

Virtual Reality Haptic Interactions

(虛擬實境與觸覺回饋互動)

Lecturer: Ray

Unity Installation



[Products](#)[Solutions](#)[Made with Unity](#)[Learning](#)[More](#)[Get started](#)

Unity for all

Start bringing your vision to life today with the Unity real-time 3D development platform.

[Get started](#)[Learn more](#)[Games](#)[Automotive,
Transportation](#)[Film, Animation
& Cinematics](#)[Architecture,
Engineering &
Construction](#)

Plans and pricing

We offer a range of plans for all industries and levels of expertise.

All plans are royalty-free.

Individual

Business

Plus

More functionality and resources to

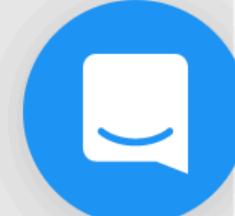
Pro

MOST POPULAR

A complete solution for professionals

Enterprise

Custom solutions at scale,
designed to support your



Individual

Business

Student

Learn the tools and workflows professionals use on the job

Free

[Sign up](#)

Eligibility:

Students enrolled in an accredited educational institution of legal age to consent to the collection and processing of their personal information, e.g., age 13 in the US, 16 in the EU. Must join the GitHub Student Developer Pack to

Personal

Start creating with the free version of Unity

Free

[Get started](#)

[Learn more](#)

Eligibility:

Revenue or funding less than \$100K in the last 12 months

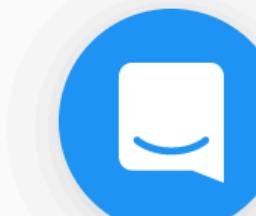
Learn Premium

Master Unity with expert-led live sessions and on-demand learning

[Try for free](#)

[Learn more](#)

Included with Plus, Pro and Enterprise plans



Start Creating with Unity

First-time Users

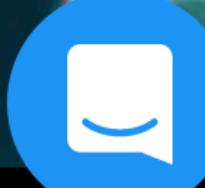
Download Unity with this fun, **guided learning path** to create your first game today!

[Start here](#)

Returning Users

Already have experience creating with Unity and want to download the **standard Unity installer** instead?

[Go here](#)



[Products](#)[Solutions](#)[Made with Unity](#)[Learning](#)[Support & Services](#)[Community](#)[Get started](#)[Asset Store](#)

Download Unity

Welcome! You're here because you want to download Unity, the world's most popular development platform for creating 2D and 3D multiplatform games and interactive experiences.

Before you download choose the version of Unity that's right for you.

[Choose your Unity + download](#)[Download Unity Hub](#)

[Learn more about the new Unity Hub here.](#)

Download Unity Beta

Get early access to our latest features, and help us improve quality by providing valuable feedback.

[Download Beta](#)

Already a customer?

We use cookies to ensure that we give you the best experience on our website. Visit our [cookie policy page](#) for more information.

[Got it](#)

System requirements

OS: Windows 7 SP1+, 8, 10, 64-bit versions only; Mac OS X 10.12+; Ubuntu 16.04, 18.04, and CentOS 7.

GPU: Graphics card with DX10 (shader model 4.0) capabilities.

[Learn more](#)

Resources

- [LTS Releases](#)
- [Older versions of Unity](#)
- [Unity 2019.3 upgrade guide](#)
- [Patch releases](#)
- [Release notes](#)
- [Engine features](#)
- [Unity User Manual](#)
- [FAQ](#)

授權協議

在安裝 Unity Hub 之前，請檢閱授權條款。



檢閱協議的其餘部分，按 [PgDn] 往下捲動頁面。

Unity Terms of Service

Last updated: May 24, 2018

Unity Technologies ApS ("Unity", "our" or "we") provides game-development and related software (the "Software"), development-related services (like [Unity Teams](#) ("Developer Services")), and various Unity communities (like [Unity Answers](#) and [Unity Connect](#) ("Communities")), provided through or in connection with our website accessible at [unity3d.com](#) or [unity.com](#)

如果接受協議的條款，按 [我同意(A)] 繼續安裝。必須要接受協議才能安裝 Unity Hub。

Unity Hub 2.3.0

我同意(A)

取消

選取安裝位置

選取 Unity Hub 要安裝的資料夾。



安裝程式會將 Unity Hub 安裝在以下資料夾。要安裝到不同的資料夾，按「瀏覽(B)...」並選擇其他資料夾。按「安裝(I)」開始安裝。

目標資料夾

C:\Program Files\Unity Hub

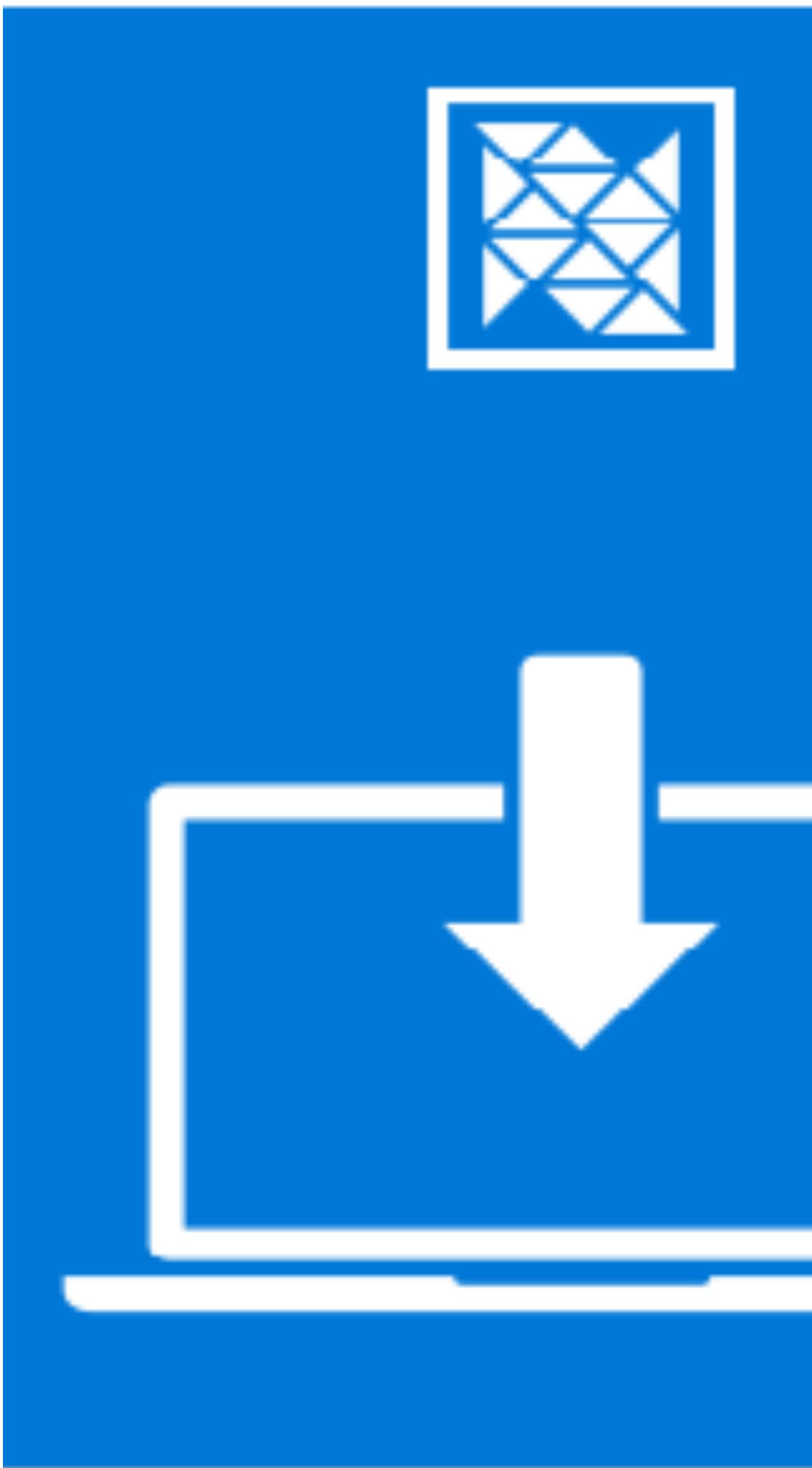
瀏覽(R)...

Unity Hub 2.3.0

< 上一步(B)

安裝(I)

取消



即將完成安裝 Unity Hub

已在電腦安裝 Unity Hub。
按 [完成(F)] 關閉安裝程式。

執行 Unity Hub(R)

< 上一步(B)

完成(F)

取消



廖昱

專案

學習

Community(社群)

安裝

安裝

選取位置

新增

2019.3.4f1

2019.2.17f1

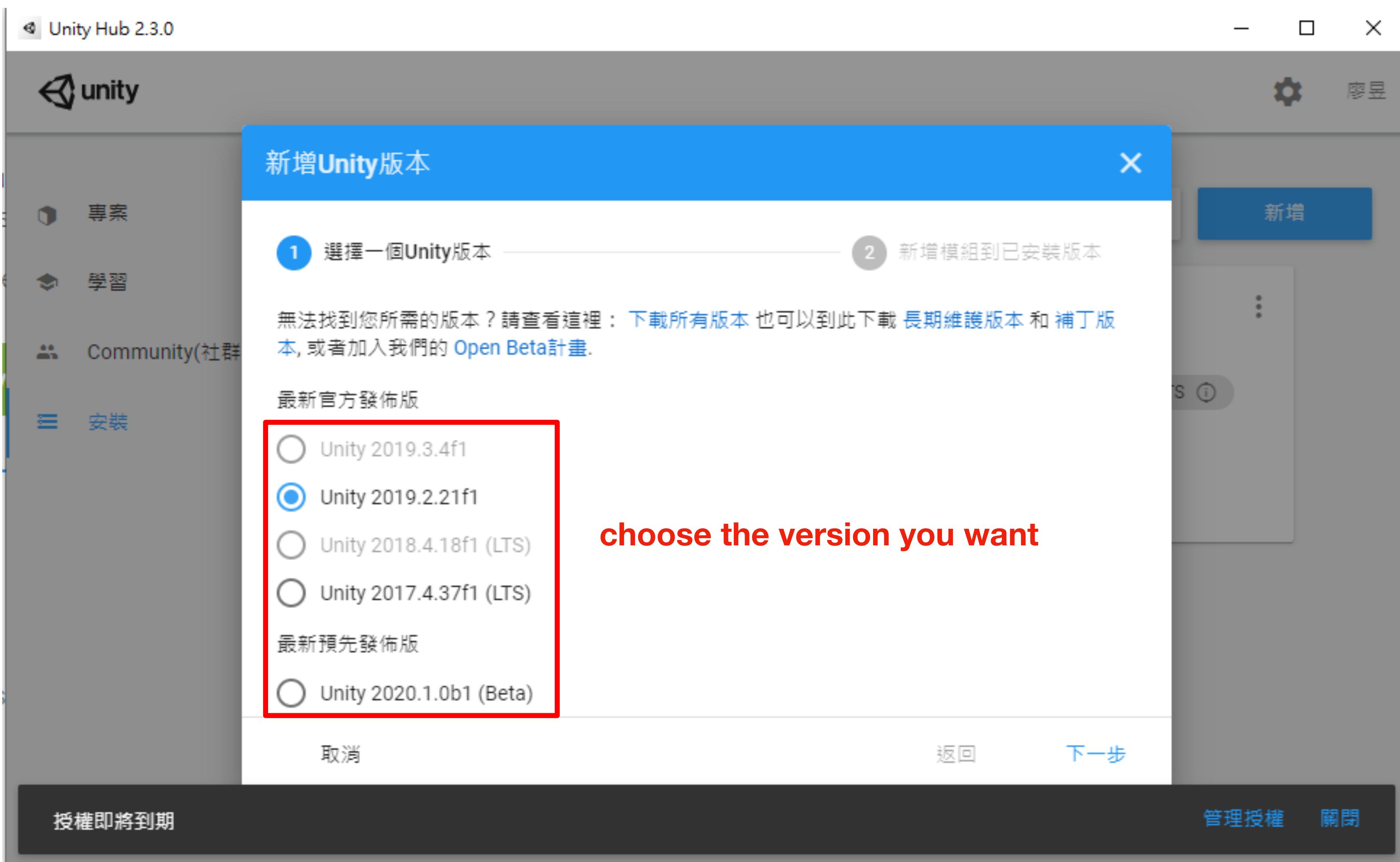
2018.4.18f1 LTS ⓘ

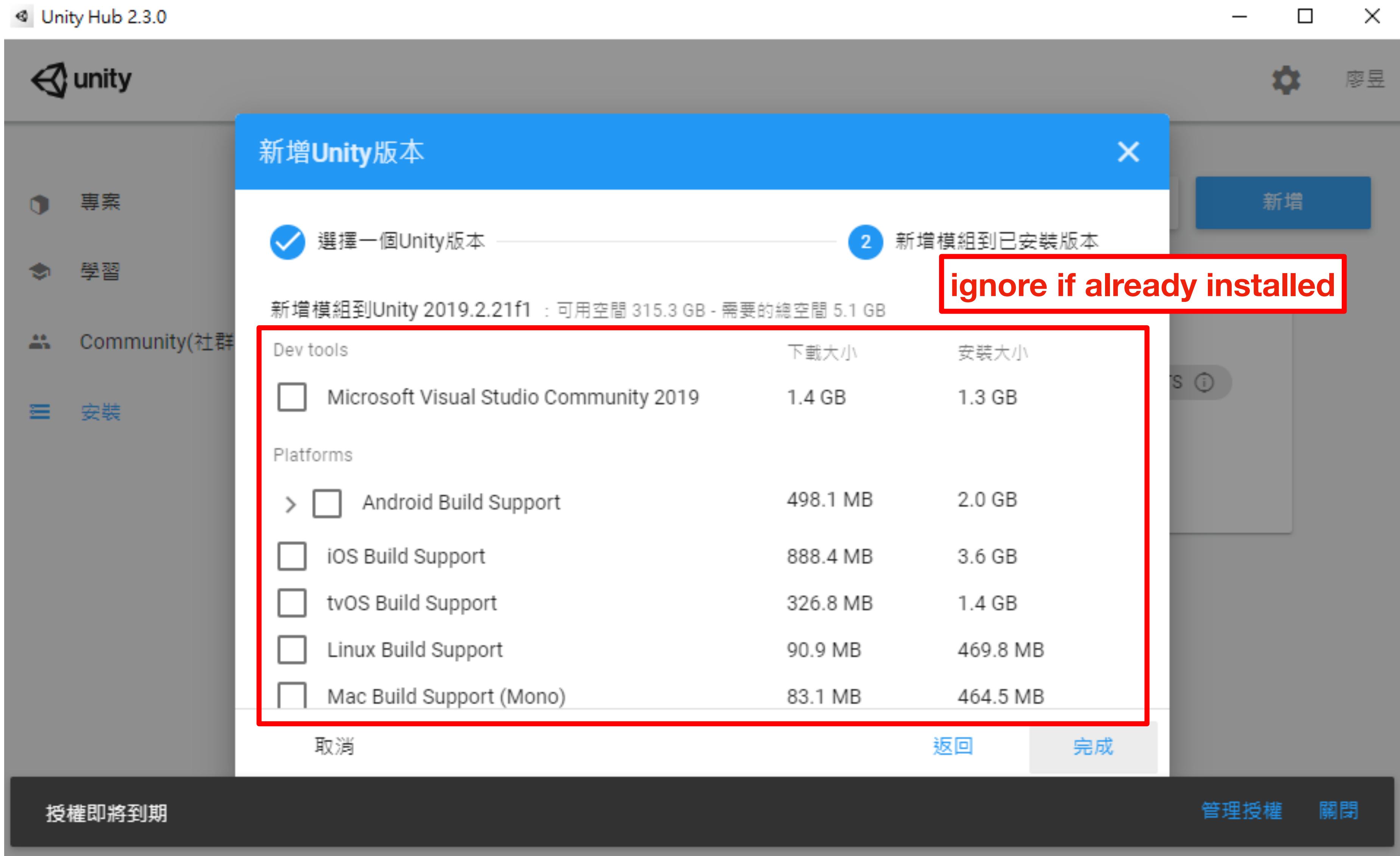
already installed

授權即將到期

管理授權 關閉

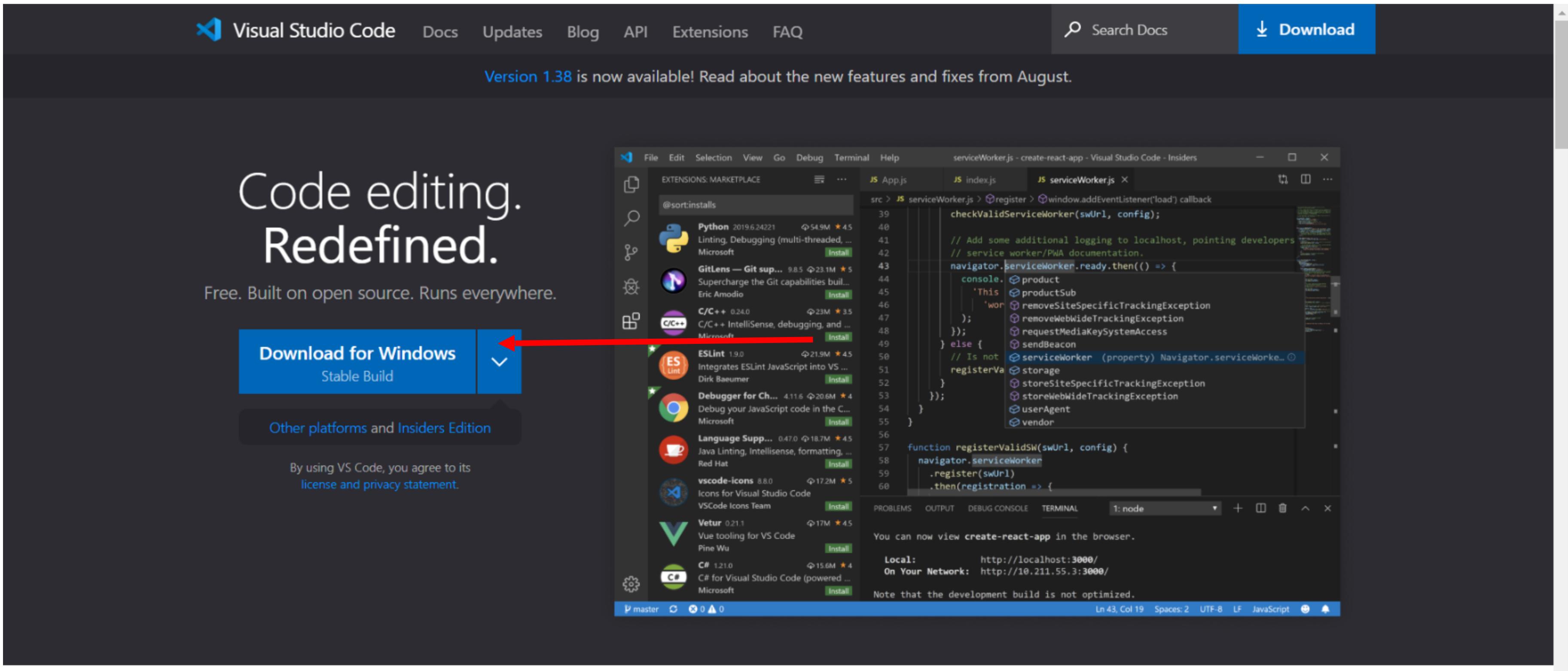
The screenshot shows the Unity Hub interface. On the left, there's a sidebar with '專案', '學習', 'Community(社群)', and a selected '安裝' tab. The main area is titled '安裝' and lists three Unity versions: '2019.3.4f1', '2019.2.17f1', and '2018.4.18f1 LTS ⓘ'. A red box highlights the first two versions. Below the list, the text 'already installed' is displayed in red. At the bottom, there's a dark bar with '授權即將到期', '管理授權', and '關閉' buttons.





Visual Studio Code

Installation (optional)



Go to: <https://code.visualstudio.com/> and download

Unity Introduction

Special thanks!!!

