

Unity LAB-2

2D UFO Tutorial

2D + Physics

iNTERACTIVE
MEDIA



課程目標

- 對於unity介面有基礎認識
- 完成一個非常簡單的2D小遊戲
 - 物理引擎，使用剛體與碰撞體的觀念
 - 觸發機制
 - 文字顯示



建立Unity的2D檔案

The screenshot shows the Unity 'New Project' dialog. At the top, it says '新專案' (New Project) and '編輯器版本: 2022.3.8f1' (Editor Version: 2022.3.8f1). Below that are buttons for 'Apple 晶片' (Apple Chip) and 'LTS'. On the left, there's a sidebar with '所有範本' (All Templates), '核心' (Core), '範例' (Examples), and '學習' (Learning). The main area shows a search bar with '搜尋所有範本' (Search all templates) and a list of templates:

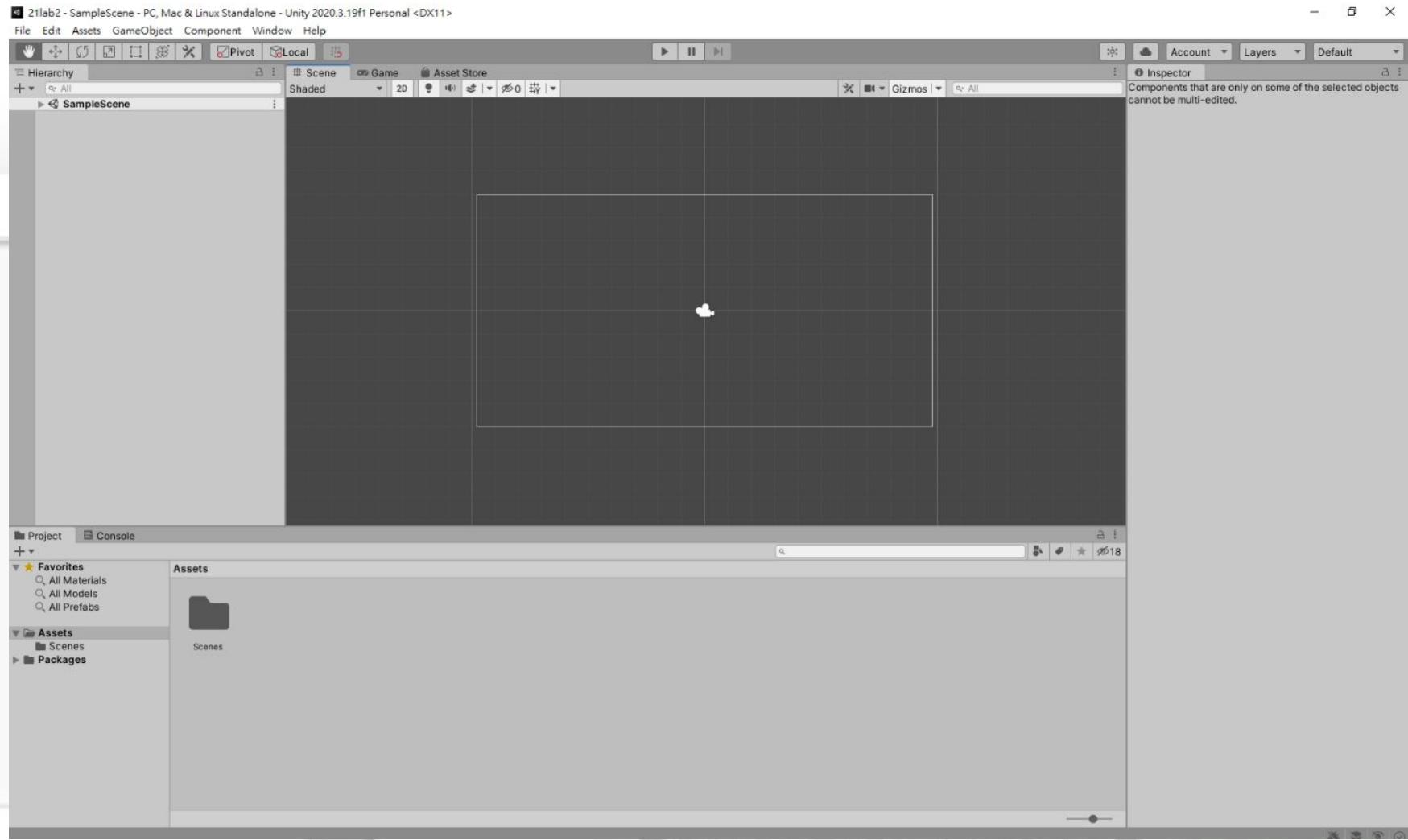
- 2D** 核心 (selected)
- 3D 核心
- 2D (URP) 核心 (with SRP badge)
- 3D Mobile 核心 (with download icon)
- 2D Mobile 核心 (with download icon)

To the right, there's a preview window showing a white square on a grid background, and a detailed description for the selected '2D' template:

2D
This is an empty project configured for 2D apps. It uses Unity's built-in renderer.

下方有 '更多資訊' (More Information)、'專案設定' (Project Settings)、'專案名稱' (Project Name: My project)、'位置' (Location)、'取消' (Cancel) 和 '建立專案' (Create Project) 按钮。

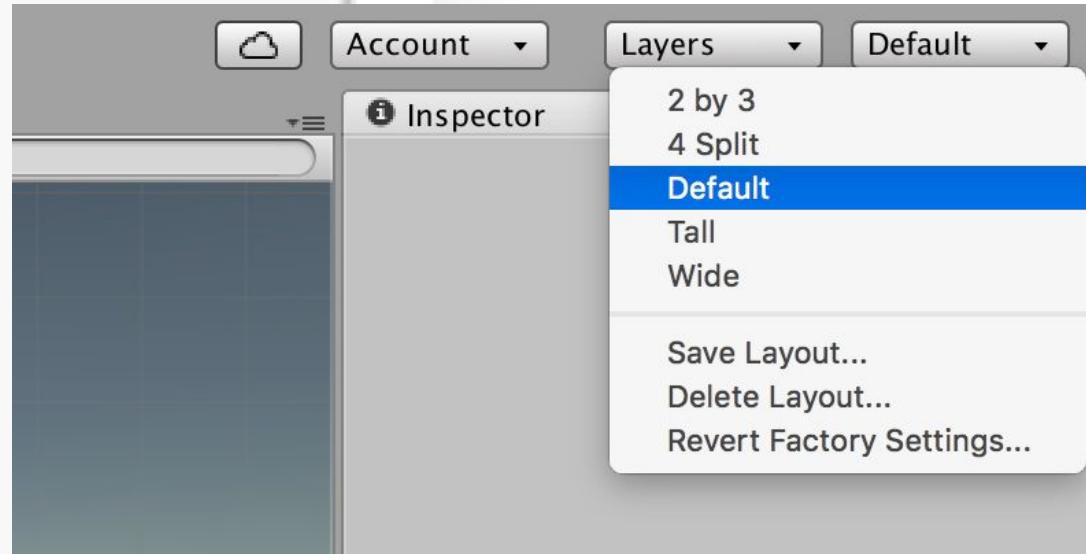
UnityEditor 介面



MEDIA

UnityEditor 介面

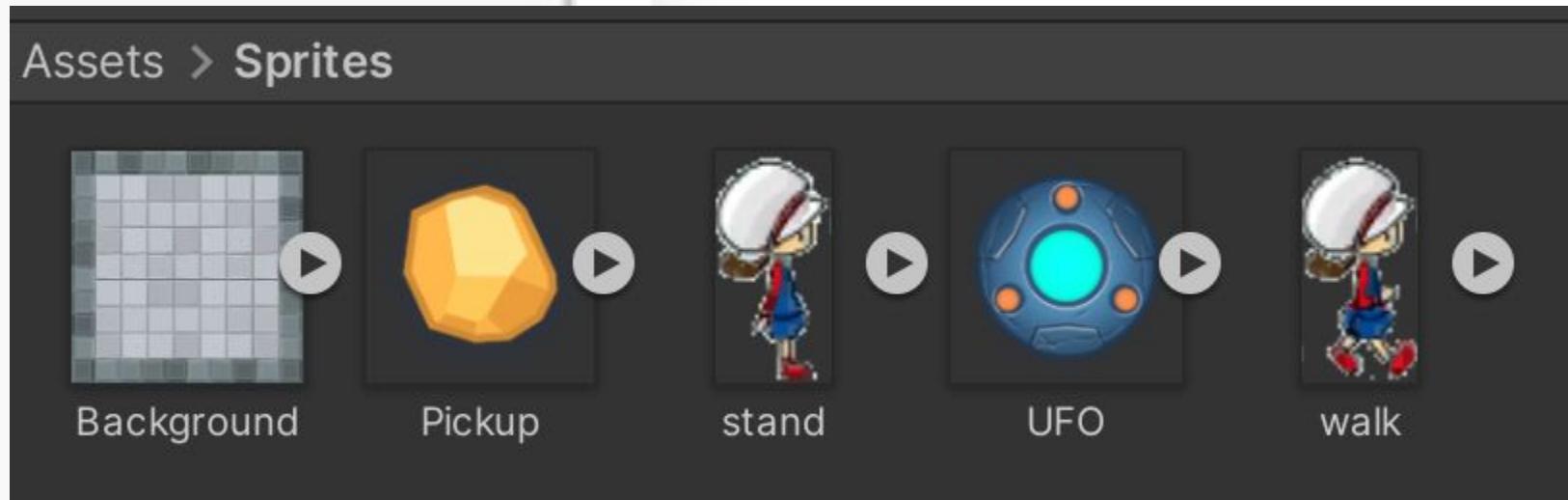
- 這些Tab可以自由移動位置
- 如果有Tab被你移到不見了
 - 右上角有選單可以歸位或其他調整



