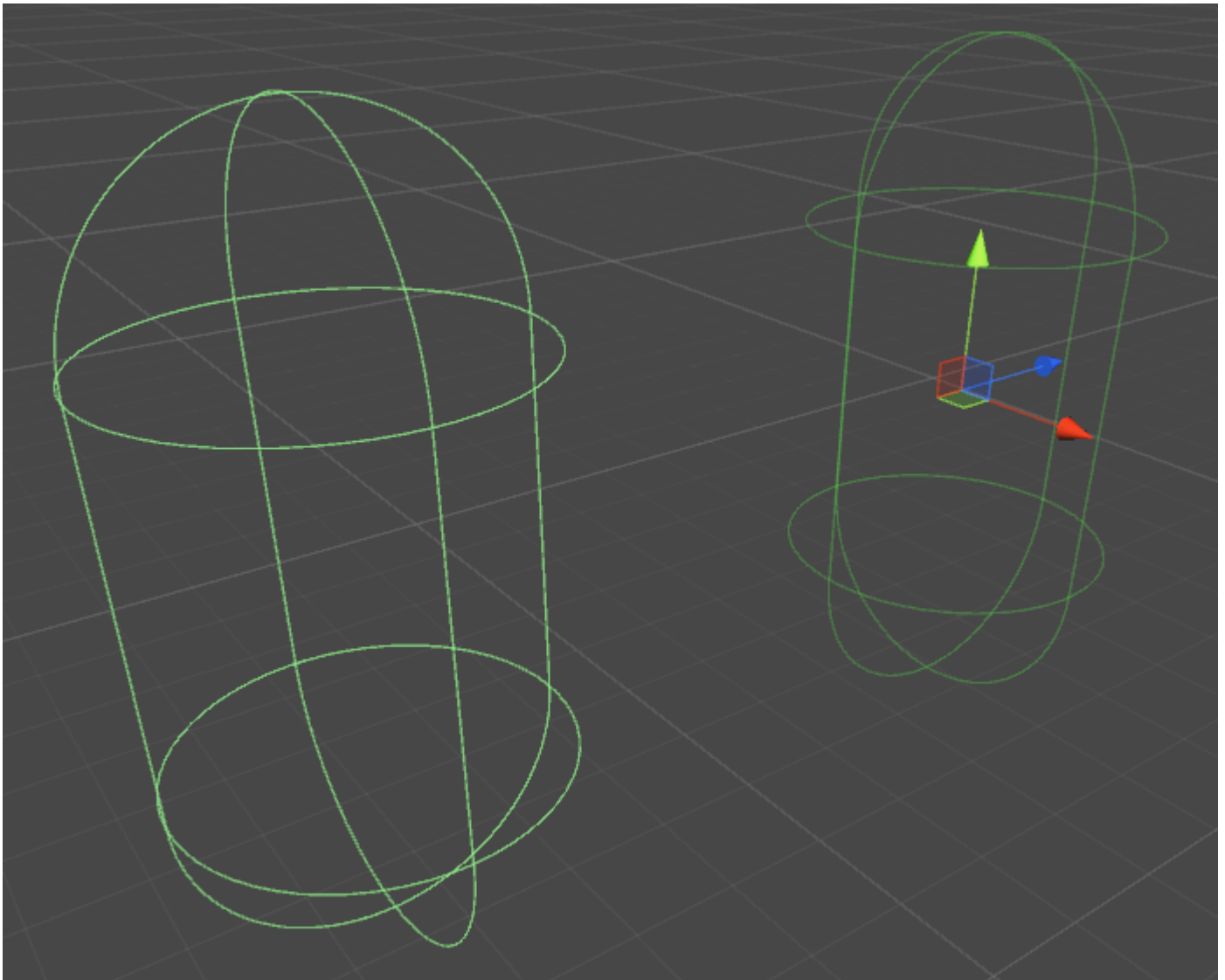
```
// Set the center of the Box Collider to the center of the GameObject.  
myBC.center = Vector3.zero;  
  
// Make the Box Collider twice as large.  
myBC.size = 2;
```

Collider.



- **Is Trigger** - Sphere Collider Trigger Collider.
- **Material** - Sphere Collider ().
- **Center** - Sphere Collider
- **Radius** - Collider

```
// Add a Sphere Collider to the current GameObject.  
SphereCollider mySC =  
SphereCollider)myGameObject.gameObject.AddComponent(typeof(SphereCollider));  
  
// Make the Sphere Collider into a Trigger Collider.  
mySC.isTrigger= true;  
  
// Set the center of the Sphere Collider to the center of the GameObject.  
mySC.center = Vector3.zero;  
  
// Make the Sphere Collider twice as large.  
mySC.radius = 2;
```



