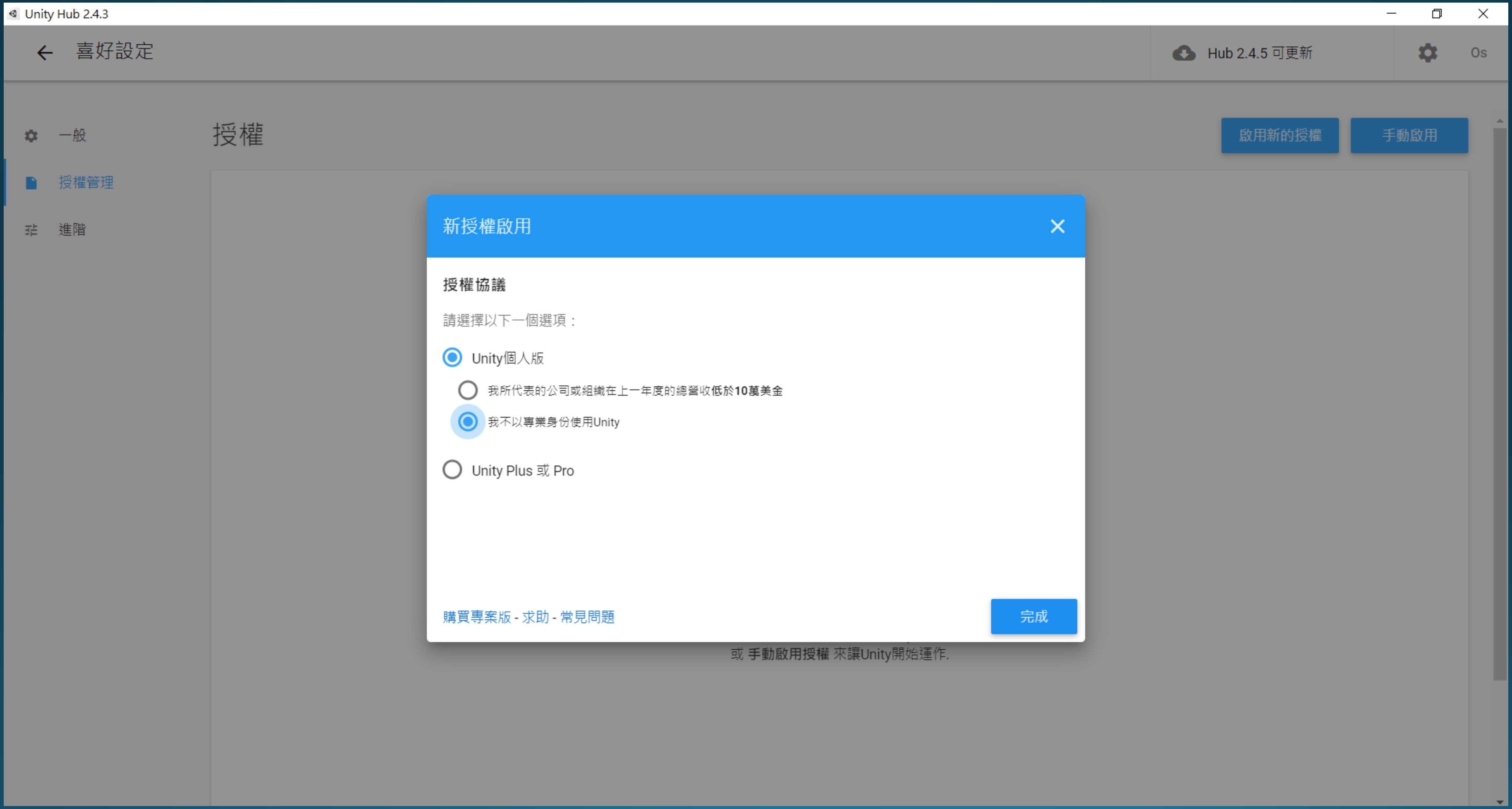
2D遊戲開發

恐龍遊戲

啟用授權



新授權啟用

授權協議

請選擇以下一個選項：

● Unity個人版

- ○ 我所代表的公司或組織在上一年度的總營收**低於10萬美金**
- ● 我不以專業身份使用Unity

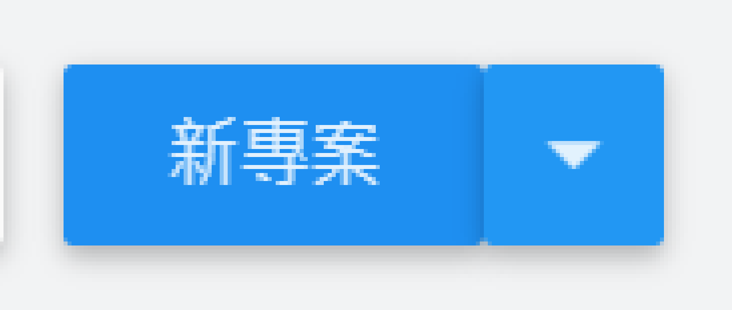
○ Unity Plus 或 Pro

購買專案版 - 求助 - 常見問題

完成

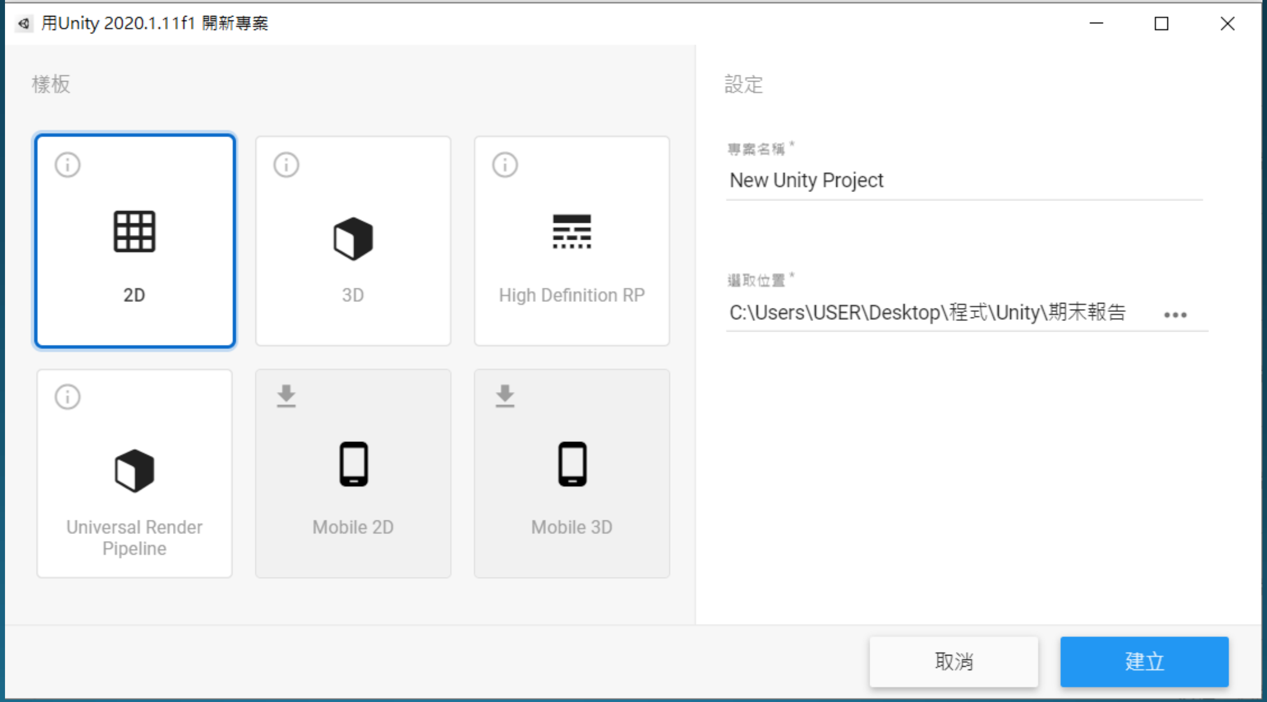
或 手動啟用授權 來讓Unity開始運作.