iNTERACTIVE
MEDIA

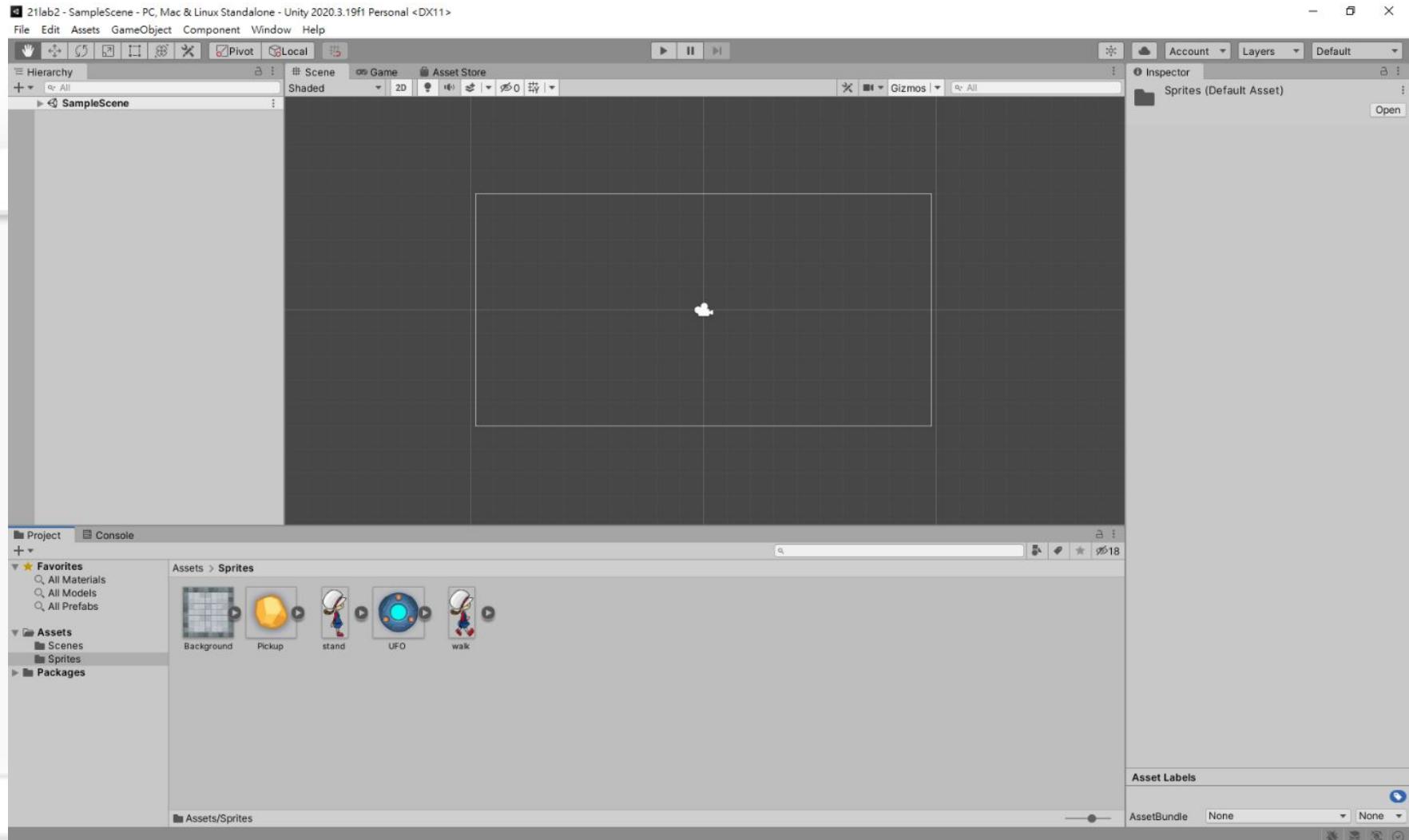
匯入素材

- 汇入第二週上課用的 [UFO_Sprites](#)
- 解壓縮後，直接從檔案總管(資料夾)把檔案拖進 Editor 介面的 Project Tab



INTERACTIVE
MEDIA

匯入後，畫面大概長這樣



MEDIA

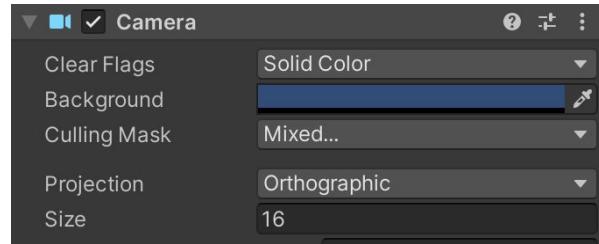
Camera

- 點擊camera物件，inspector會顯示細節



Camera Projection

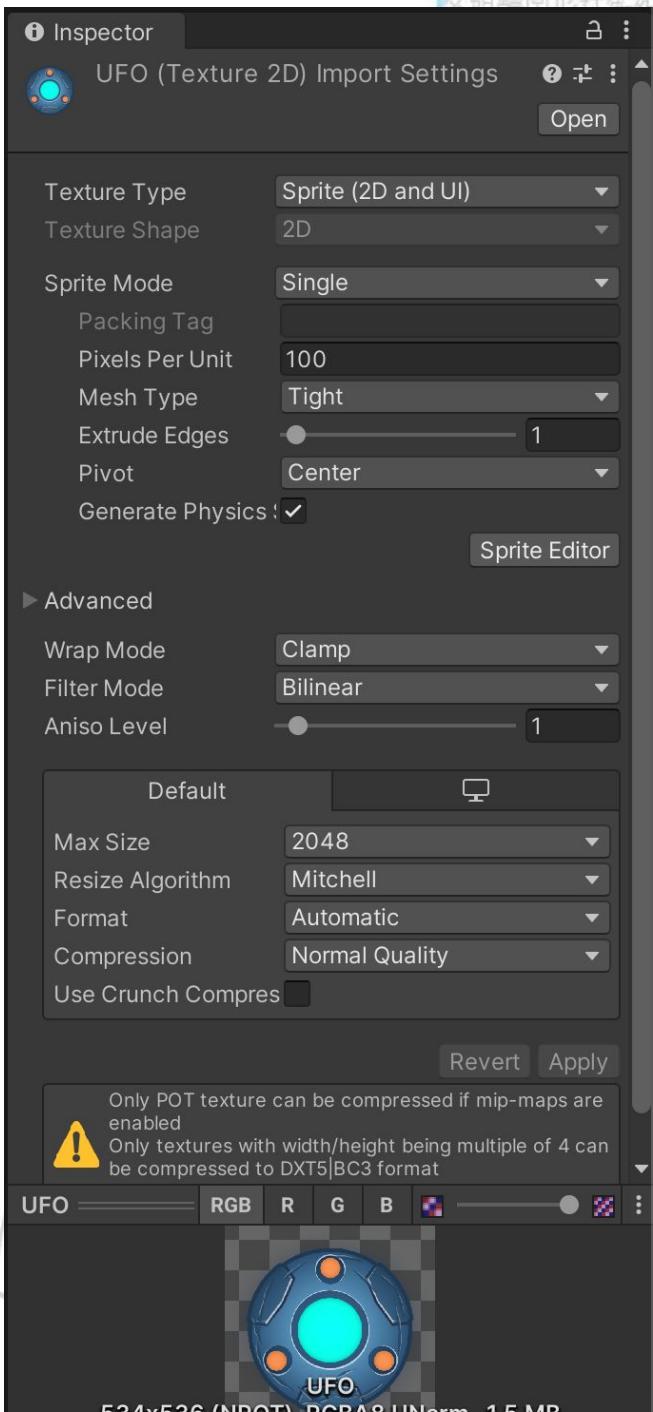
- Orthographic VS Perspective
 - 2D遊戲主要使用Orthographic
 - Camera Size調到16



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

放入素材

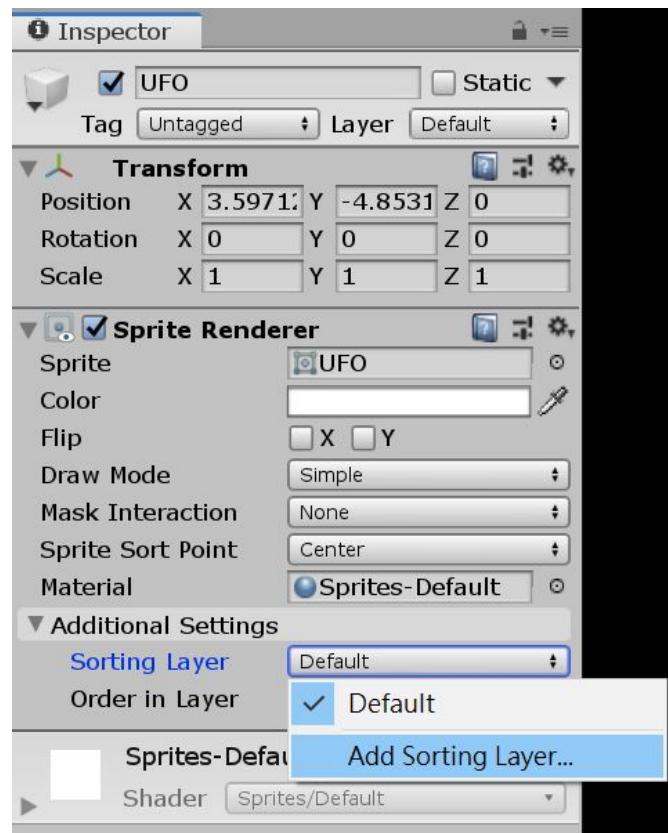
- 確認素材Texture Type
 - 先點要使用的素材
 - 看看inspector的texture type是不是 **Sprite(2D and UI)** 不是就自己改一下
- 將素材拖曳進場景