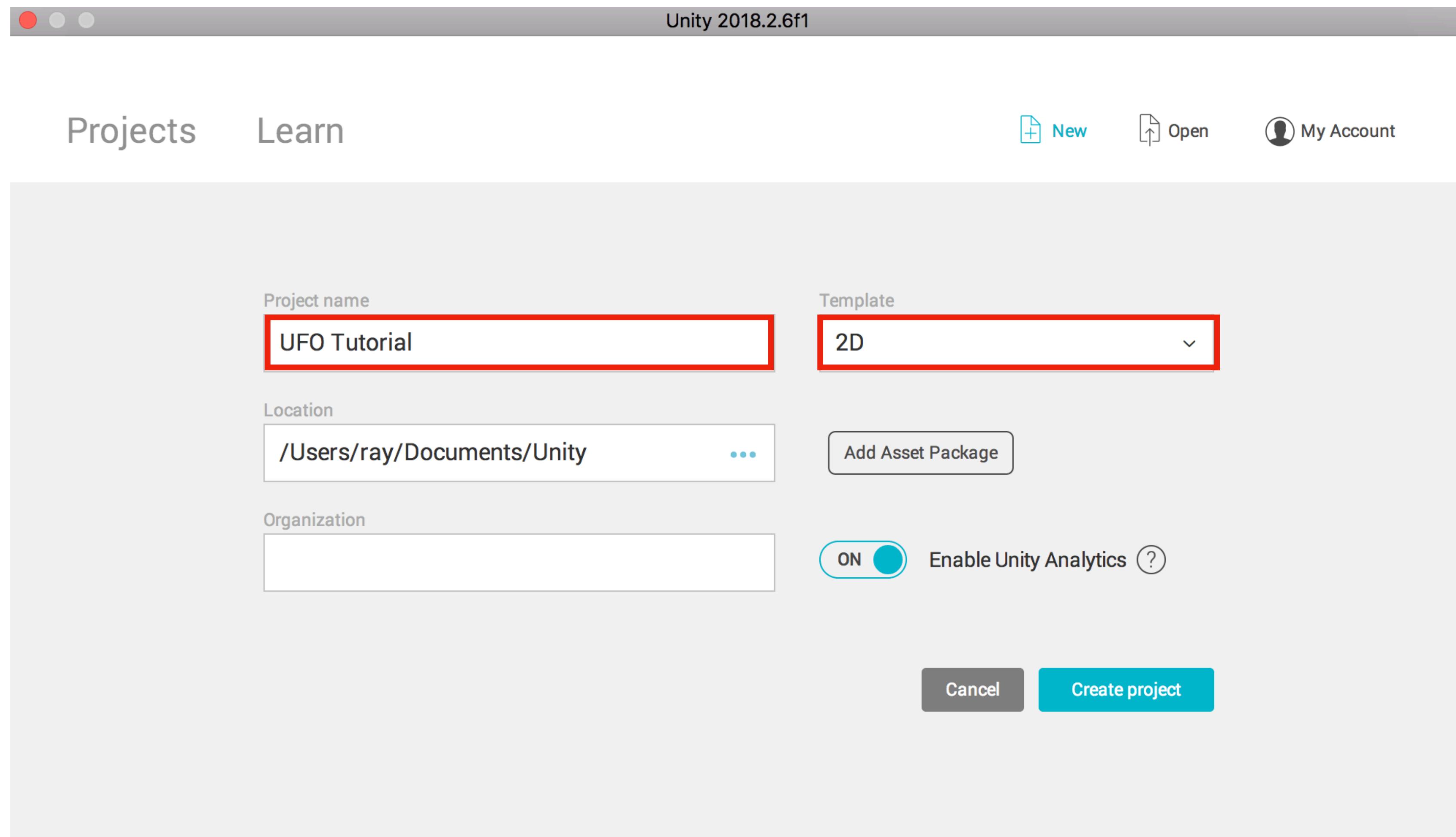


赤燭遊戲 包子

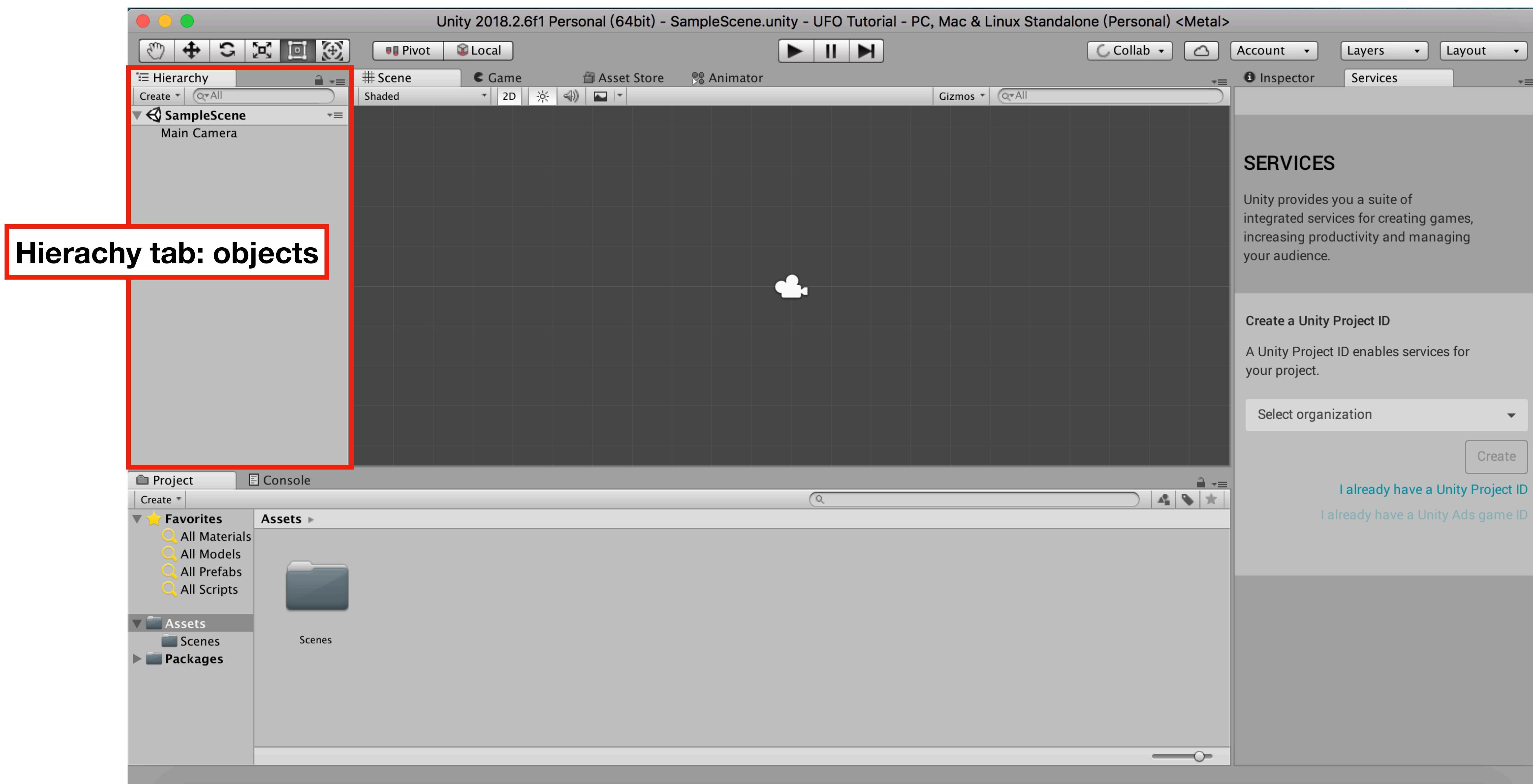
Unity Introduction

Tutorials for learning Unity:
<https://learn.unity.com/tutorials>

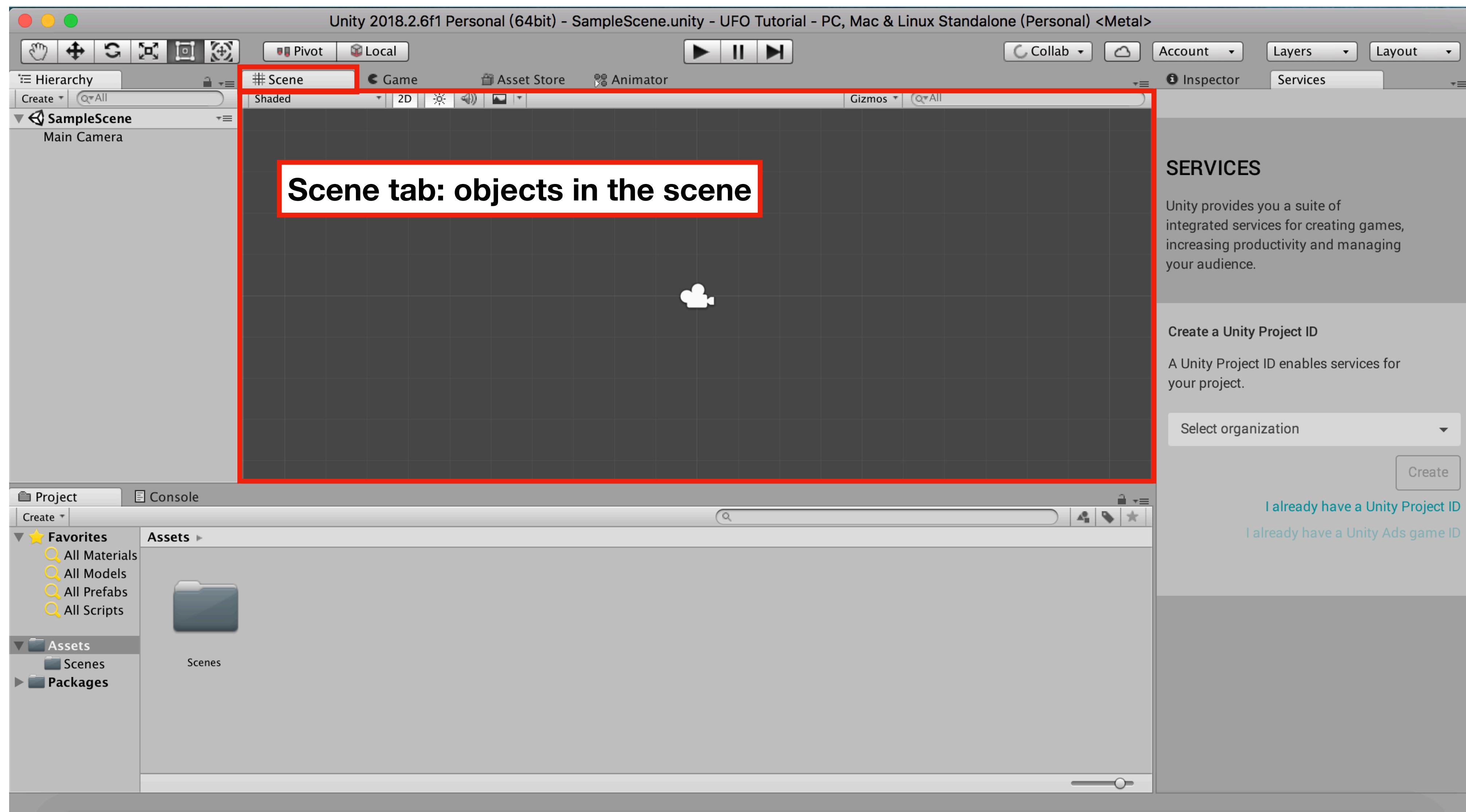
Create a Project



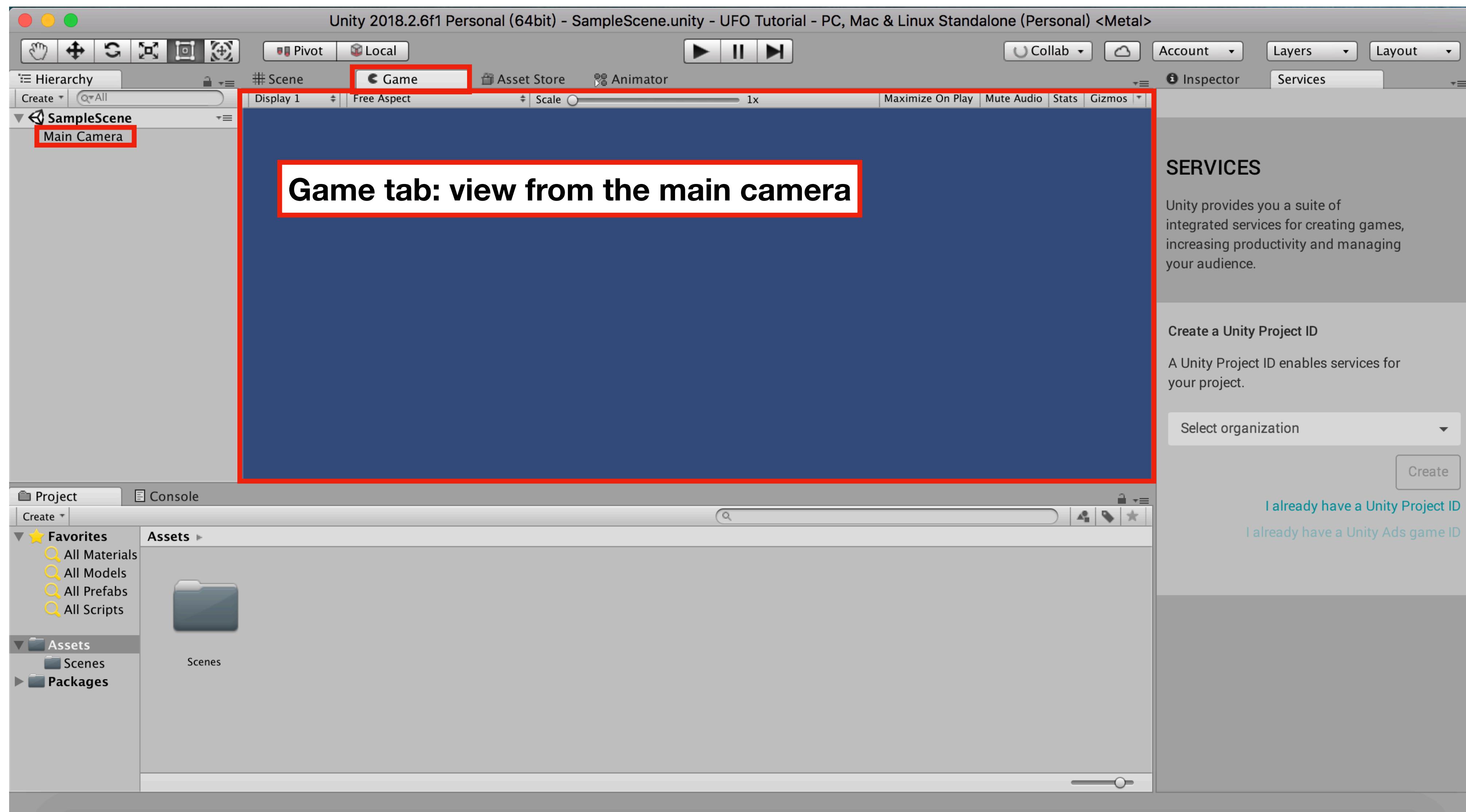
Unity IDE



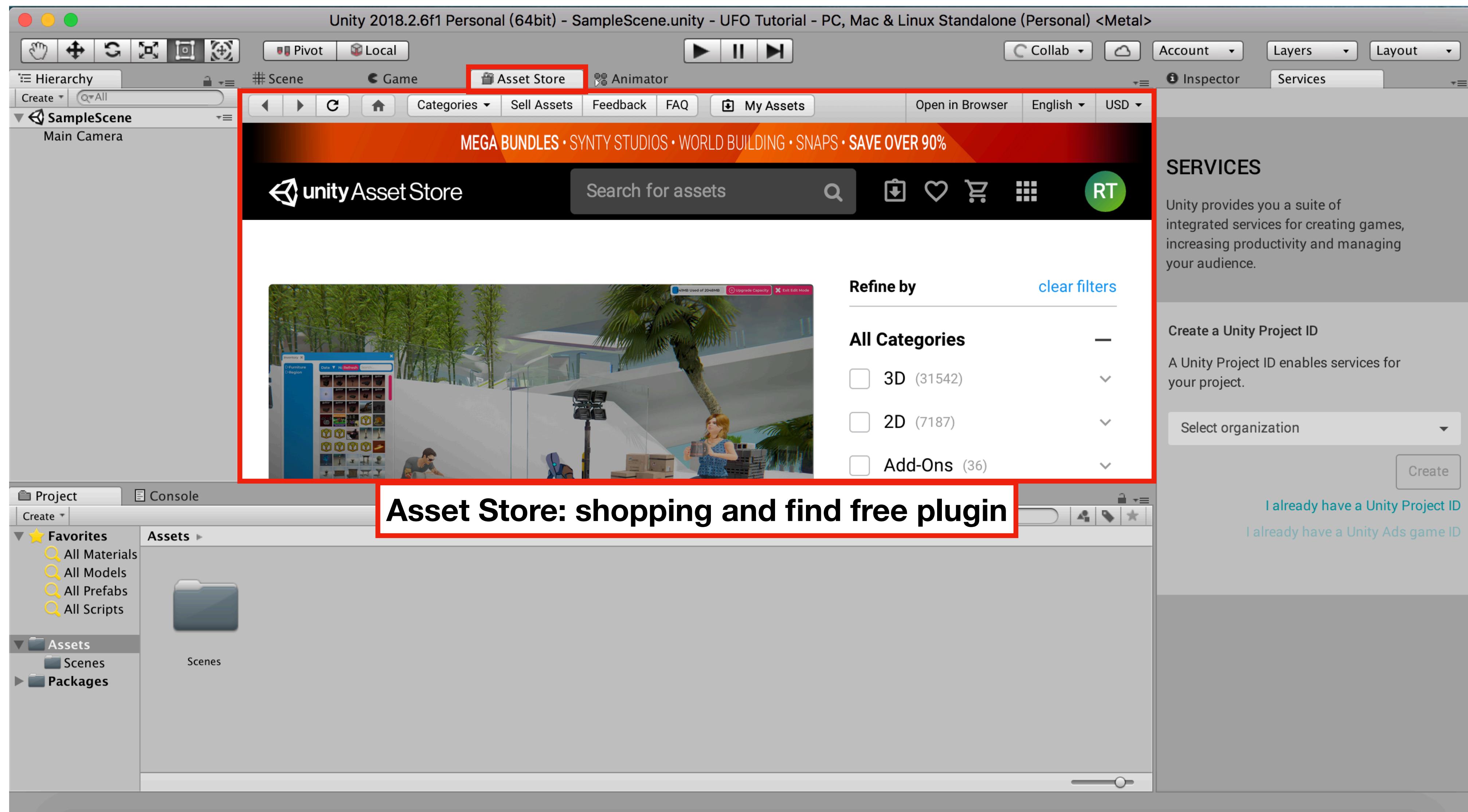
Unity IDE



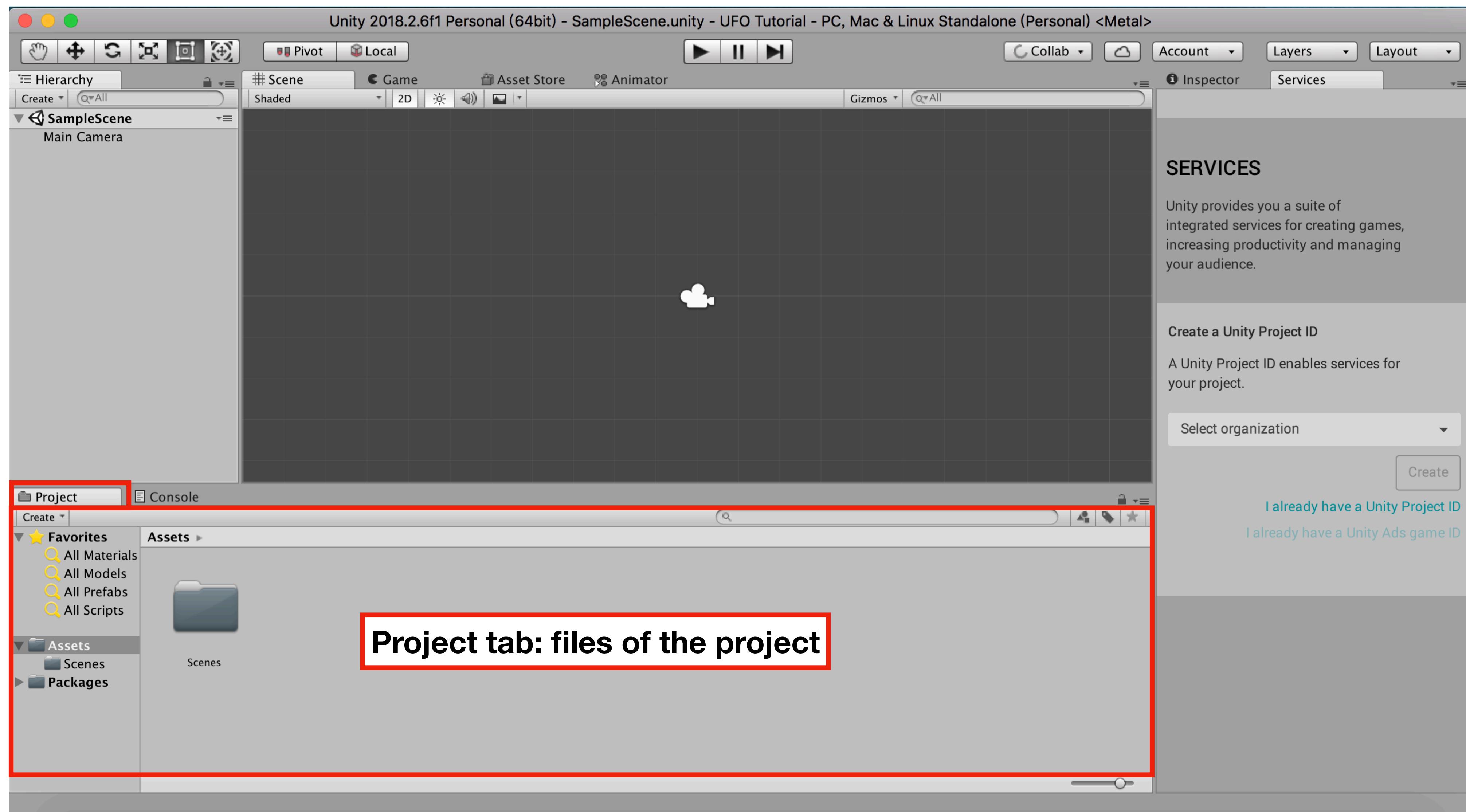
Unity IDE



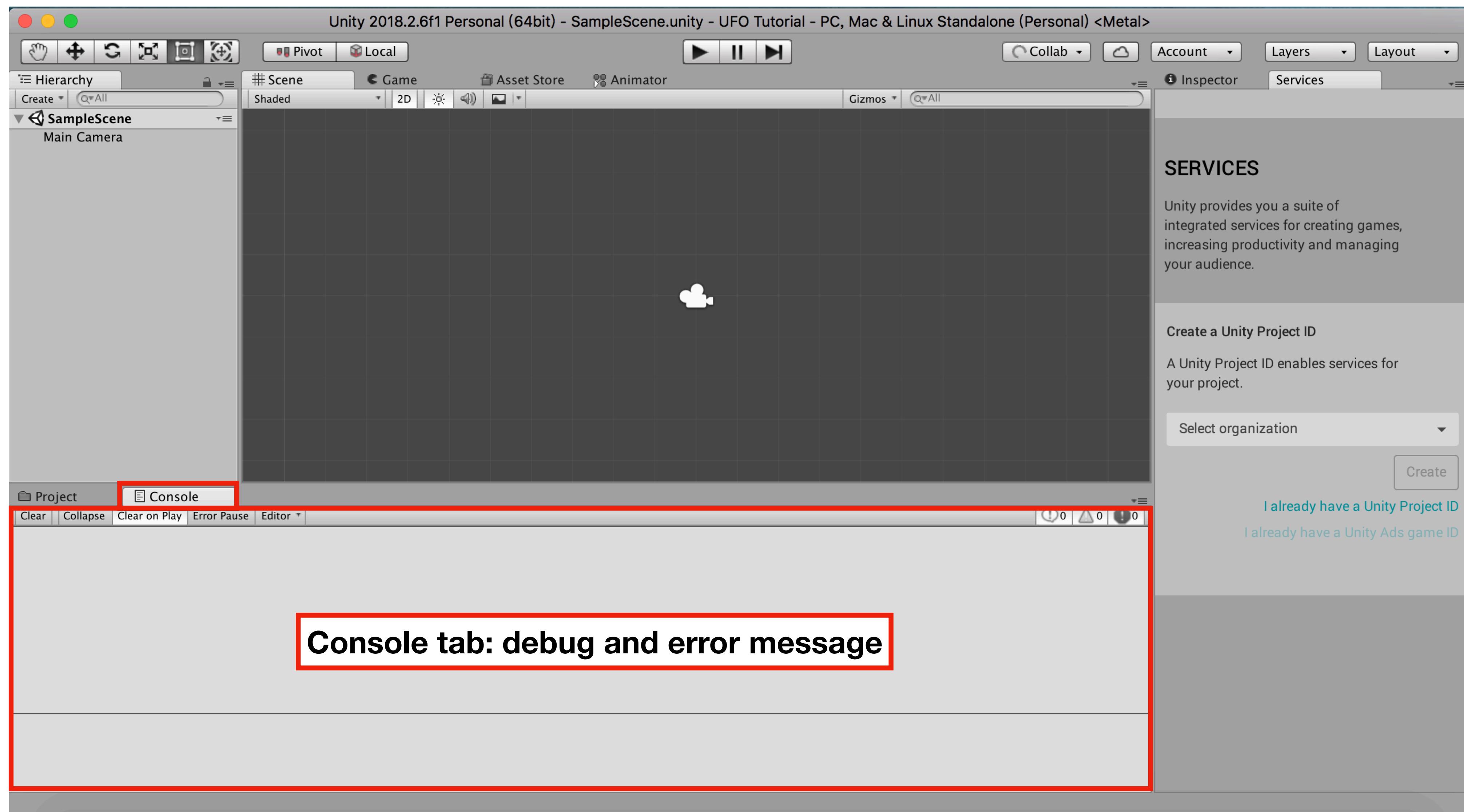
Unity IDE



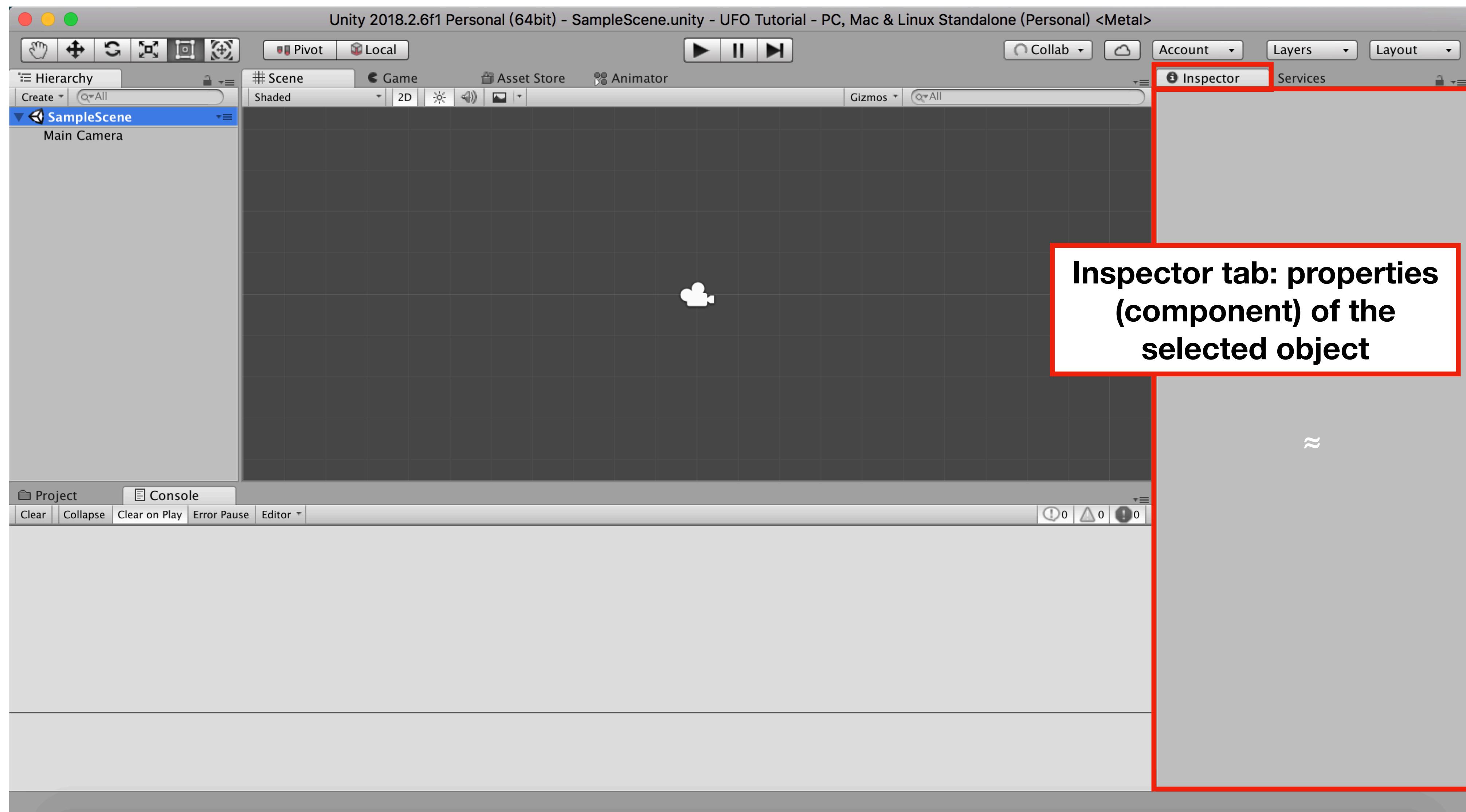
Unity IDE



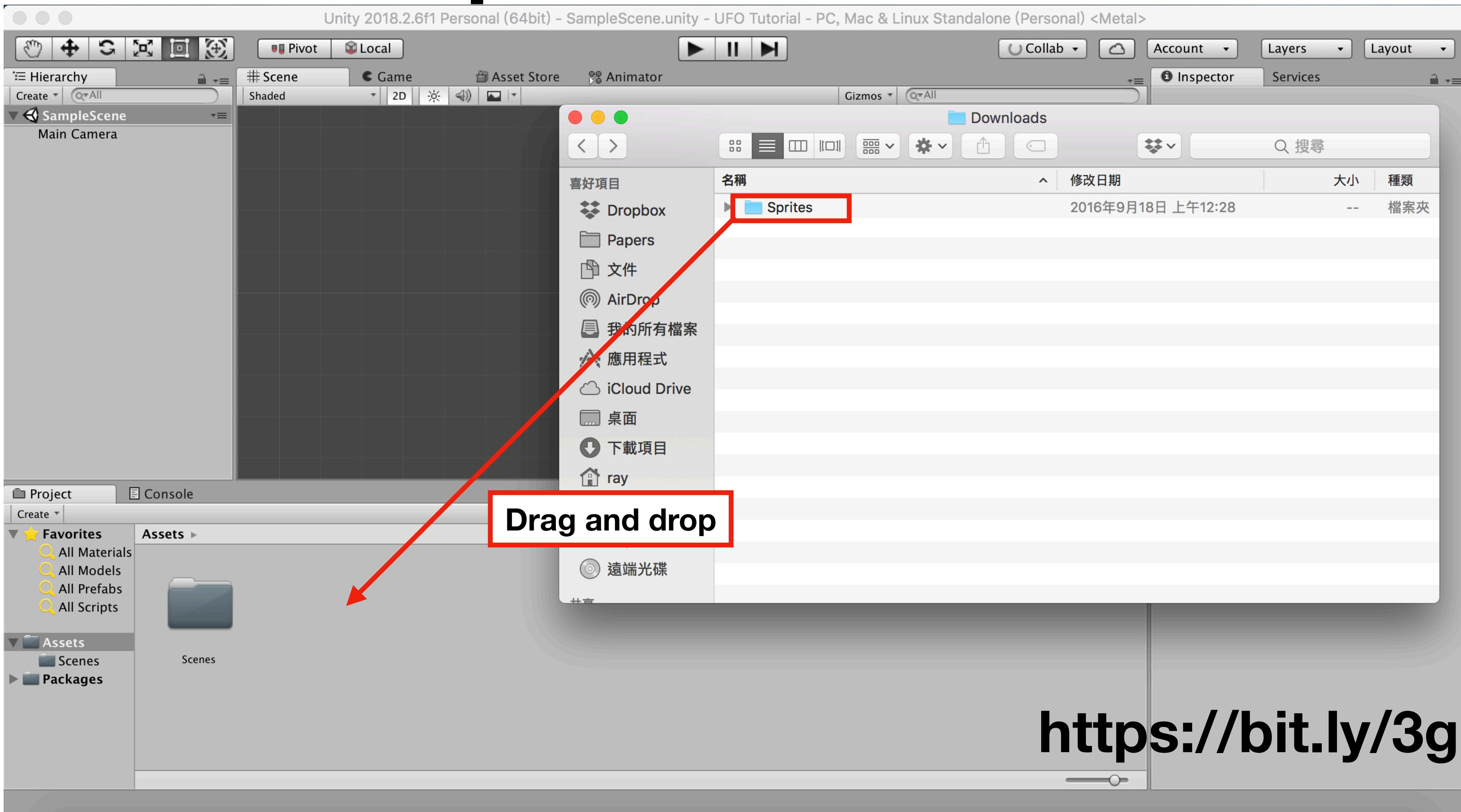
Unity IDE



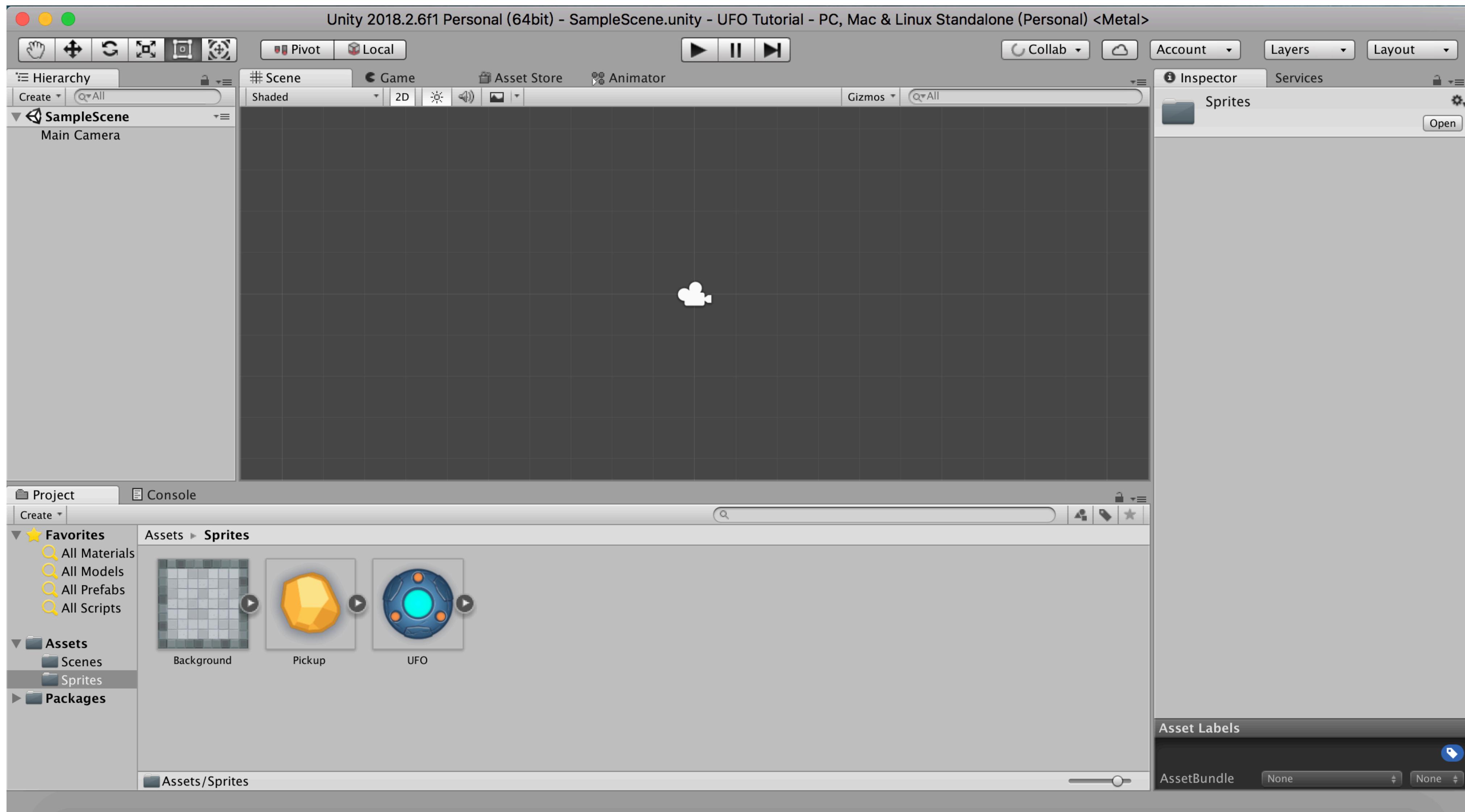
Unity IDE



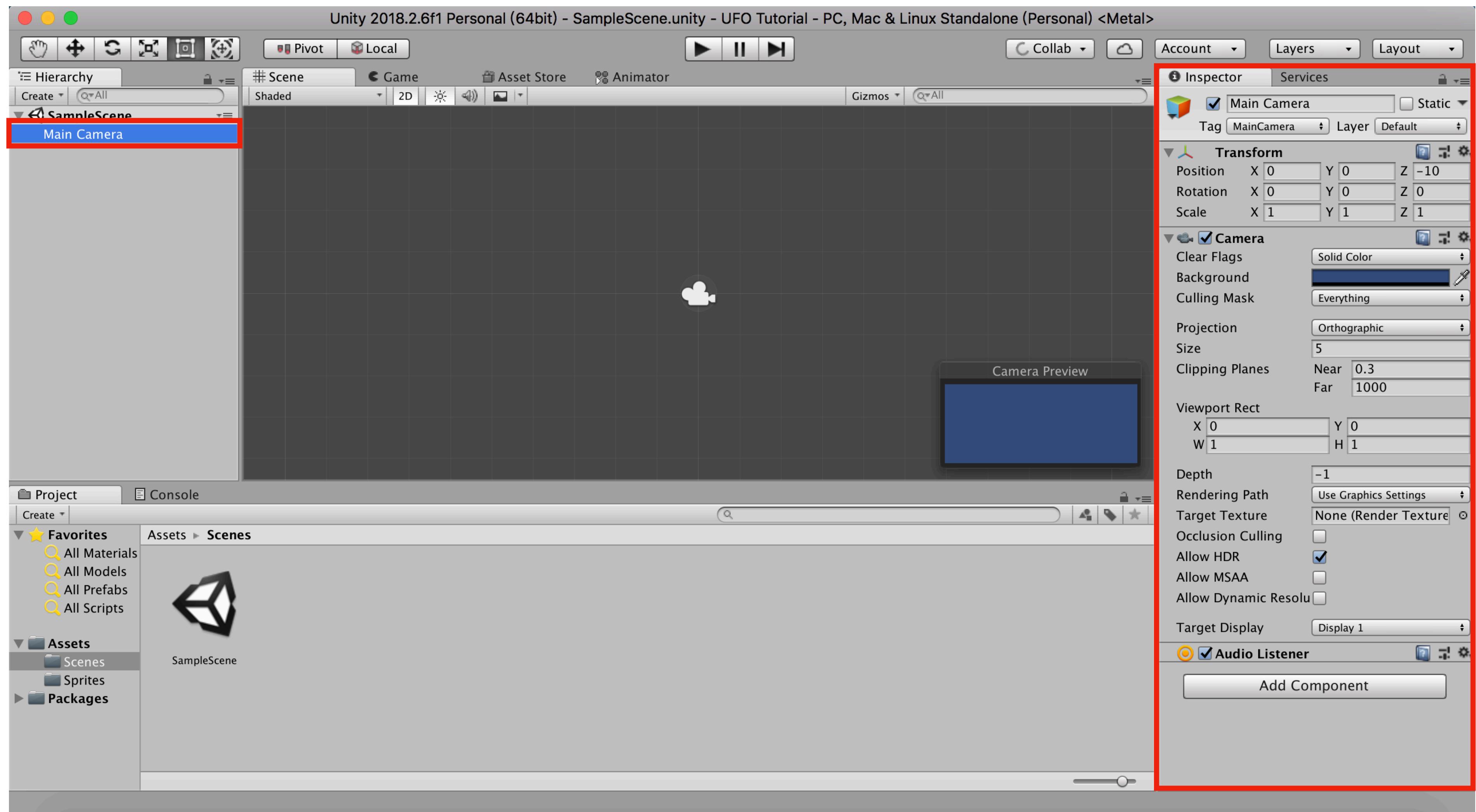
Import Materials



Import Materials



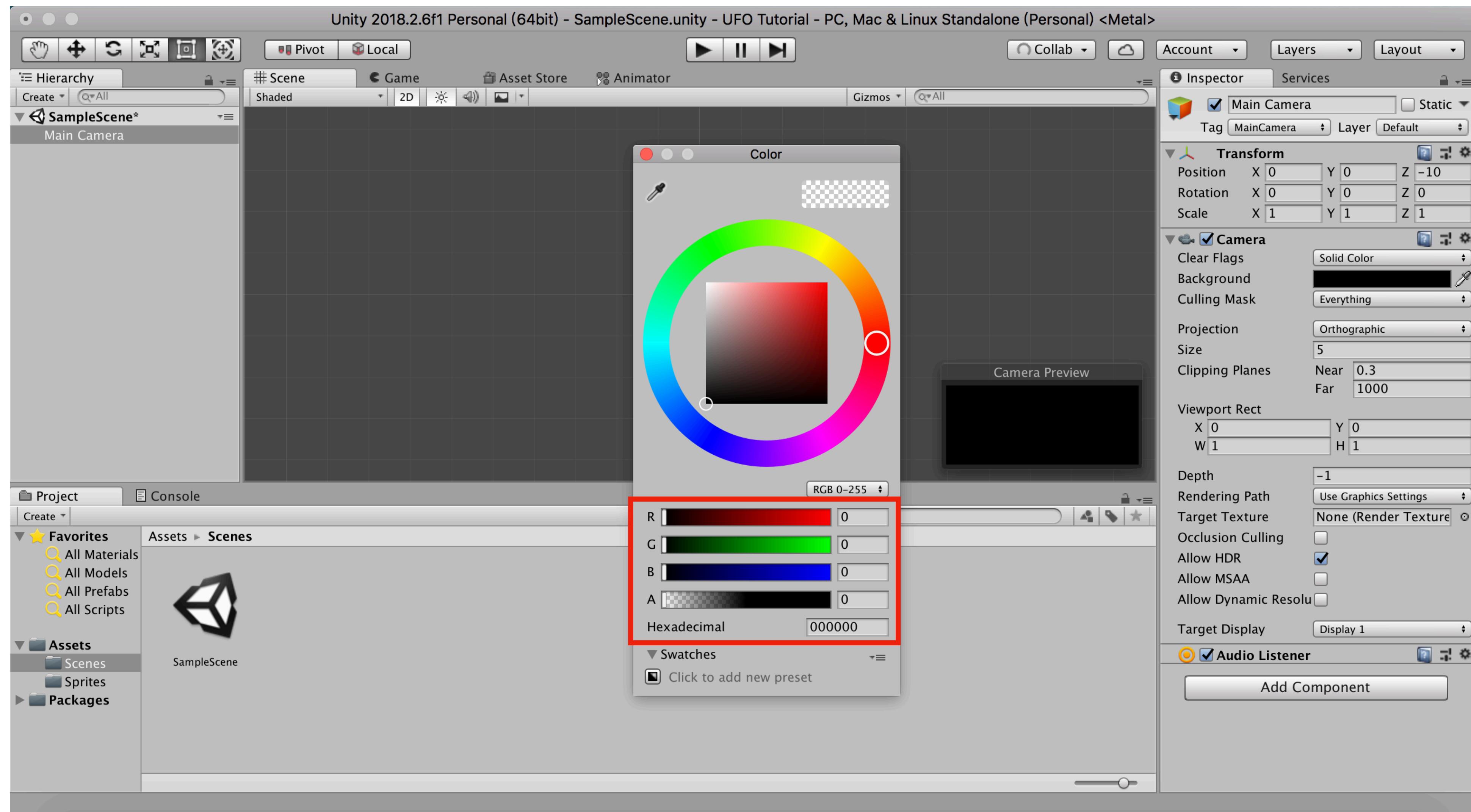
Main Camera



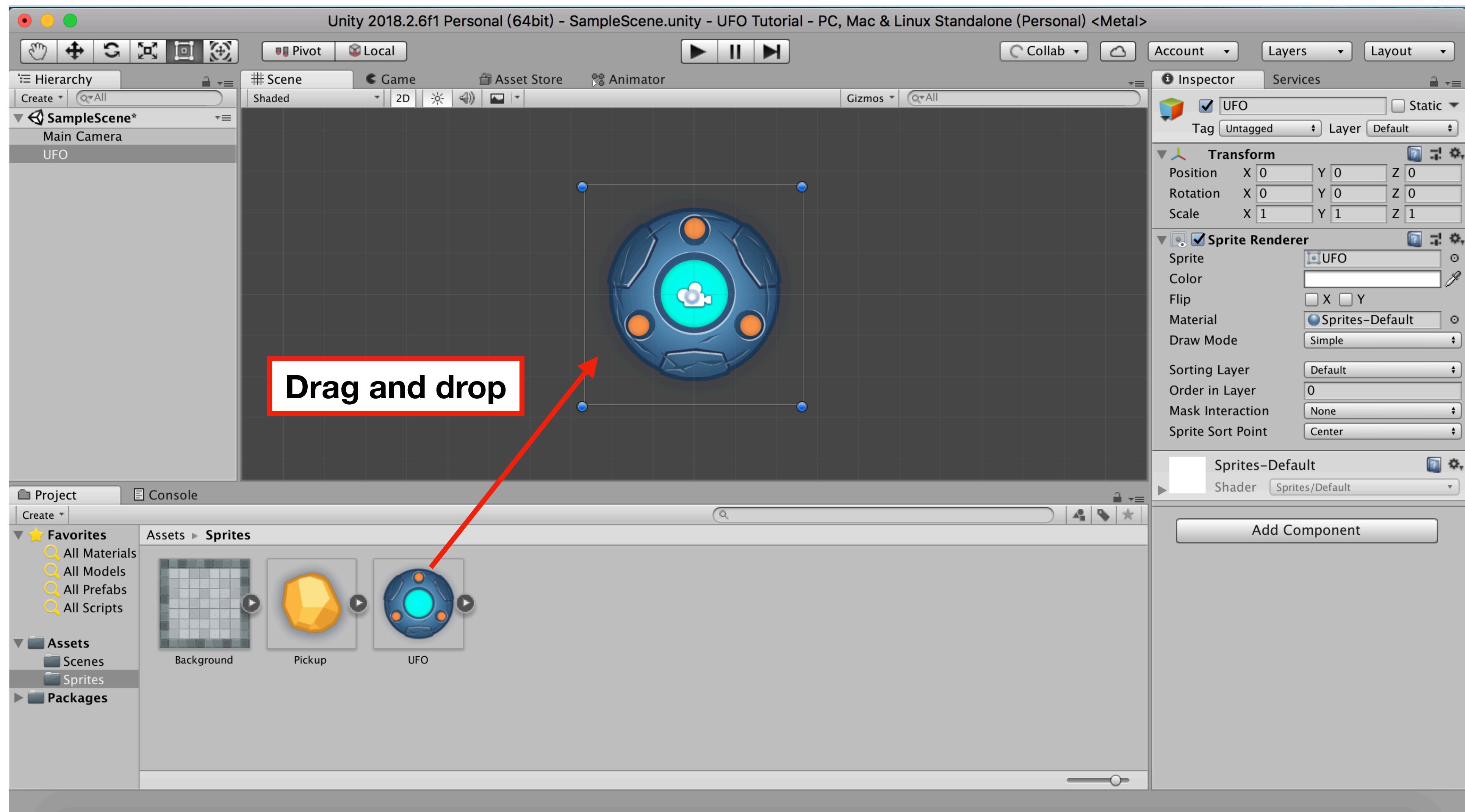
Main Camera



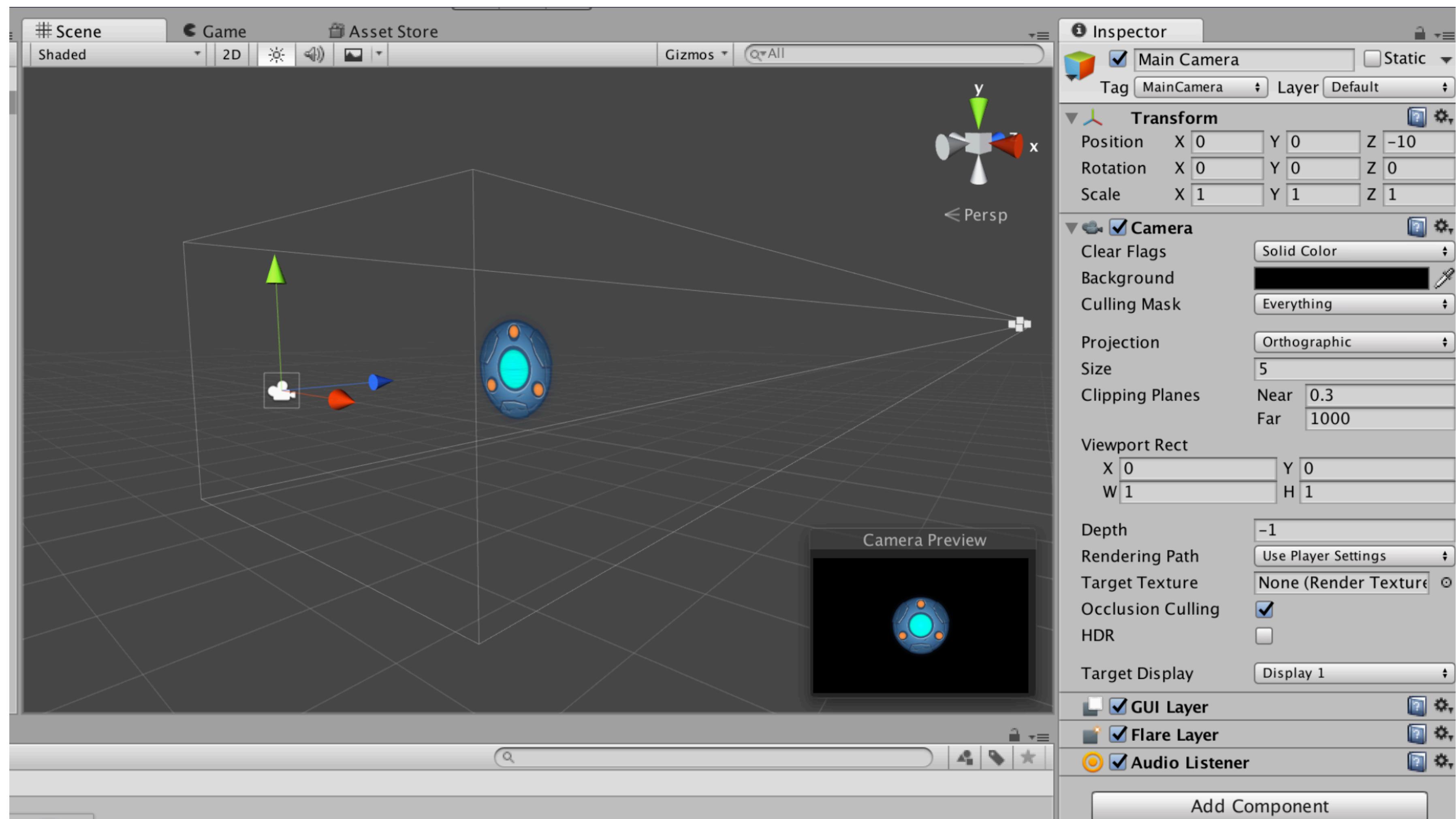
Make Background Black



Add Assets in the Scene



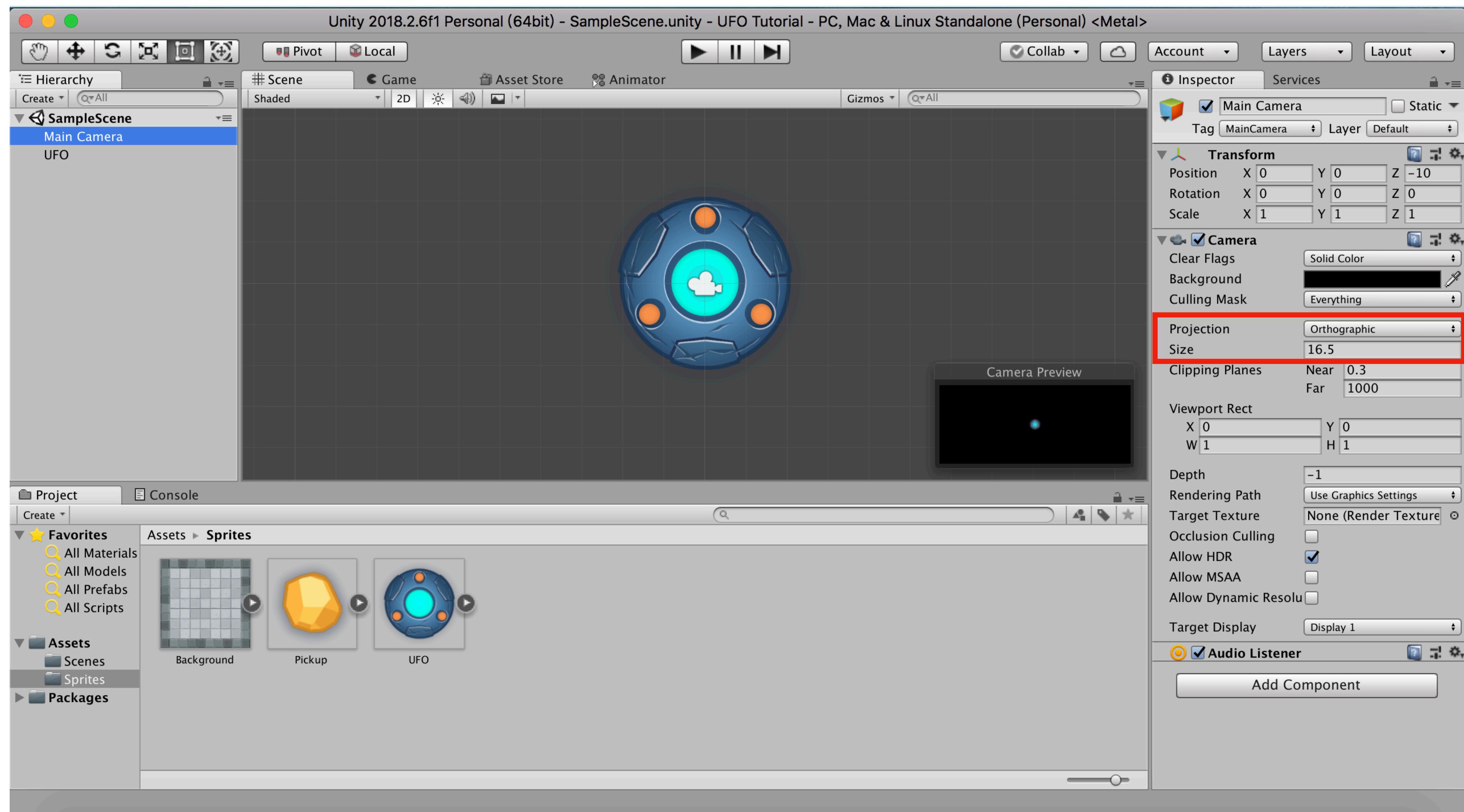
How it looks now