- **Is Is Trigger** - Capsule Collider Trigger Collider.
- **Material** - Capsule Collider .
- -
- -
- **Height** - Collider
- -

```
// Add a Capsule Collider to the current GameObject.  
CapsuleCollider myCC =  
CapsuleCollider)myGameObject.gameObject.AddComponent(typeof(CapsuleCollider));  
  
// Make the Capsule Collider into a Trigger Collider.  
myCC.isTrigger= true;  
  
// Set the center of the Capsule Collider to the center of the GameObject.  
myCC.center = Vector3.zero;
```

```
// Make the Sphere Collider twice as tall.
myCC.height= 2;

// Make the Sphere Collider twice as wide.
myCC.radius= 2;

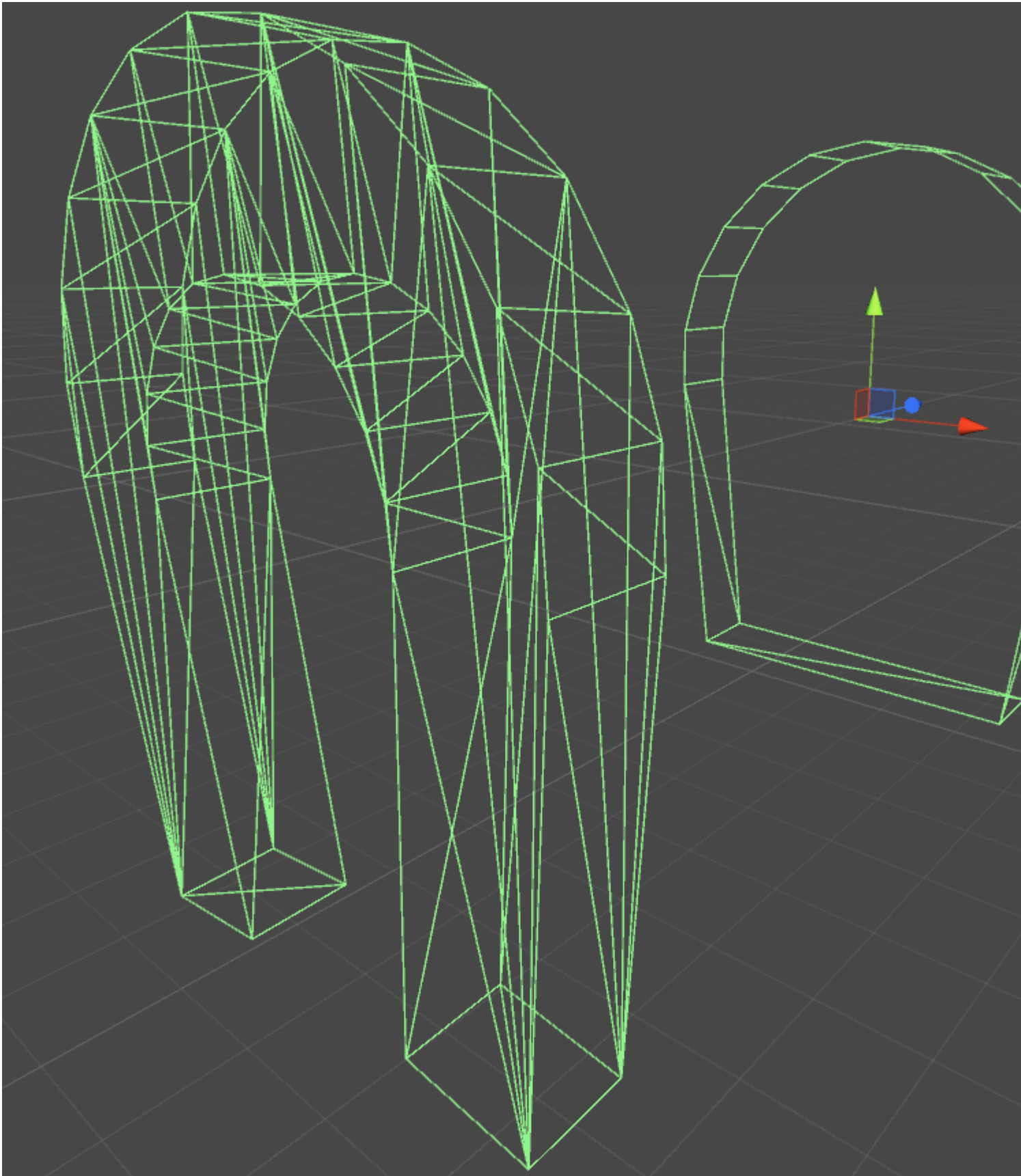
// Set the axis of lengthwise orientation to the X axis.
myCC.direction = 0;

// Set the axis of lengthwise orientation to the Y axis.
myCC.direction = 1;

// Set the axis of lengthwise orientation to the Y axis.
myCC.direction = 2;
```