建立專案



點選右上新專案

選擇2D，並建立


用Unity 2020.1.11f1 開新專案

樣板

設定

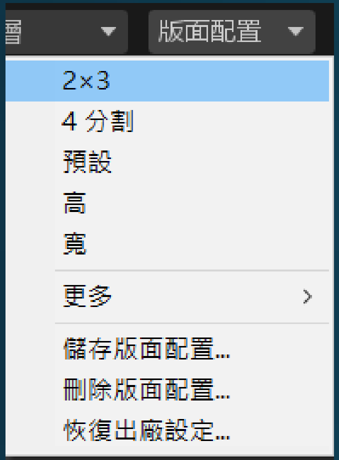
專案名稱 *

New Unity Project

選取位置 *

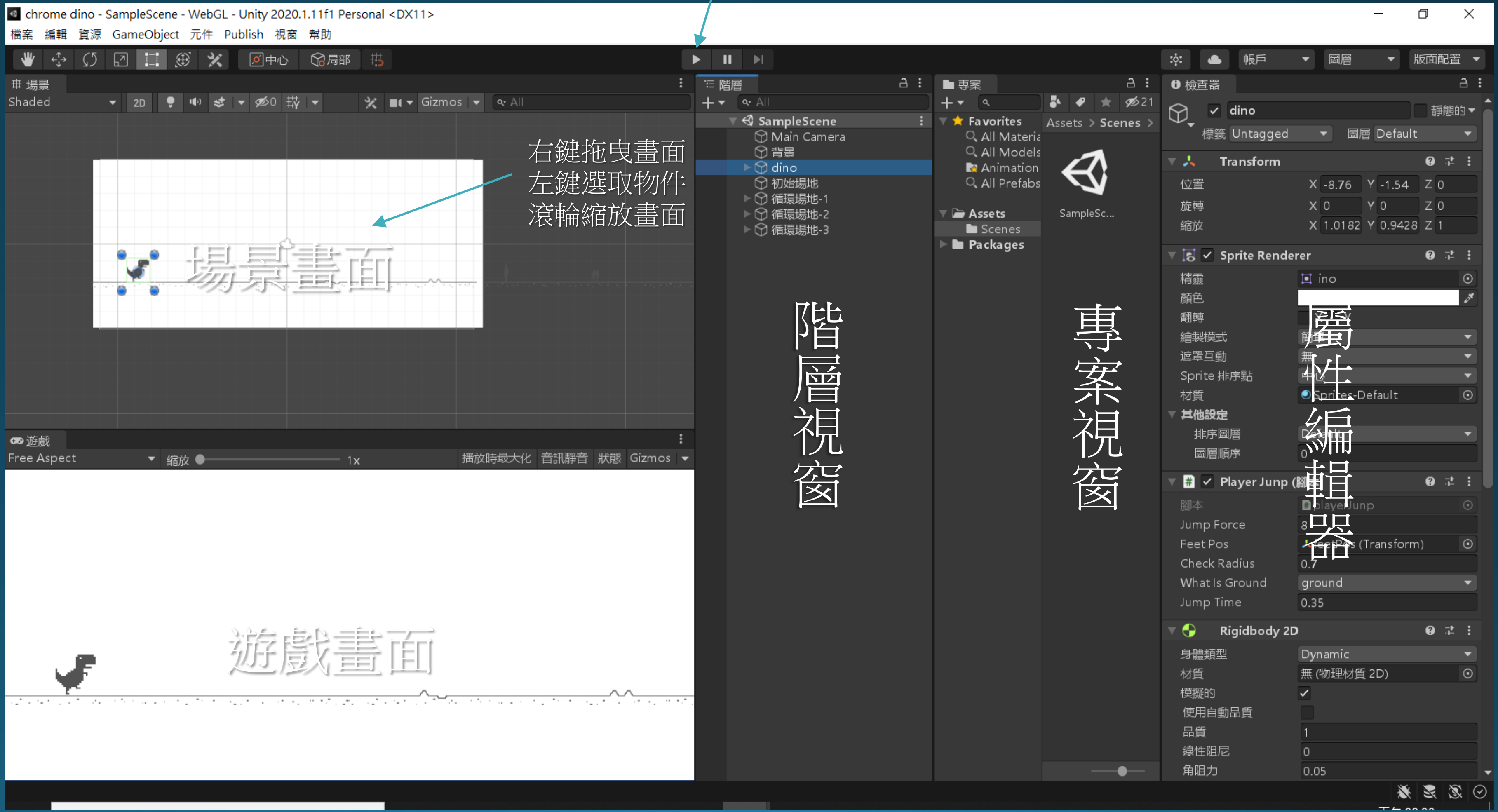
C:\Users\USER\Desktop\程式\Unity\期末報告

基本介面認識



點選右上版面配置
選擇2x3

執行程式



dino 靜態的
標籤 Untagged 圖層 Default

Transform
位置 X -8.76 Y -1.54 Z 0
旋轉 X 0 Y 0 Z 0
縮放 X 1.0182 Y 0.9428 Z 1

Sprite Renderer
精靈 ino
顏色
翻轉
繪製模式
遮罩互動
Sprite 排序點
材質 Sprites-Default
其他設定
排序圖層
圖層順序 0

Player Jump
腳本 playerJunp
Jump Force 8
Feet Pos (Transform)
Check Radius 0.7
What Is Ground ground
Jump Time 0.35