2D圖層順序

- Inspector tab那邊
- 作法一：**Sorting Layer**
 - 建立Background和Player圖層
 - 調整順序。越下方的layer在畫面最前方(越不會被遮蔽)
- 作法二：**Order in Layer**
 - 調整的數字，數字越大越上層

Order in Layer 0



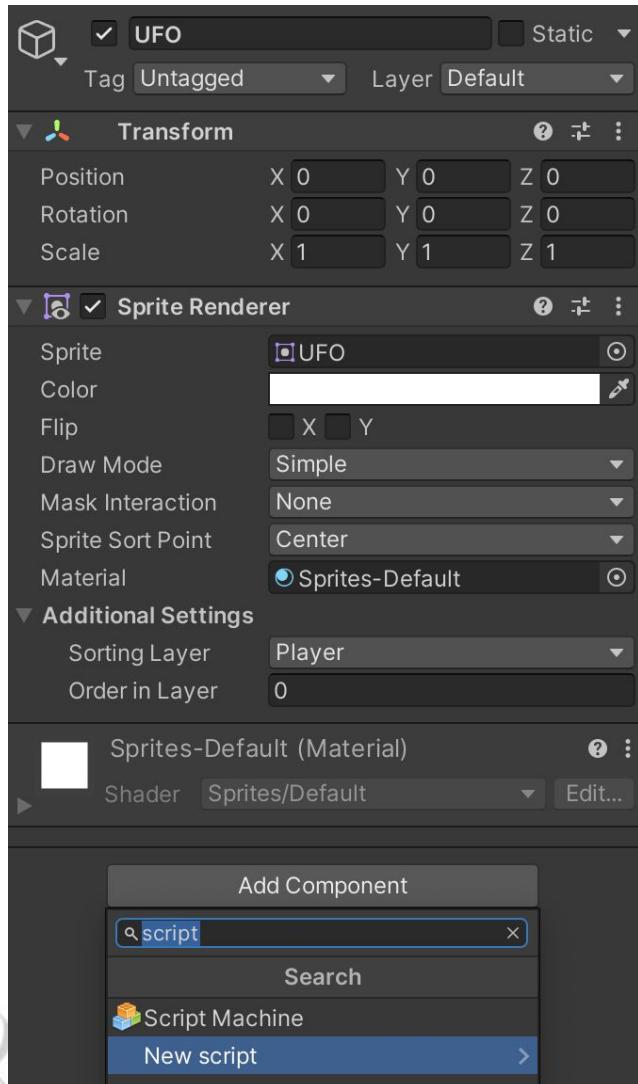
實作時間(3min)

1. 建立專案
2. 下載素材檔案並匯入
3. 設定素材

iNTERACTIVE
MEDIA

加入script

- 法一 : 在project tab有
create→C# Script, 或在
project欄右鍵create→C#
Script
- 法二 : 在你選定的物件屬性直
接加入script
- 先在UFO上面加一個move的
script



move.cs

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

小提醒

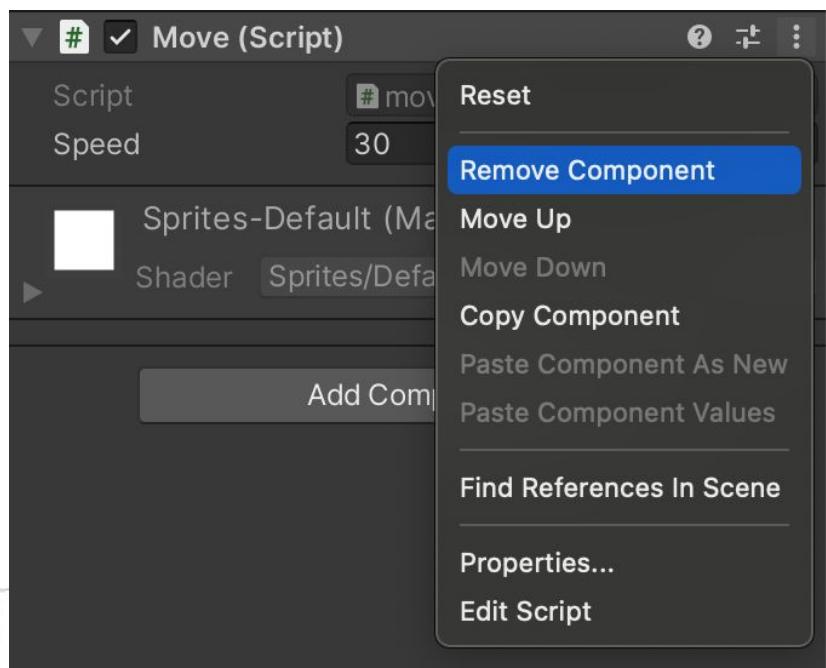
- 呈上頁的 **public** 變數
 - 將變數設定為public就可以直接在Unity UI介面上進行設定
- Unity 本身是一個整合程式、企劃、美術、音樂、音效的開發環境
 - 因此程式人員更要了解其他人的需求，撰寫適合的工具讓他人使用
- 所以寫code不只要注意可閱讀性、效率、架構，更要想到其他人使用這份code的方便性
- 在本學期的各個遊戲企劃也是一樣的，你也要為了與組員協作著想

INTERACTIVE
MEDIA

實作時間(2min)

1. 建立控制UFO的script

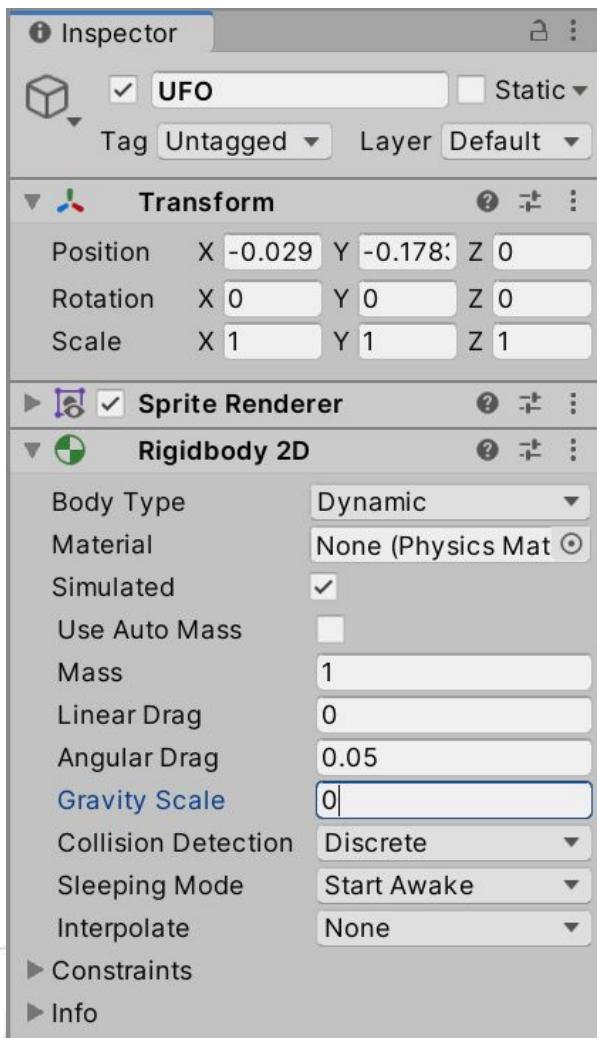
- 剛剛的move.cs內的移動模式比較像是讓物件移動位置(不符合物理的感覺)
 - 所以先把UFO component中的move.cs刪掉
 - 新增UFOController.cs, 以用符合物理的模式讓UFO移動



i
MEDIA

讓UFO增加物理特性(1)

- 點擊物件後 inspector
- Add Component
 - **Rigidbody2D** (剛體)
- Rigidbody2D 的
 - Gravity Scale先調到0