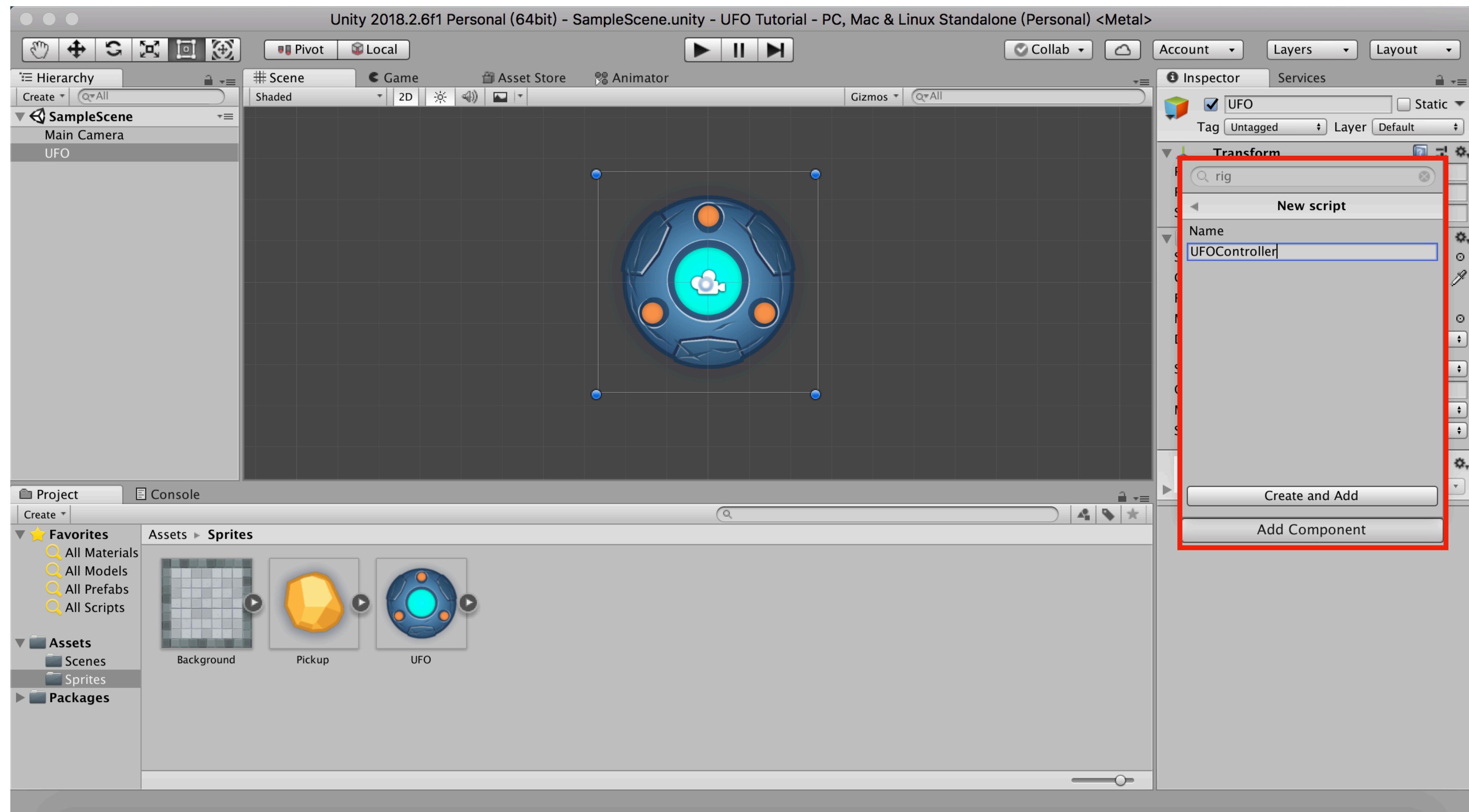
Orthographic v.s. Perspective

Orthographic	<p>視野範圍在空間中是一個巨大的矩形。</p> <p>視野範圍由矩形的長寬高決定。</p> <p>物件不論遠近看起來都一樣大。</p> <p>Camera移動時，不論遠景或近景，物件的相對位置都不會位移。</p>
Perspective	<p>視野範圍在空間中是一個四角錐。</p> <p>Camera在角錐尖端。</p> <p>視野範圍由角錐的張開角度決定。</p> <p>物件近的較大，遠的較小</p> <p>Camera移動時，遠景動比較慢，近景動比較快。</p>

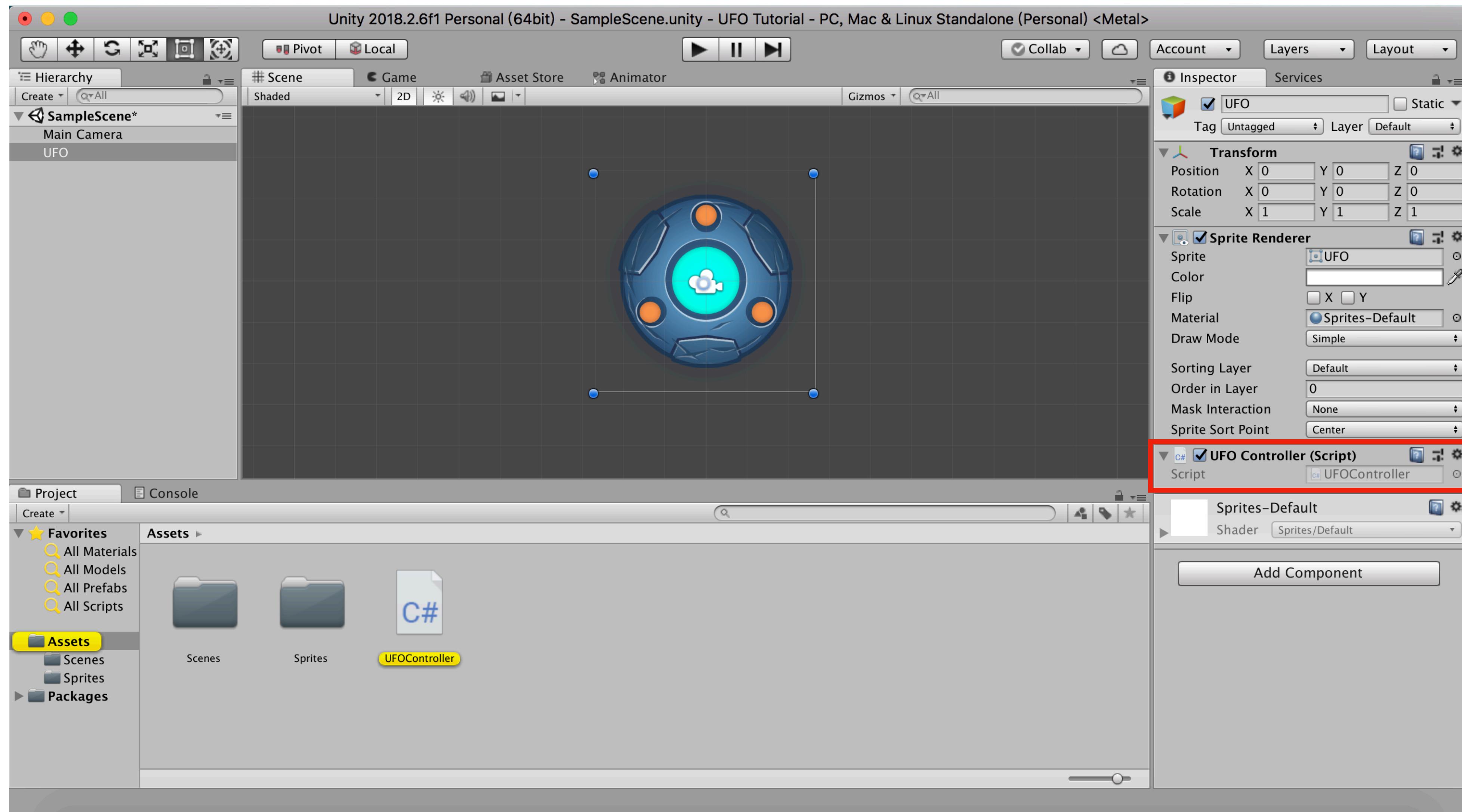
Main Camera



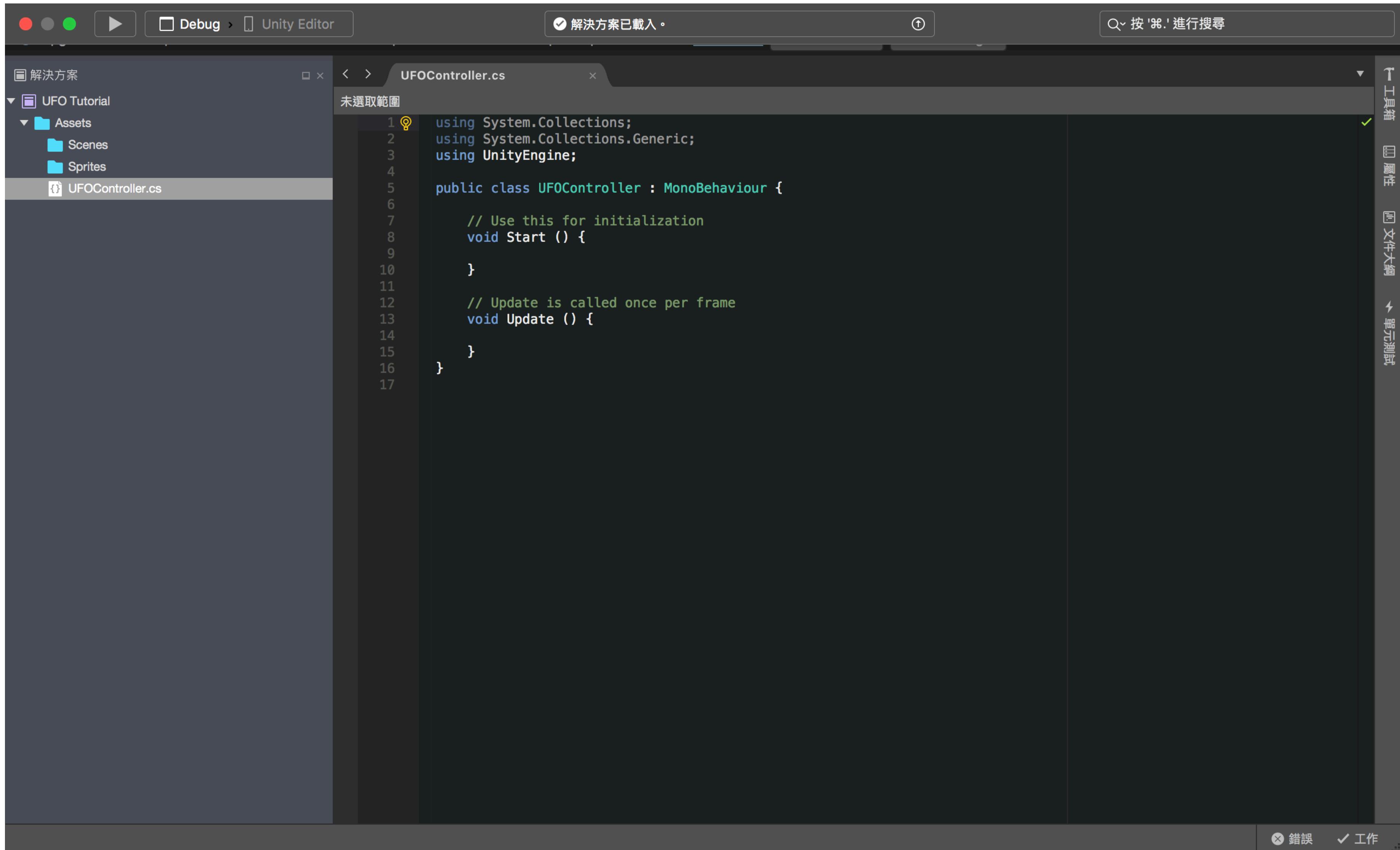
Add Script



Add Script



Script

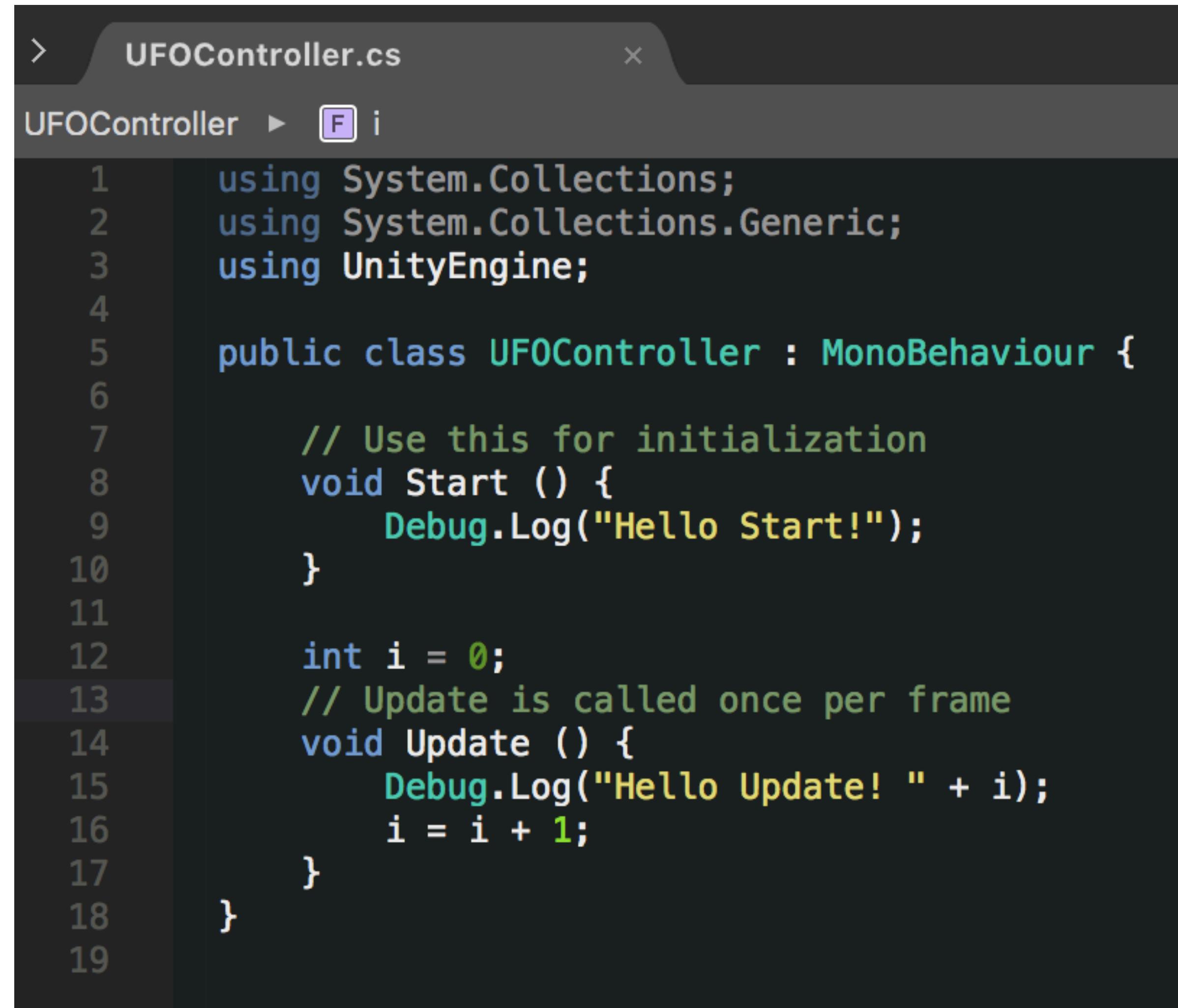


The screenshot shows the Unity Editor interface with the following details:

- Top Bar:** Includes icons for switching between scenes, a play button, and "Debug > Unity Editor".
- Status Bar:** Displays the message "解決方案已載入。" (Solution loaded).
- Solution Explorer:** Shows the project structure under "UFO Tutorial": Assets (Scenes, Sprites), and a selected file "UFOController.cs".
- Code Editor:** Displays the "UFOController.cs" script content:

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class UFOController : MonoBehaviour {
6
7     // Use this for initialization
8     void Start () {
9
10    }
11
12     // Update is called once per frame
13     void Update () {
14
15    }
16
17 }
```
- Right Sidebar:** Contains tabs for "工具箱" (Toolbox), "屬性" (Properties), "文件大綱" (Hierarchy), "單元測試" (Unit Test), and "工作" (Work) (indicated by a checkmark).
- Bottom Status:** Shows "錯誤" (Error) and "工作" (Working).

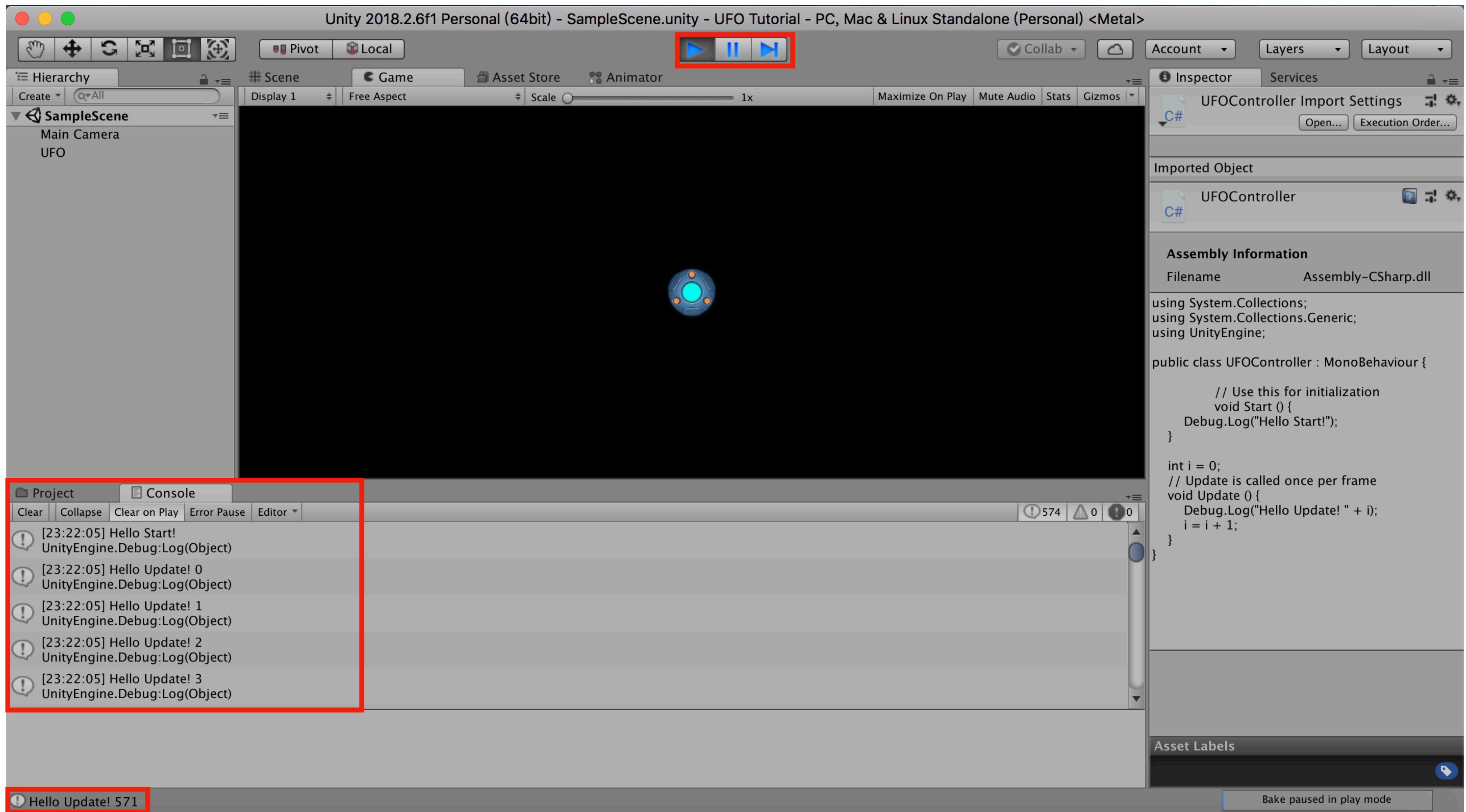
Hello~~



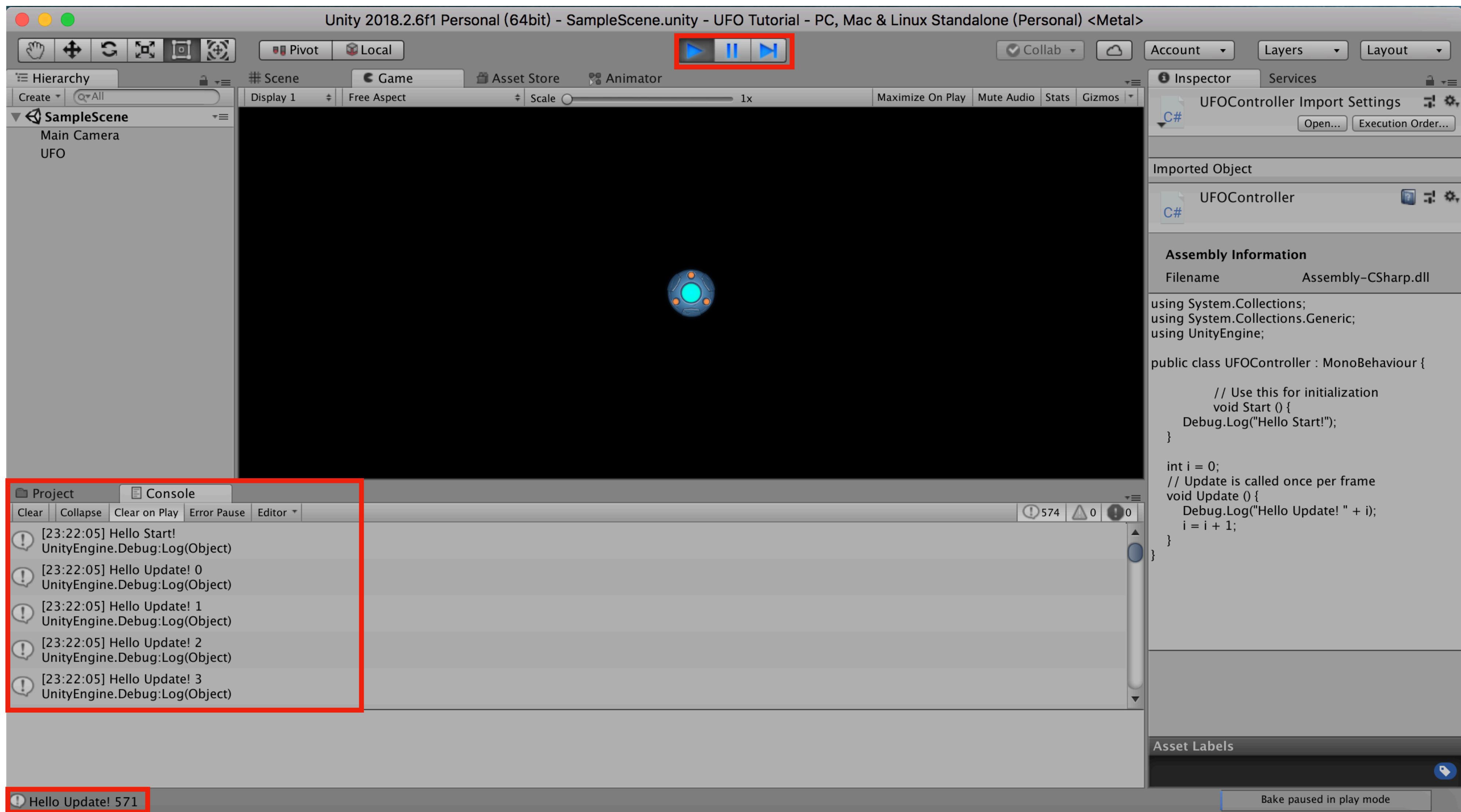
```
>  UFOController.cs  x
UFOController ▶  F i
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class UFOController : MonoBehaviour {
6
7      // Use this for initialization
8      void Start () {
9          Debug.Log("Hello Start!");
10     }
11
12     int i = 0;
13     // Update is called once per frame
14     void Update () {
15         Debug.Log("Hello Update! " + i);
16         i = i + 1;
17     }
18 }
19
```

<https://bit.ly/3ddKF9W>

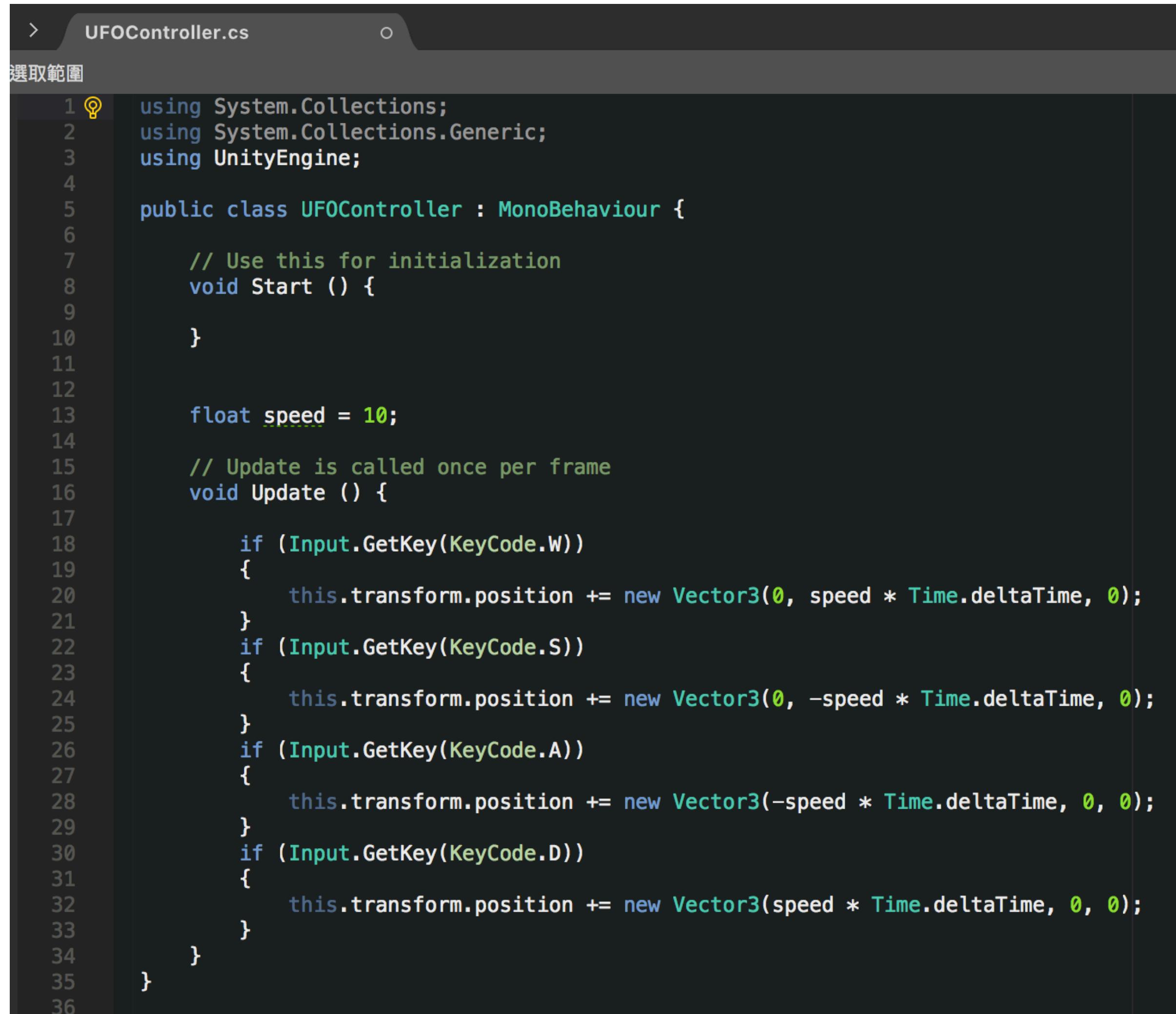
Hello~~



Move UFO



WSAD Move UFO

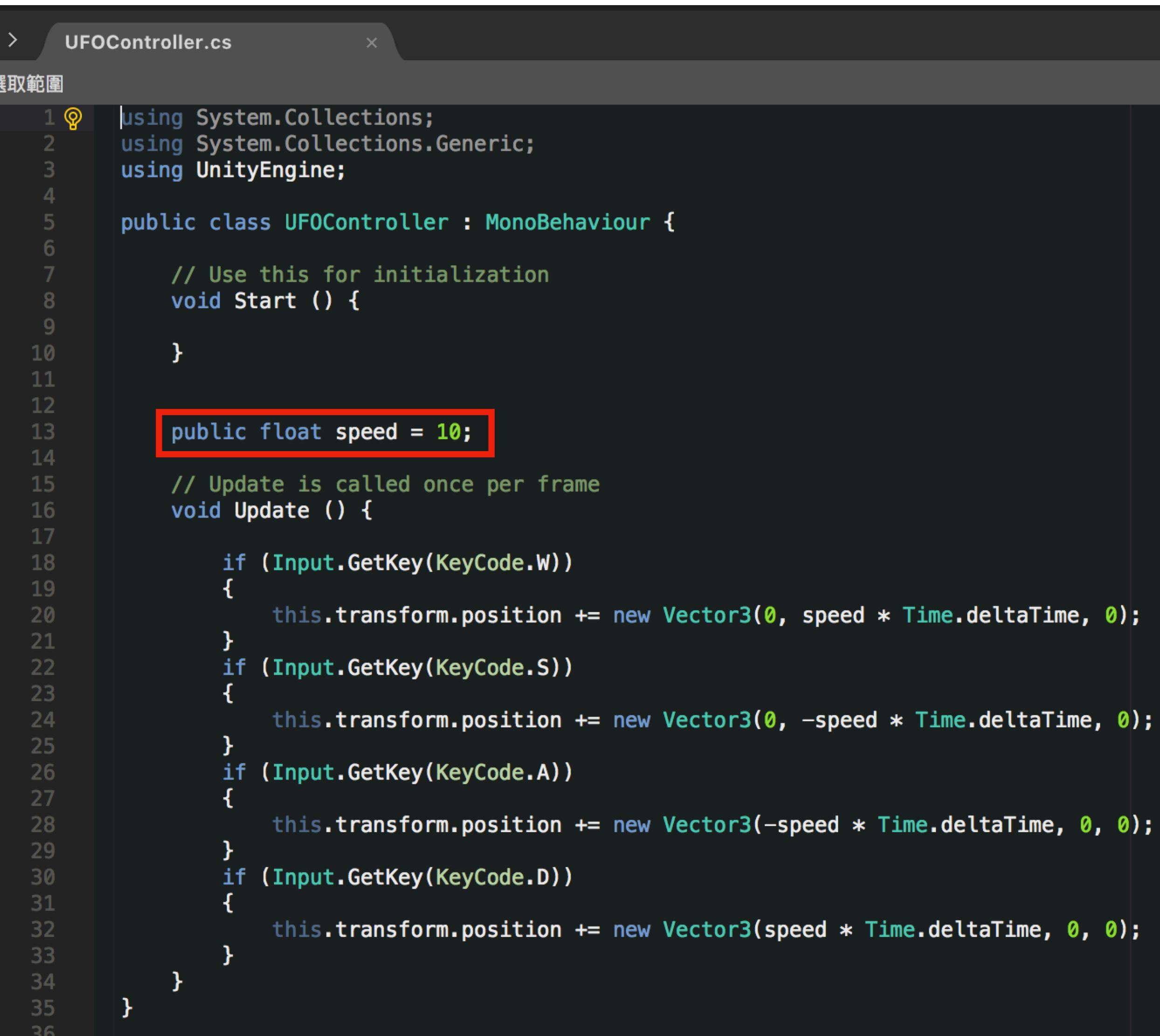


The image shows a screenshot of a code editor with a dark theme. The title bar says "UFOController.cs". The code is a C# script for a Unity MonoBehaviour. It defines a class "UFOController" with a constructor and two methods: Start() and Update(). The Update() method contains logic to move a transform based on WSAD key inputs. The code uses Input.GetKeyDown for key detection and Vector3 for movement.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class UFOController : MonoBehaviour {
6
7     // Use this for initialization
8     void Start () {
9
10    }
11
12
13    float speed = 10;
14
15    // Update is called once per frame
16    void Update () {
17
18        if (Input.GetKey(KeyCode.W))
19        {
20            this.transform.position += new Vector3(0, speed * Time.deltaTime, 0);
21        }
22        if (Input.GetKey(KeyCode.S))
23        {
24            this.transform.position += new Vector3(0, -speed * Time.deltaTime, 0);
25        }
26        if (Input.GetKey(KeyCode.A))
27        {
28            this.transform.position += new Vector3(-speed * Time.deltaTime, 0, 0);
29        }
30        if (Input.GetKey(KeyCode.D))
31        {
32            this.transform.position += new Vector3(speed * Time.deltaTime, 0, 0);
33        }
34    }
35}
36
```

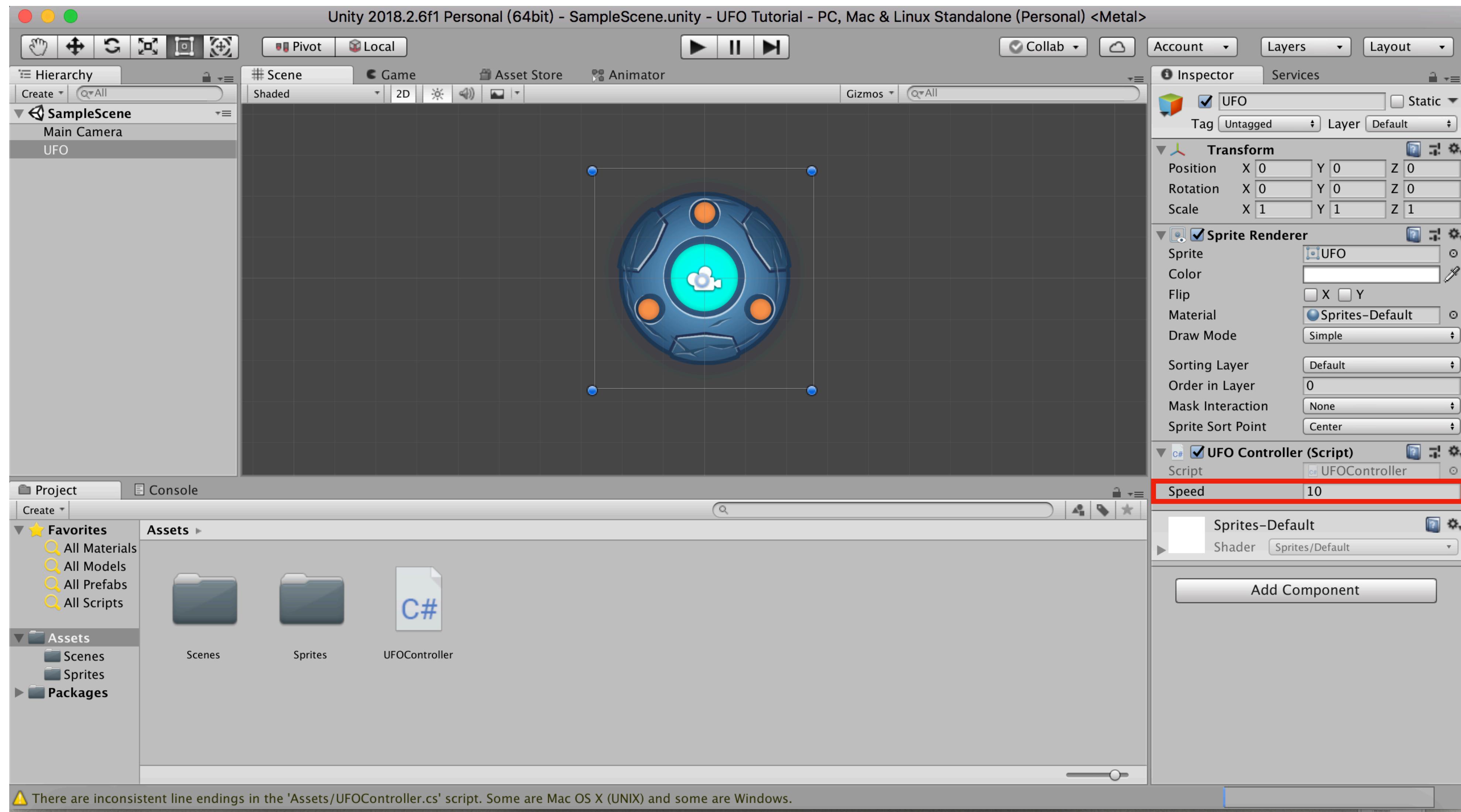
<https://bit.ly/39YZECB>

Public Variables

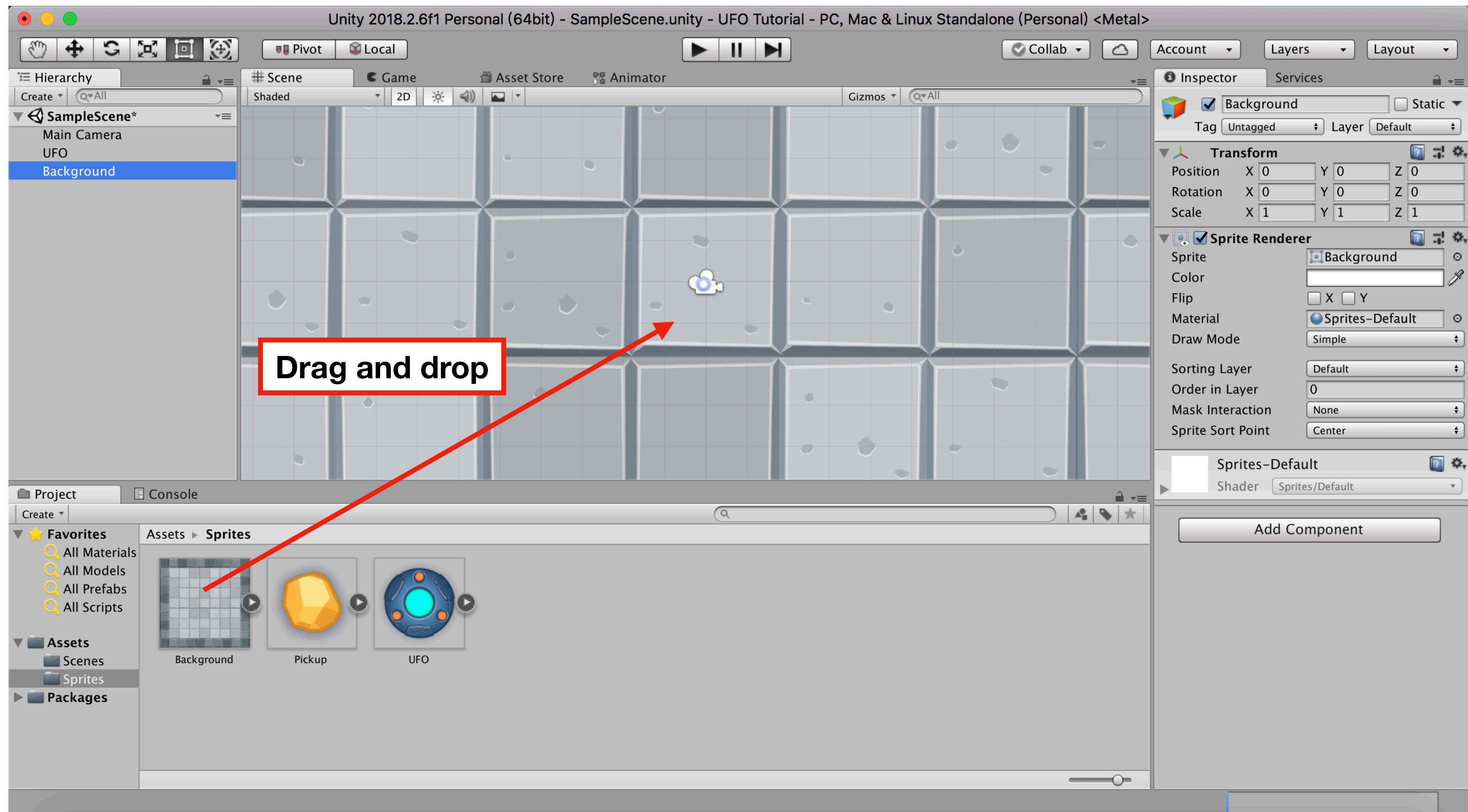


```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class UFOController : MonoBehaviour {
6
7     // Use this for initialization
8     void Start () {
9
10    }
11
12
13    public float speed = 10;
14
15    // Update is called once per frame
16    void Update () {
17
18        if (Input.GetKey(KeyCode.W))
19        {
20            this.transform.position += new Vector3(0, speed * Time.deltaTime, 0);
21        }
22        if (Input.GetKey(KeyCode.S))
23        {
24            this.transform.position += new Vector3(0, -speed * Time.deltaTime, 0);
25        }
26        if (Input.GetKey(KeyCode.A))
27        {
28            this.transform.position += new Vector3(-speed * Time.deltaTime, 0, 0);
29        }
30        if (Input.GetKey(KeyCode.D))
31        {
32            this.transform.position += new Vector3(speed * Time.deltaTime, 0, 0);
33        }
34    }
35}
36
```