Rigidbody 2D
身體類型 Dynamic
材質 無 (物理材質 2D)
模擬的 ☑
使用自動品質 □
品質 1
線性阻尼 0
角阻力 0.05

右鍵拖曳畫面
左鍵選取物件
滾輪縮放畫面

場景畫面

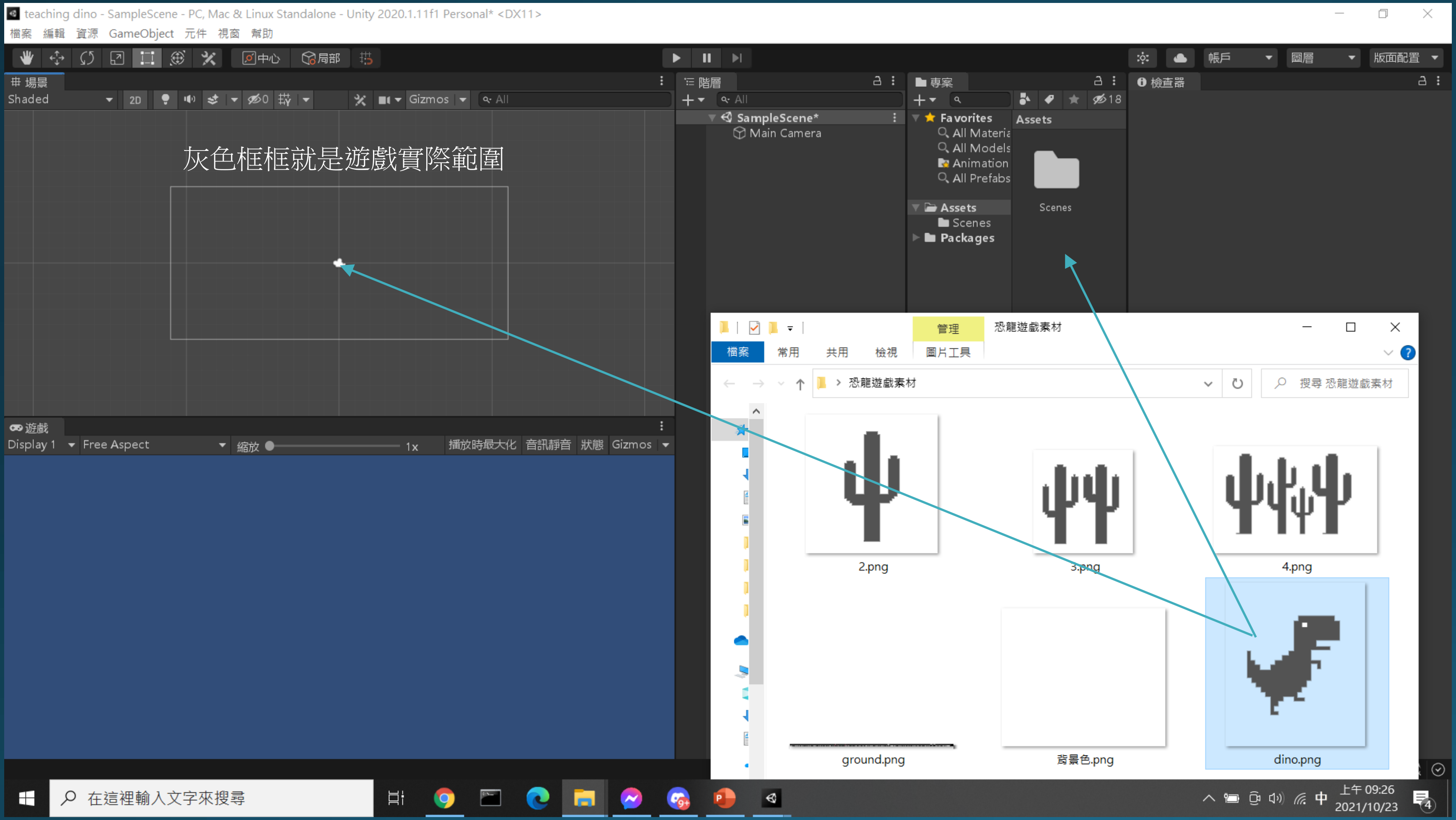
遊戲畫面

階層視窗

專案視窗

屬性編輯器

創建物件

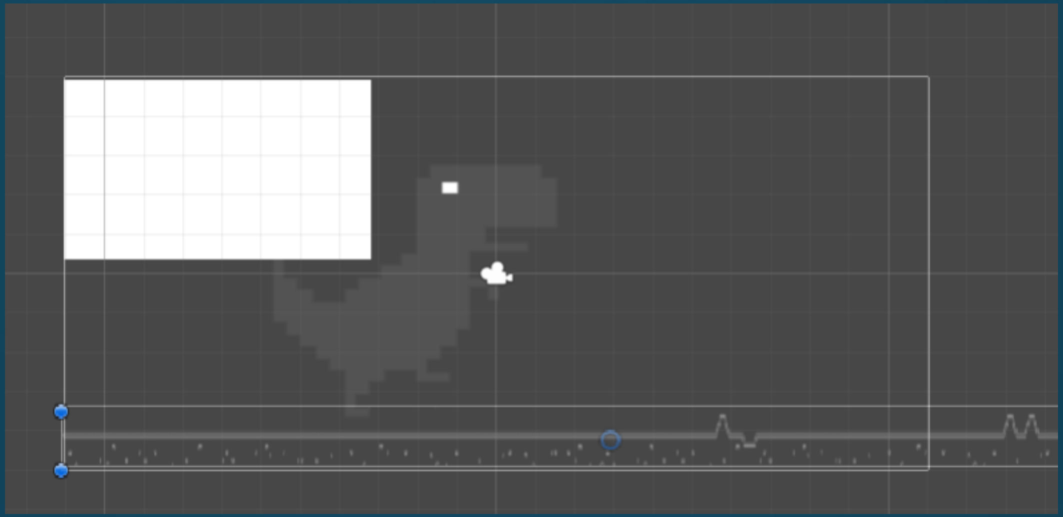


灰色框框就是遊戲實際範圍

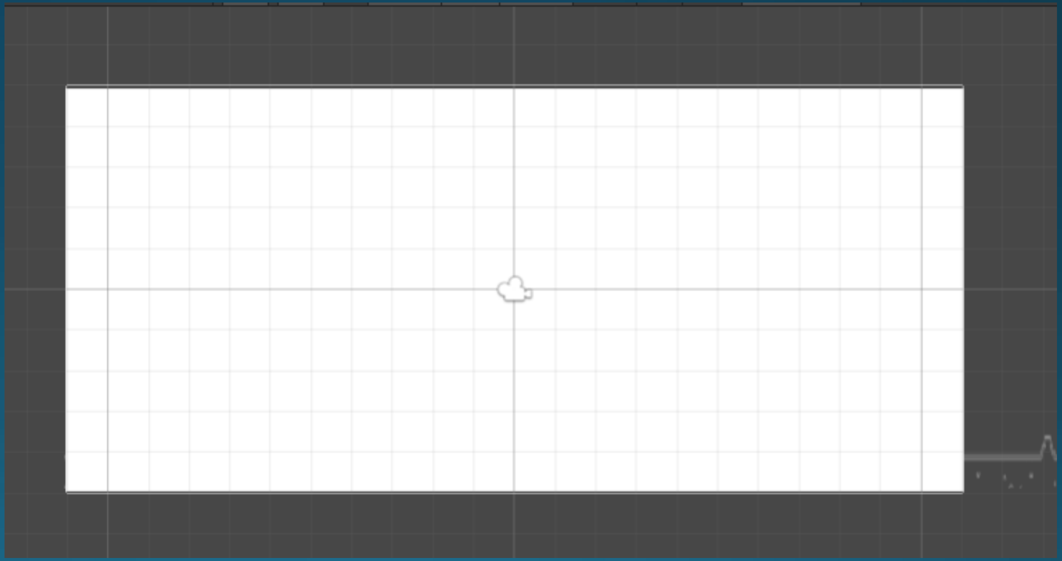
如果要創建物件可以將圖片拖入場景畫面或專案視窗中

背景建立

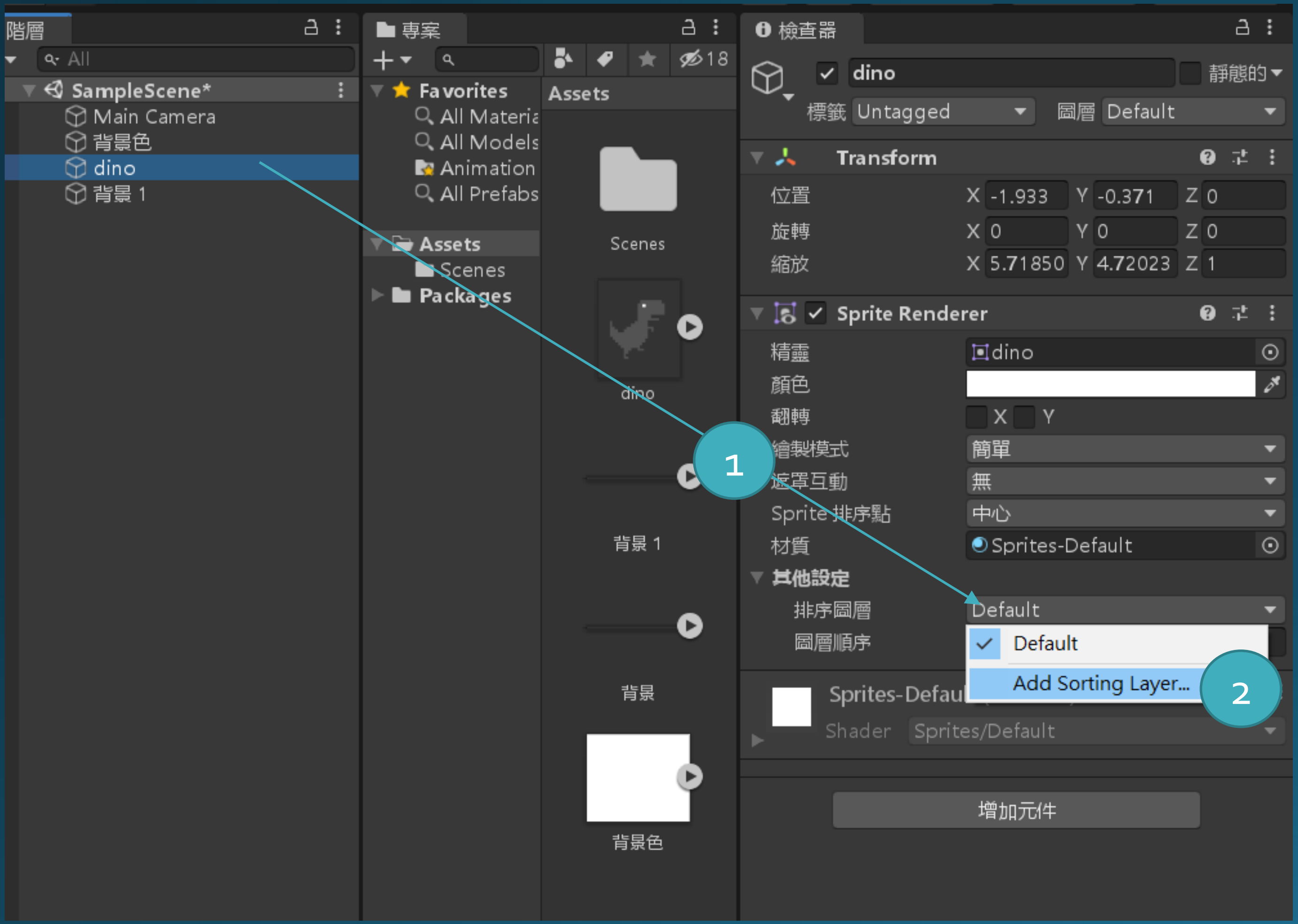
將背景和背景色拉進unity



再將背景色填滿灰色框框
這時會發現背景色把所有東西蓋住了