INTEC

MEDIA

UFOController.cs

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  // References
6  public class UFOController : MonoBehaviour
7  {
8      public float forceValue = 2f;
9      Rigidbody2D rbody2D;
10     // Use this for initialization
11     void Start()
12     {
13         rbody2D = this.GetComponent<Rigidbody2D>();
14     }
15
16     // Update is called once per frame
17     void Update()
18     {
19         Vector2 force2D = Vector2.zero;
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         rbody2D.AddForce(force2D);
37     }
38 }
```

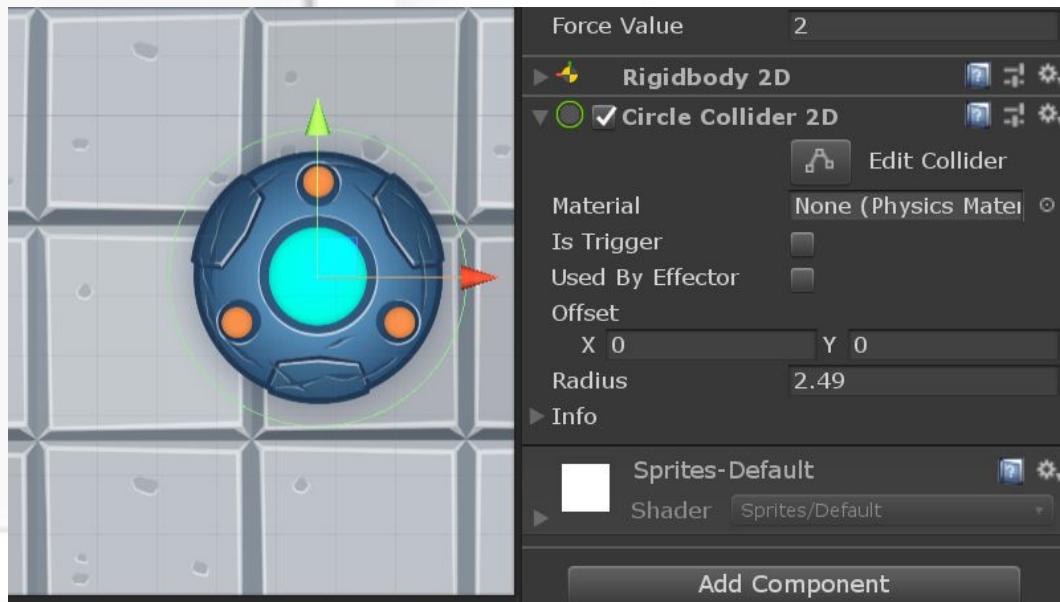
rbody2D.AddForce()

使用力量去推動UFO的位置, UFO會產生加速度跟慣性

INTERACTIVE
MEDIA

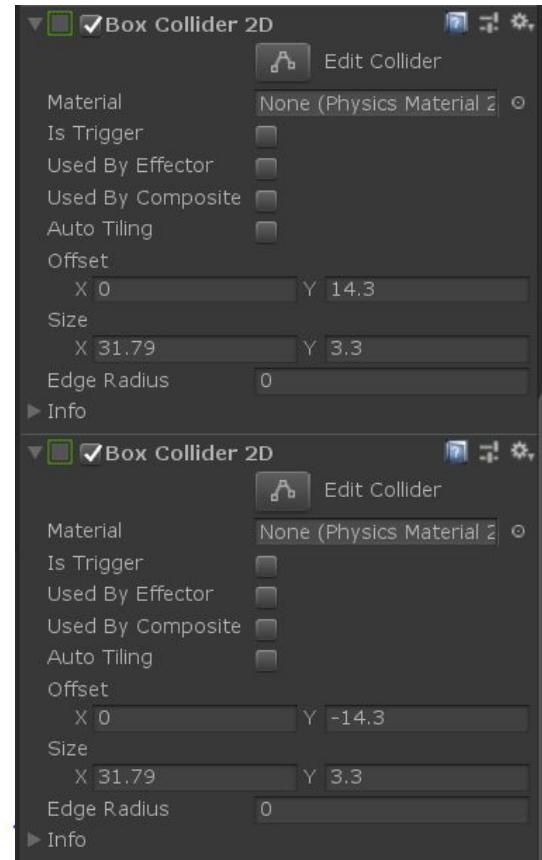
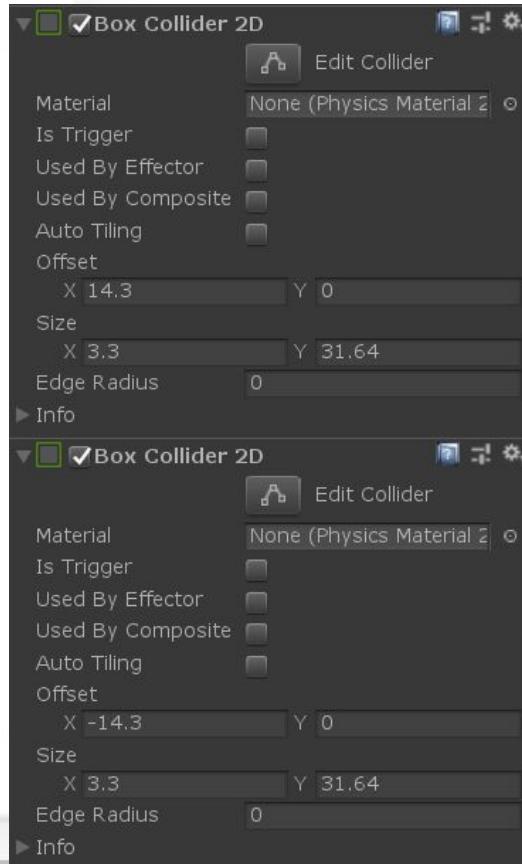
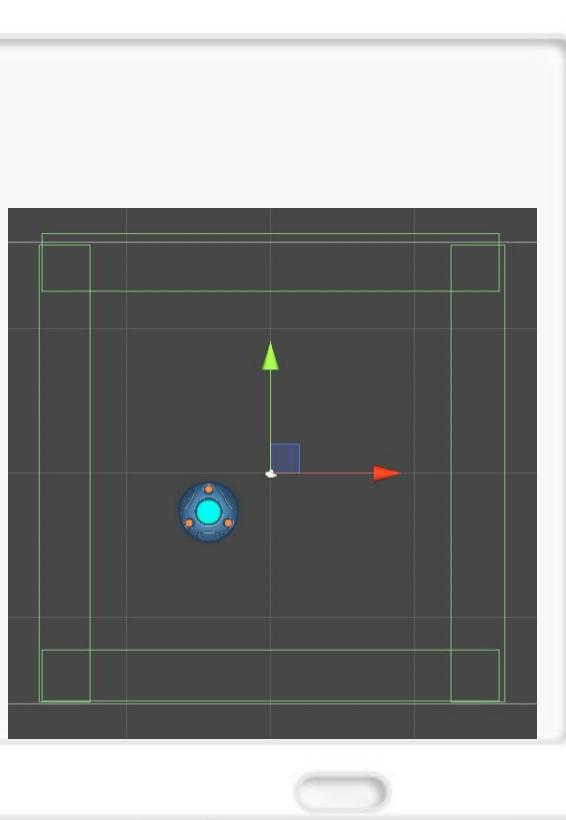
讓UFO增加物理特性(2)

- 點擊物件後 inspector→AddComponent
- 搜尋 “**collider**” (碰撞器)
- 選取適合的collider - **Circle Collider 2D**
- 調整Collider 的radius使其符合物件外輪廓



遊戲邊界

- 點選Background物件
- 加上四個BoxCollider2D



小體驗(本頁面上課不會帶到)

- 可以將Main Camara放到UFO的子物件
 - (在hierarchy中將Main Camara拖曳至UFO處)
- 按play 你就會看到鏡頭跟著UFO跑
- 撞到牆以後鏡頭會開始瘋狂旋轉，是因為UFO transform 中的rotation撞到牆後會改變(這是物理特性才有的)
- 如果你是用前面寫的move.cs作為移動的腳本，UFO撞到牆就不會改變UFO的方向(因為不是用剛體的受力在進行位移，只是單純在變更UFO位置而已)

INTERACTIVE
MEDIA

實作時間(5min)

1. 移除move.cs
2. 新增UFOController.cs
3. 設定UFO物理特性 -
Rigidbody & Collider
4. 設定背景邊界

加上金塊

- 放金塊到scene中 (金塊呢?)
- **sorting layer** 加上“Pickup”
- 現在圖層優先度：
 Pickup>Player>Background
- 金塊的sorting layer改成Pickup
- 金塊附一新腳本“pickupController.cs”

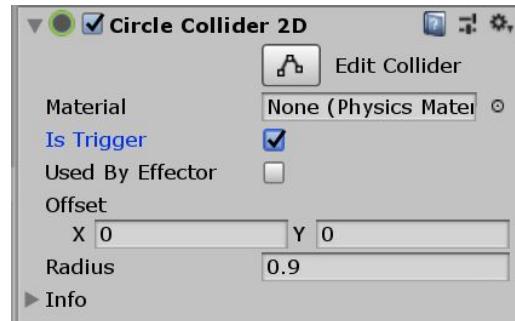
pickupController.cs

```
public class pickupController : MonoBehaviour {  
    public float rotateAngle = 45f;  
    // Use this for initialization  
    void Start () {  
  
    }  
  
    // Update is called once per frame  
    void Update () {  
        this.transform.Rotate(new Vector3(0, 0, rotateAngle * Time.deltaTime));  
    }  
}
```

以Z軸為軸心旋轉

讓UFO撿金塊

- 將金塊加上CircleCollider2D
 - Radius約0.9 勾上 is trigger
- 將UFOController.cs加入一個函式