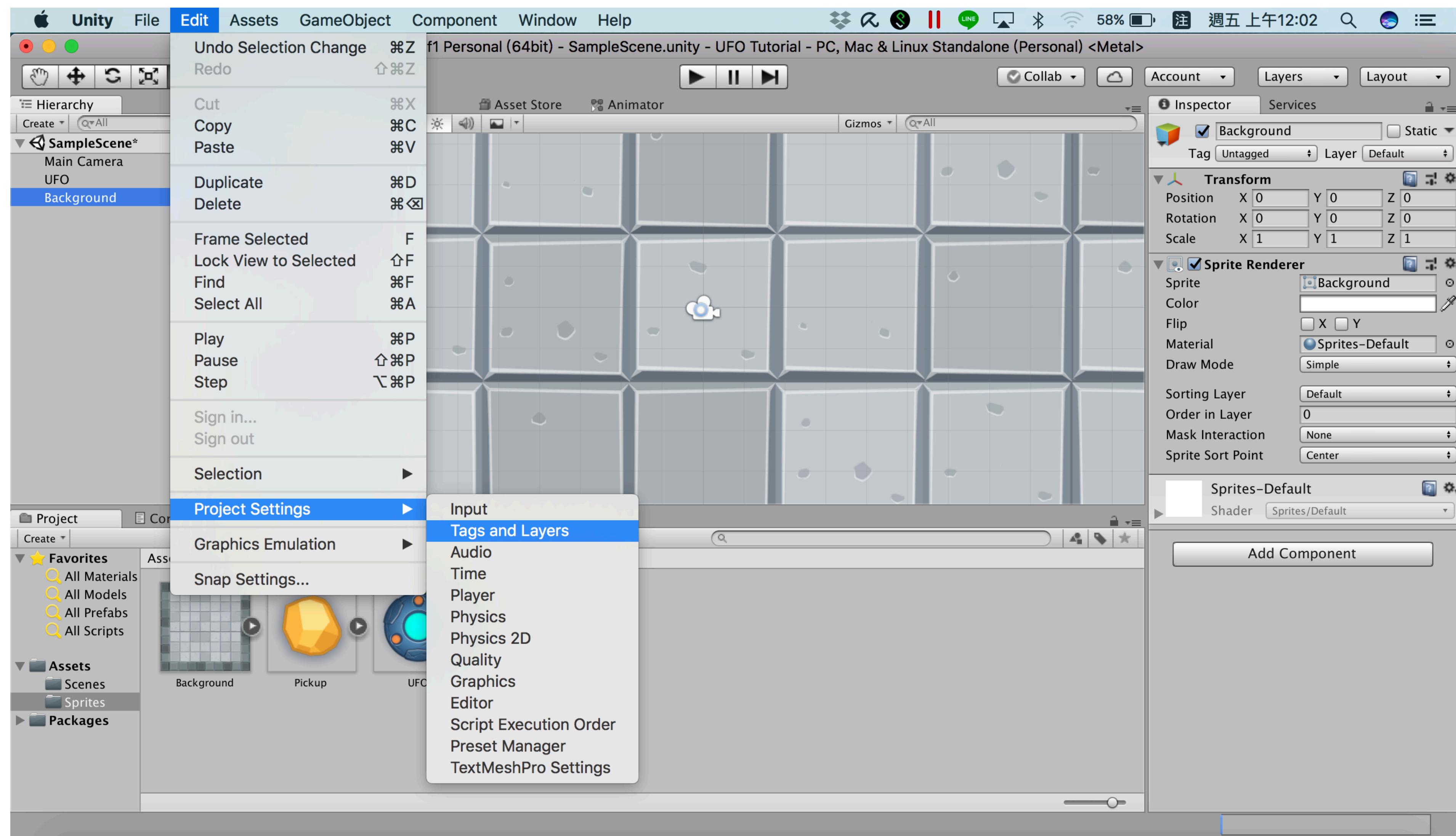
Public Variables



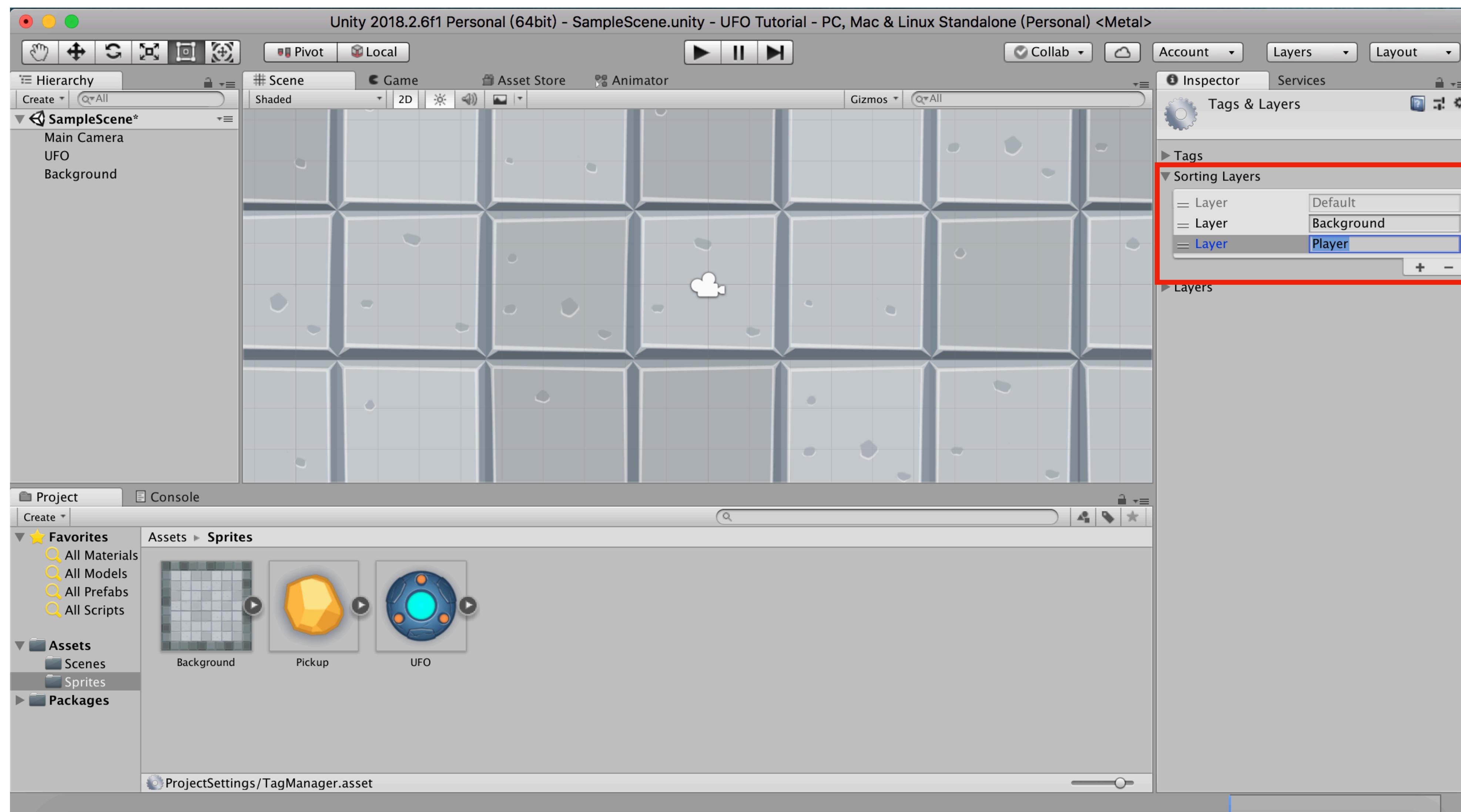
Background



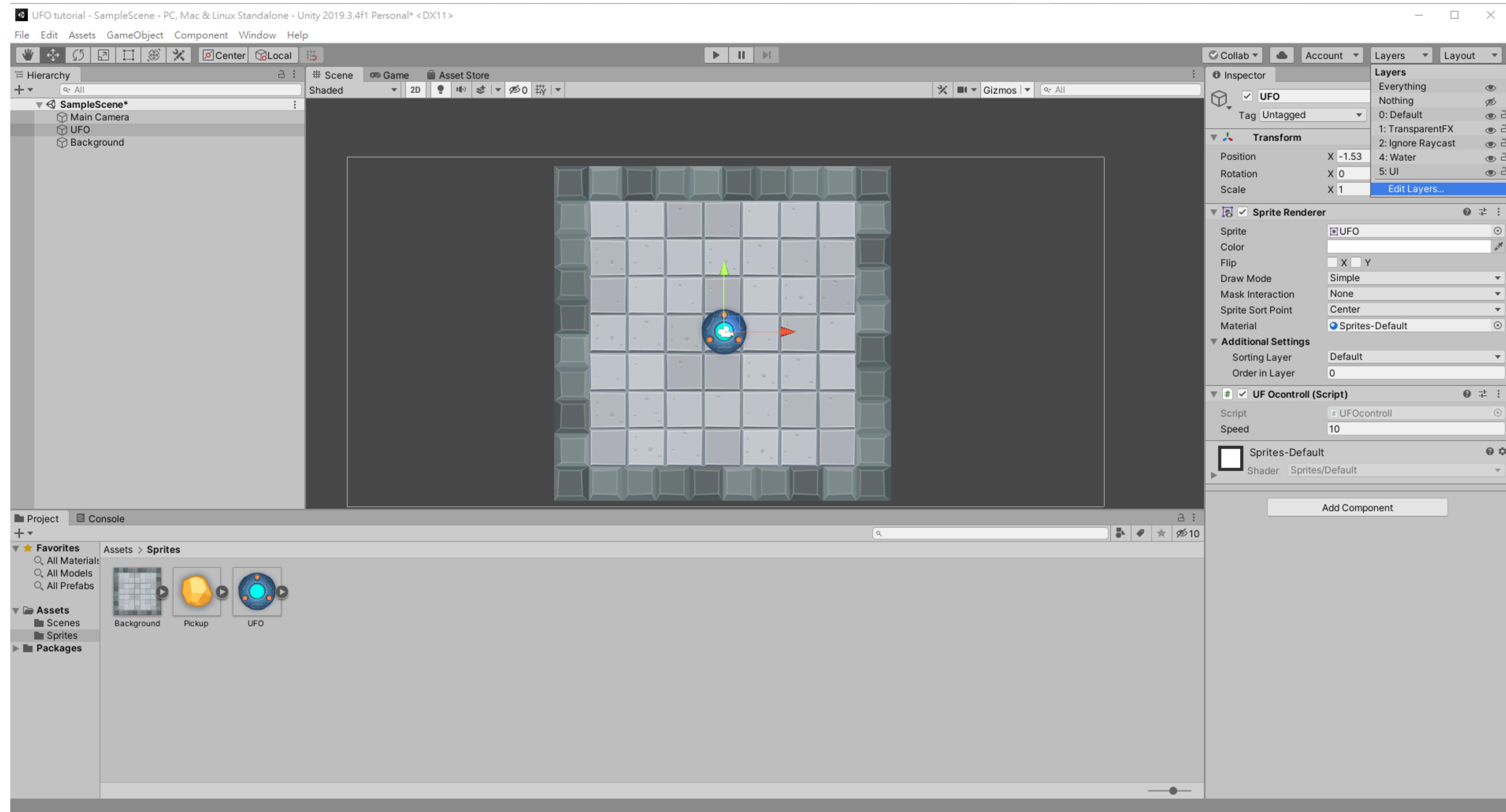
Tags and Layers



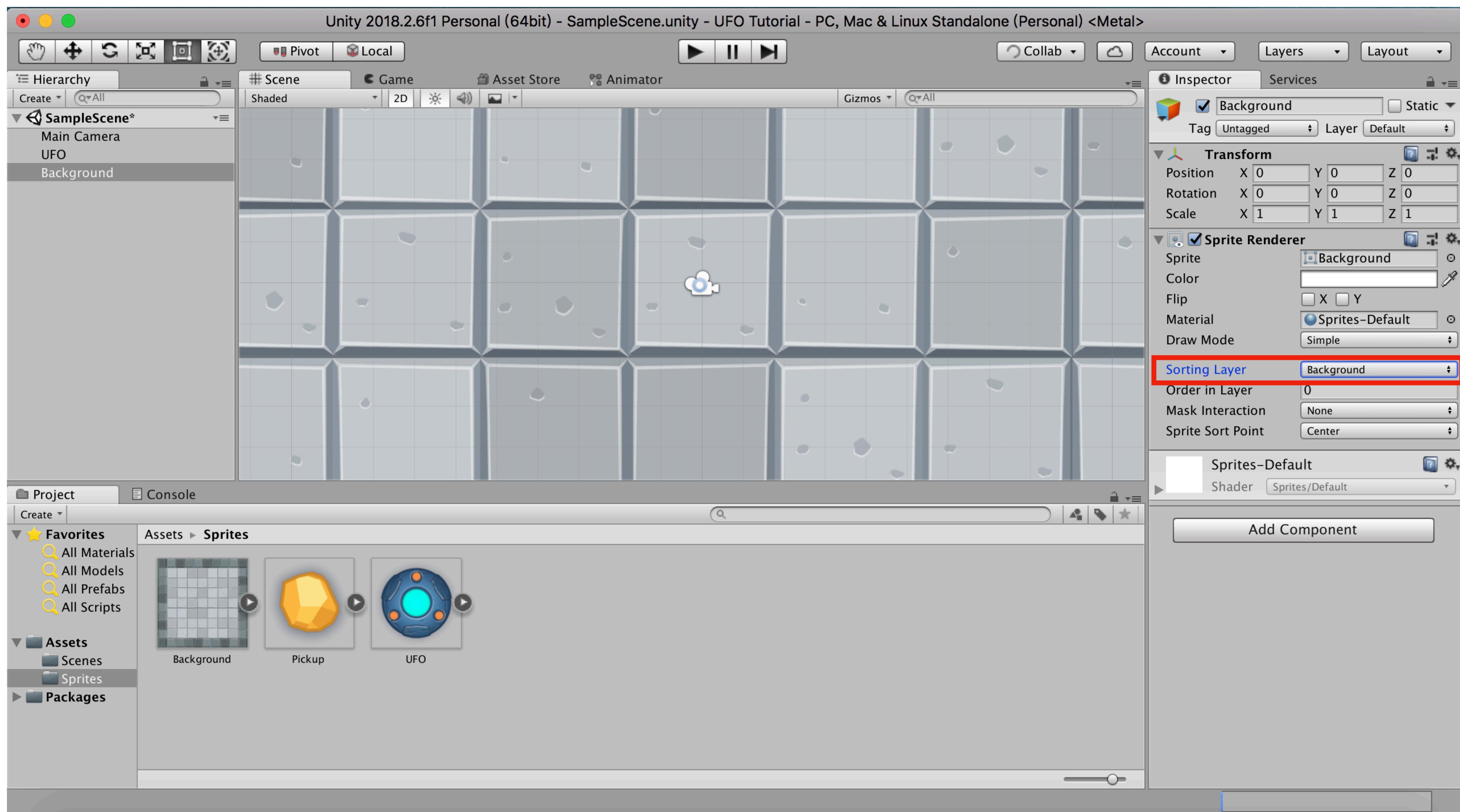
Tags and Layers



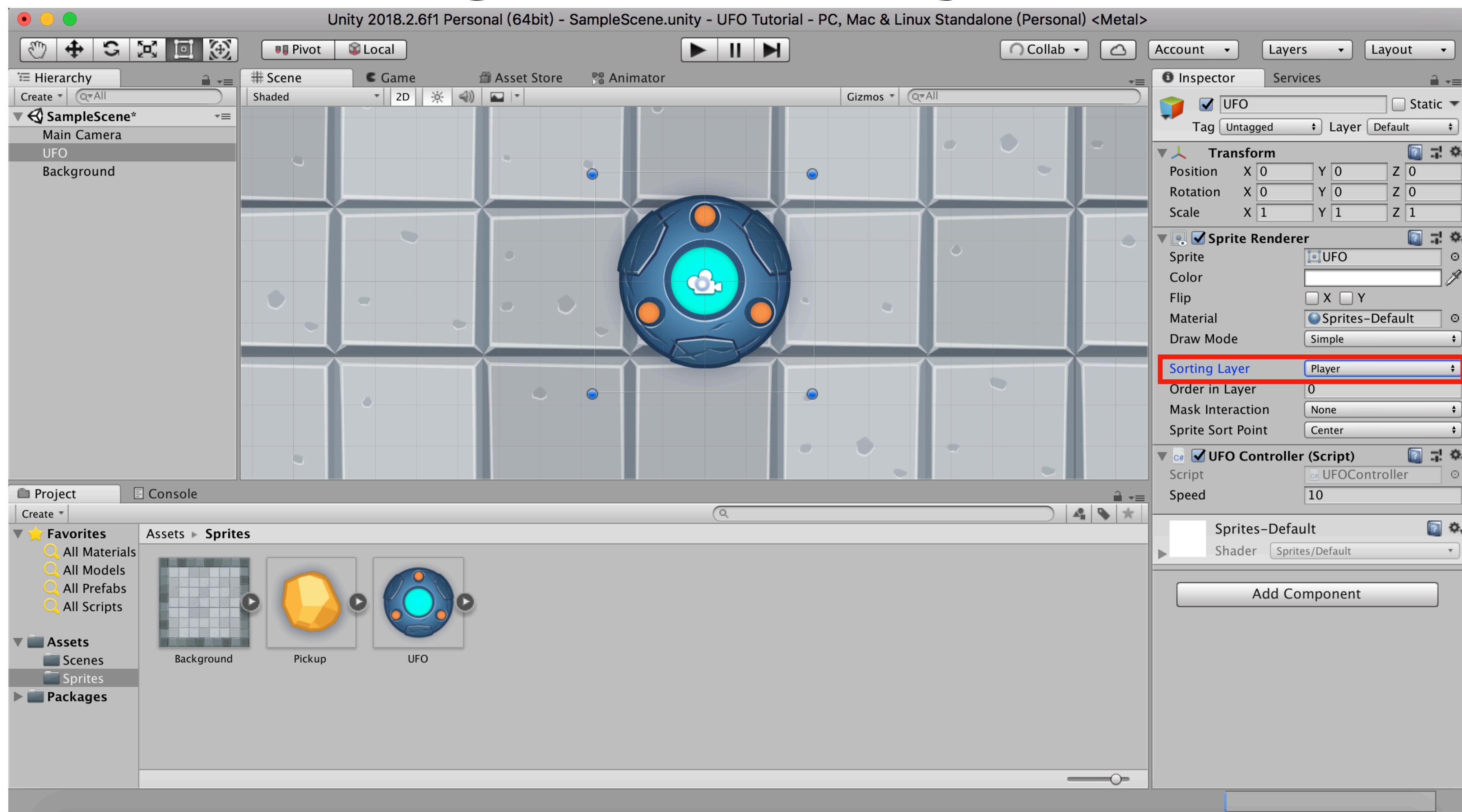
Tags and Layers



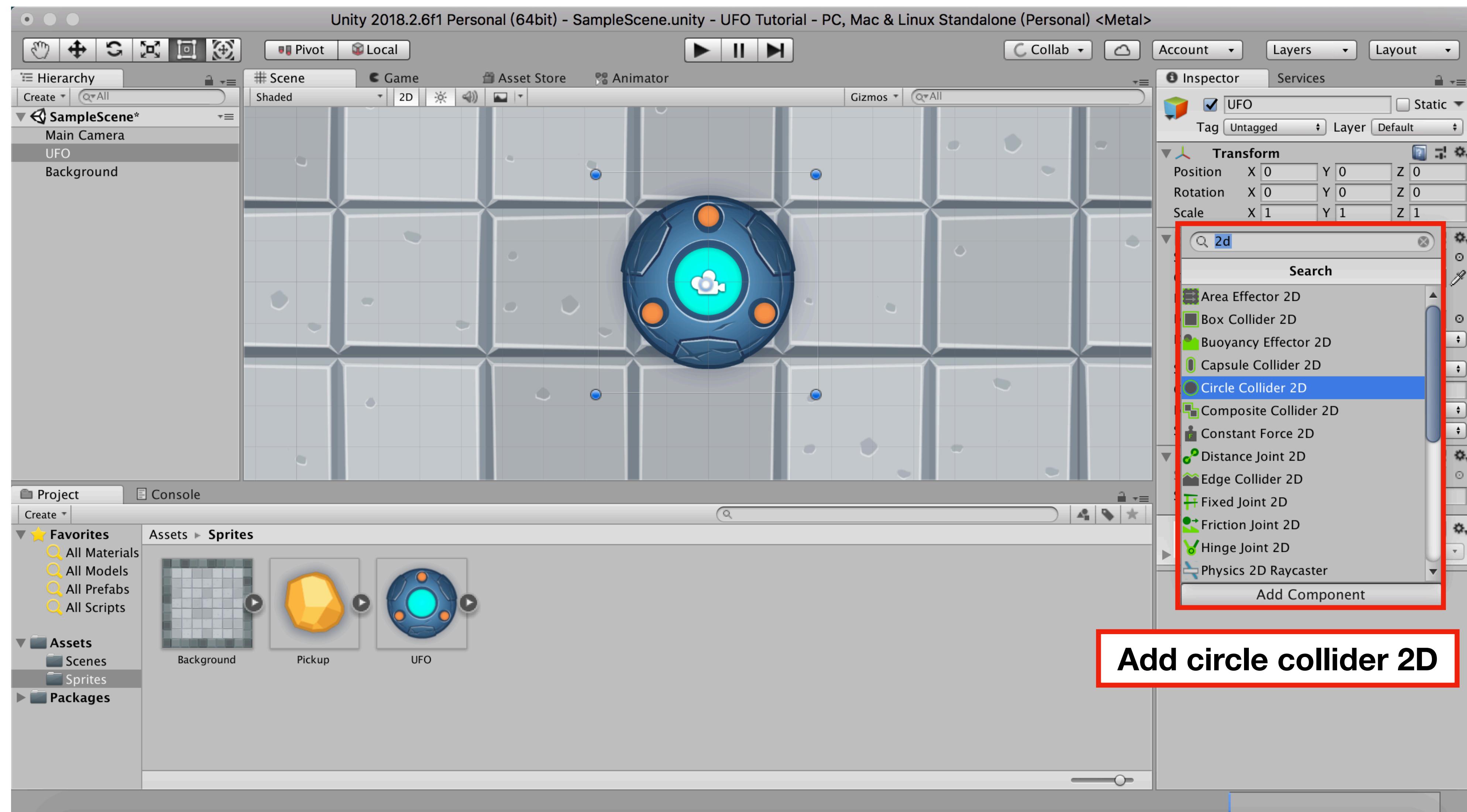
Tags and Layers



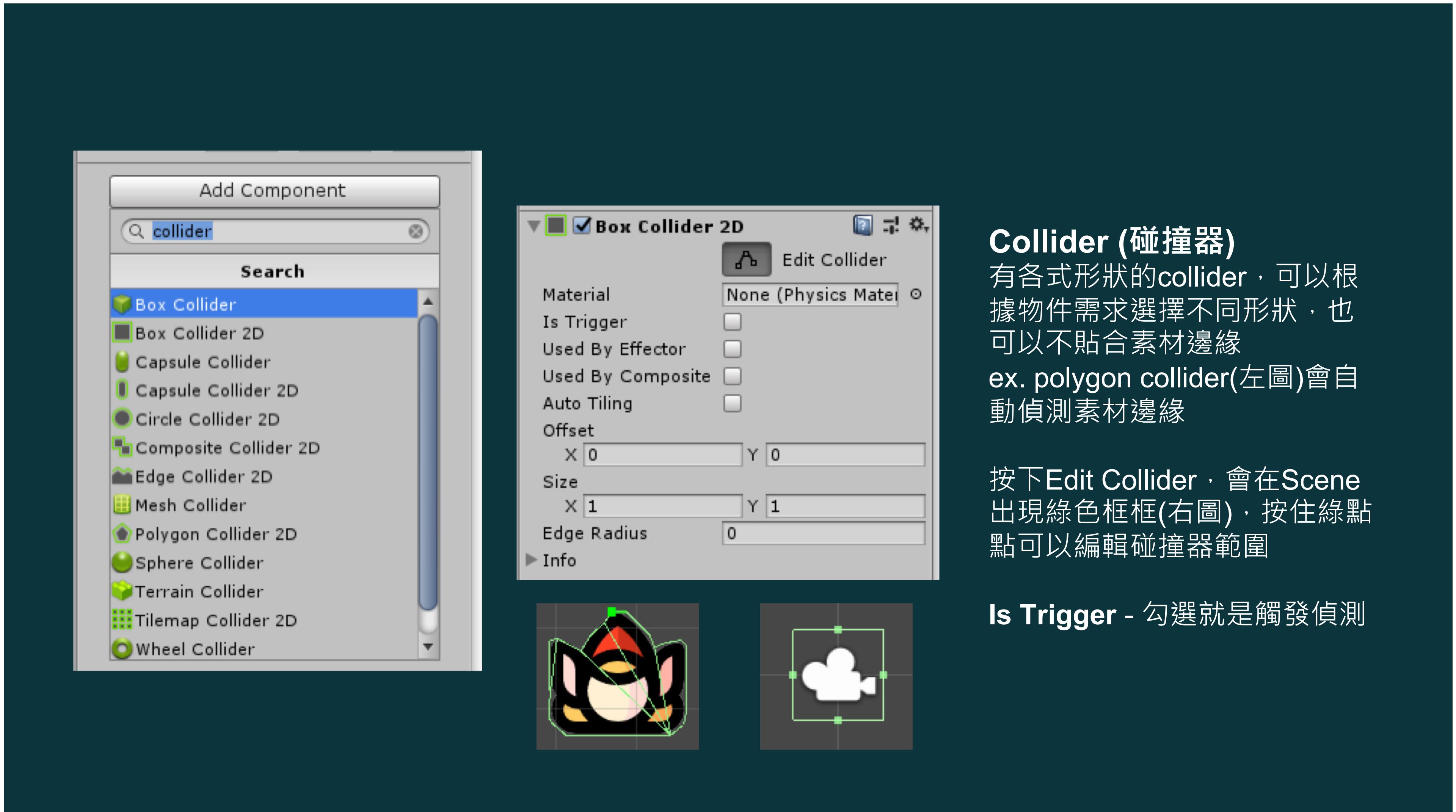
Tags and Layers



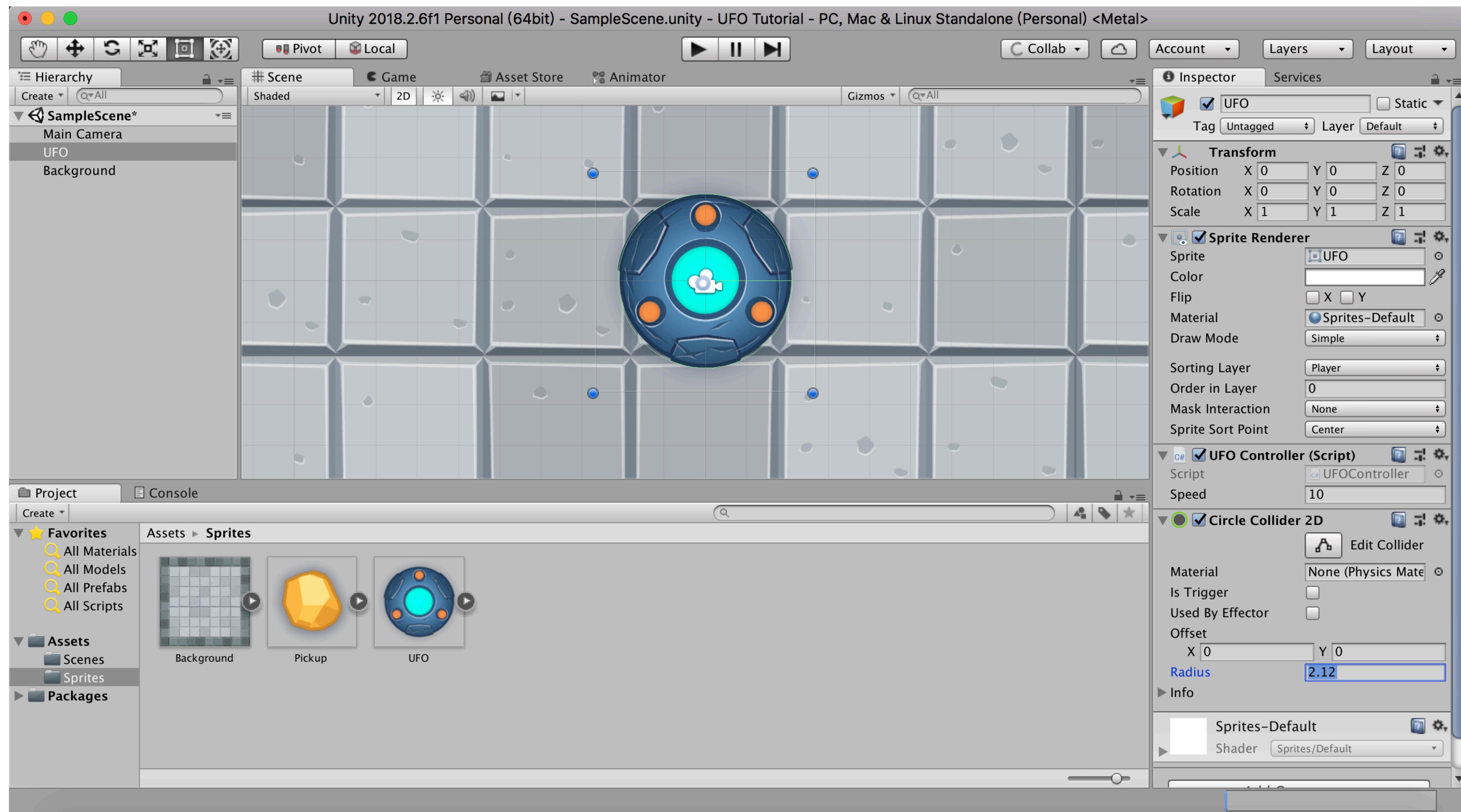
Collider



Collider



Collider



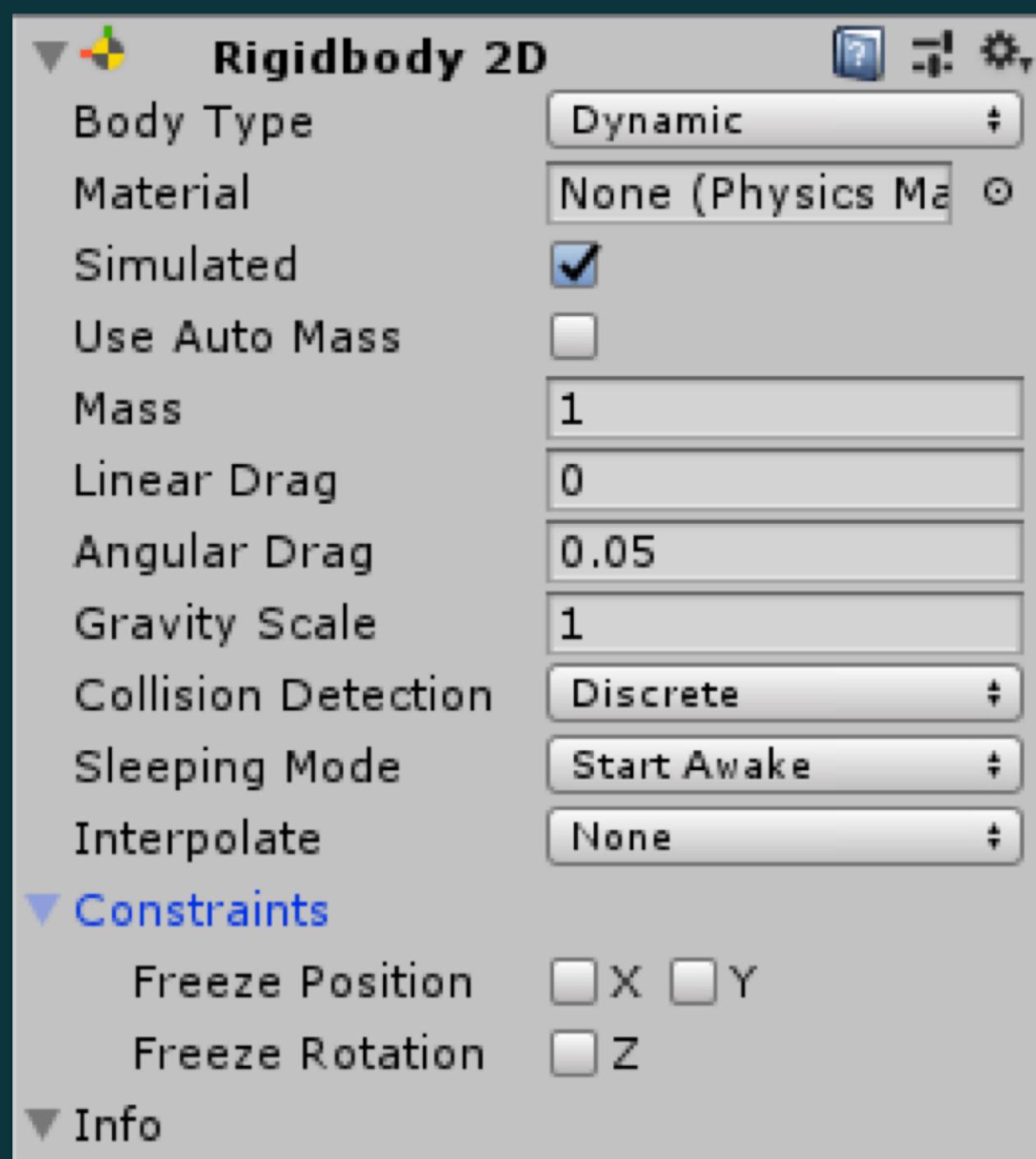
Rigidbody



Rigidbody

Rigidbody (剛體)

透過加上質量、摩擦力、重力等物理數值讓虛擬物件像現實物體般運動



Body Type :

Dynamic - 可以被遊戲中的任何物件移動

Kinematic - 只有玩家和程式可以控制它的移動，其他遊戲物件都不行

Mass - 物件的虛擬質量

Linear Drag - 物體位移時的空氣阻力。物體受力後位移的容易程度，越大越不容易被移動

Angular Drag - 物體旋轉時的角阻力。受力後旋轉的容易程度，越大越不容易被旋轉

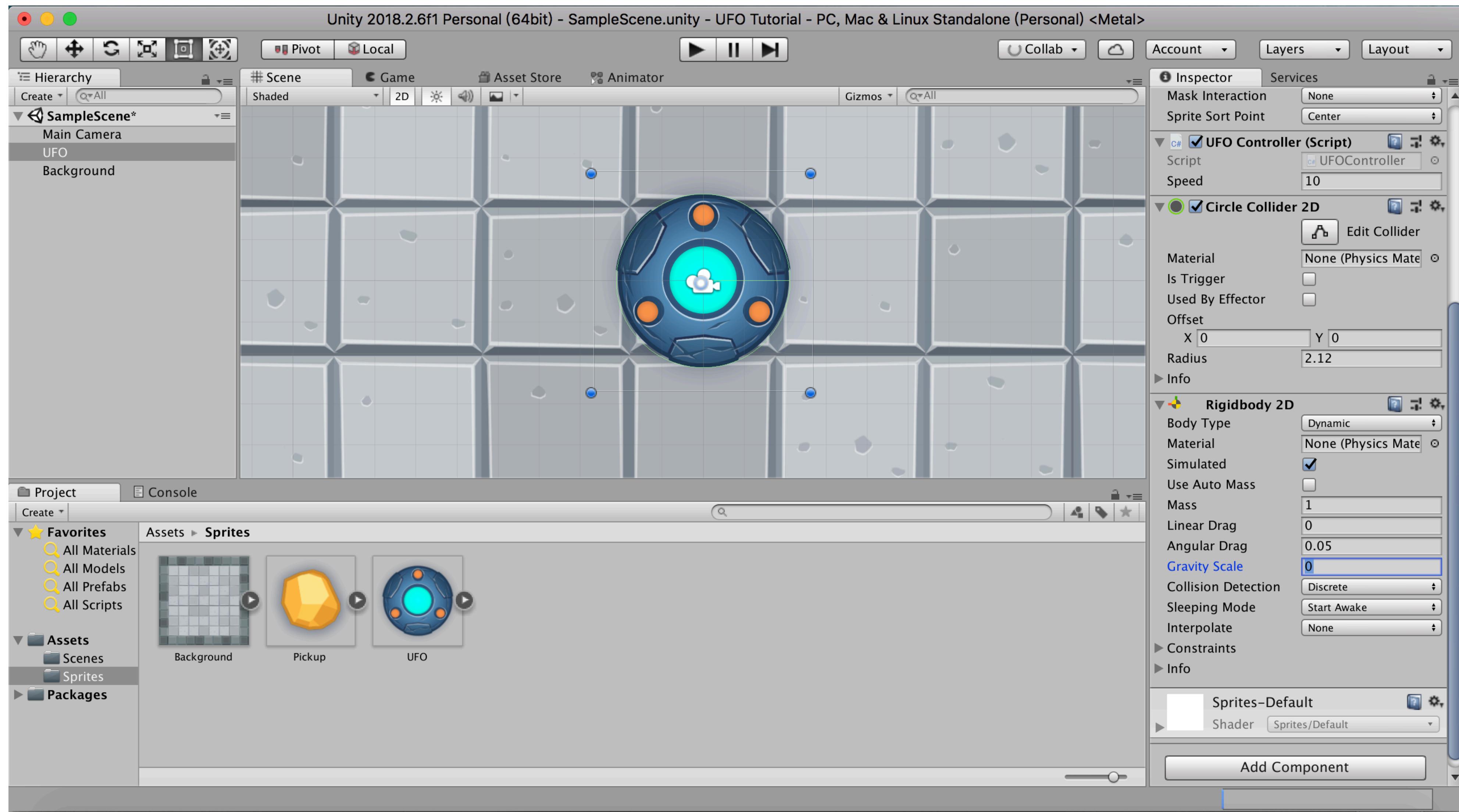
Gravity Scale - 重力=1時，遊戲執行後物件就會往下掉落；=0時，沒有受力就停在原地；=-1會往上飄

Constrains :