調整圖層

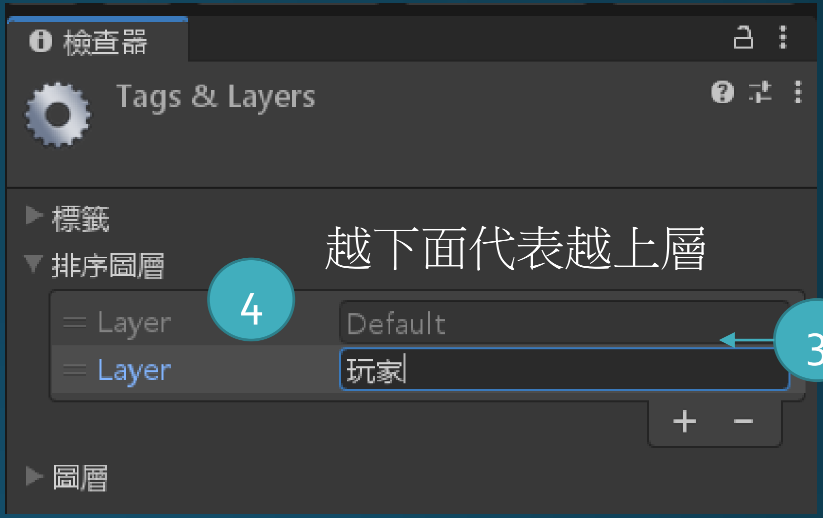


dino 靜態的
標籤 Untagged 圖層 Default

Transform
位置 X -1.933 Y -0.371 Z 0
旋轉 X 0 Y 0 Z 0
縮放 X 5.71850 Y 4.72023 Z 1

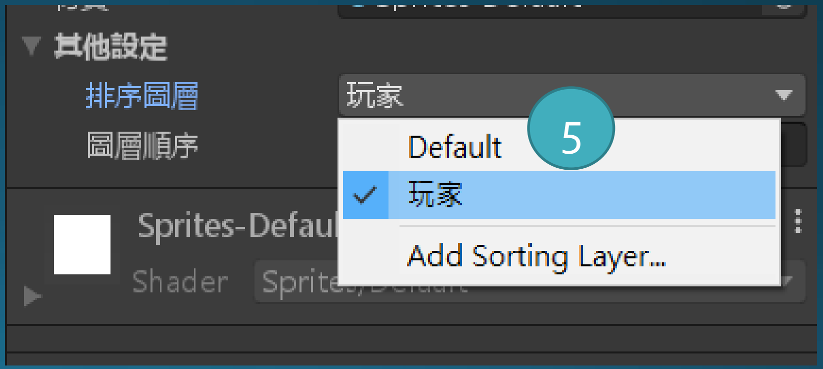
Sprite Renderer
精靈 dino
顏色
翻轉 X Y
繪製模式 簡單
遮罩互動 無
Sprite 排序點 中心
材質 Sprites-Default
其他設定
排序圖層 Default
圖層順序
✓ Default
Add Sorting Layer...

Sprites-Defau
Shader Sprites/Default



越下面代表越上層