```

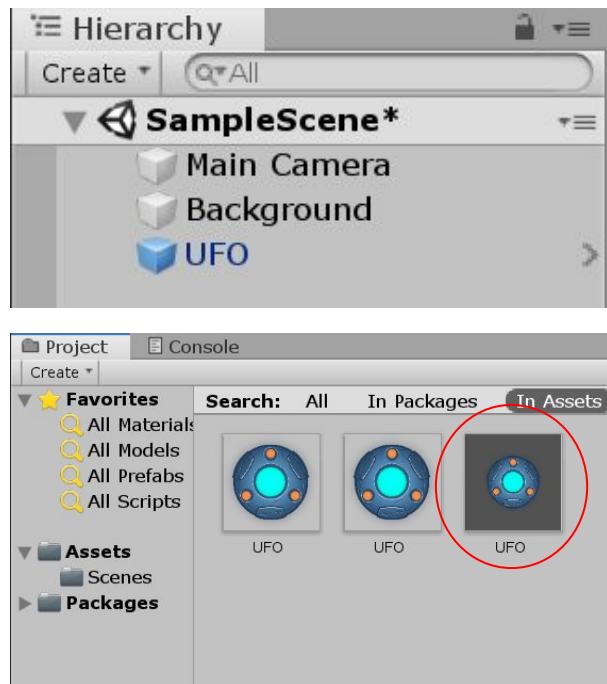
27         force2D.x += forcevalue;
28     }
29     rigidbody2D.AddForce (force2D);
30 }
31 void OnTriggerEnter2D(Collider2D other){
32     Destroy (other.gameObject);
33 }
34 }
35

```

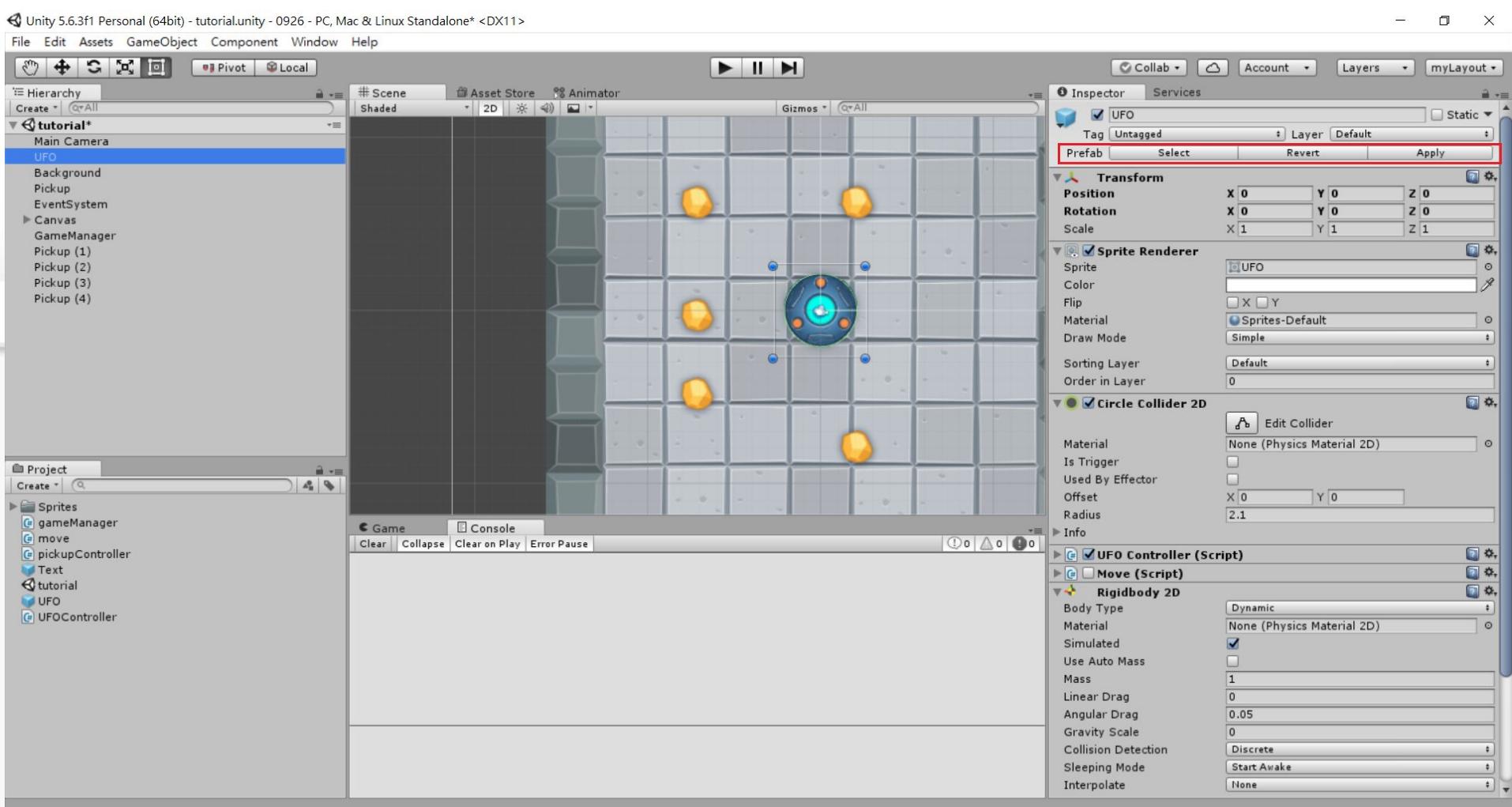
- 接著就可以將UFO移到金塊處讓金塊消失

Prefab (預設體)

- 預先設定屬性的物件
- 用於生成固定性質的固定物件
- How
 - 將場景中的某一物件拖曳至Project欄位
 - 被你拖曳到project的物件會變得藍色的，代表他成為了prefab，之後你對該物件的修改，可以透過inspector欄位上方的prefab row進行設置
- 往後可以直接將帶有固定屬性的物件直接(從Project tab)拉入場景



iNTERACTIVE
MEDIA



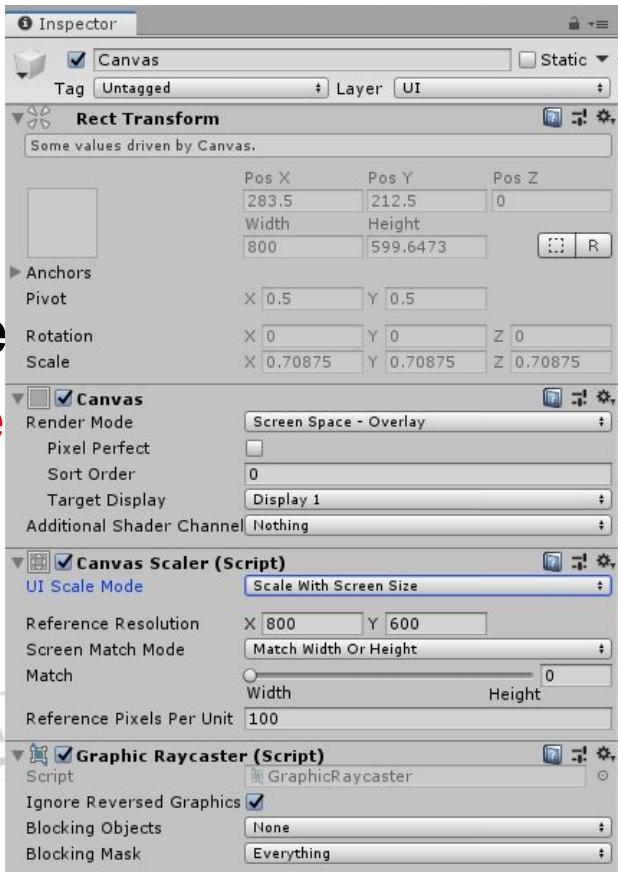
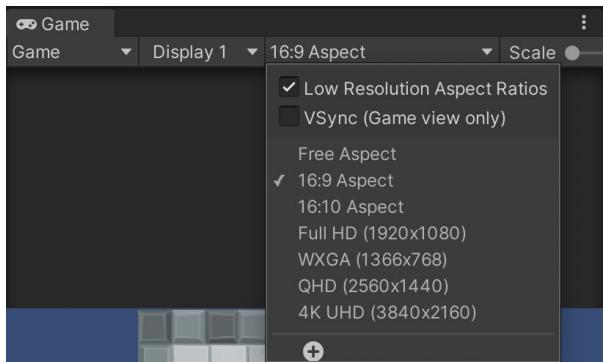
iNTERACTIVE
MEDIa

實作時間(5min)

1. 加入金塊
2. 新增pickupController.cs
3. 設定金塊物理特性 - Collider
4. 新增UFO撿金塊函式
5. 設定Prefab

紀錄分數 - UI & Text

- Game Tab的比例調成16:9
- 建立Text
 - GameObject→UI→Text
 - 產生的同時會自動建立Canvas
- 更改UI Scale Mode
 - Canvas Scalar(Script) → UI Scale Mode換成 Scale with Screen Size



No UI->Text

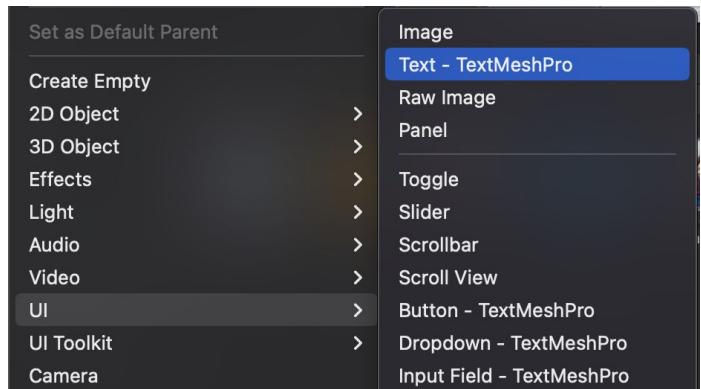
unity 2021 Its only UI-> Text - TextMeshPro

* simple solution
add

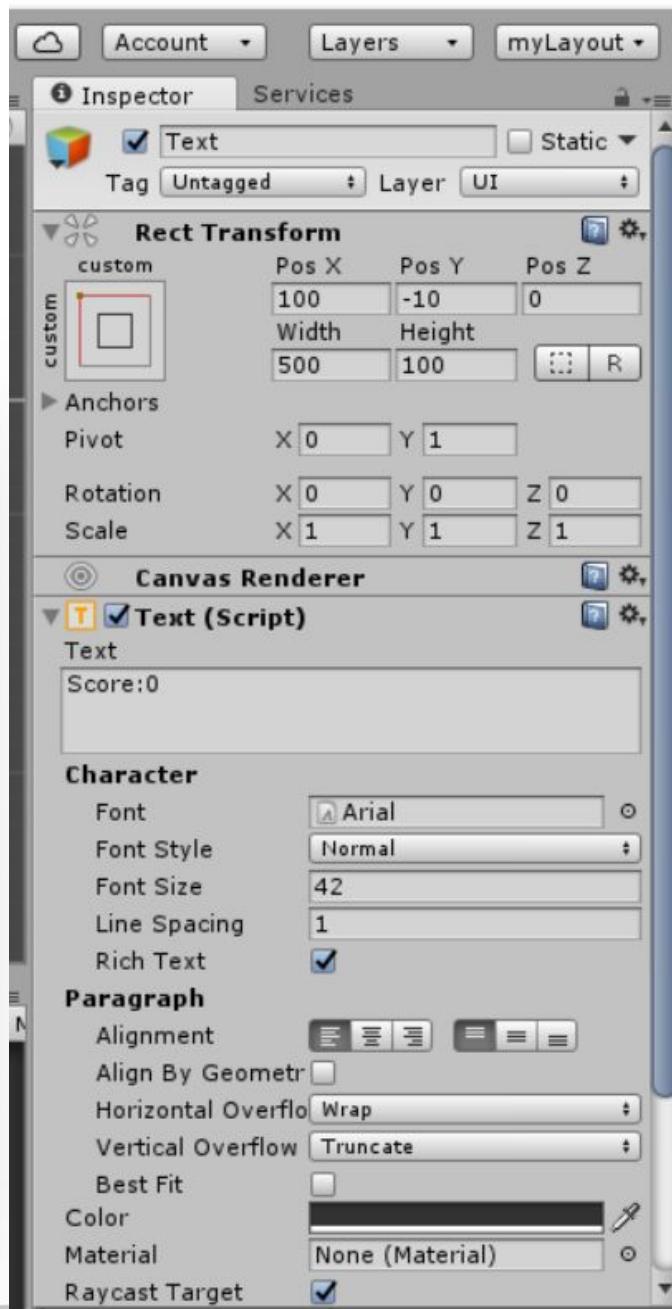
```
using UnityEngine;  
  