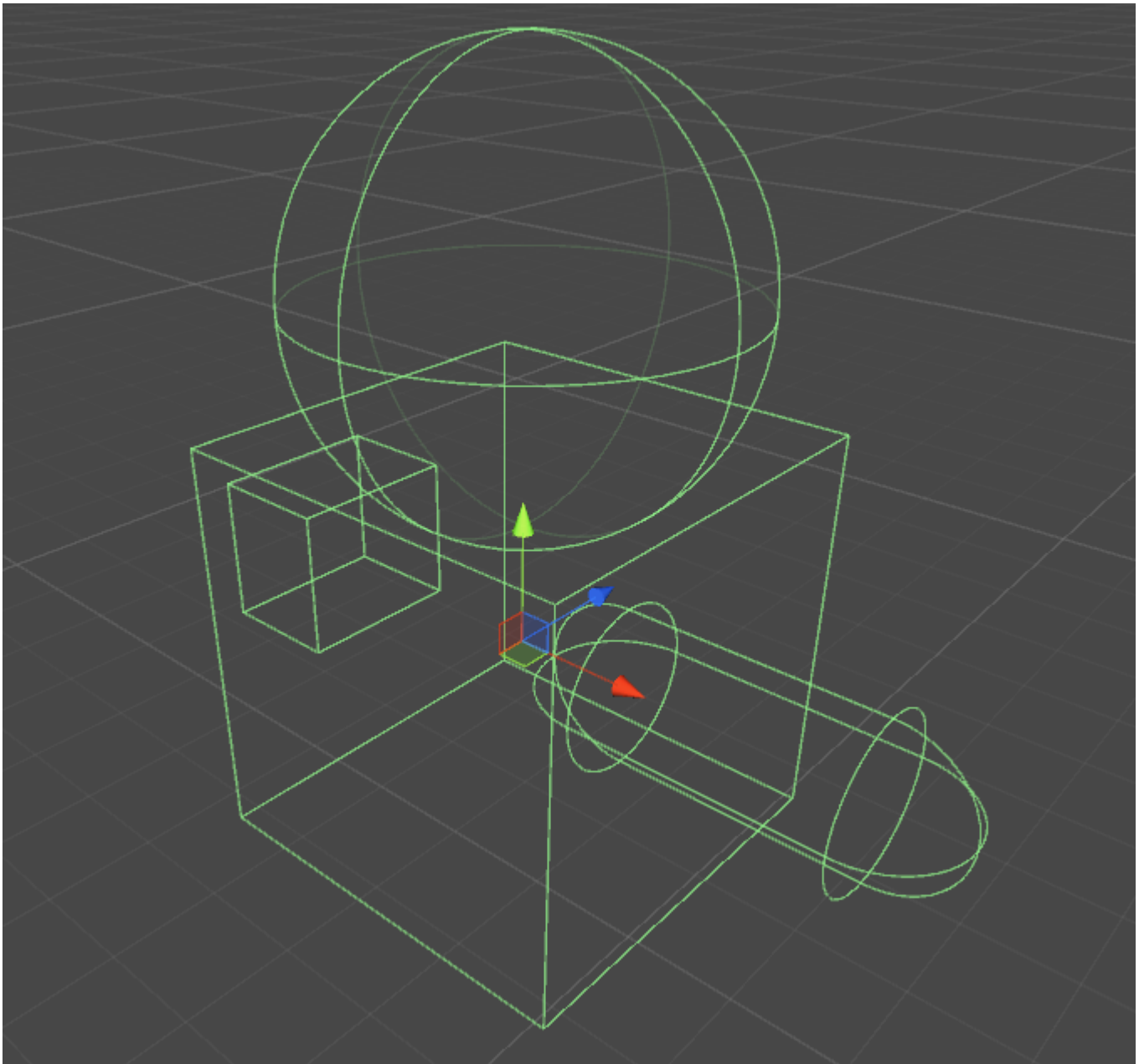
-
- -
 - -
 - -
 - - Y
 - - ,
 - -
 - **Spring** - Wheel
 - - .
 - - 0.5, 0 , 1 .
 - / -

Collider.



- **Is Is Trigger** - Box Collider Trigger Collider.
- **Material** - Box Collider .
- **Mesh** - Collider .
-

GameObject Collider Compound Collider .



Nvidia PhysX . "" .

- - (), inertia .
- - , .
- Wheel Damping Rate - " " .
- Suspension Distance - ()
- Force App Point Distance - rigidbody
- -
- - $K /$.
- = *

50 100 . 4 2,000kg 500kg . 75 37,500 / .

- - . " " " " .

(m / s) .

- Extremum Slip - (m / s).
- (Extremum Value) - .

Extremum Slip 0.2m 2m / s . 2m / s 6 5mph. . 2m / s () .

() .

$$() = * ()$$

1, + . 1 . .7.9 1.0 .

2.5 . , , , , .

. Asymptote Slip 0.5m / s 2m / s , . , . .

. .

GTA . , . , . .

, .

wheel colliders Nvidia

<http://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/guide/Manual/Vehicles.html> .

Colliders

- OnTriggerEnter()
- OnTriggerStay()
- OnTriggerExit()

OnTriggerEnter() , OnTriggerStay() OnTriggerExit() Collider . , . GameObject . , , .

Collider

GameObject () . GameObject .

```
void OnTriggerEnter(Collider other)
{
    //Check collider for specific properties (Such as tag=item or has component=item)
}
```

: <https://riptutorial.com/ko/unity3d/topic/4405/>

37: API

CullingGroups

```
using UnityEngine;
using System;
public interface ICullingGroupManager
{
    int ReserveSphere();
    void ReleaseSphere(int sphereIndex);
    void SetPosition(int sphereIndex, Vector3 position);
    void SetRadius(int sphereIndex, float radius);
    void SetCullingEvent(int sphereIndex, Action<CullingGroupEvent> sphere);
}
```