Layer Default
Layer 玩家

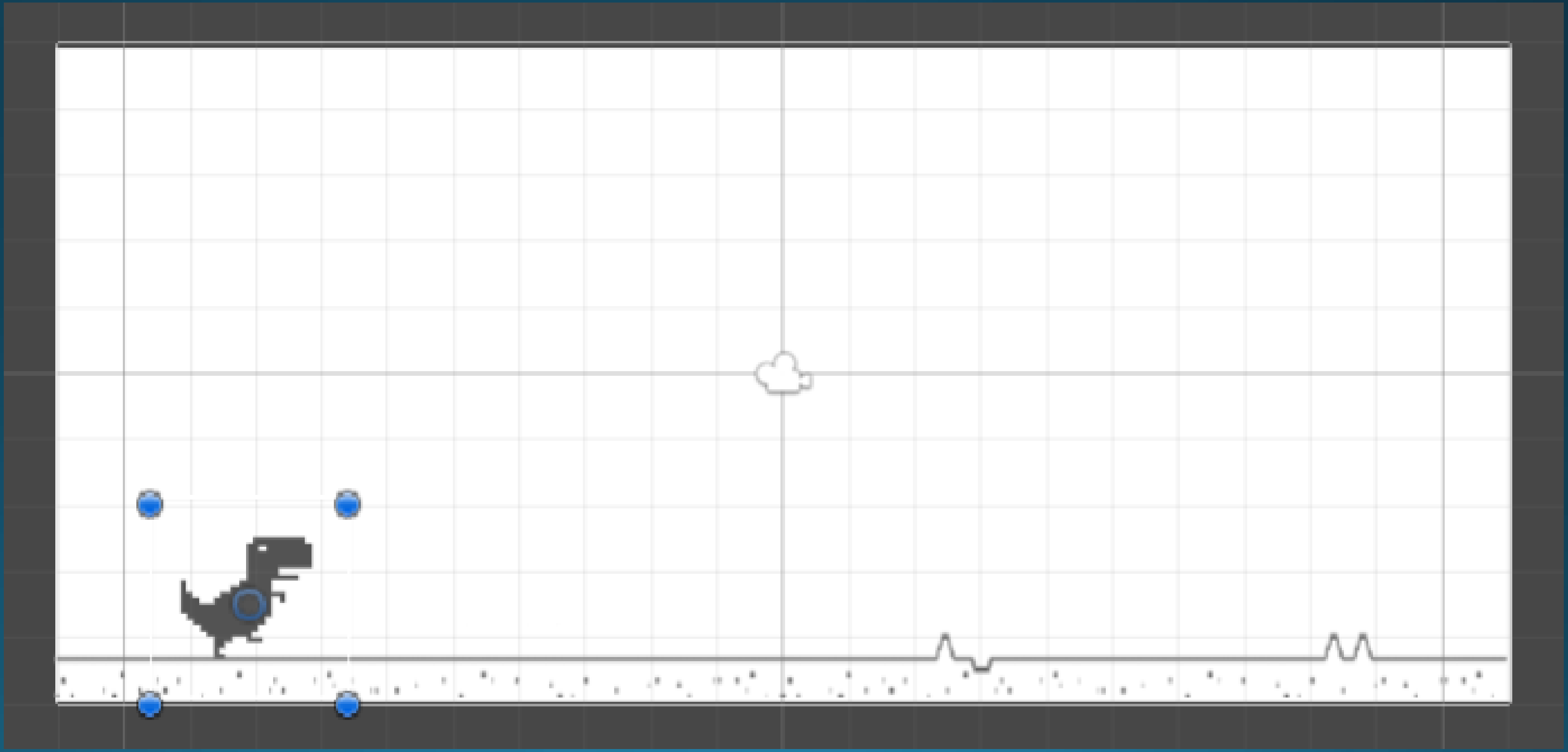


其他設定
排序圖層 玩家
圖層順序
Default
✓ 玩家
Add Sorting Layer...

Sprites-Defau
Shader Sprites/Default

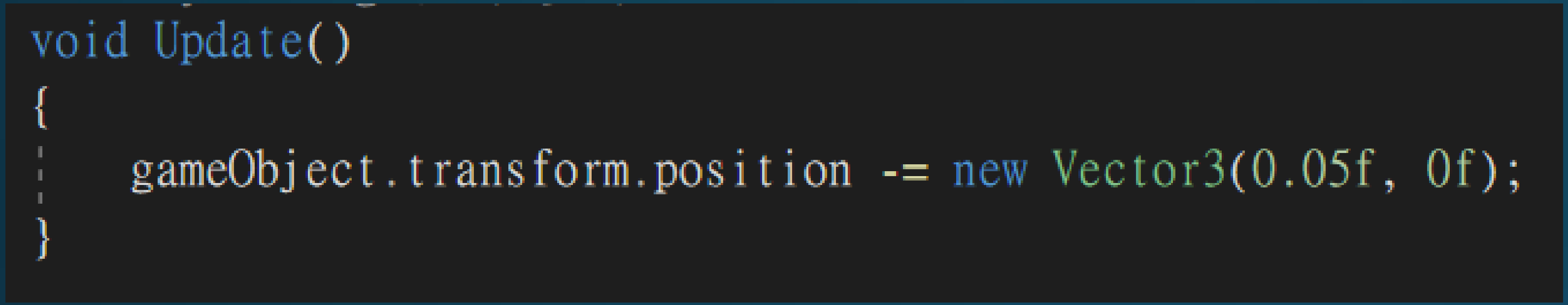
練習

新增一個背景圖層讓背景移至最上層



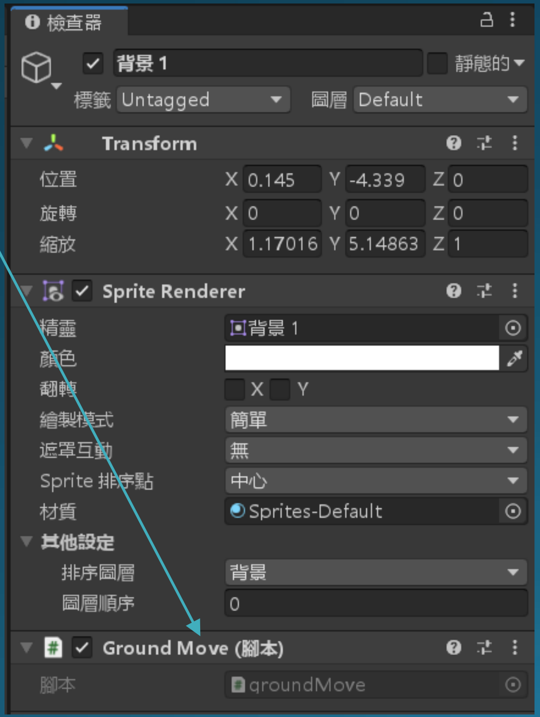
背景移動

專案視窗->Assets右鍵->建立->C#腳本(名字用英文)->點兩下打開



```
void Update()
{
    gameObject.transform.position -= new Vector3(0.05f, 0f);
}
```