Freeze Position - 鎖住該軸平移

Freeze Rotation - 鎖住旋轉方向，2D只有Z軸(突出螢幕)可以旋轉

Gravity = 0

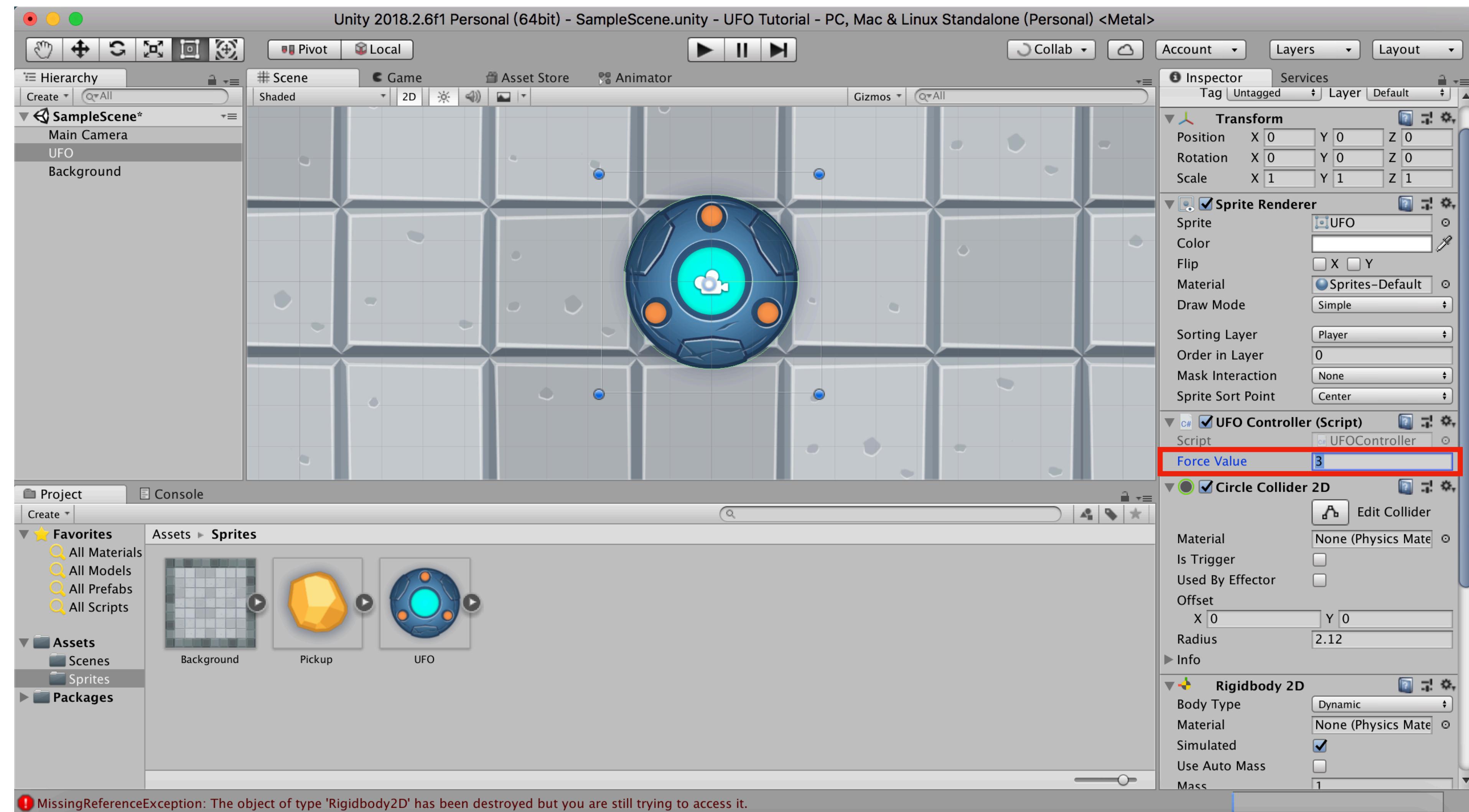


Move in Force

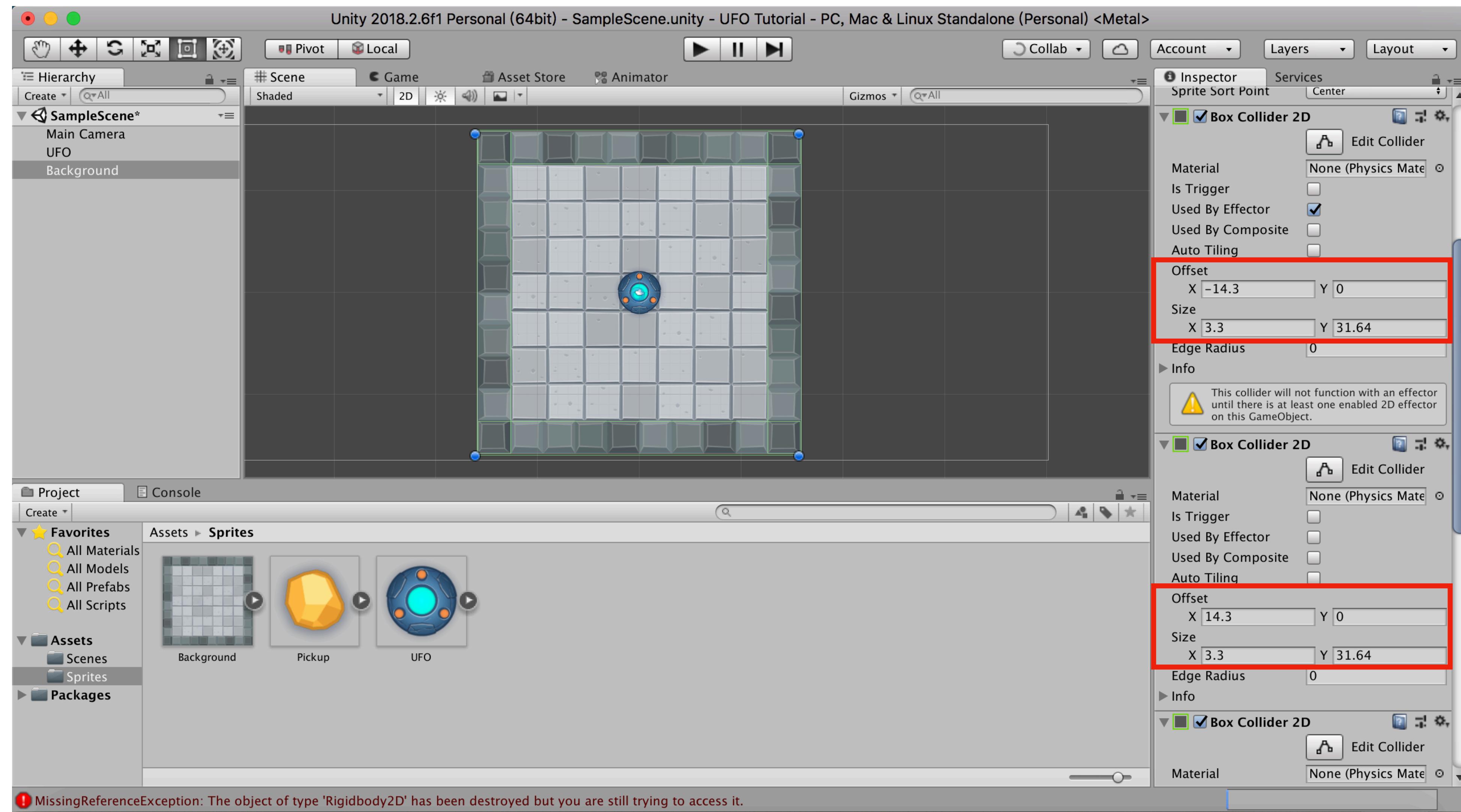
```
1 ⑧  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class UFOController : MonoBehaviour {
6
7      // Use this for initialization
8      void Start () {
9          rigidbody2D = this.GetComponent<Rigidbody2D>();
10     }
11
12
13     public float forceValue;
14     private Rigidbody2D rigidbody2D = null;
15
16     // Update is called once per frame
17     void Update () {
18
19         Vector2 force2D = Vector2.zero;
20
21         if (Input.GetKey(KeyCode.W))
22         {
23             force2D.y += forceValue;
24         }
25         if (Input.GetKey(KeyCode.S))
26         {
27             force2D.y -= forceValue;
28         }
29         if (Input.GetKey(KeyCode.A))
30         {
31             force2D.x -= forceValue;
32         }
33         if (Input.GetKey(KeyCode.D))
34         {
35             force2D.x += forceValue;
36         }
37
38         rigidbody2D.AddForce(force2D);
39     }
40 }
41 }
```

<https://bit.ly/2wkRXIk>

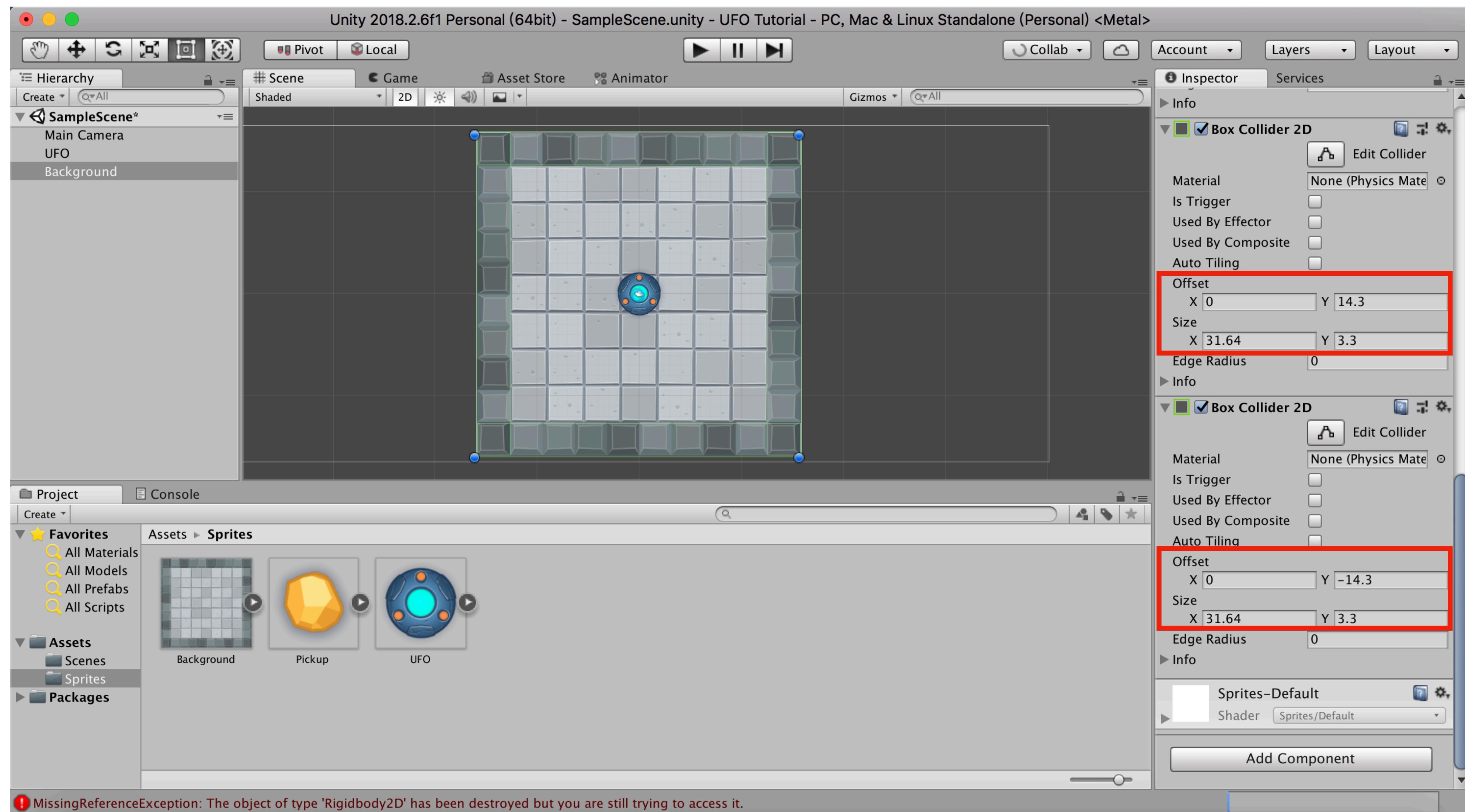
Force value = 3



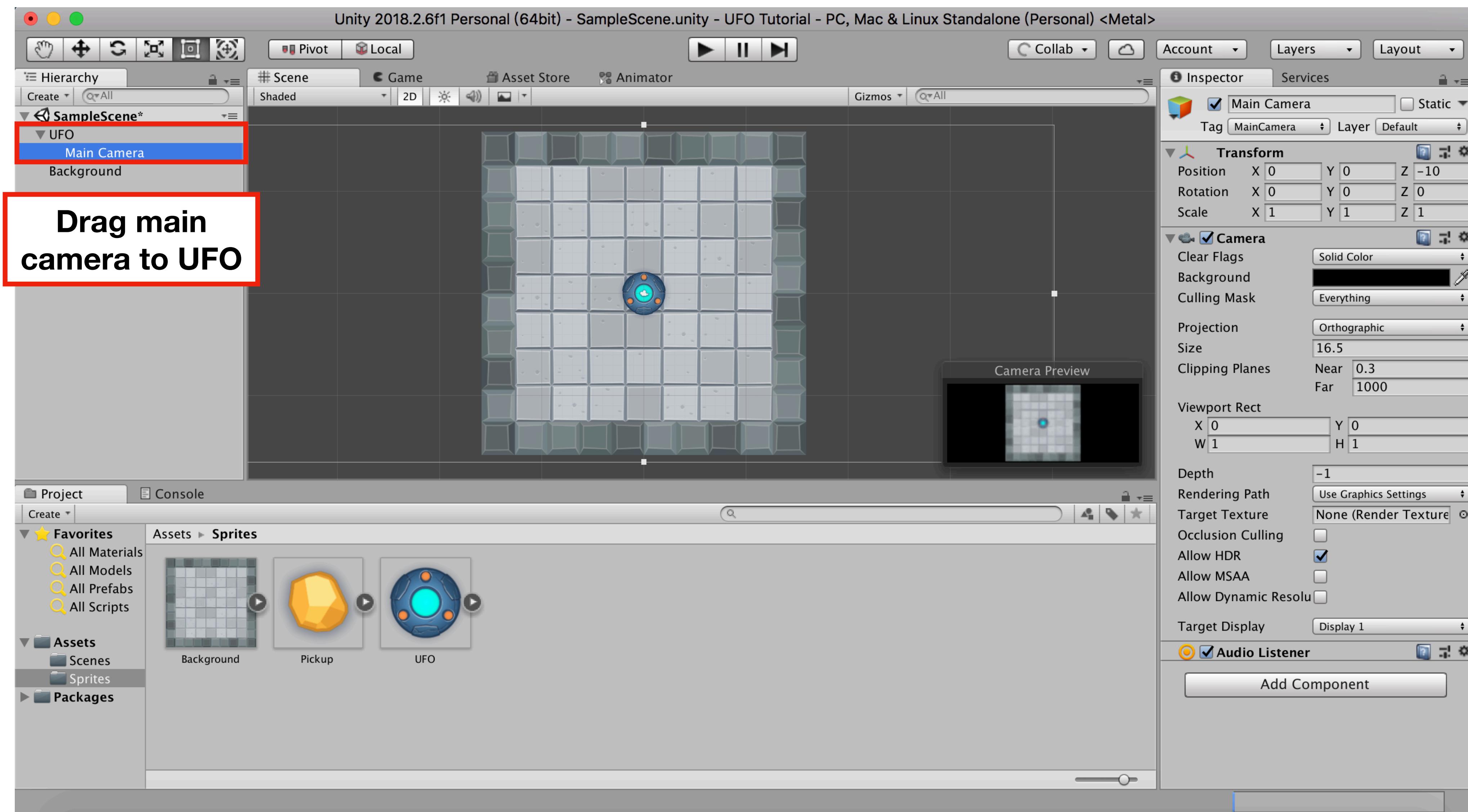
Box Colliders



Box Colliders



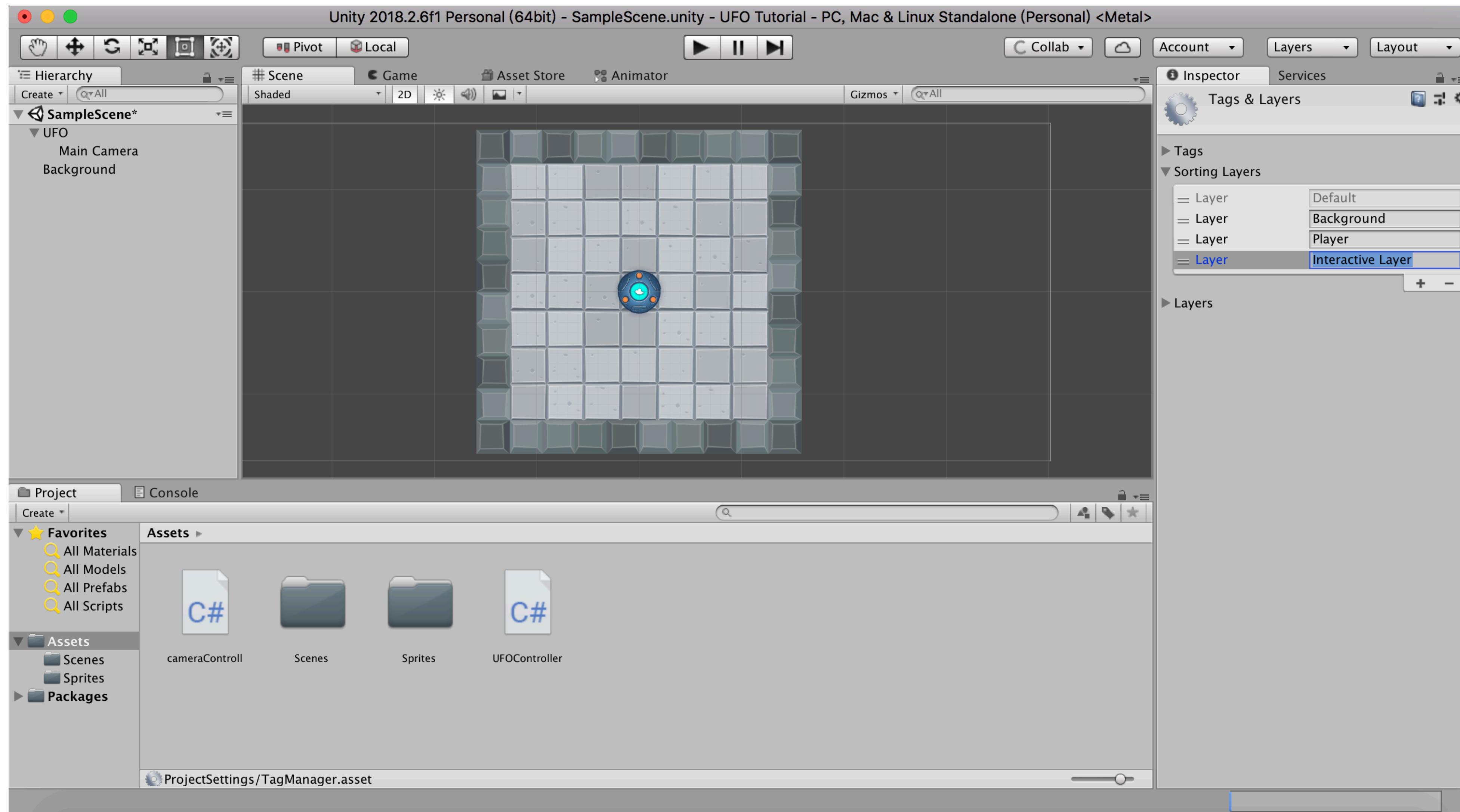
Camera on UFO



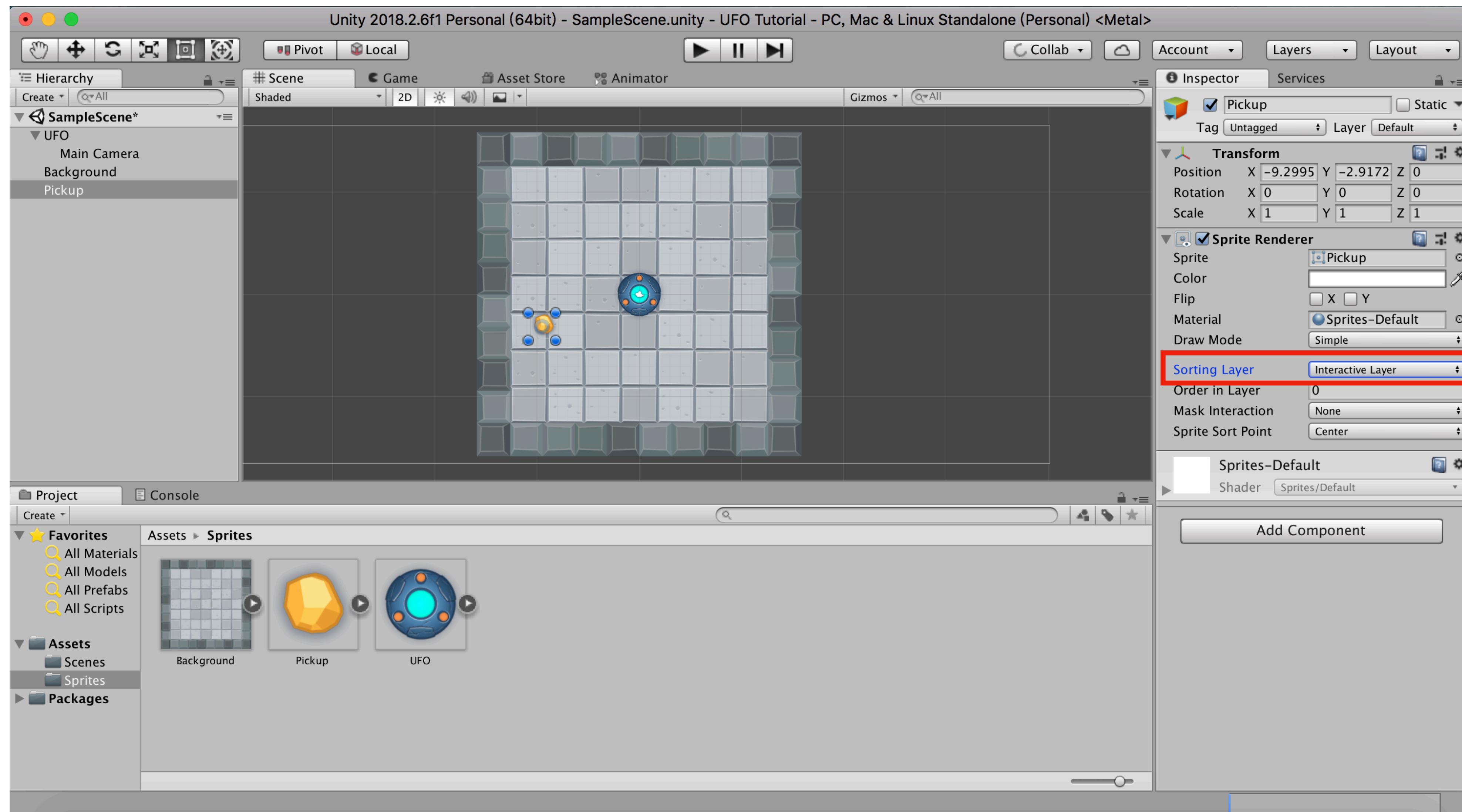
cameraControll

```
1 ①  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class cameraControll : MonoBehaviour {
6
7      // Use this for initialization
8      void Start () {
9
10     }
11
12      // Update is called once per frame
13      void Update () {
14          this.transform.eulerAngles = Vector3.zero;
15      }
16
17 }
```