Examples

CullingGroups

```
using UnityEngine;
using System.Linq;

public class CullingGroupBehaviour : MonoBehaviour
{
    CullingGroup localCullingGroup;

    MeshRenderer[] meshRenderers;
    Transform[] meshTransforms;
    BoundingSphere[] cullingPoints;

    void OnEnable()
    {
        localCullingGroup = new CullingGroup();

        meshRenderers = FindObjectsOfType<MeshRenderer>()
            .Where((MeshRenderer m) => m.gameObject != this.gameObject)
            .ToArray();

        cullingPoints = new BoundingSphere[meshRenderers.Length];
        meshTransforms = new Transform[meshRenderers.Length];

        for (var i = 0; i < meshRenderers.Length; i++)
        {
            meshTransforms[i] = meshRenderers[i].GetComponent<Transform>();
            cullingPoints[i].position = meshTransforms[i].position;
            cullingPoints[i].radius = 4f;
        }

        localCullingGroup.onStateChanged = CullingEvent;
    }
}
```

```

    localCullingGroup.SetBoundingSpheres(cullingPoints);
    localCullingGroup.SetBoundingDistances(new float[] { 0f, 5f });
    localCullingGroup.SetDistanceReferencePoint (GetComponent<Transform>().position);
    localCullingGroup.targetCamera = Camera.main;
}

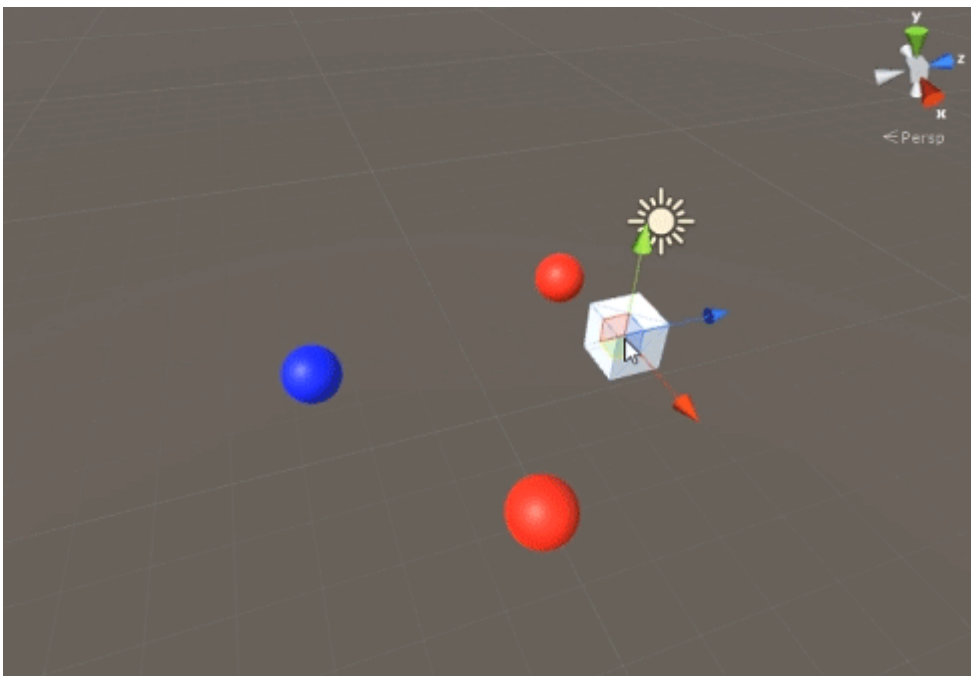
void FixedUpdate()
{
    localCullingGroup.SetDistanceReferencePoint (GetComponent<Transform>().position);
    for (var i = 0; i < meshTransforms.Length; i++)
    {
        cullingPoints[i].position = meshTransforms[i].position;
    }
}

void CullingEvent (CullingGroupEvent sphere)
{
    Color newColor = Color.red;
    if (sphere.currentDistance == 1) newColor = Color.blue;
    if (sphere.currentDistance == 2) newColor = Color.white;
    meshRenderers[sphere.index].material.color = newColor;
}

void OnDisable()
{
    localCullingGroup.Dispose();
}
}

```

GameObject () Play . GameObject .



```

using UnityEngine;
using System.Linq;