將腳本拖進背景裡面



☑ 背景 1 □ 靜態的

標籤 Untagged 圖層 Default

Transform

位置 X 0.145 Y -4.339 Z 0

旋轉 X 0 Y 0 Z 0

縮放 X 1.17016 Y 5.14863 Z 1

☑ Sprite Renderer

精靈 背景 1

顏色

翻轉 □ X □ Y

繪製模式 簡單

遮罩互動 無

Sprite 排序點 中心

材質 Sprites-Default

其他設定

排序圖層 背景

圖層順序 0

☑ Ground Move (腳本)

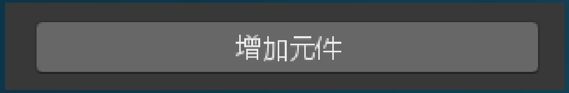
腳本 groundMove

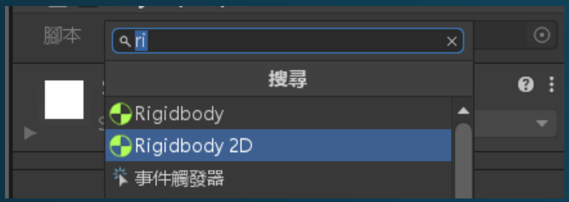
多個場景移動練習

讓多個背景可以一直循環向左

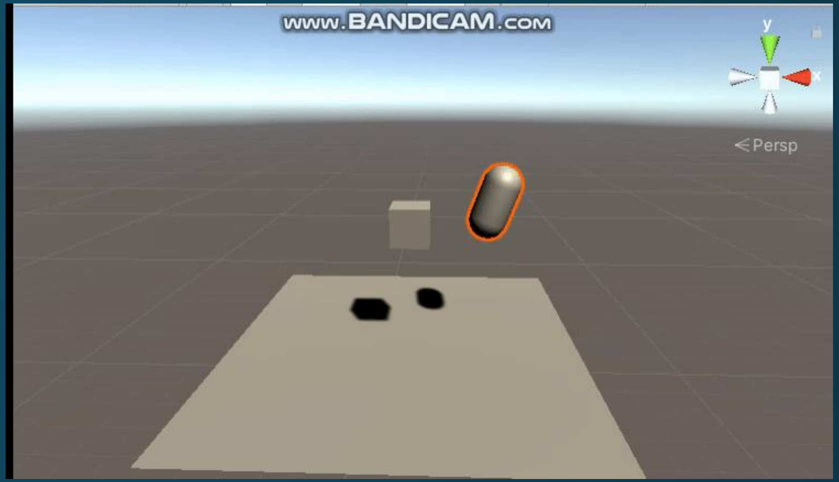
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ground_manage : MonoBehaviour
{

    public bool start_ground;

    void Update()
    {
        gameObject.transform.position -= new Vector3(0.05f, 0f, 0f);
        if (gameObject.transform.position.x <= -23f)
        {
            if(start_ground == true)
            {
                gameObject.SetActive(false);
            }
            else
            {
                gameObject.transform.position = new Vector3(46.64f, -2.25f, 0f);
            }
        }
    }
 }
```

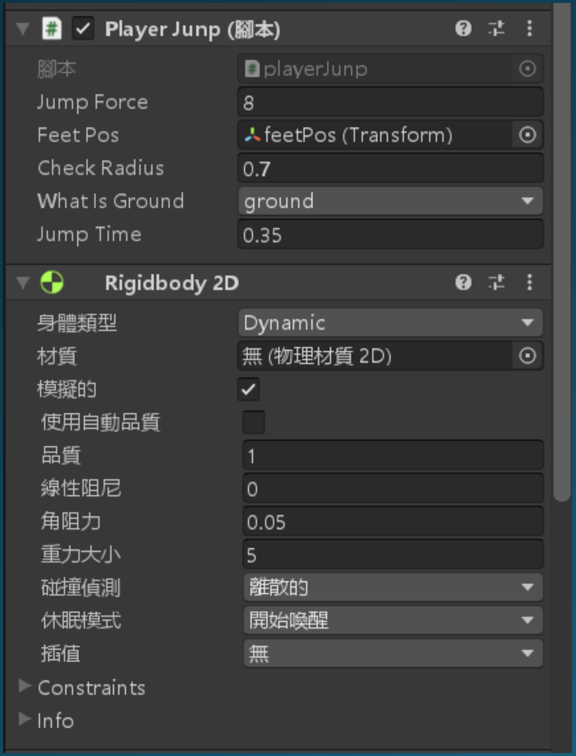
控制恐龍





Rigidbody2D是unity內建的物件物理效果





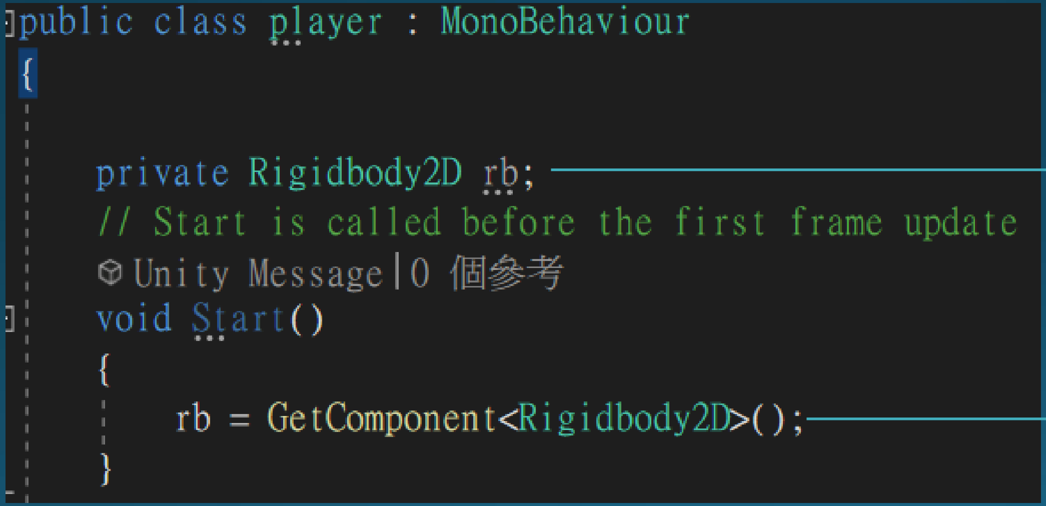
Player Junp (腳本)

腳本	playerJunp
Jump Force	8
Feet Pos	feetPos (Transform)
Check Radius	0.7
What Is Ground	ground
Jump Time	0.35

Rigidbody 2D

身體類型	Dynamic
材質	無 (物理材質 2D)
模擬的	☑
使用自動品質	□
品質	1
線性阻尼	0
角阻力	0.05
重力大小	5
碰撞偵測	離散的
休眠模式	開始喚醒
插值	無

- Constraints
- Info



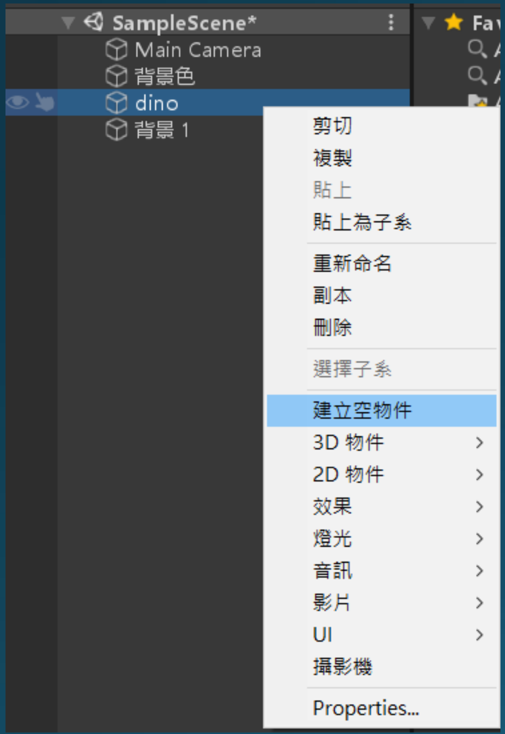
```
public class player : MonoBehaviour
{