Add Coin in the Scene



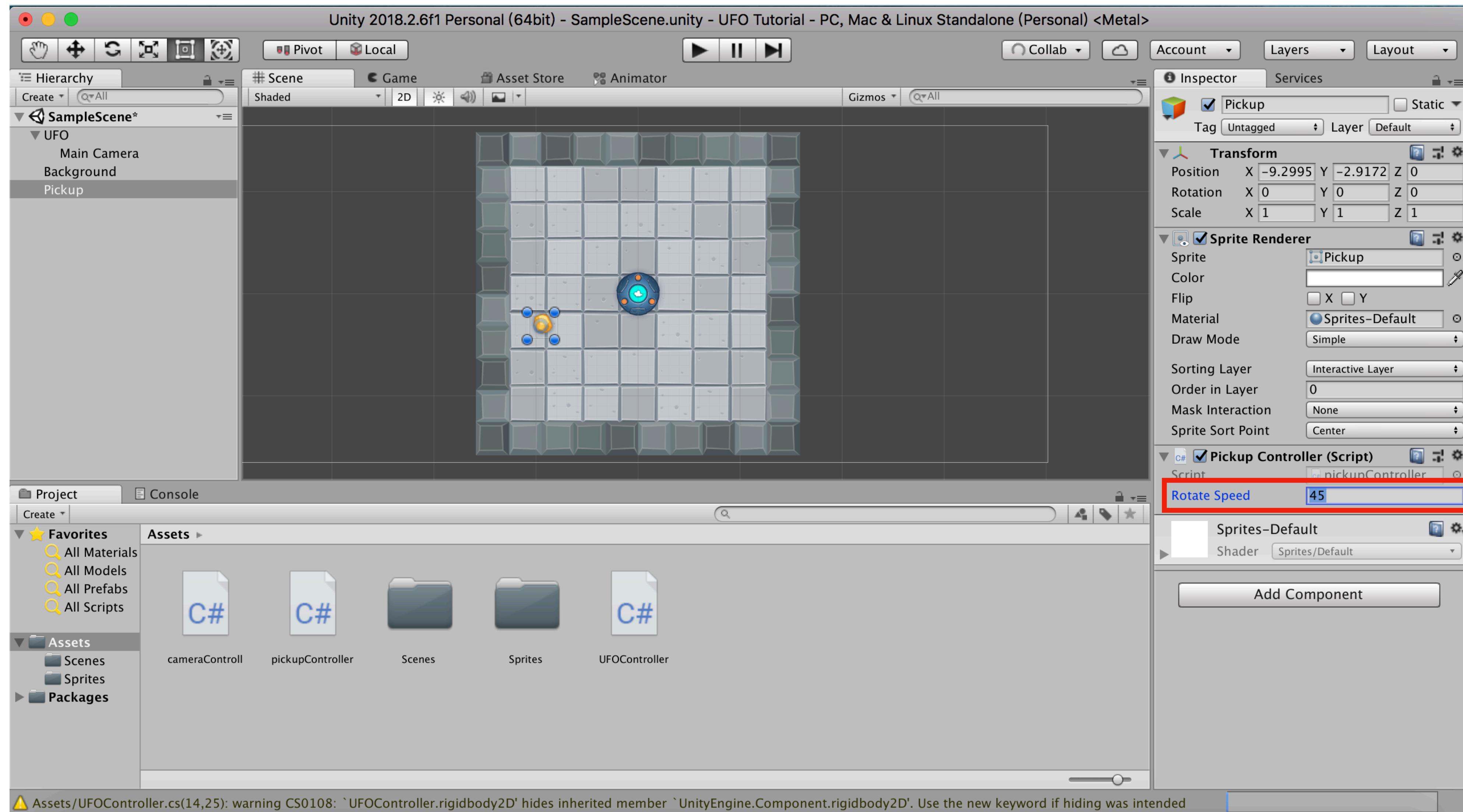
Add Coin in the Scene



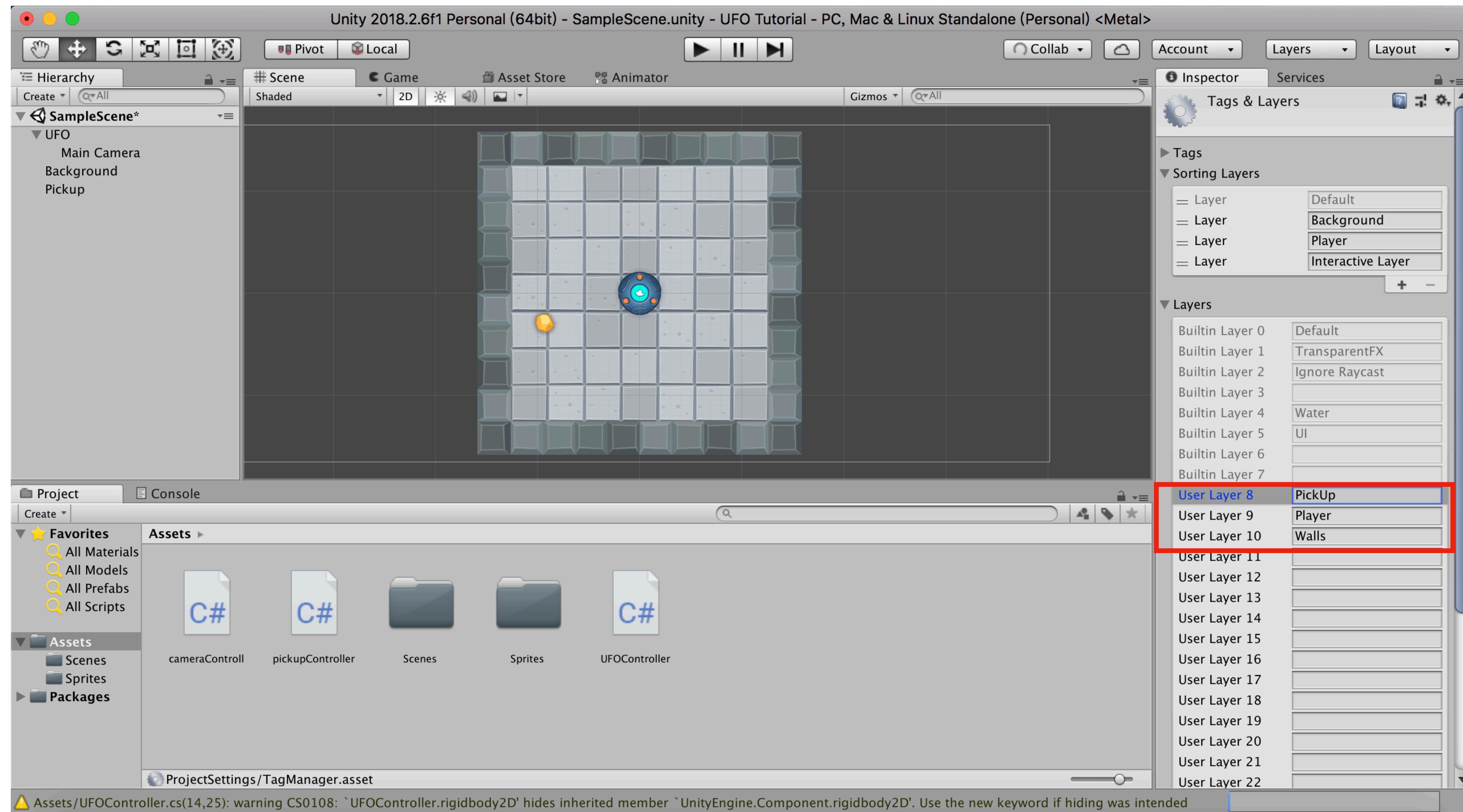
pickupController

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class pickupController : MonoBehaviour {
6
7      public float RotateSpeed;
8
9      // Use this for initialization
10     void Start () {
11
12     }
13
14     // Update is called once per frame
15     void Update () {
16         this.transform.Rotate(new Vector3(0, 0, RotateSpeed * Time.deltaTime));
17     }
18 }
19 }
```

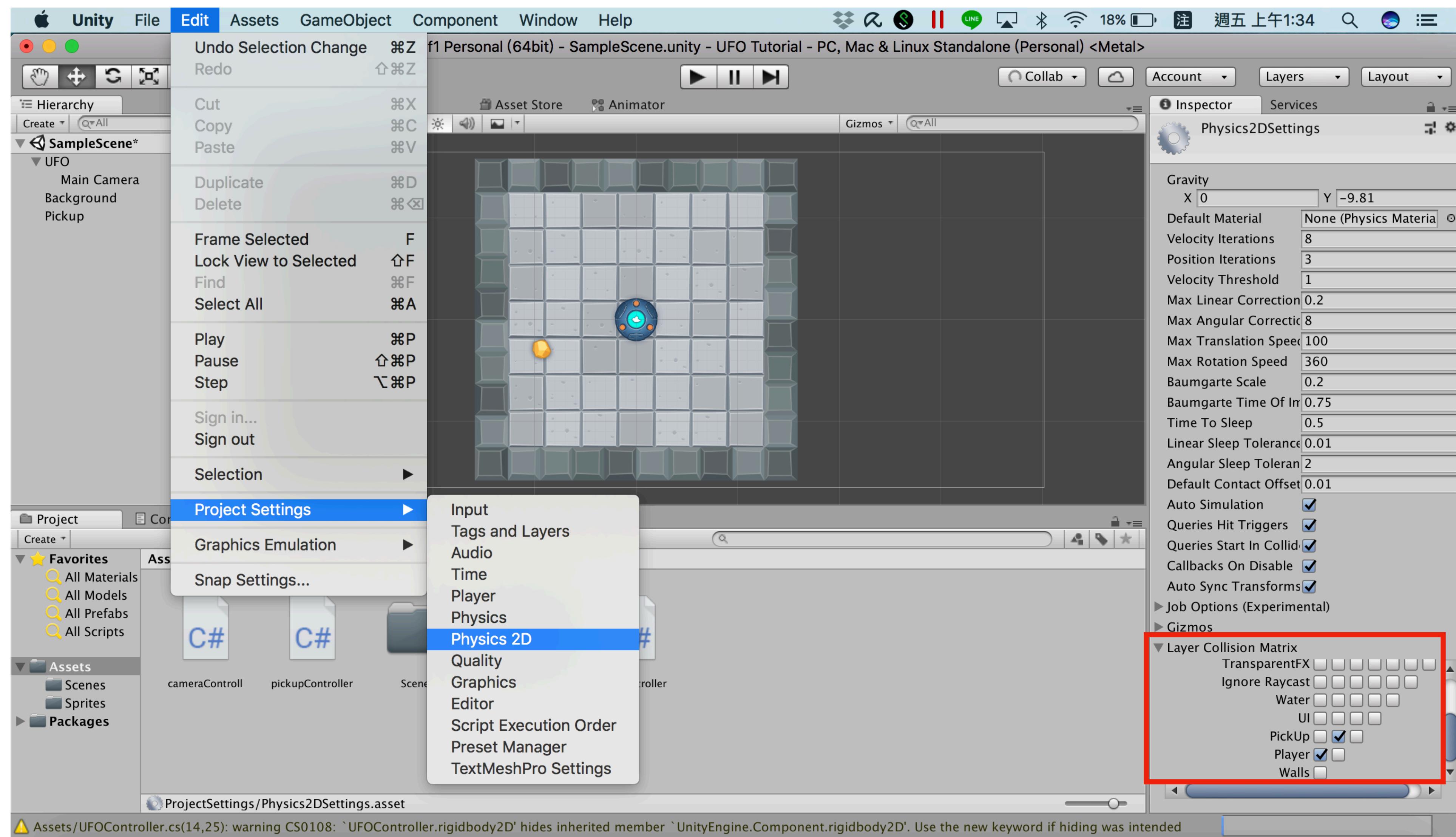
RotateSpeed = 45



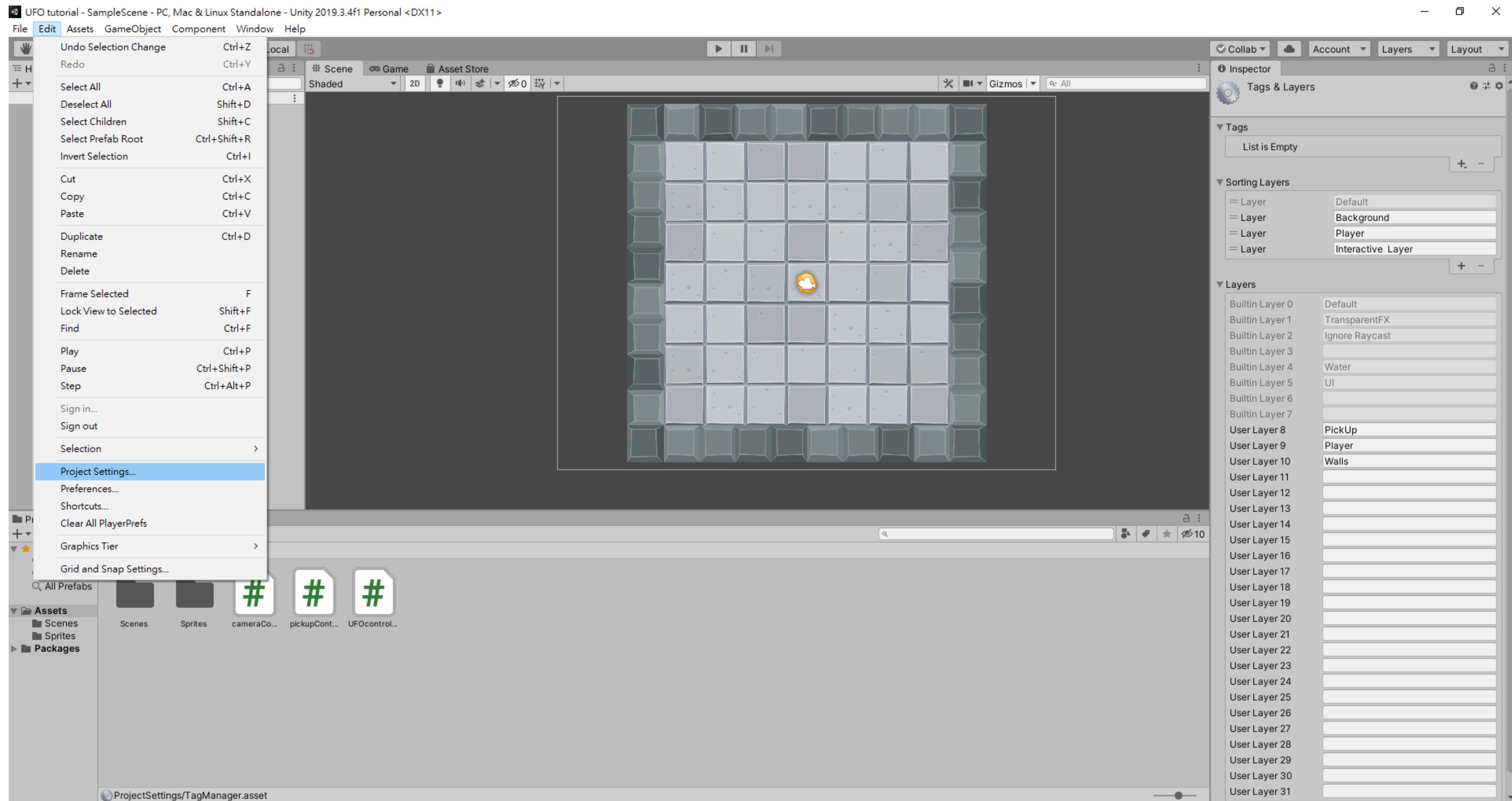
Layers



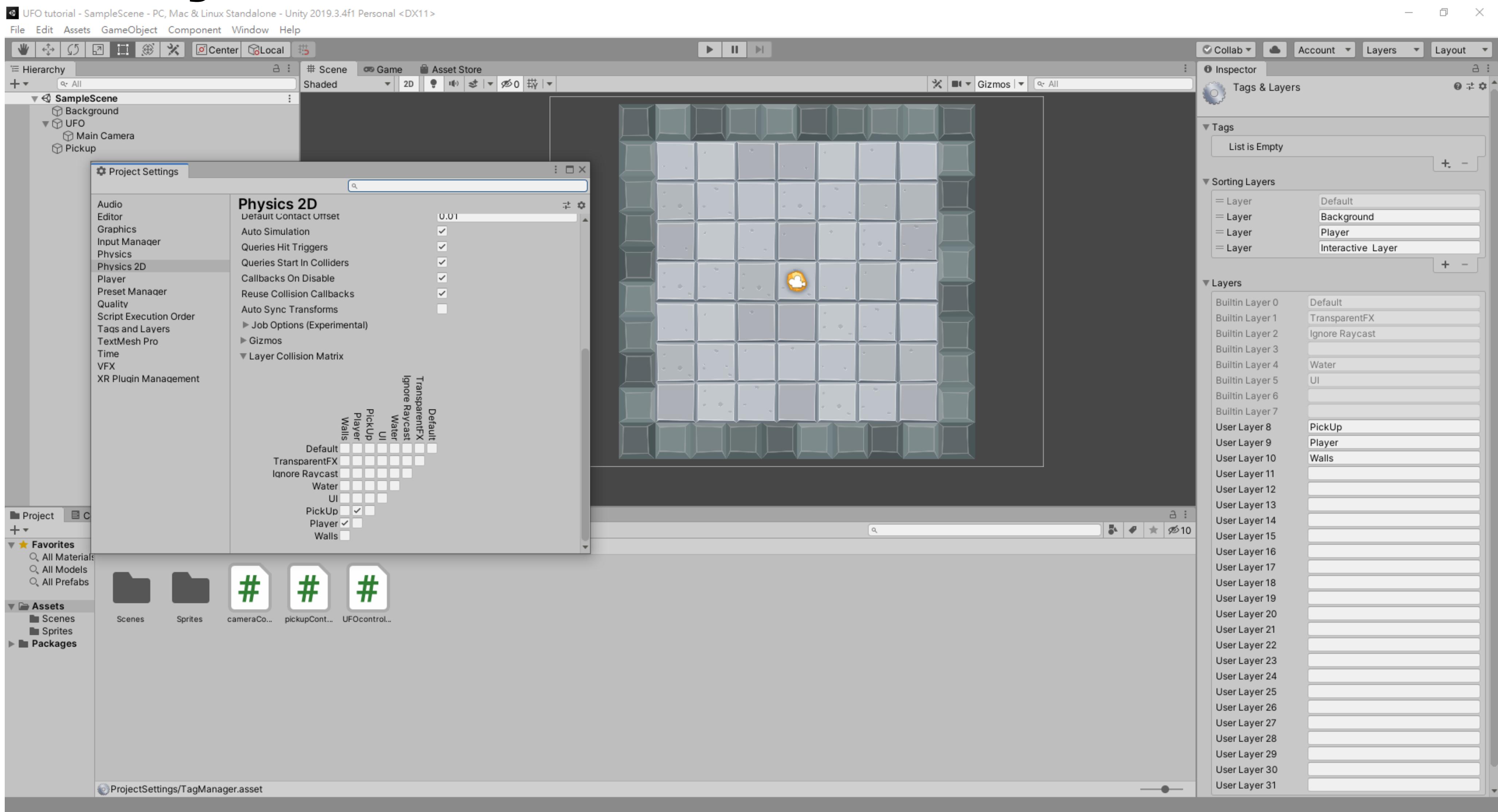
Physics 2D and Collision Matrix



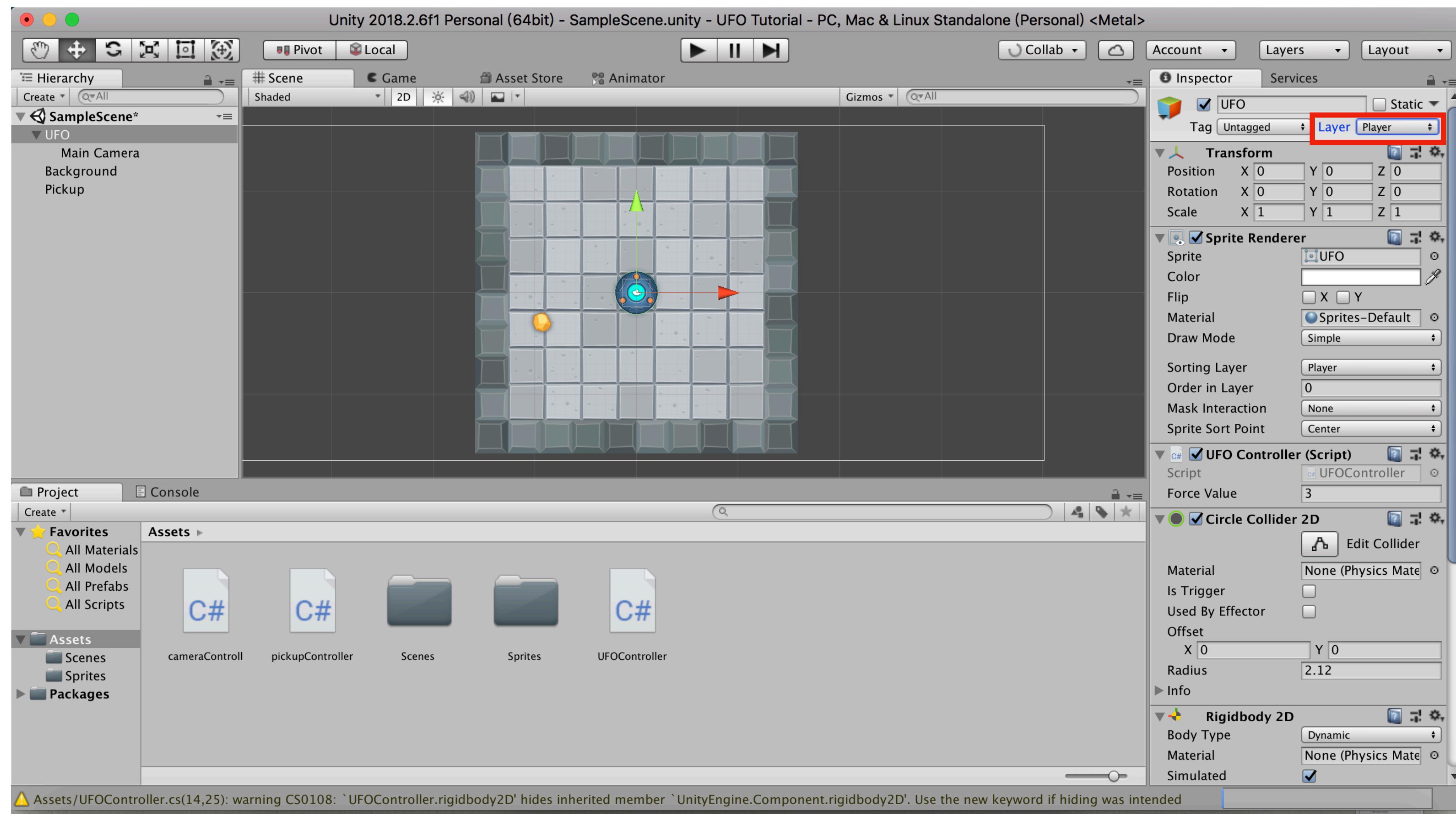
Physics 2D and Collision Matrix



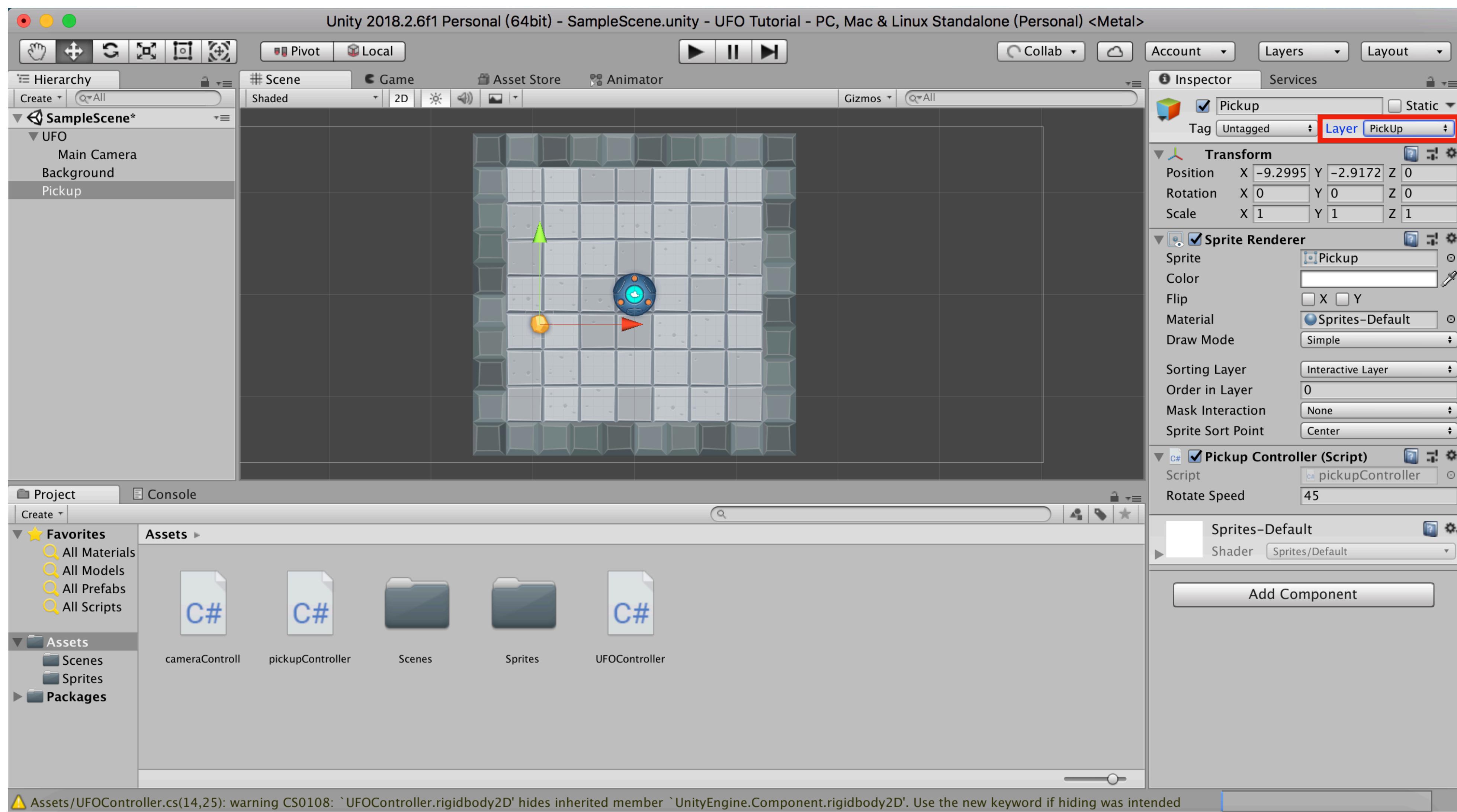
Physics 2D and Collision Matrix



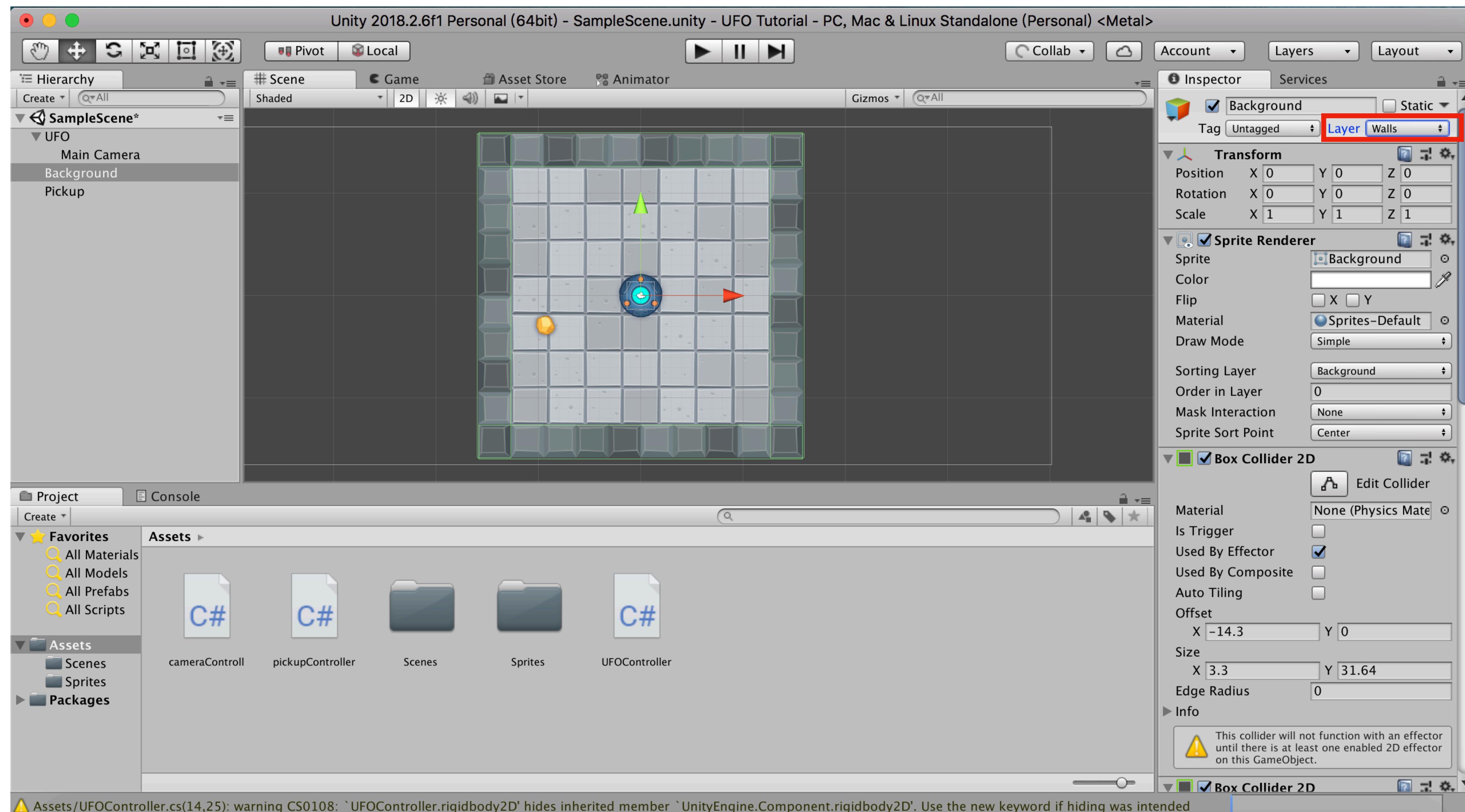
Set Layer



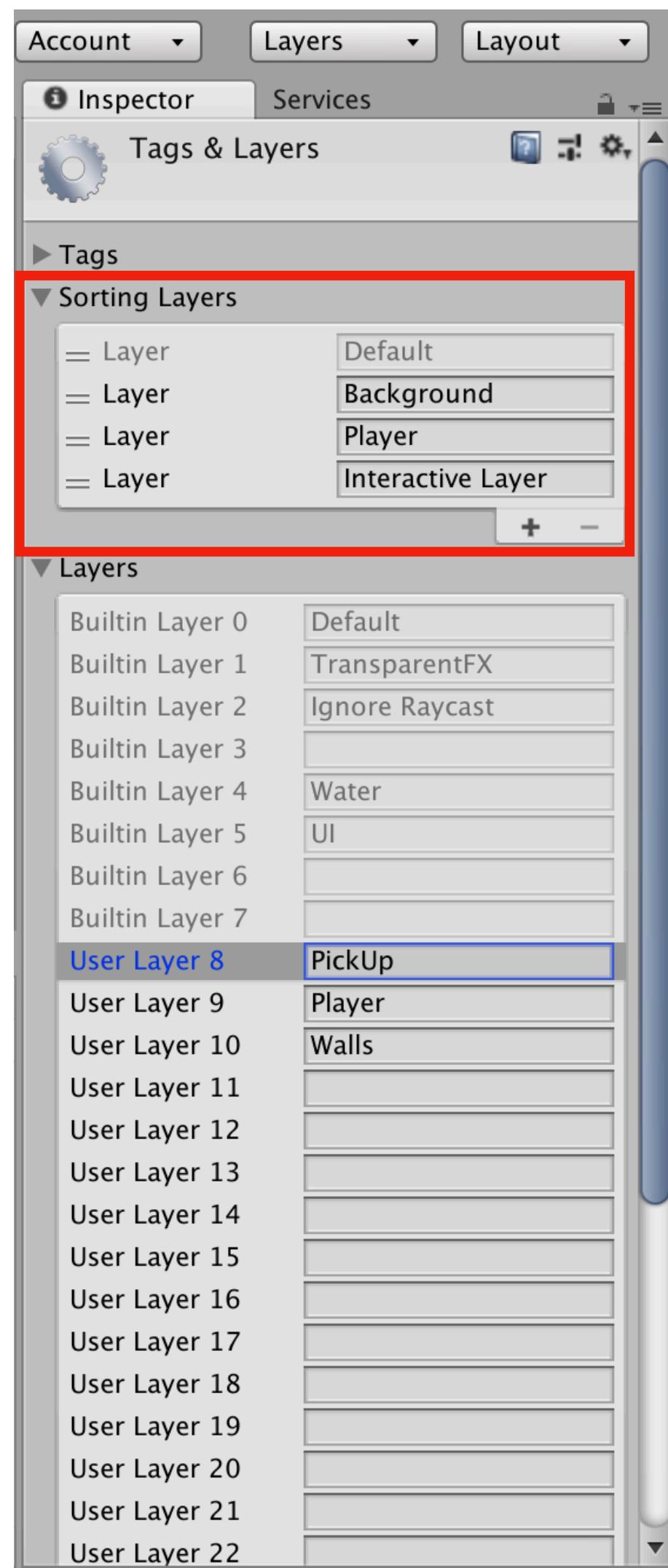
Set Layer



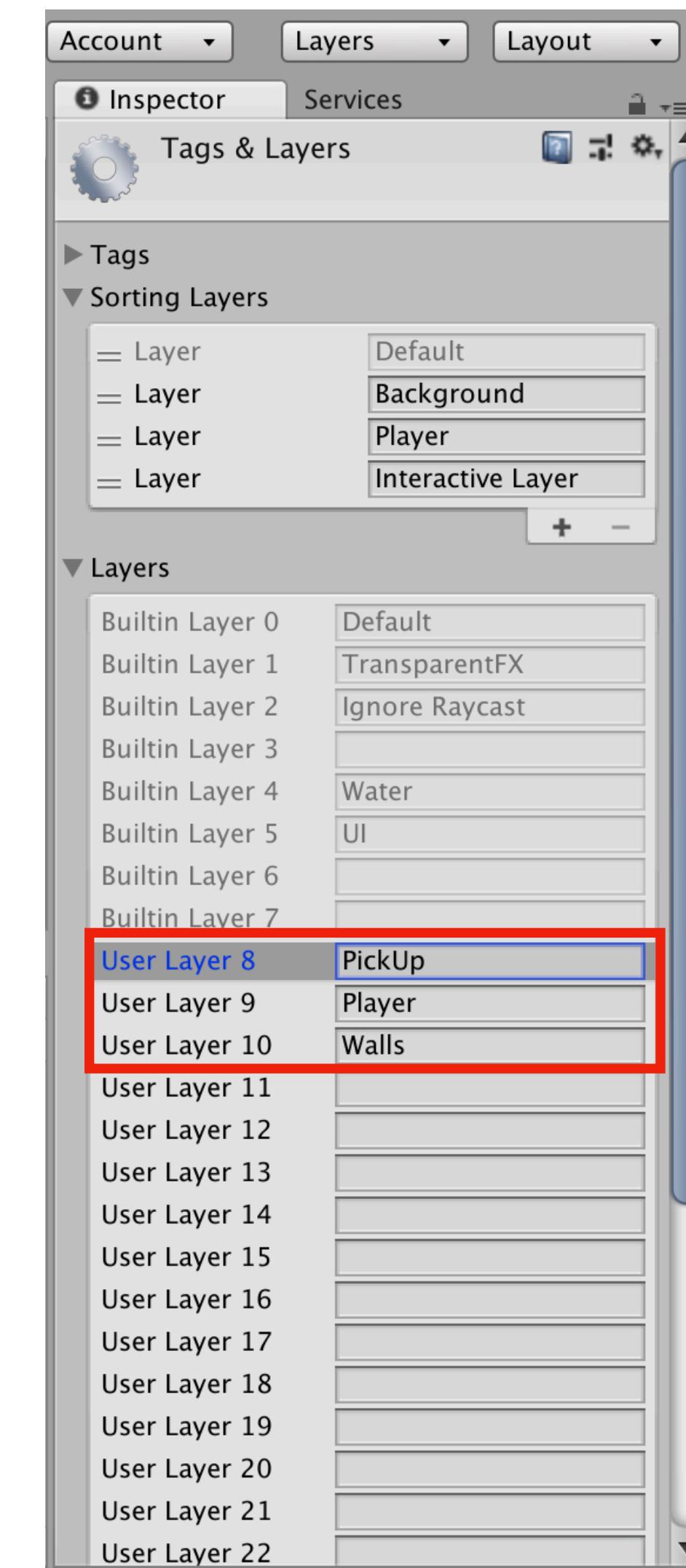
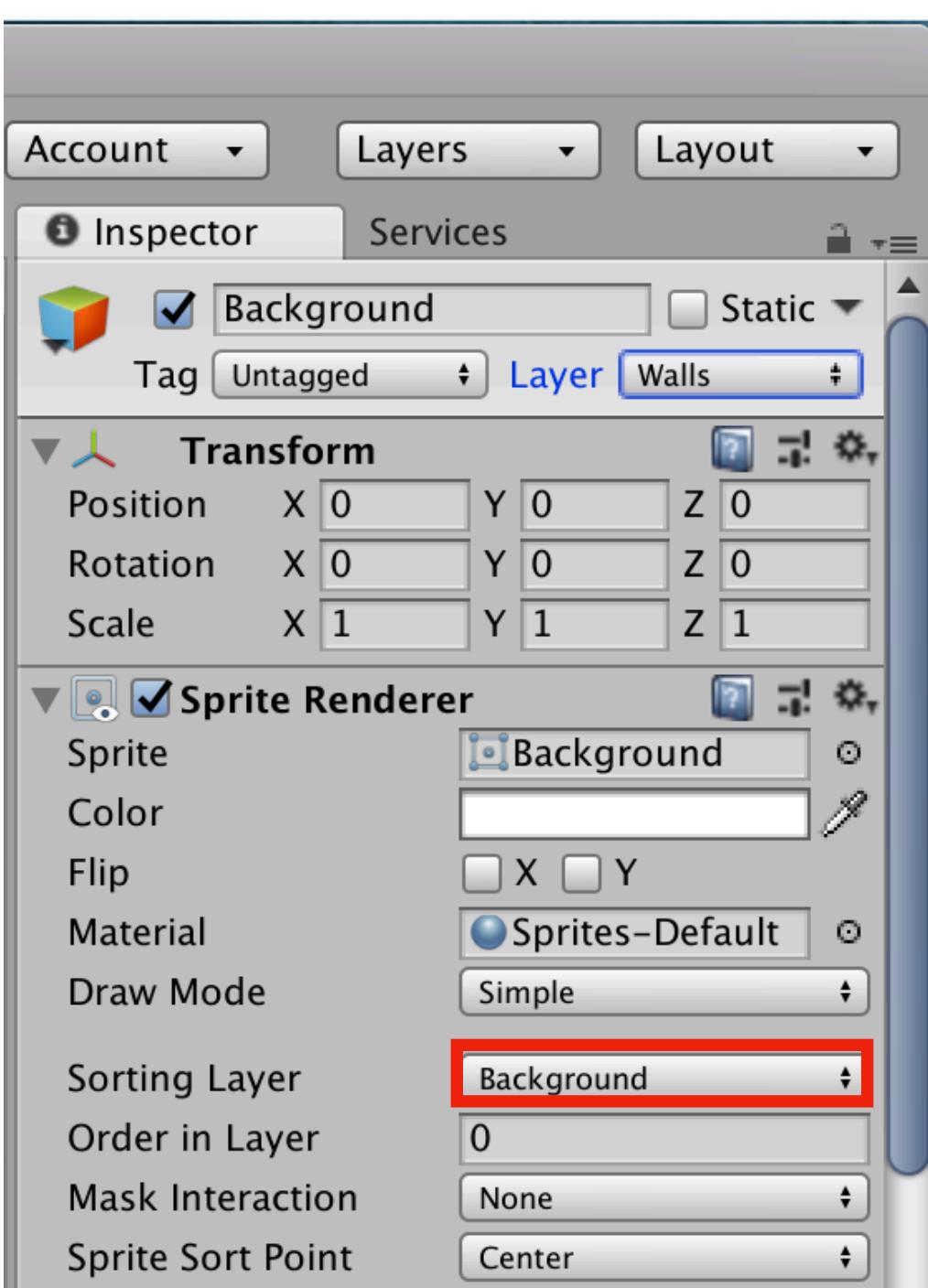
Set Layer



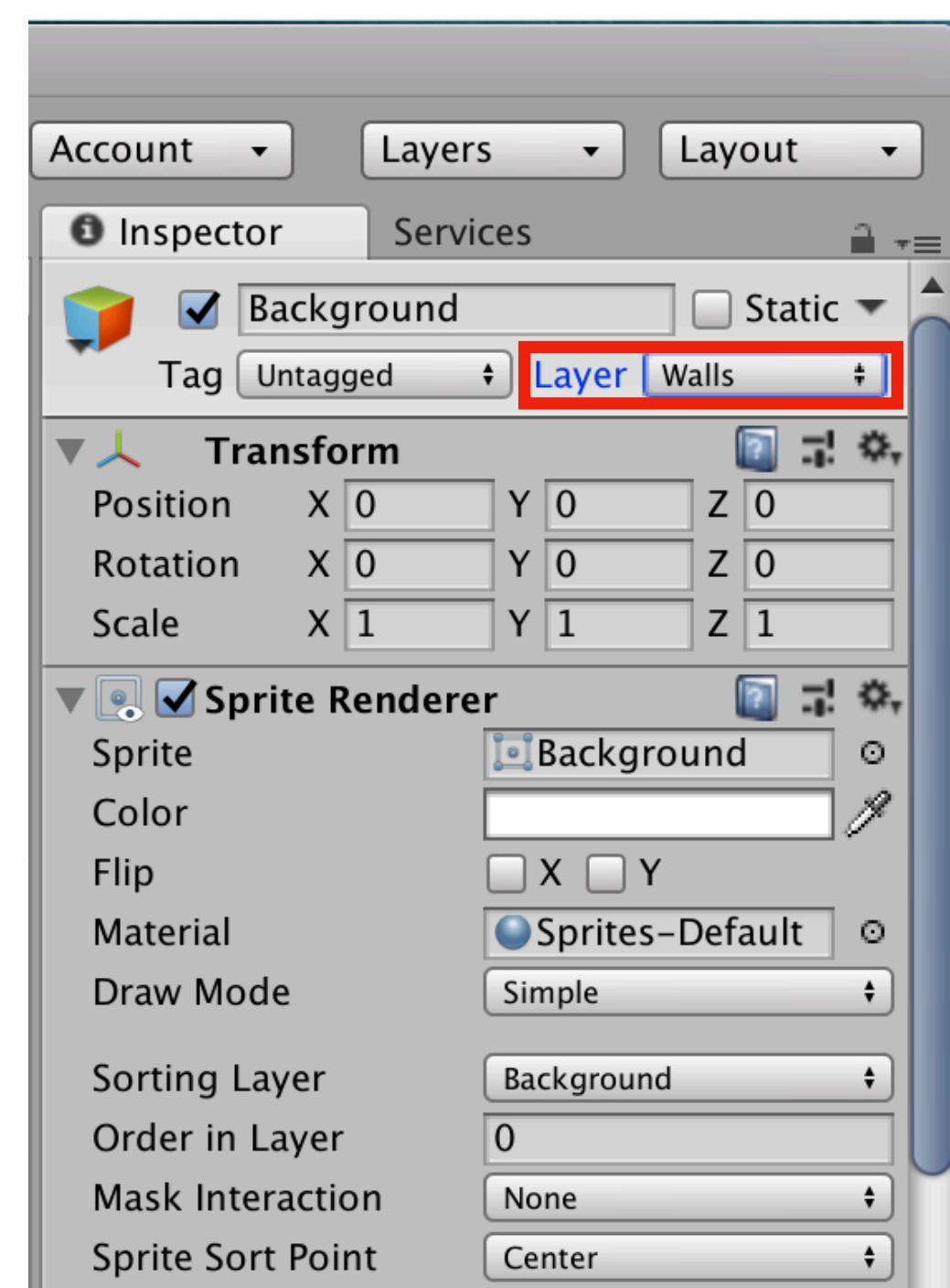
Sorting Layer v.s. Layer



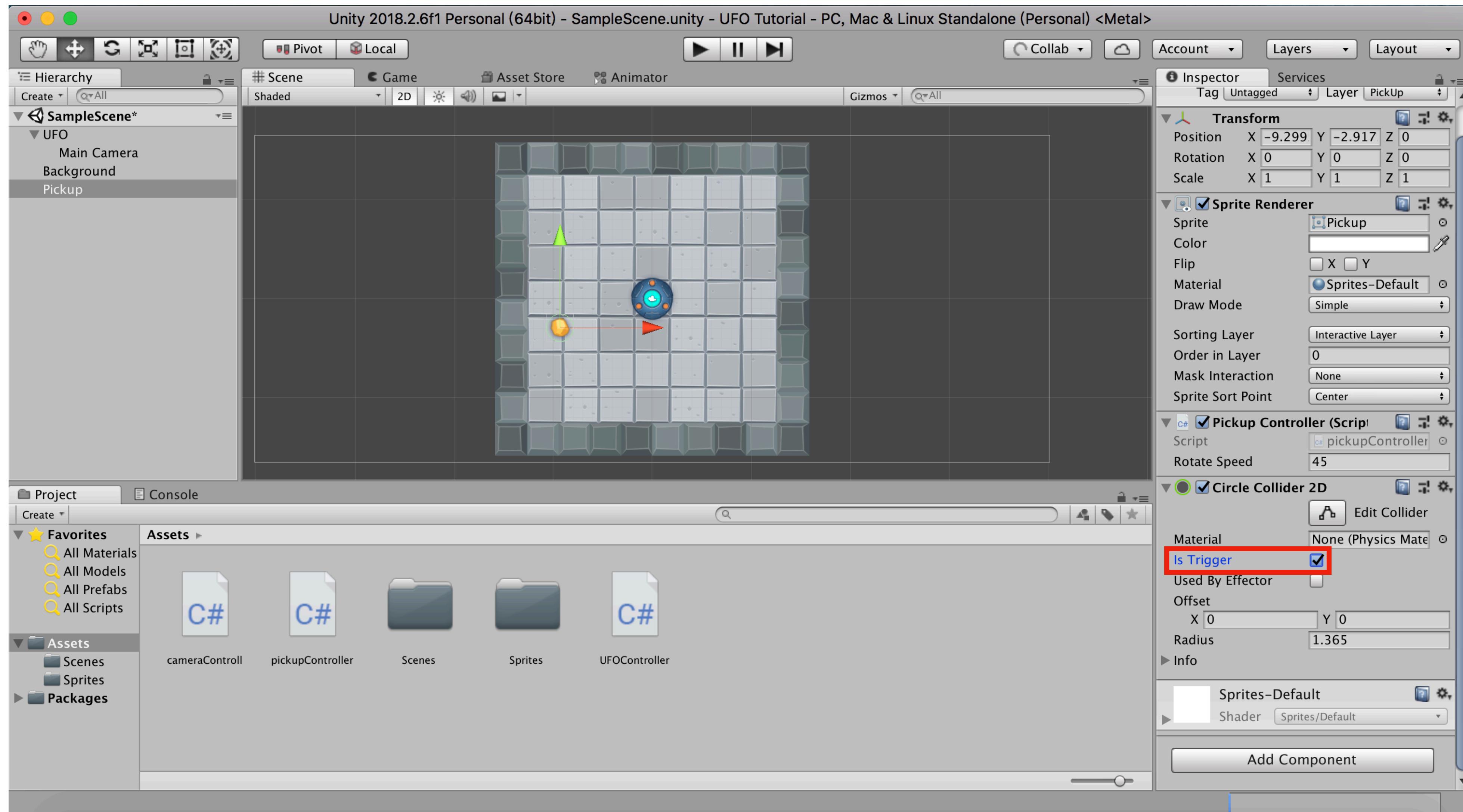
圖層前後



碰撞用



Collider Is Trigger



Collider and Trigger

碰撞器(Collider) & 觸發器(Trigger)

碰撞器給予物件碰撞範圍，讓它可以和別的物件產生碰撞偵測(會撞到彈開或被擋住)，可以用下列函式執行碰撞偵測：

OnCollisionEnter2D() -> 進入碰撞範圍瞬間時執行，只執行一次

OnCollisionStay2D() -> 停留在碰撞範圍時不斷執行

OnCollisionExit2D() -> 離開碰撞範圍瞬間時執行，只執行一次

觸發器則是只有偵測，沒有碰撞(會穿過去)，下列函式可以執行觸發偵測，用法同上：

OnTriggerEnter2D()

OnTriggerStay2D()

OnTriggerExit2D()

Collider and Trigger

何時會啟動碰撞和觸發：

任一方有**rigidbody** + 雙方都有**collider**，會有**碰撞偵測**
但若是**kinematic rigidbody**，則碰撞無效

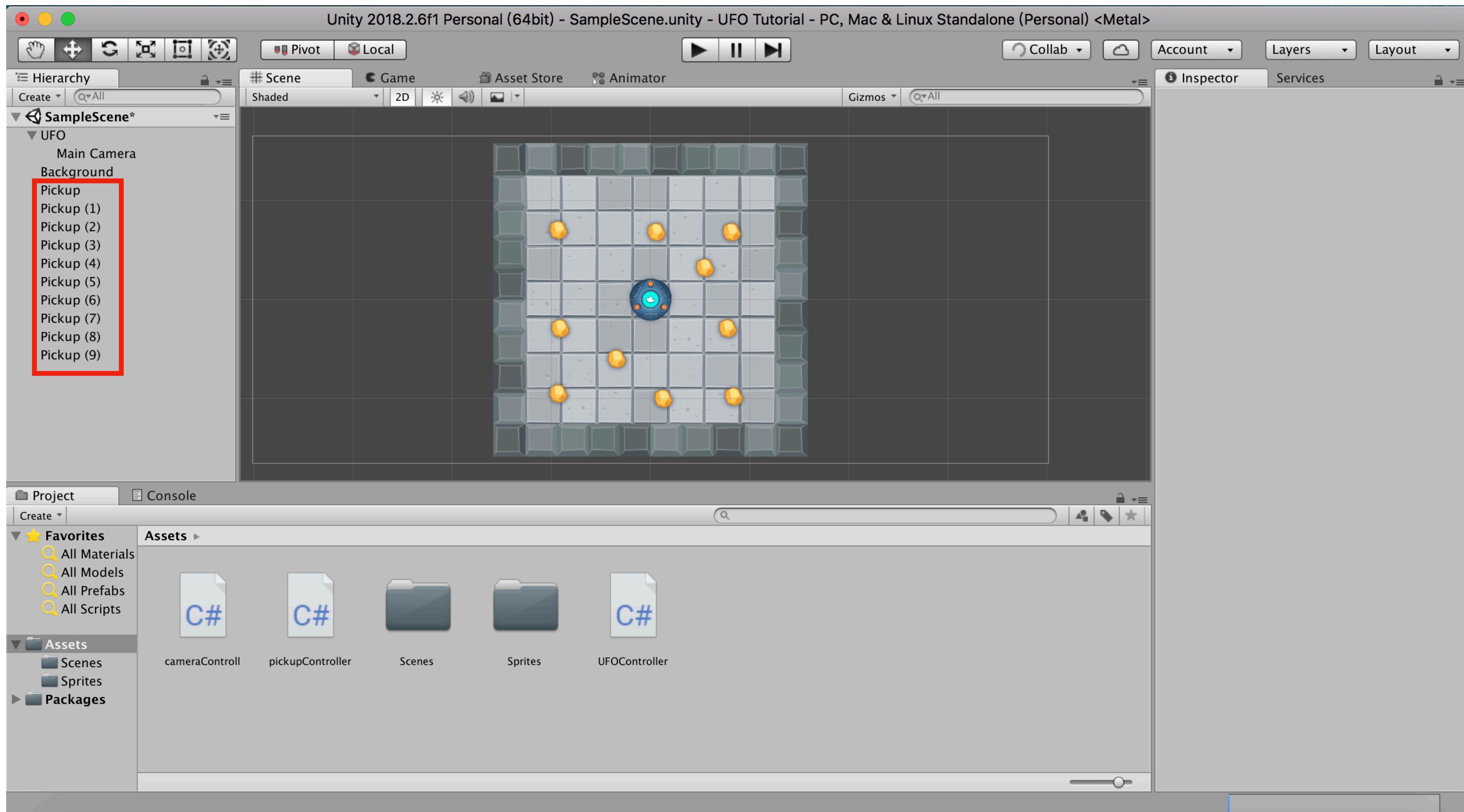
任一方有**rigidbody** + 雙方都有**collider** + 任一方有勾**trigger**，會
有**觸發偵測**

OnTriggerEnter2D

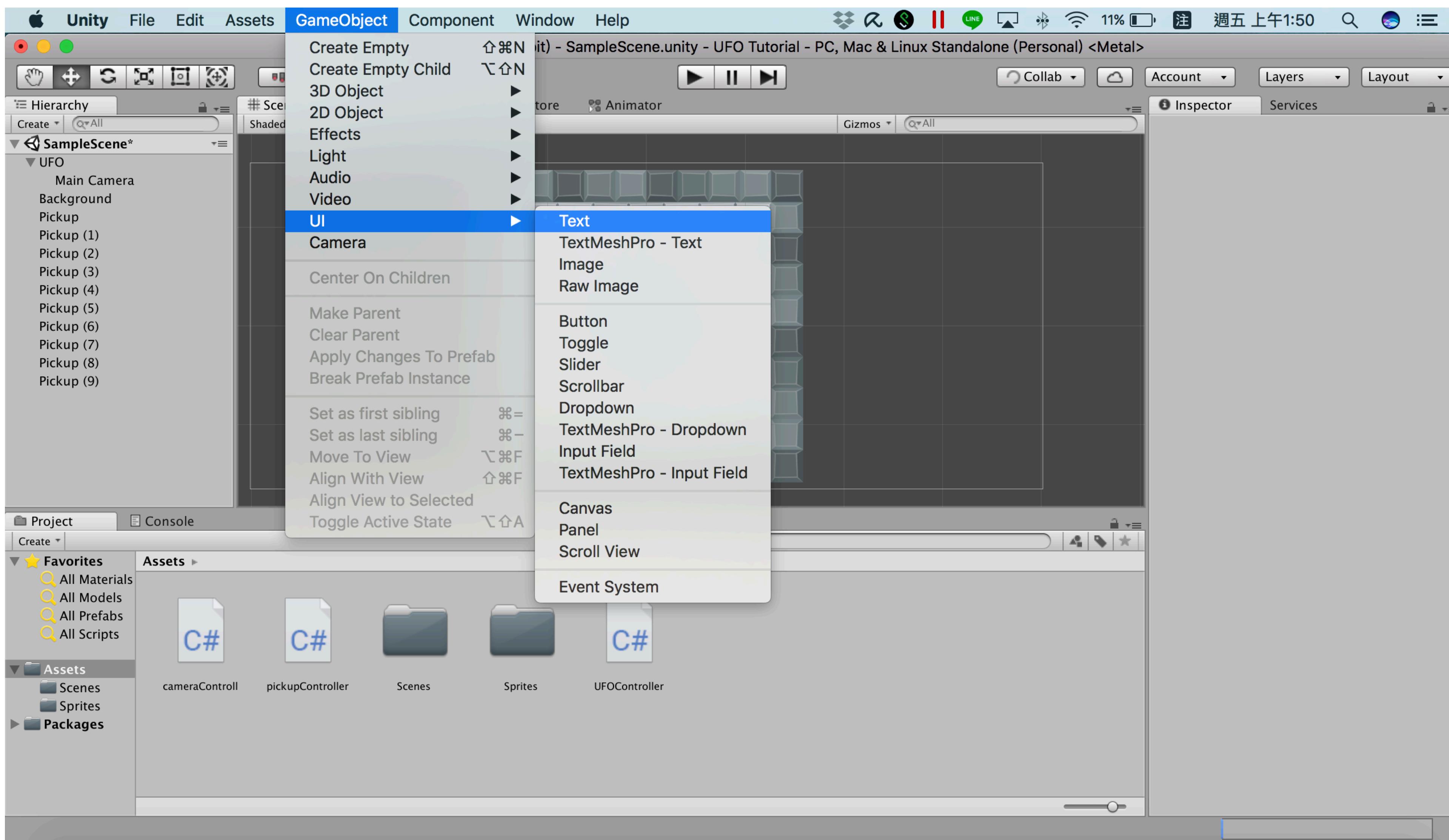
```
15      // Update is called once per frame
16      void Update () {
17
18          Vector2 force2D = Vector2.zero;
19
20          if (Input.GetKey(KeyCode.W))
21          {
22              force2D.y += forceValue;
23          }
24          if (Input.GetKey(KeyCode.S))
25          {
26              force2D.y -= forceValue;
27          }
28          if (Input.GetKey(KeyCode.A))
29          {
30              force2D.x -= forceValue;
31          }
32          if (Input.GetKey(KeyCode.D))
33          {
34              force2D.x += forceValue;
35          }
36
37
38          rigidbody2D.AddForce(force2D);
39      }
40
41      void OnTriggerEnter2D(Collider2D other)
42      {
43          other.gameObject.SetActive(false);
44      }
45
46 }
```

<https://bit.ly/2U0kYC5>

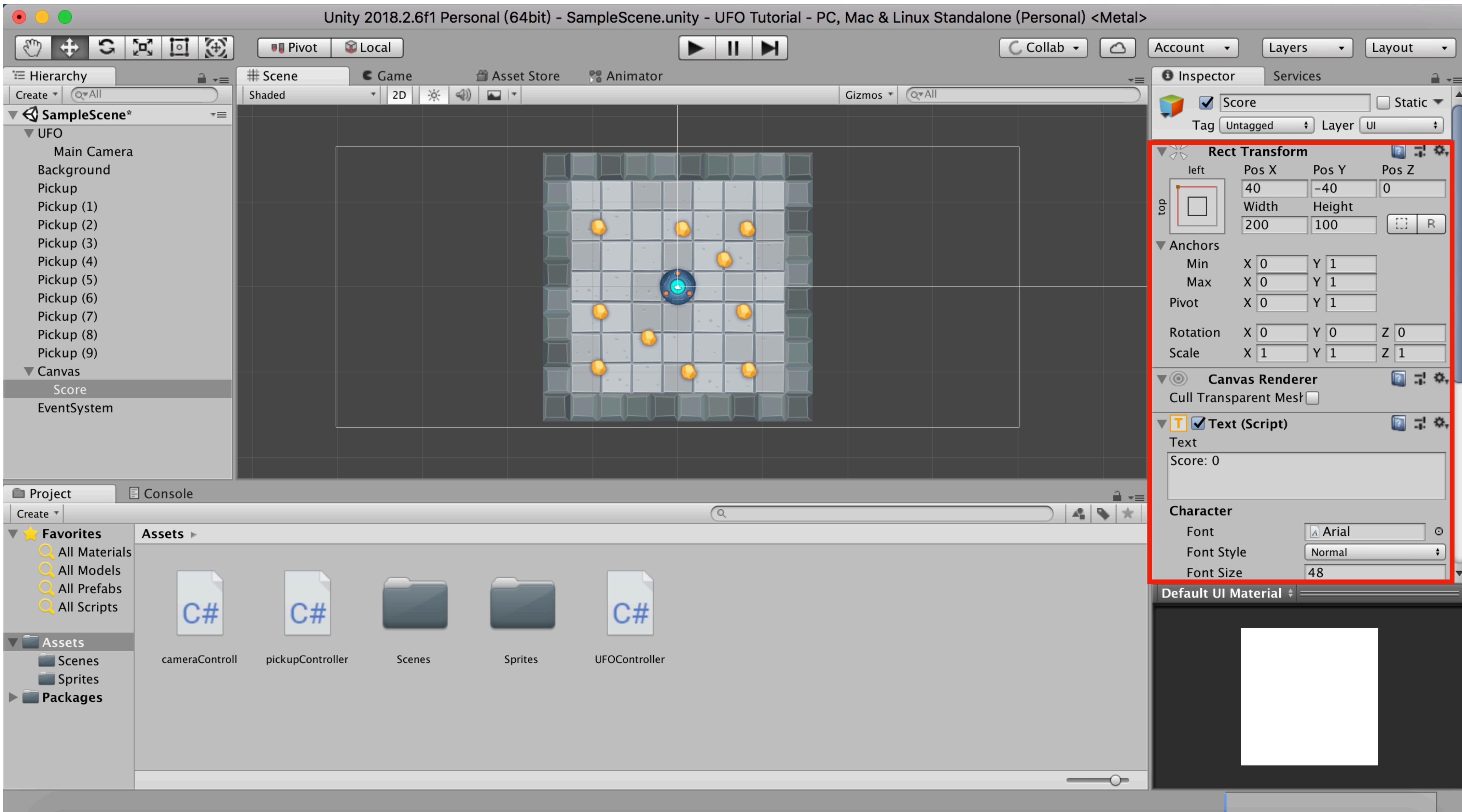
Copy PickUp



UI Text for Showing Score



Text on Canvas



ScoreManager