```

```

public class CullingGroupCameraBehaviour : MonoBehaviour
{
    CullingGroup localCullingGroup;

    MeshRenderer[] meshRenderers;

    void OnEnable()
    {
        localCullingGroup = new CullingGroup();

        meshRenderers = FindObjectsOfType<MeshRenderer>()
            .Where((MeshRenderer m) => m.gameObject != this.gameObject)
            .ToArray();

        BoundingSphere[] cullingPoints = new BoundingSphere[meshRenderers.Length];
        Transform[] meshTransforms = new Transform[meshRenderers.Length];

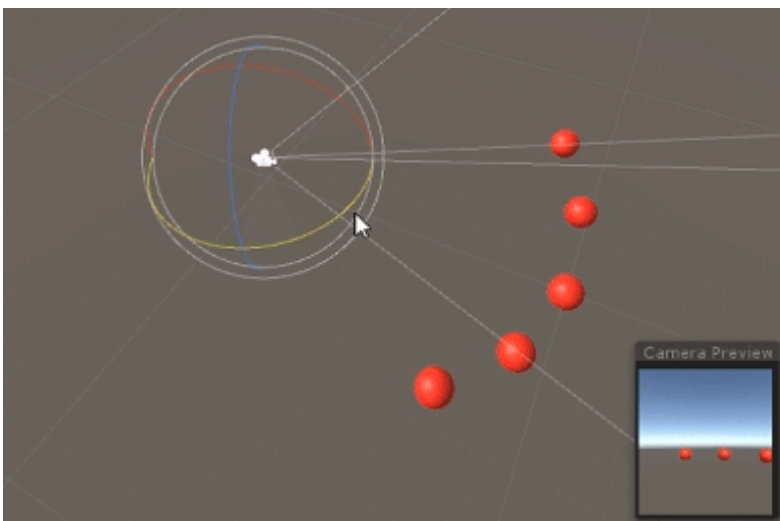
        for (var i = 0; i < meshRenderers.Length; i++)
        {
            meshTransforms[i] = meshRenderers[i].GetComponent<Transform>();
            cullingPoints[i].position = meshTransforms[i].position;
            cullingPoints[i].radius = 4f;
        }

        localCullingGroup.onStateChanged = CullingEvent;
        localCullingGroup.SetBoundingSpheres(cullingPoints);
        localCullingGroup.targetCamera = Camera.main;
    }

    void CullingEvent(CullingGroupEvent sphere)
    {
        meshRenderers[sphere.index].material.color = sphere.isVisible ? Color.red :
Color.white;
    }

    void OnDisable()
    {
        localCullingGroup.Dispose();
    }
}

```



MeshRenderer MonoBehaviour.OnBecameVisible() . GameObject, Vector3
CullingGroups .

. "" , "" "" .

```
cullingGroup.SetBoundingDistances(new float[] { 0f, 10f, 100f});
```

. .

.

, . . .

```
float cullingPointArea = Mathf.PI * (cullingPointRadius * cullingPointRadius);  
float boundingArea = Mathf.PI * (boundingDistance * boundingDistance);  
float combinedRadius = Mathf.Sqrt((cullingPointArea + boundingArea) / Mathf.PI);
```

API : <https://riptutorial.com/ko/unity3d/topic/4574/--api>

38:

- StartCoroutine (IEnumerator);
- StartCoroutine (string methodName, object value = null);
- StopCoroutine (methodName);
- StopCoroutine (IEnumerator);
- StopAllCoroutines ();

coroutines .

- CPU .
- MoveNext MoveNext coroutine Unity . 5.4.0b13 . ()

YieldInstructions

YieldInstruction YieldInstruction .

```
IEnumerator TickEverySecond()
{
    var wait = new WaitForSeconds(1f); // Cache
    while(true)
    {
        yield return wait; // Reuse
    }
}
```

yield null .

Examples

(: Unity) " " .

.

Unity .

. . 0.02632, 0.021167, 0.029778 . 1 1 "" . . .

, : Unity ?

, Update () . (- . .)

.

" " .

() .


```
, "1" ., 1 1 .( .) 1 . .
```

:

.

```
// do something  
yield return null; // wait until next frame  
// do something
```

:

```
// do something  
yield return null; // wait until three frames from now  
yield return null;  
yield return null;  
// do something
```

30 :

```
// do something  
yield return new WaitForSeconds (0.5f); // wait for a frame in about .5 seconds  
// do something
```

:

```
while (true)  
{  
    // do something  
    yield return null; // wait until the next frame  
}
```

Unity "Update" . "do something" .

Ticker GameObject

```
using UnityEngine;  
using System.Collections;  
  
public class Ticker:MonoBehaviour {  
  
    void OnEnable()  
    {  
        StartCoroutine(TickEverySecond());  
    }  
  
    void OnDisable()  
    {  
        StopAllCoroutines();  
    }  
  
    IEnumerator TickEverySecond()  
    {
```

```

var wait = new WaitForSeconds(1f); // REMEMBER: IT IS ONLY APPROXIMATE
while(true)
{
    Debug.Log("Tick");
    yield return wait; // wait for a frame, about 1 second from now
}
}
}

```

```

IEnumerator TickFiveSeconds()
{
    var wait = new WaitForSeconds(1f);
    int counter = 1;
    while(counter < 5)
    {
        Debug.Log("Tick");
        counter++;
        yield return wait;
    }
    Debug.Log("I am done ticking");
}

```

"" , "" . yield break .

```

IEnumerator ShowExplosions()
{
    ... show basic explosions
    if(player.xp < 100) yield break;
    ... show fancy explosions
}

```

```

void OnDisable()
{
    // Stops all running coroutines
    StopAllCoroutines();
}

```

coroutine :

```
StartCoroutine("YourAnimation");
```

StopCoroutine .

```
StopCoroutine("YourAnimation");
```

coroutine IEnumerator StartCoroutine Coroutine StopCoroutine .

```

public class SomeComponent : MonoBehaviour
{
    Coroutine routine;

    void Start () {
        routine = StartCoroutine(YourAnimation());
    }

    void Update () {
        // later, in response to some input...
        StopCoroutine(routine);
    }

    IEnumerator YourAnimation () { /* ... */ }
}

```

MonoBehaviour

MonoBehaviour .