    private Rigidbody2D rb;
    // Start is called before the first frame update
    Unity Message|0 個參考
    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
    }
```

建立一個Rigidbody2D的變數rb

將Rigidbody2D導入rb

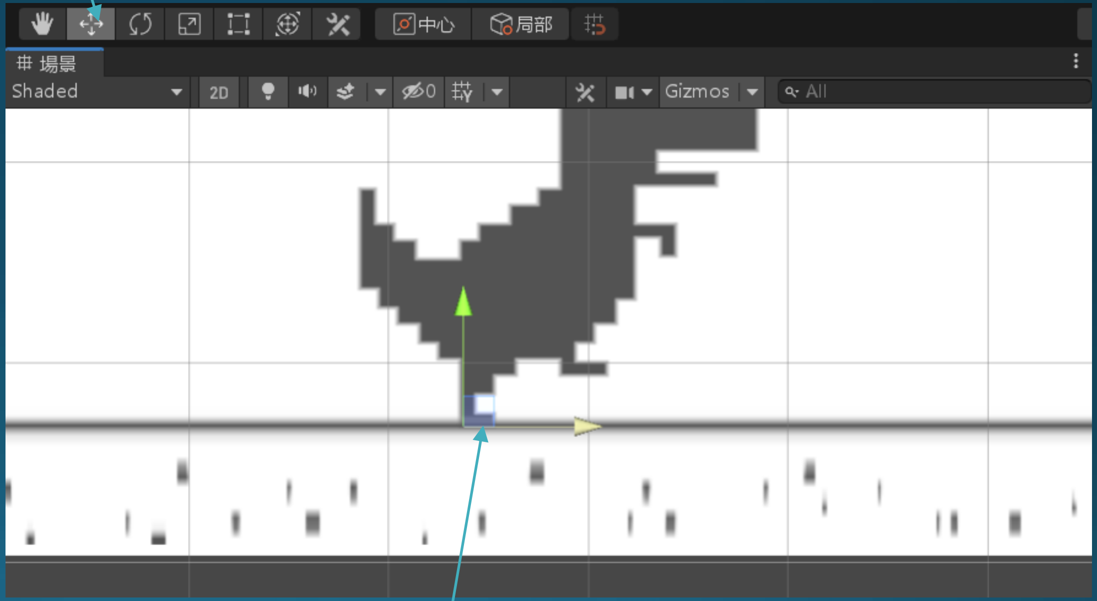
控制恐龍

新增一個空物件來偵測是否碰到地面



點選恐龍
選擇建立空物件

選擇

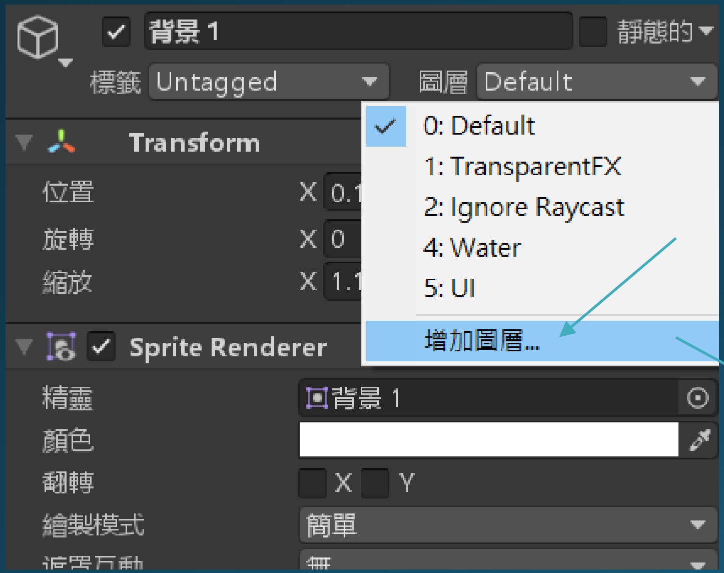


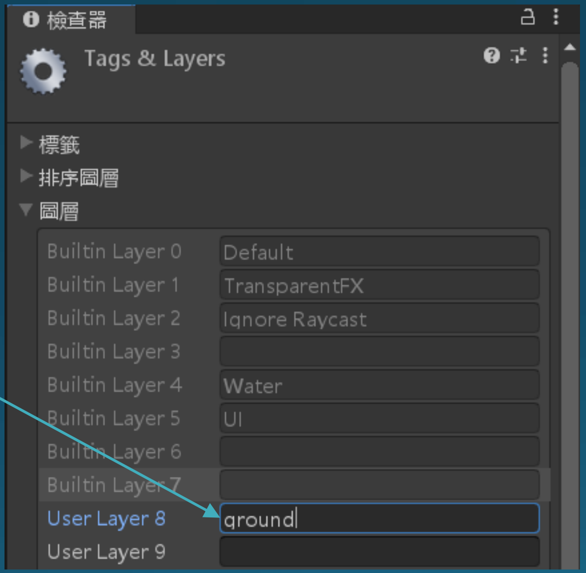
把空物件拖到恐龍腳下

控制恐龍

新增一個地面分類

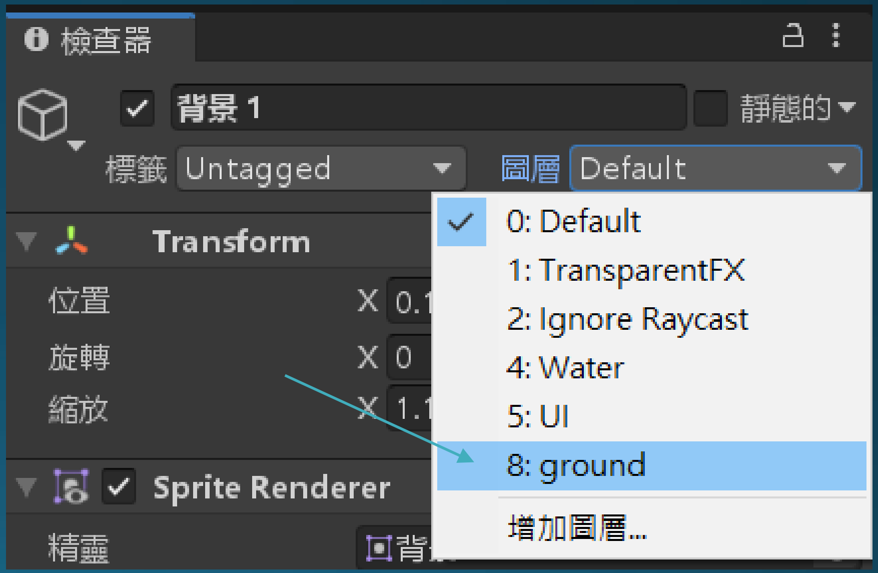
選擇背景





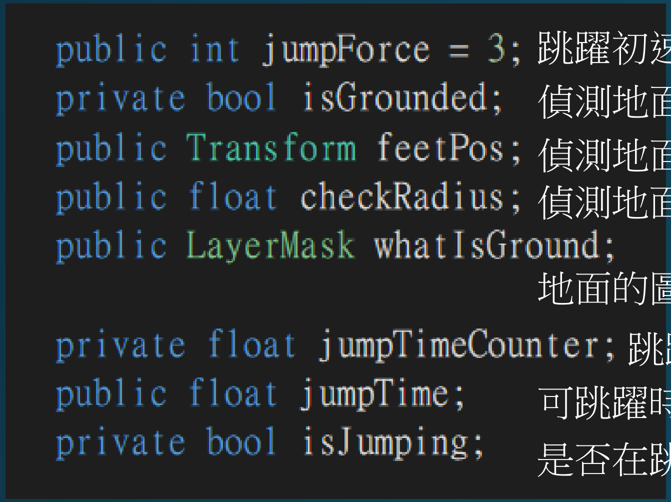
Builtin Layer 0 Default
Builtin Layer 1 TransparentFX
Builtin Layer 2 Ignore Raycast
Builtin Layer 3
Builtin Layer 4 Water
Builtin Layer 5 UI
Builtin Layer 6
Builtin Layer 7
User Layer 8 ground
User Layer 9

將背景的圖層改為ground



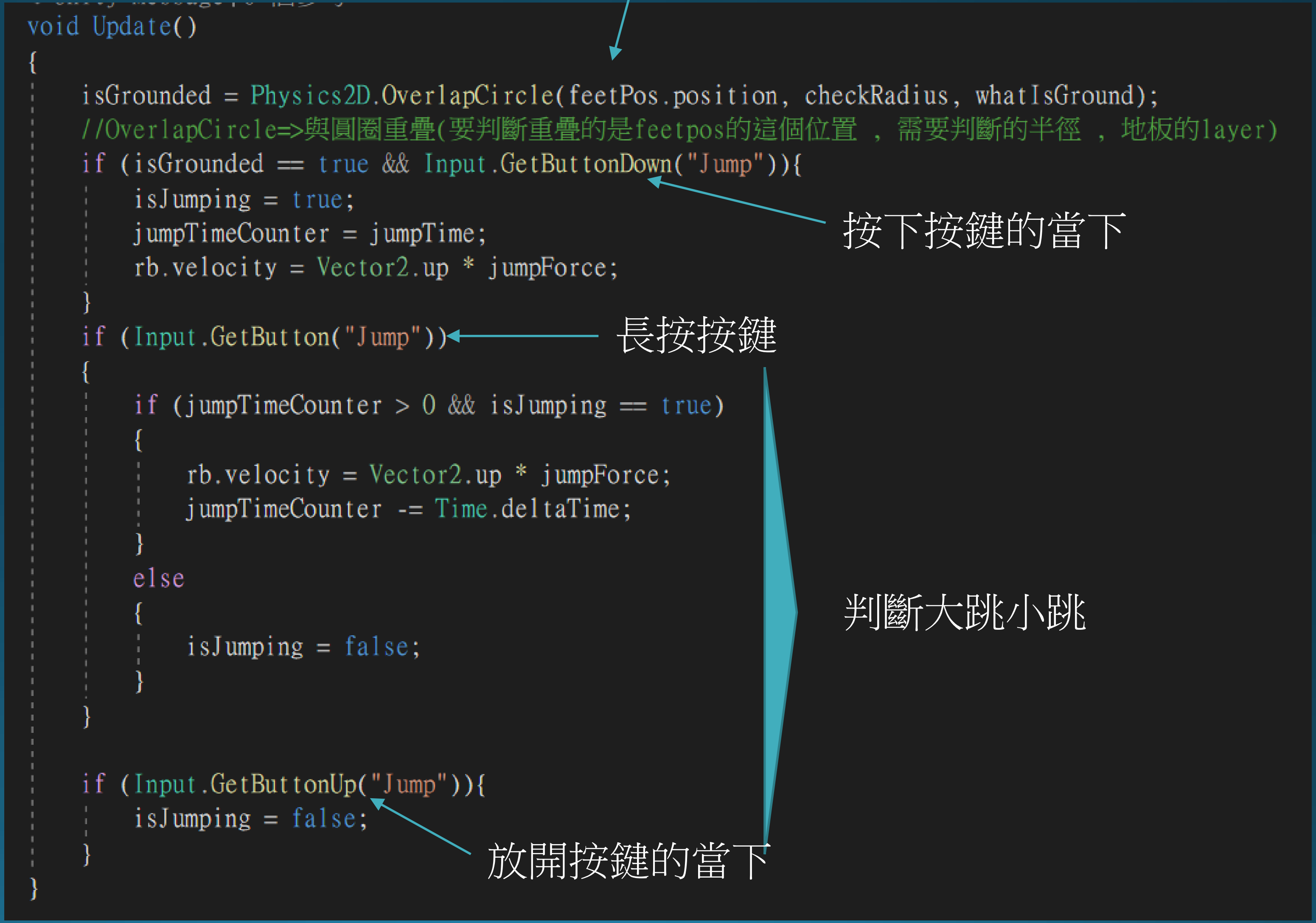
控制恐龍

讓恐龍跳