using UnityEngine.UI;
```

```
using Text = TMPro.TextMeshProUGUI;
```

```
public class ScoreTextController : MonoBehaviour {  
  
    public int initialScore = 0;
```

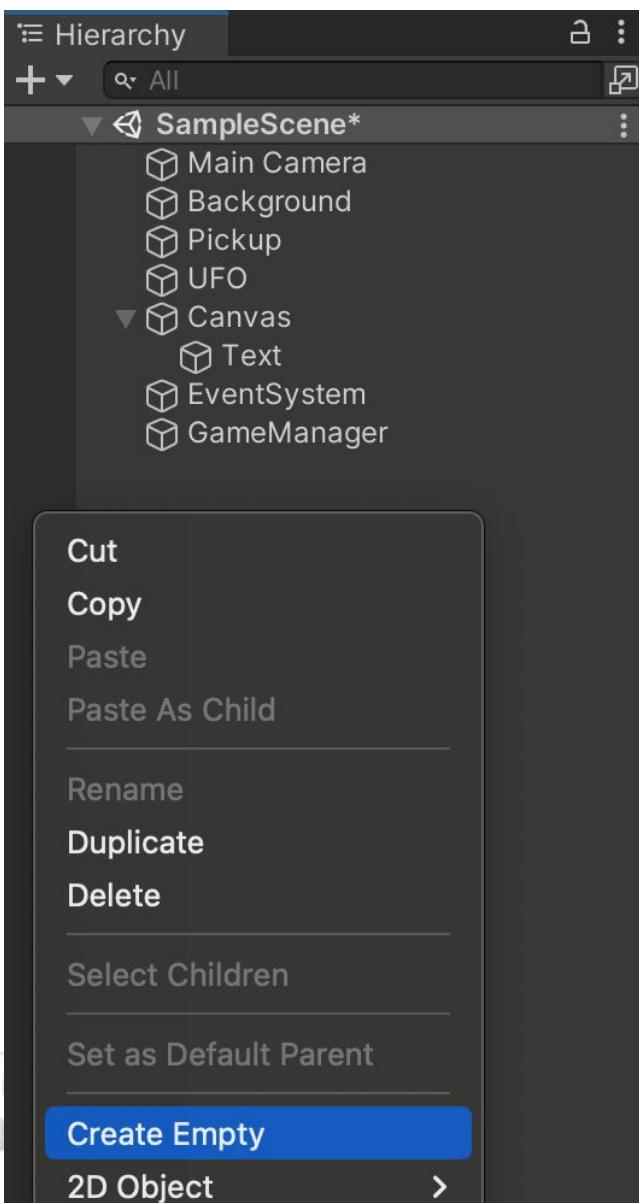


Text的設定

VE
MEDIA

分數與遊戲的連結

- 在hierarchy建立一個空物件
 - CreateEmpty 命名為 GameManager
- 在GameManager加上script
 - 「gameManager.cs」



流程控制 gameManager.cs

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class gameManager : MonoBehaviour
7  {
8      public Text scoreText;
9      int score;
10     // Use this for initialization
11
12     void Start()
13     {
14         score = 0;
15
16         // Update is called once per frame
17         void Update()
18         {
19
20             public void addScore(int n)
21             {
22                 score += n;
23                 scoreText.text = "Score: " + score;
24             }
25     }
```

回editor將剛剛建立的Text拉到gameManager的script的scoreText上

ERACTIVE
MEDIA

UFOController.cs修改

- 碰到金塊加分
 - 先前只有寫碰到金塊，金塊消失
 - 要呼叫執行碰到金塊加分函式
- 在OnTriggerEnter2D函數中加入這行內容
 - GameObject.Find("GameManager").GetComponent<gameManager>().addScore(100);

```
void OnTriggerEnter2D(Collider2D other)
{
    Destroy(other.gameObject);

    GameObject.Find("GameManager").GetComponent<gameManager>().addScore(100);
}
```

實作時間 (5min)

1. 加入UI - Text
2. 設定Canvas與Text
3. 建立分數與遊戲的連結-GameManager
4. 寫GameManager的腳本
5. 修改觸發事件UFOController

RUN!

- 測試自己的UFO是否可以觸碰到金塊並觸發加分的機制



iNTERACTIVE
MEDIa

GameManager

gameManager

```
int score;  
public Text scoreText;  
  
public void addScore(int n)
```

Canvas

Text
char text;

PickUp

ufoController

UFO

OnTriggerEnter2D(Collider2D other)

```
{  
    Destroy(other.gameObject);  
    GameObject.Find("GameManager").GetComponent<gameManager>().addScore(100);  
}
```

作業

- 加入背景
- 加入Player並且使其可以有物理性質的移動
- 物體被觸碰後消失
- 加入計分功能
- 做完練習之後可以使用螢幕錄影將影片交到moodle上(影片要錄到的內容包括Player物理性移動，觸碰物體後物體消失並且UI計分發生改變)



以下正課已介紹

INTERACTIVE MEDIA

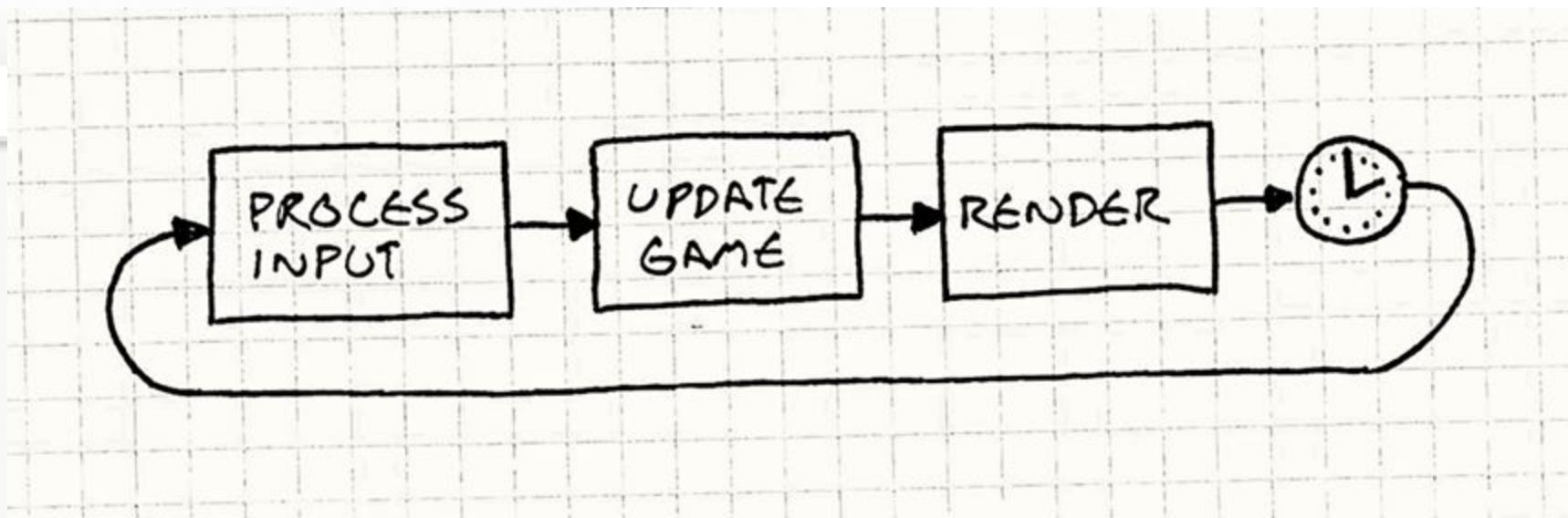
Unity的概念講解

iNTERACTIVE
MEDIa

Script-基本函式

函數名	被呼叫時機
Awake()	在腳本(Script)檔被創建並載入場景的時候呼叫。
Start()	在創建後的下一個Update()被呼叫前執行。
Update()	每個禎(frame)更新時呼叫。
FixedUpdate()	在固定的時間呼叫。
OnGUI()	用於處理GUI事件，每個禎(Frame)更新時執行。
OnCollisionEnter()	碰撞體(Collider)/剛體(Rigidbody)和其他的碰撞體(Collider)/剛體(Rigidbody)接觸的一開始執行。
OnCollisionExit()	當碰撞體(Collider)/剛體(Rigidbody)和其他的碰撞體(Collider)/剛體(Rigidbody)終止接觸的瞬間執行。
OnCollisionStay()	當碰撞體(Collider)/剛體(Rigidbody)和其他的碰撞體(Collider)/剛體(Rigidbody)碰撞時持續執行
OnTriggerEnter()	當本身碰撞體進入觸發器(Trigger)時執行。
OnTriggerExit()	當本身碰撞體離開觸發器(Trigger)的瞬間執行。
OnTriggerStay()	本身碰撞體與觸發器(Trigger)持續接觸的情況下，每個禎(Frame)執行。