1. ()
2. OnBecameVisible ()
3. OnLevelWasLoaded ()

```

using UnityEngine;
using System.Collections;

public class RotateObject : MonoBehaviour
{
    IEnumerator OnBecameVisible()
    {
        var tr = GetComponent<Transform>();
        while (true)
        {
            tr.Rotate(new Vector3(0, 180f * Time.deltaTime));
            yield return null;
        }
    }

    void OnBecameInvisible()
    {
        StopAllCoroutines();
    }
}

```

coroutines

.

" " .

, Unity .

```
"" . "" . .
```

```
IEnumerator BeginRace()  
{  
    yield return StartCoroutine(PrepareRace());  
    yield return StartCoroutine(Countdown());  
    yield return StartCoroutine(StartRace());  
}
```

BeginRace ...

```
StartCoroutine(BeginRace());
```

```
" " . . UI " " . AI .
```

```
yield return StartCoroutine(PrepareRace());  
yield return StartCoroutine(Countdown());  
yield return StartCoroutine(StartRace());
```

```
., IEnumerator . IEnumerator BeginRace . "" StartCoroutine .
```

```
StartCoroutine(BeginRace());
```

coroutine . coroutines . .

```
// run various routines, one after the other  
IEnumerator OneAfterTheOther( params IEnumerator[] routines )  
{  
    foreach ( var item in routines )  
    {  
        while ( item.MoveNext() ) yield return item.Current;  
    }  
  
    yield break;  
}
```

```
... . IEnumerator .
```

```
IEnumerator PrepareRace()  
{  
    // codesay, crowd cheering and camera pan around the stadium  
    yield break;  
}  
  
IEnumerator Countdown()  
{  
    // codesay, animate your countdown on UI  
    yield break;  
}  
  
IEnumerator StartRace()  
{  
    // codesay, camera moves and light changes and launch the AIs  
    yield break;  
}
```

```
}
```

```
StartCoroutine( MultipleRoutines( PrepareRace(), Countdown(), StartRace() ) );
```

```
IEnumerator[] routines = new IEnumerator[] {  
    PrepareRace(),  
    Countdown(),  
    StartRace() };  
StartCoroutine( MultipleRoutines( routines ) );
```

. Unity .

```
yield return StartCoroutine(PrepareRace());  
yield return StartCoroutine(Countdown());  
yield return StartCoroutine(StartRace());
```

```
yield return null; // wait until sometime in the next frame
```

```
//wait for a few frames  
yield return null;  
yield return null;
```

n . .

```
yield return new WaitForSeconds(n);
```

"WaitForSeconds" .

```
,,, , . .
```

```
yield return StartCoroutine(coroutine);
```

```
..:
```

```
StartCoroutine(Test());
```

```
"" .
```

```
:
```

```
Debug.Log("A");  
StartCoroutine(LongProcess());  
Debug.Log("B");
```

A B . . :

```
Debug.Log("A");
yield return StartCoroutine(LongProcess());
Debug.Log("B");
```

A B .

. :

```
Debug.Log("A");
StartCoroutine(LongProcess());
Debug.Log("B");
```

LongProcess . . . , Unity "coroutines" .

.

```
void Start() {
    string url = "http://google.com";
    WWW www = new WWW(url);
    StartCoroutine(WaitForRequest(www));
}

IEnumerator WaitForRequest(WWW www) {
    yield return www;

    if (www.error == null) {
        //use www.data);
    }
    else {
        //use www.error);
    }
}
```

: Unity ; WaitForFixedUpdate() . (Unity WaitForEndOfFrame()) . Unity .

: [https://riptutorial.com/ko/unity3d/topic/3415/-](https://riptutorial.com/ko/unity3d/topic/3415/)

39:

- Quaternion.LookRotation (Vector3 forward [, Vector3 up]);
- Quaternion.AngleAxis (float , Vector3 axisOfRotation);
- float angleBetween = Quaternion.Angle (Quaternion rotation1, Quaternion rotation2);

Examples

90, 180, 45, 30 " ". (1). Unit circle 3D .

(), Unity .

```
// Create a quaternion that represents 30 degrees about X, 10 degrees about Y
Quaternion rotation = Quaternion.Euler(30, 10, 0);

// Using a Vector
Vector3 EulerRotation = new Vector3(30, 10, 0);
Quaternion rotation = Quaternion.Euler(EulerRotation);

// Convert a transform's Quaternion angles to Euler angles
Quaternion quaternionAngles = transform.rotation;
Vector3 eulerAngles = quaternionAngles.eulerAngles;
```

?