```
1 ⑧  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class scoreManager : MonoBehaviour {
7
8      private Text _text;
9      private float _currentScore = 0;
10     const string ScorePrefix = "Score : ";
11
12     // Use this for initialization
13     void Start () {
14         _text = this.GetComponent<Text>();
15         _text.text = ScorePrefix + _currentScore;
16     }
17
18     // Update is called once per frame
19     void Update () {
20
21     }
22
23     public void AddScore(int score)
24     {
25         _currentScore += score;
26         _text.text = ScorePrefix + _currentScore;
27     }
28 }
29
```

<https://bit.ly/2xS3FKY>

ScoreManager (for newer Unity versions that cannot use Text)

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  using TMPro;
6
7  public class ScoreManager : MonoBehaviour
8  {
9      private TMP_Text _text;
10     private float _currentScore = 0;
11     const string ScorePrefix = "Score : ";
12
13     // Use this for initialization
14     void Start () {
15         _text.text = ScorePrefix + _currentScore;
16     }
17
18     // Update is called once per frame
19     void Update () {
20
21     }
22
23     public void AddScore(int score)
24     {
25         _currentScore += score;
26         _text.text = ScorePrefix + _currentScore;
27     }
28 }
```

Using TMPro for TMP_TEXT

No Get.Componentent is required.

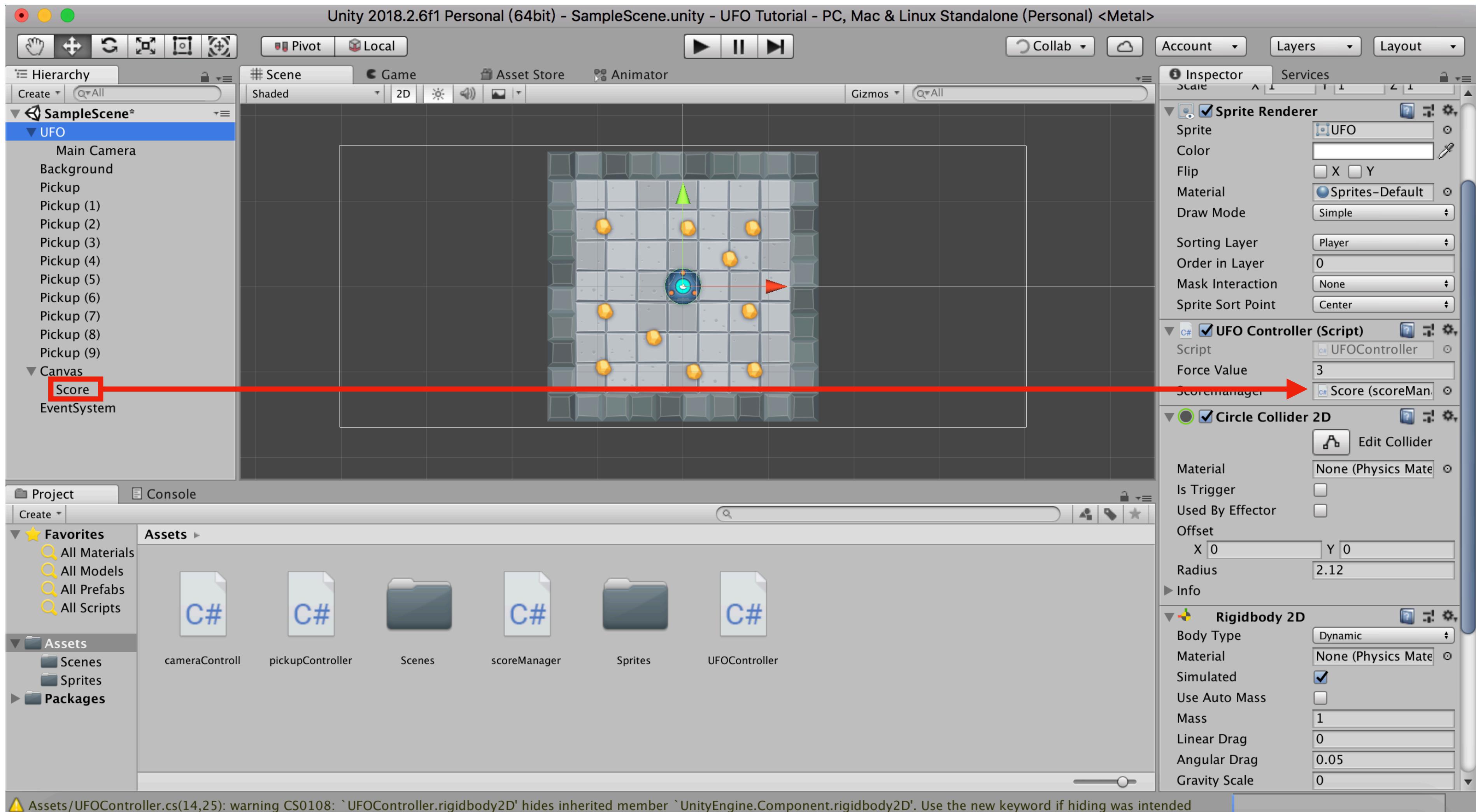
<https://bit.ly/3RtygDC>

UFOController

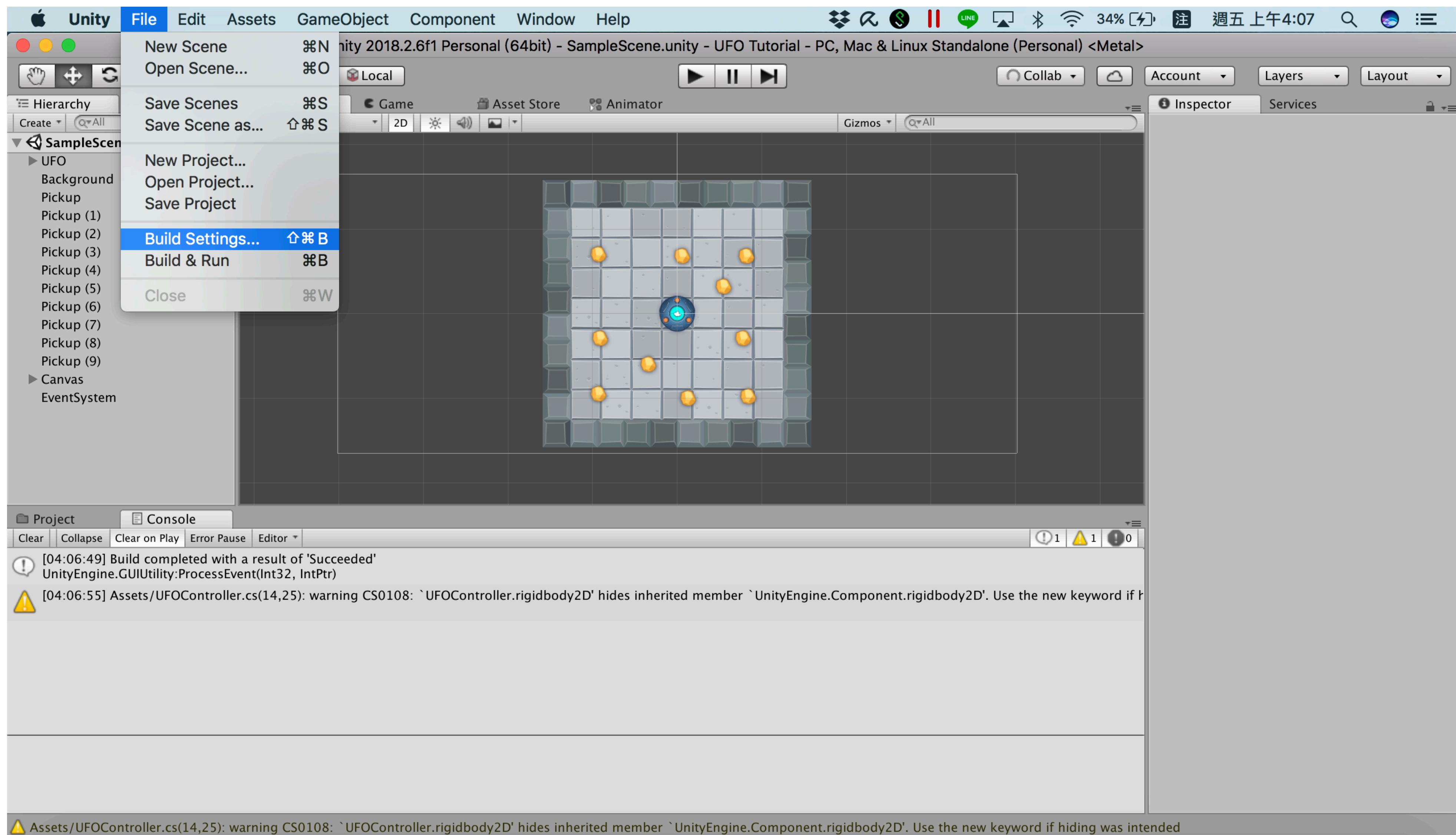
```
14  
15     public float forceValue;  
16     private Rigidbody2D rigidbody2D = null;  
17     public scoreManager scoremanager;  
18  
19     // Update is called once per frame  
20     void Update()  
21     {  
22  
23         Vector2 force2D = Vector2.zero;  
24  
25         if (Input.GetKey(KeyCode.W))  
26         {  
27             force2D.y += forceValue;  
28         }  
29         if (Input.GetKey(KeyCode.S))  
30         {  
31             force2D.y -= forceValue;  
32         }  
33         if (Input.GetKey(KeyCode.A))  
34         {  
35             force2D.x -= forceValue;  
36         }  
37         if (Input.GetKey(KeyCode.D))  
38         {  
39             force2D.x += forceValue;  
40         }  
41  
42         rigidbody2D.AddForce(force2D);  
43     }  
44  
45     void OnTriggerEnter2D(Collider2D other)  
46     {  
47         scoremanager.AddScore(100);  
48         other.gameObject.SetActive(false);  
49     }  
50 }
```

<https://bit.ly/2Uj3fVw>

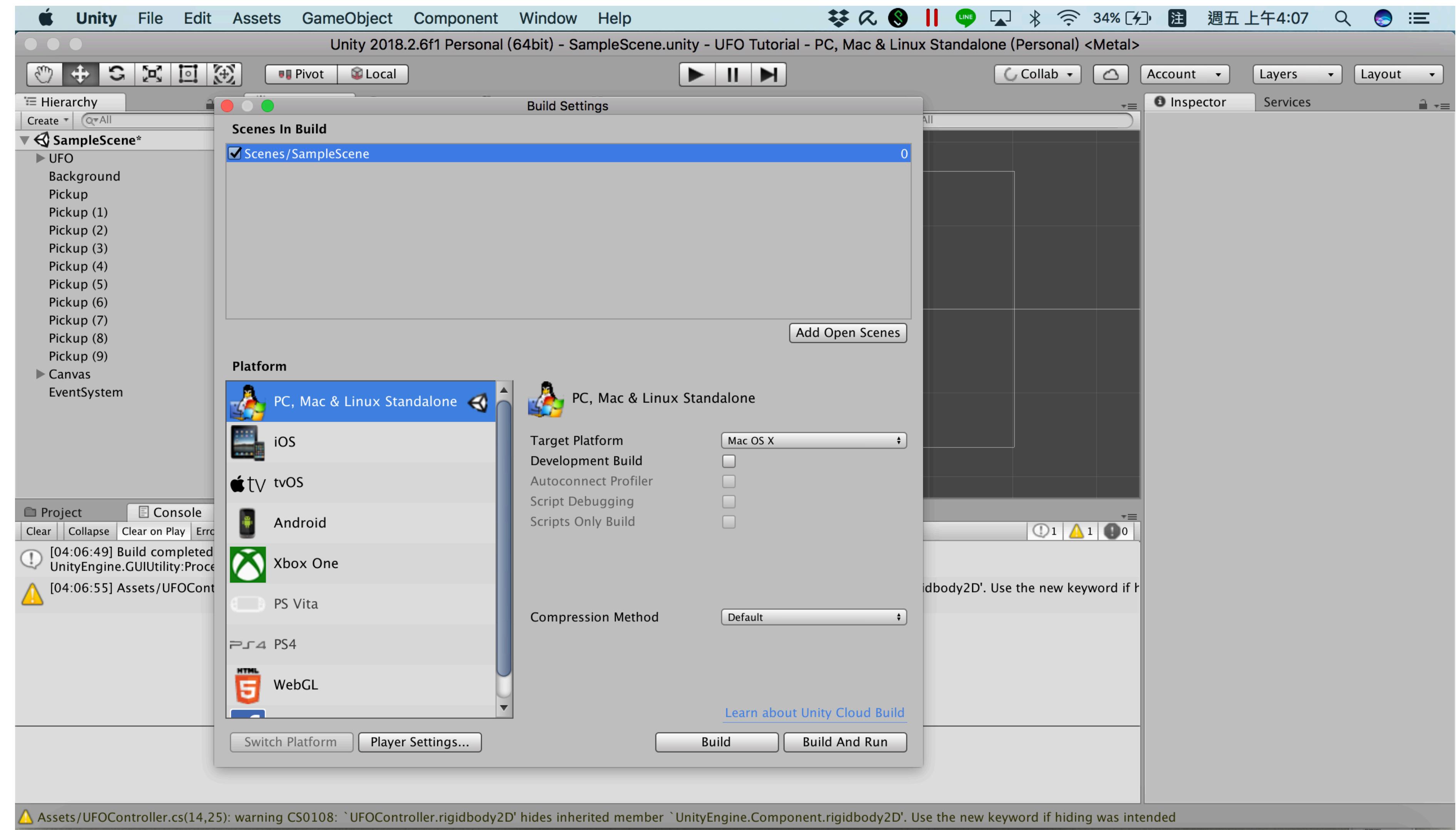
Drag to Assign



Build



Build



Today's class link:

[https://www.youtube.com/live/
LFaomTsvPIU?si=qYKm8qed1wK5Kf6L](https://www.youtube.com/live/LFaomTsvPIU?si=qYKm8qed1wK5Kf6L)

Syllabus

1	Introduction	9	Midterm
2	Data storage	10	Loop
3	Operating systems	11	Function
4	Algorithm	12	Array
5	Day off (10/10)	13	OOP
6	Network	14	Unity 2D game control
7	Variables, operators and operands, statements	15	Final exam
8	Reviewing and practicing	16	Final project

Grading Policy

- Midterm (30%)
- Final Exam (30%)
- Final Project (30%)
- Participation (10%)

Final Exam

- Week 15 (12/18)
- 2:10pm~4:10pm

Final Project Requirements (12/25)

- Original project or revised from current tutorial projects are both ok (with different score levels).
- If using tutorial projects (<https://learn.unity.com/tutorials>), please describe what you revise or improve.
- Each one has 3 min. presentation/demo time + 3 Q&A.
- A report in **PDF** includes the flow of the code, which should be uploaded on Moodle.
- Source code/project are required.