Game loop

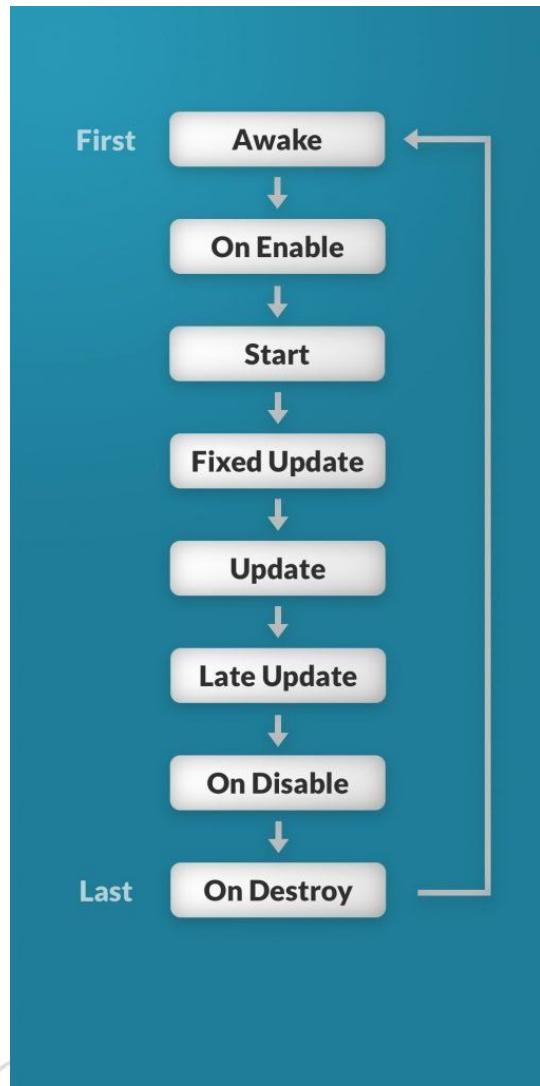


<http://gameprogrammingpatterns.com/game-loop.html>

INTERACTIVE
MEDIA

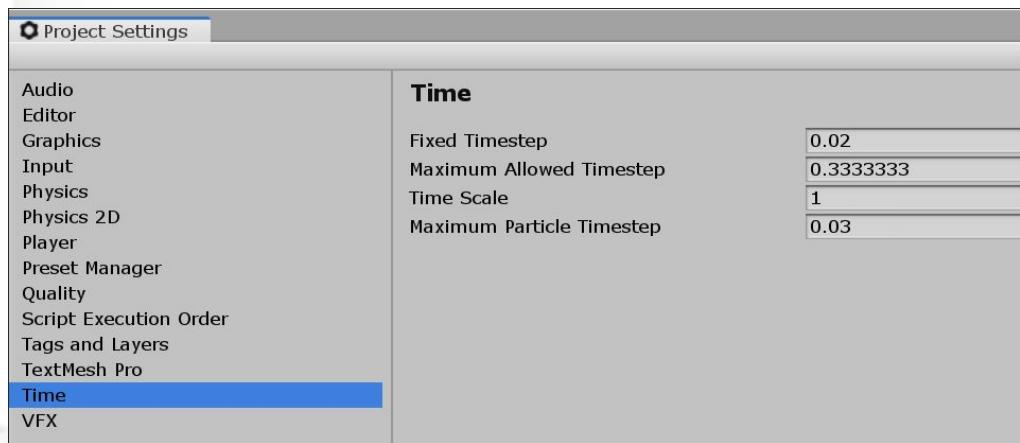
Awake VS Start

- Awake():
 - 遊戲開始後(play鍵以後), 每一個物件在其**active**的那一刻就會執行這個函式(有些物件可能一開始不是active的, 在其轉變成active狀態的時候會立即執行awake)
- Start():
 - 物件在其**active**「後」, 遊戲進行的下一幀(frame)前會呼叫此函式



Update VS FixedUpdate

- **Update():**
 - 在遊戲進行的每一偵會執行一次(執行的次數受到電腦實際的速度影響)
- **FixedUpdate():**
 - 固定的時間會執行一次, 時間可以在下面做更動
 - Edit → Project settings → Time → Fixed Timestep