. 3 . @ 2:09

Quaternion.LookRotation(Vector3 forward [, Vector3 up]) Y 'up' Quaternion . Vector3.up .

```
// Find a game object in the scene named Target
public Transform target = GameObject.Find("Target").GetComponent<Transform>();

// We subtract our position from the target position to create a
// Vector that points from our position to the target position
// If we reverse the order, our rotation would be 180 degrees off.
Vector3 lookVector = target.position - transform.position;
Quaternion rotation = Quaternion.LookRotation(lookVector);
transform.rotation = rotation;
```

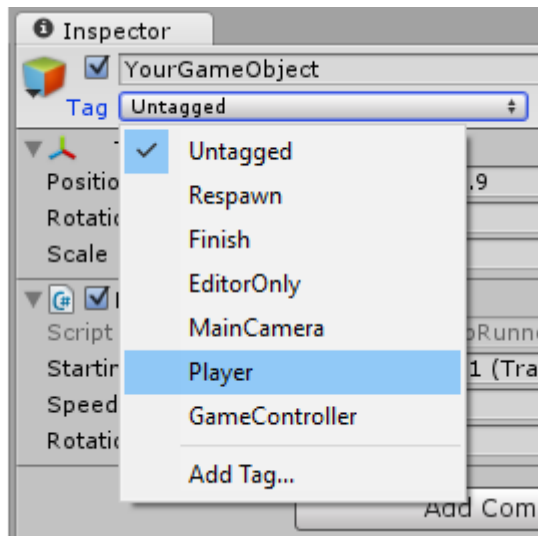
: <https://riptutorial.com/ko/unity3d/topic/1782/>

40:

GameObject . GameObject .
"Untagged" GameObject .

Examples

Tags & Layers .



, static string .

static string . static string .

```
using UnityEngine;

public class Tagging : MonoBehaviour
{
    static string tagUntagged = "Untagged";
    static string tagPlayer = "Player";
    static string tagEnemy = "Enemy";

    /// <summary>Represents the player character. This game object should
    /// be linked up via the inspector.</summary>
    public GameObject player;
    /// <summary>Represents all the enemy characters. All enemies should
    /// be added to the array via the inspector.</summary>
    public GameObject[] enemy;

    void Start ()
```



```

{
    // We ensure that the game object this script is attached to
    // is left untagged by using the default "Untagged" tag.
    gameObject.tag = tagUntagged;

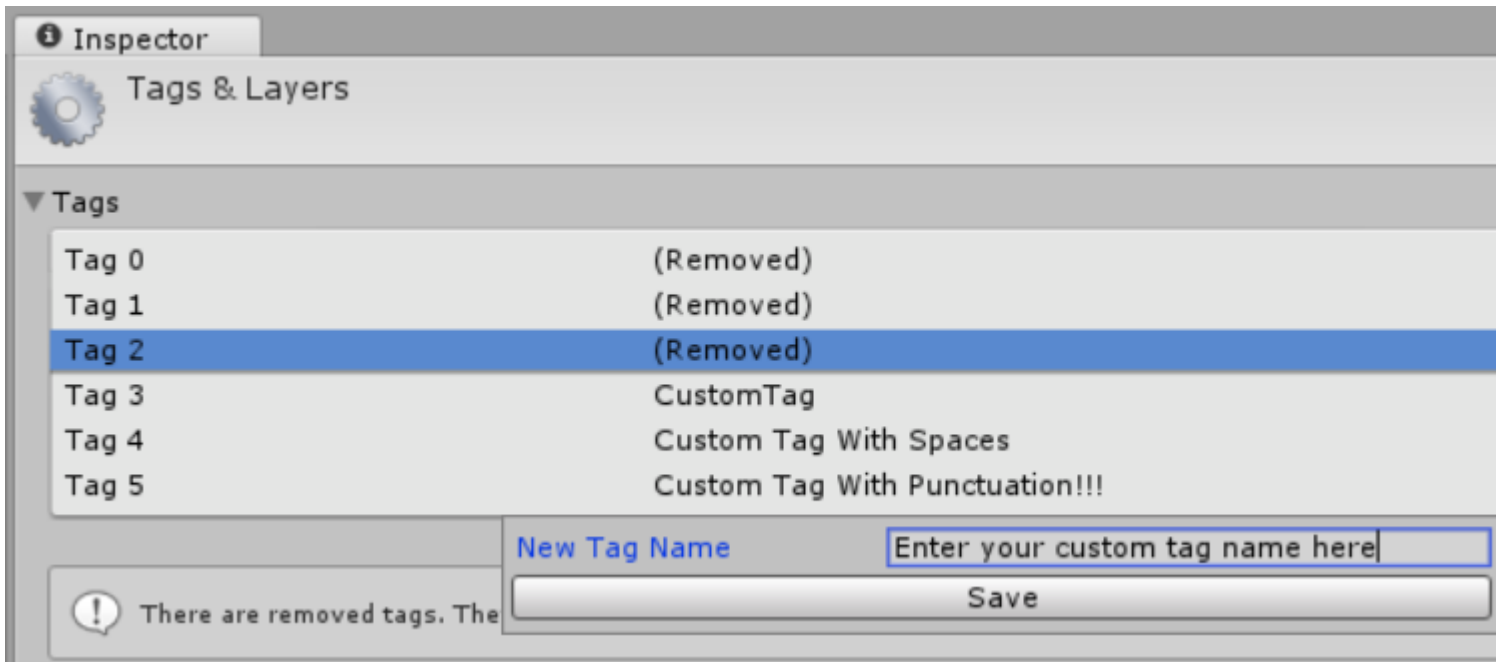
    // We ensure the player has the player tag.
    player.tag = tagUntagged;

    // We loop through the enemy array to ensure they are all tagged.
    for(int i = 0; i < enemy.Length; i++)
    {
        enemy[i].tag = tagEnemy;
    }
}
}

```

Inspector

. " ..." . > > .



+ □□ □□□ . - . "()" .