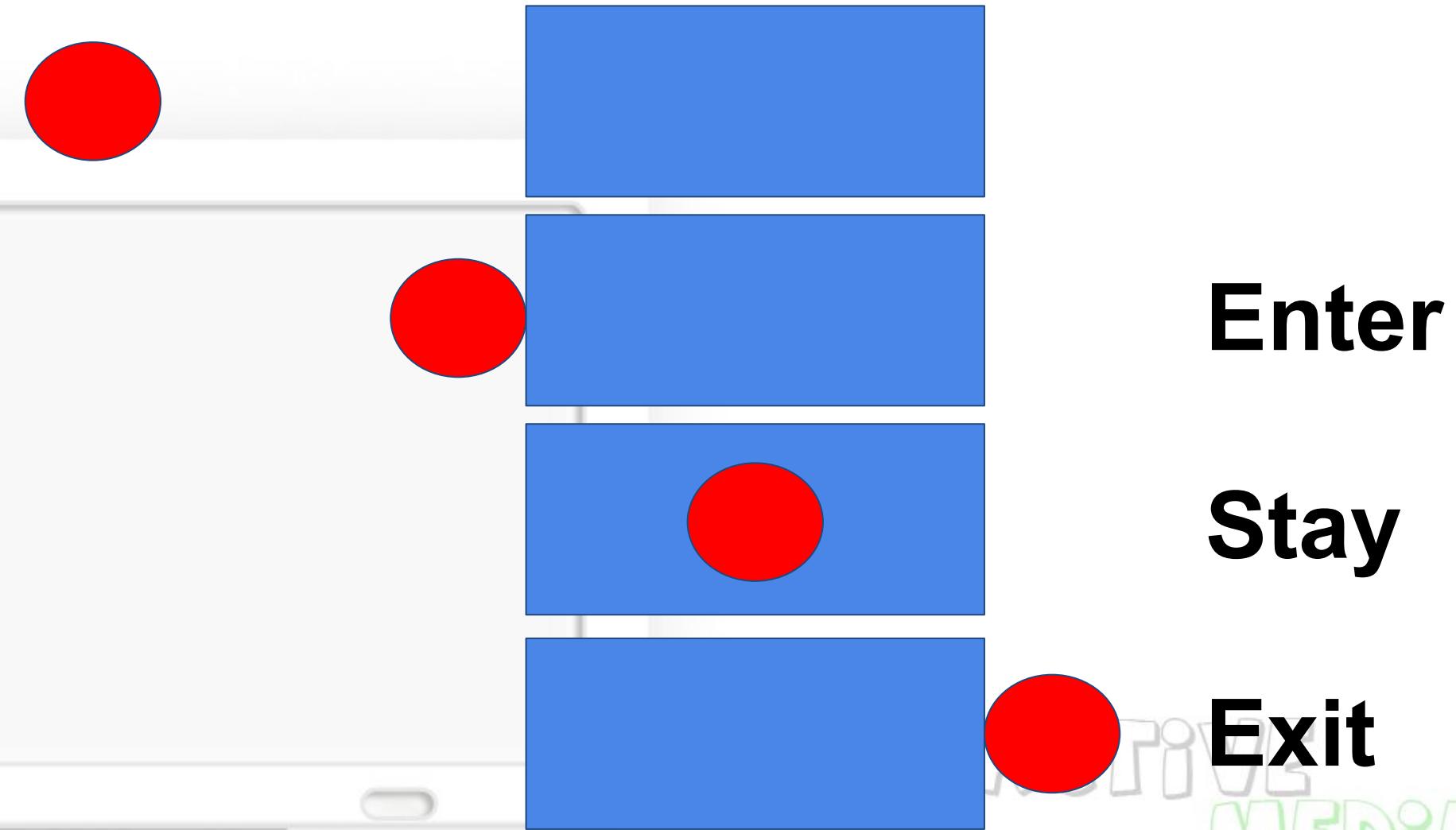


physic2D

motion simulation + collision

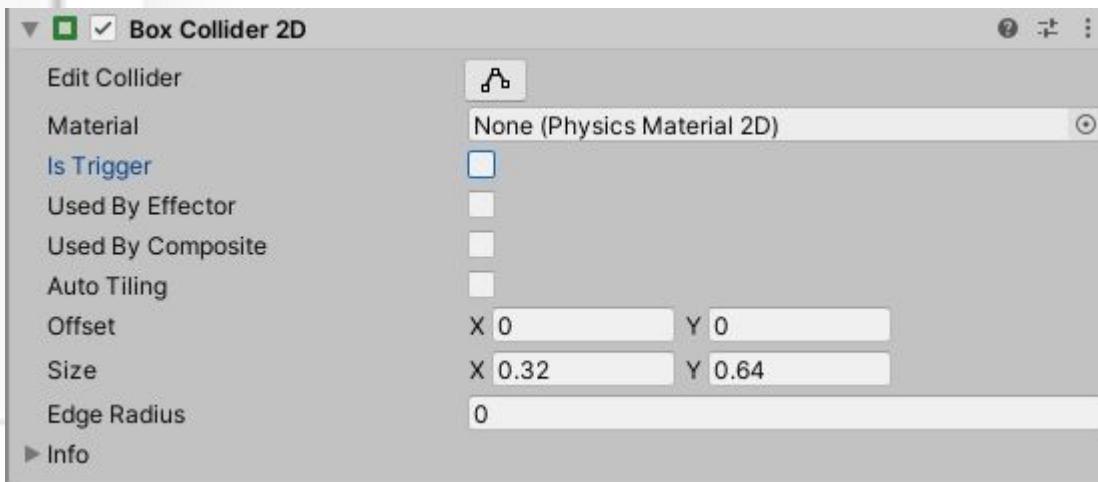
INTERACTIVE MEDIA

OnCollisionEnter/Stay/Exit



Collision(碰撞)與Trigger(觸發)

- **碰撞**: 兩個物件產生物理碰撞，會觸發
OnCollisionEnter/Stay/Exit函數
- **觸發**: 取消所有物理碰撞，兩物件接觸時會觸發
OnTriggerEnter/Stay/Exit
- 換言之兩物件接觸時**一定不可能同時觸發**Collision和Trigger signal



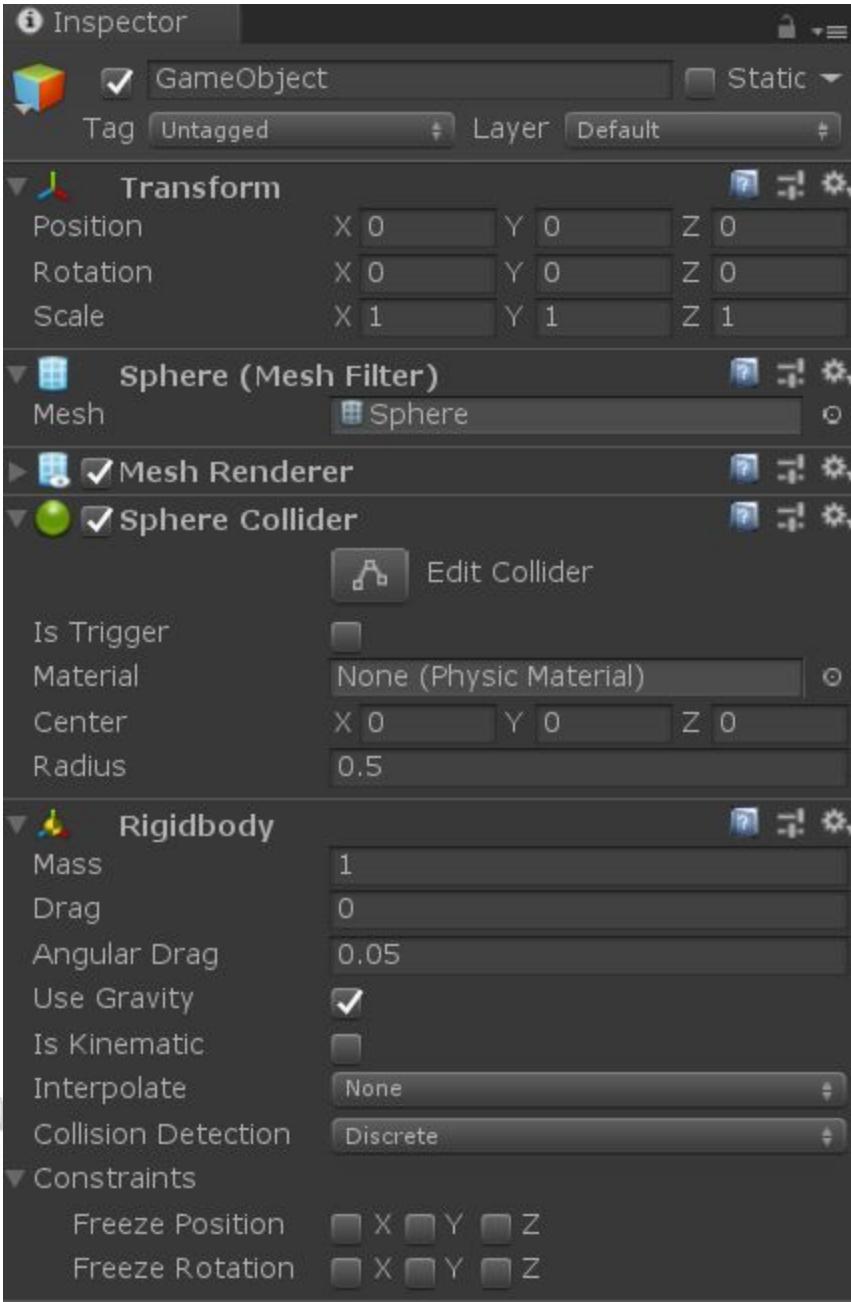
physics2D

static: 未加入rigidbody

Rigidbody: 允許物理運算。Ex:受力(重力、阻力、慣性)和速度。
(add component->Rigidbody)

Kinematic: 不使用物理運算，但可對其他剛體進行物理運算。
(Rigidbody->click 'Is Kinematic')

備忘有reference連結
有興趣自己看>_O



Collision(碰撞)與Trigger(觸發)

- 六種碰撞器的分別

Static Collider

Rigidbody Collider

Kinematic Rigidbody Collider

- 左半邊代表collider在接觸時會產生碰撞訊號

Static Trigger Collider

Rigidbody Trigger Collider

Kinematic Rigidbody Trigger Collider

- 右半邊Trigger代表在collider的屬性中勾選了is trigger, 使得該碰撞器不會發生碰撞, 而會被穿過, 被其他物件接觸時產生觸發訊號

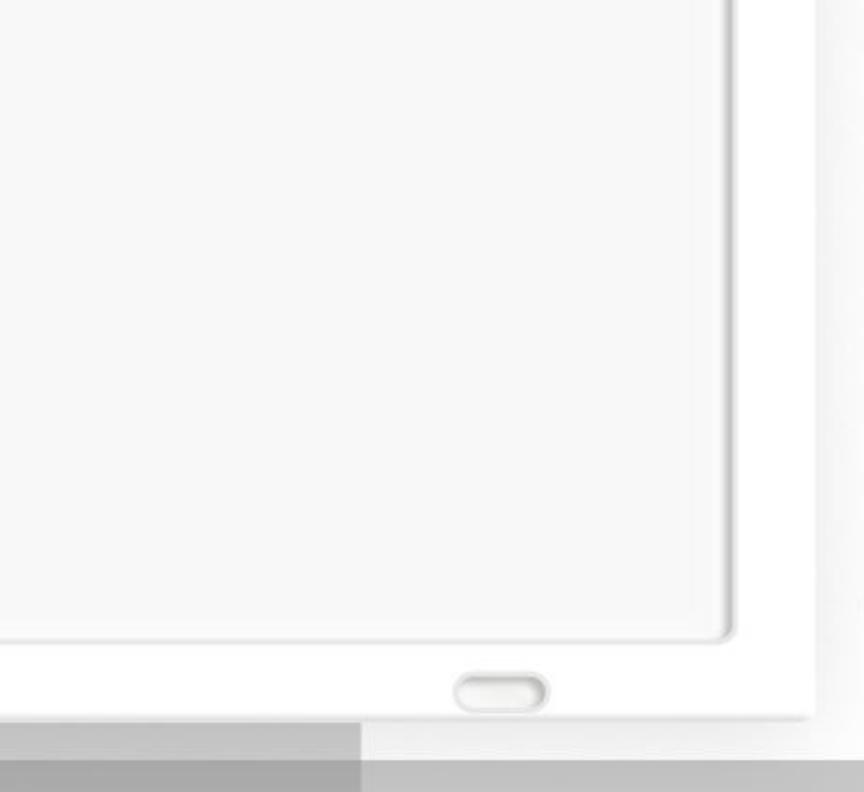
宣告

- public, static
 - int a; //a只會在該腳本被使用、修改
 - public int a; // a可以在editor的介面中直接使用與修改
 - public type function(type argv...) {} //可在其他類別取用
 - public static a; //假設宣告a的腳本為abc.cs, a可以在其他腳本中寫abc.a 來使用與修改
- Random ← 寫遊戲肯定會用到
 - float a = Random.value; // between 0 ~ 1
 - float a = Random.Range(1.0f , 3.0f);
- Float型態的變數後面的 f 是必須的
 - float a = 2.5; //error
 - float a = 2.5f; //OK

INTERACTIVE
MEDIA

Script - the most important

- <https://docs.unity3d.com/ScriptReference/>
- 寫程式基本上不太可能所有method都會
- 不會的、不確定的都可以去上面查



iNTERACTIVE
MEDIa