/ . " " "" .

GameObject :

.

GameObject

GameObject.FindGameObjectWithTag(string tag) . . . GameObject .

```

///

```

```
///

---


```

GameObject

```
GameObject.FindGameObjectsWithTag(string tag) . . .
GameObject.FindGameObjectWithTag(string tag) , GameObject
GameObject.FindGameObjectsWithTag(string tag) , .
```

```
///
```

GameObject .

```
if (go.Tag == "myTag")
{
    //Stuff
}
```

Update () Unity () .

```
if (go.CompareTag("myTag")
{
    //Stuff
}
```

```
public static class Tags
{
    public const string Player = "Player";
    public const string MyCustomTag = "MyCustomTag";
}
```

```
if (go.CompareTag(Tags.MyCustomTag)
{
    //Stuff
}
```

.

.

```
public enum Tags
{
    Player, Enemies, MyCustomTag;
}
```

enum toString() .

```
if (go.CompareTag (Tags.MyCustomTag.toString())
{
    //Stuff
}
```

: <https://riptutorial.com/ko/unity3d/topic/5534/>

S. No		Contributors
1	unity3d	Alexey Shimansky, Chris McFarland, Community, Desutoroiya, driconmax, Fjámínġ ómġbíé, James Radvan, josephsw, Linus Juhlin, Luís Fonseca, Maarten Bicknese, martinhodler, matiaslauriti, Mike B, Minzkraut, PlanetVaster, R.K123, S. Tarik Çetin, Skyblade, SourabhV, SP., tenpn, tim, user3071284
2	Android Plugins 101 -	Venkat at Axiom Studios
3	GameObjects	Pierrick Bignet, S. Tarik Çetin, volvis
4	MonoBehaviour	matiaslauriti, Skyblade, Thundernerd, user3797758
5	ScriptableObject	volvis
6	Unity Profiler	Amitayu Chakraborty, ForceMagic, RamenChef, Skyblade
7	Unity Git	Commodore Yournero, Hacky, James Radvan, matiaslauriti, Max Yankov, Maxim Kamalov, Pierrick Bignet, Ricardo Amores, S. Tarik Çetin, S.Richmond, Skyblade, Thulani Chivandikwa, YsenGrimm, yummpasta
8	(VR)	4444, Airwarfare, Guglie, pew., Pratham Sehgal, tim
9		Chris McFarland, Ed Marty, lase, matiaslauriti, S. Tarik Çetin, Thulani Chivandikwa, Thundernerd, ıolæz əyɫ qoq, volvis
10		ıolæz əyɫ qoq
11		David Martinez, driconmax, Rafiwui, RamenChef
12		Fjámínġ ómġbíé
13		David Darias, Fehr, James Radvan, JohnTube, matiaslauriti, Maxim Kamalov, Simon Heinen, SP., Tiziano Coroneo, Umair M, volvis, Zze,
14		Ian Newland
15		driconmax, Meinkraft, Skyblade, user3570542, volvis, wouterrobot
16		Arijoon, dreadnought, Light Drake, RamenChef, Skyblade

17		user3797758 , volvis
18		Airwarfare , Skyblade
19		eunoia , F̂lámínġ óm̂bíé , jack jay
20	3	driconmax , F̂lámínġ óm̂bíé , Gnemlock
21		ADB , Jean Vitor , matiaslauriti , S. Tarık Çetin , Skyblade , Thundernerd , Xander Luciano
22	(UI)	Helium , matiaslauriti , Maxim Kamalov , Programmer , RamenChef , Skyblade , Umair M
23		David Martinez , devon t , F̂lámínġ óm̂bíé , Maxim Kamalov , tim
24		4444 , Thundernerd
25	()	gman , Skyblade , volvis
26		Pierrick Bignet , Skyblade , Thundernerd , l̂l̂æz æɥl qoq , volvis
27		R4mbi , l̂l̂æz æɥl qoq
28		4444 , Fiery Raccoon , Guglie
29		Programmer , Skyblade , l̂l̂æz æɥl qoq
30		JakeD , Trent , zwcloud
31		F̂lámínġ óm̂bíé
32		glaubergft , MadJlzz , Skyblade , Venkat at Axiom Studios
33		Brandon Mintern , Dávid Florek , F̂lámínġ óm̂bíé , gman , Gnemlock , Guglie , James Radvan , Jean Vitor , josephsw , Lich , matiaslauriti , Skyblade , Thulani Chivandikwa , l̂l̂æz æɥl qoq , Woltus , yumypasta
34	(IMGUI)	Skyblade , Soaring Code
35		Ed Marty , EvilTak , F̂lámínġ óm̂bíé , Grigory , JohnTube , Skyblade , Thulani Chivandikwa , volvis
36		F̂lámínġ óm̂bíé , jjhavokk , Xander Luciano
37	API	volvis
38		agiro , Fattie , Fehr , Giuseppe De Francesco , Problematic , Skyblade , Thulani Chivandikwa , Thundernerd , l̂l̂æz æɥl qoq ,

		volvis
39		matiaslauriti , Tiziano Coroneo , Xander Luciano , yummypasta
40		Arijoon , Augure , glaubergft , Gnemlock , MadJlzz , Skyblade , Trent