```
public int jumpForce = 3; 跳躍初速度
private bool isGrounded;  偵測地面
public Transform feetPos; 偵測地面的物件
public float checkRadius;  偵測地面的半徑
public LayerMask whatIsGround;
                          地面的圖層

private float jumpTimeCounter; 跳躍時間計時
public float jumpTime;     可跳躍時間
private bool isJumping;    是否在跳躍
```

偵測空物件的位置有沒有碰到地面



```
void Update()
{
    isGrounded = Physics2D.OverlapCircle(feetPos.position, checkRadius, whatIsGround);
    //OverlapCircle=>與圓圈重疊(要判斷重疊的是feetpos的這個位置 , 需要判斷的半徑 , 地板的layer)
    if (isGrounded == true && Input.GetButtonDown("Jump")){
        isJumping = true;
        jumpTimeCounter = jumpTime;
        rb.velocity = Vector2.up * jumpForce;
    }
    if (Input.GetButton("Jump"))
    {
        if (jumpTimeCounter > 0 && isJumping == true)
        {
            rb.velocity = Vector2.up * jumpForce;
            jumpTimeCounter -= Time.deltaTime;
        }
        else
        {
            isJumping = false;
        }
    }

    if (Input.GetButtonUp("Jump")){
        isJumping = false;
    }
}
```

按下按鍵的當下

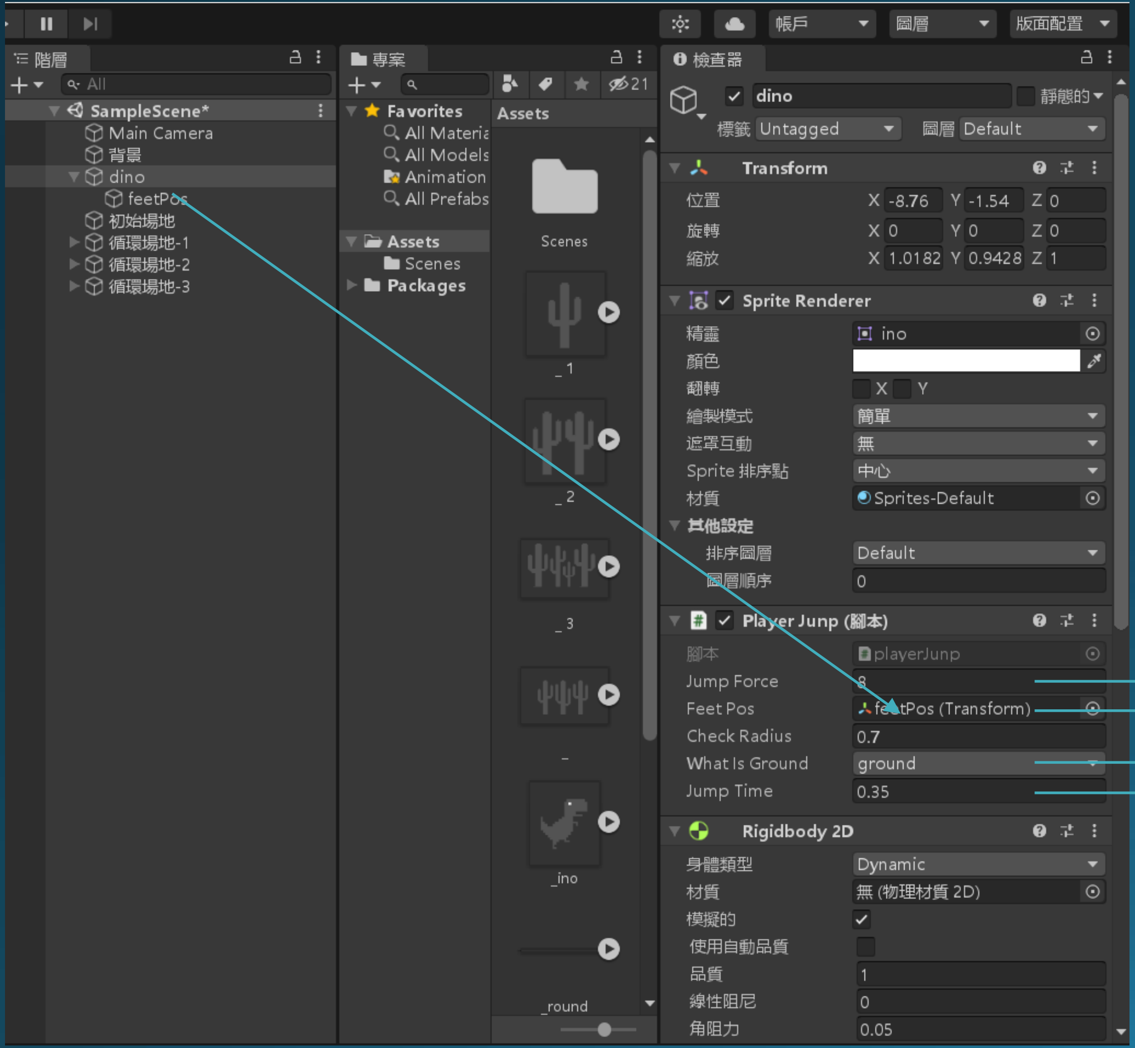
長按按鍵

判斷大跳小跳

放開按鍵的當下

控制恐龍

設定數值



dino 靜態的
標籤 Untagged 圖層 Default

Transform
位置 X -8.76 Y -1.54 Z 0
旋轉 X 0 Y 0 Z 0
縮放 X 1.0182 Y 0.9428 Z 1

☑ **Sprite Renderer**
精靈 ino
顏色
翻轉 □ X □ Y
繪製模式 簡單
遮罩互動 無
Sprite 排序點 中心
材質 Sprites-Default
其他設定
排序圖層 Default
圖層順序 0

☑ **Player Junp (腳本)**
腳本 playerJunp
Jump Force 8
Feet Pos feetPos (Transform)
Check Radius 0.7
What Is Ground ground
Jump Time 0.35

Rigidbody 2D
身體類型 Dynamic
材質 無 (物理材質 2D)
模擬的 ☑
使用自動品質 □
品質 1
線性阻尼 0
角阻力 0.05

跳躍初速度

將偵測地面的物件托進來

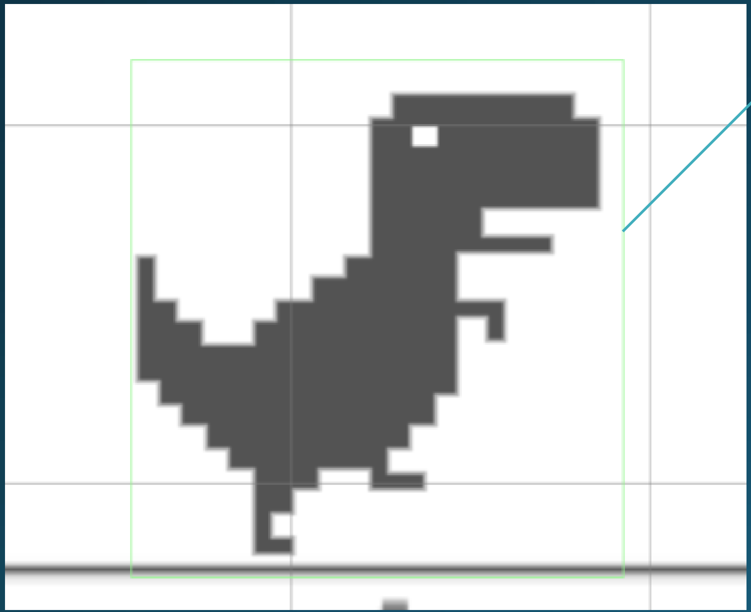
選擇地面的圖層

可跳躍時間

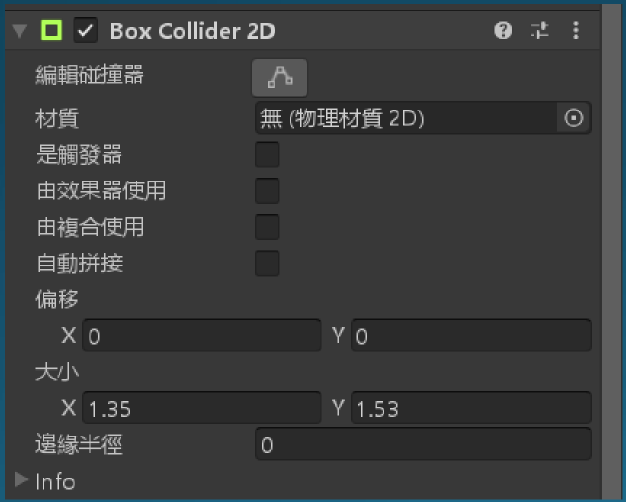
碰撞設定

現在執行後會發現恐龍會直接掉下去
這是因為我們設定什麼是地面後，卻沒告訴恐龍遇到地面可以在上面跑

恐龍、地面->增加元件->Box Collider 2D



綠色框框即是碰撞範圍



☑ Box Collider 2D

編輯碰撞器

材質 無 (物理材質 2D)

是觸發器 □

由效果器使用 □

由複合使用 □

自動拼接 □

偏移 X 0 Y 0

大小 X 1.35 Y 1.53

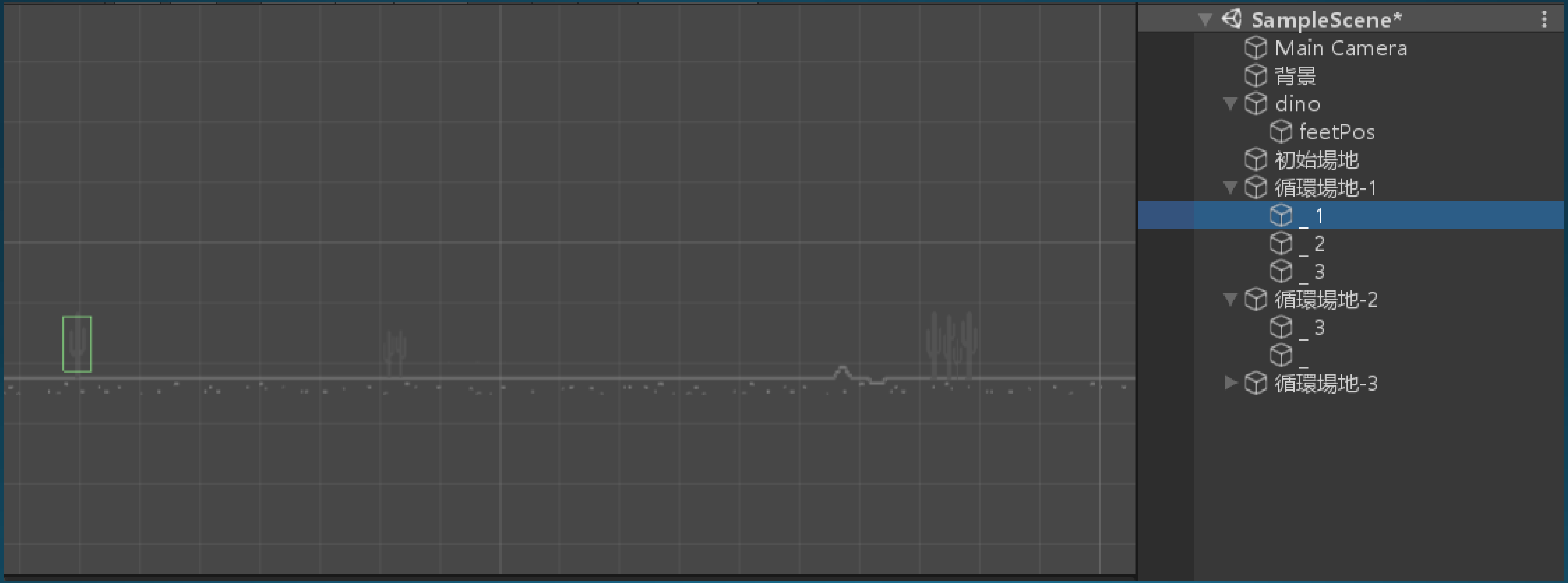
邊緣半徑 0

▶ Info

由這裡來更改碰撞範圍

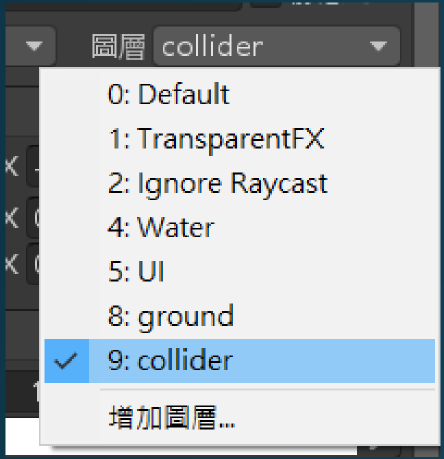
練習

新增仙人掌到背景上，並將仙人掌裝上碰撞器(觸發器要打勾)

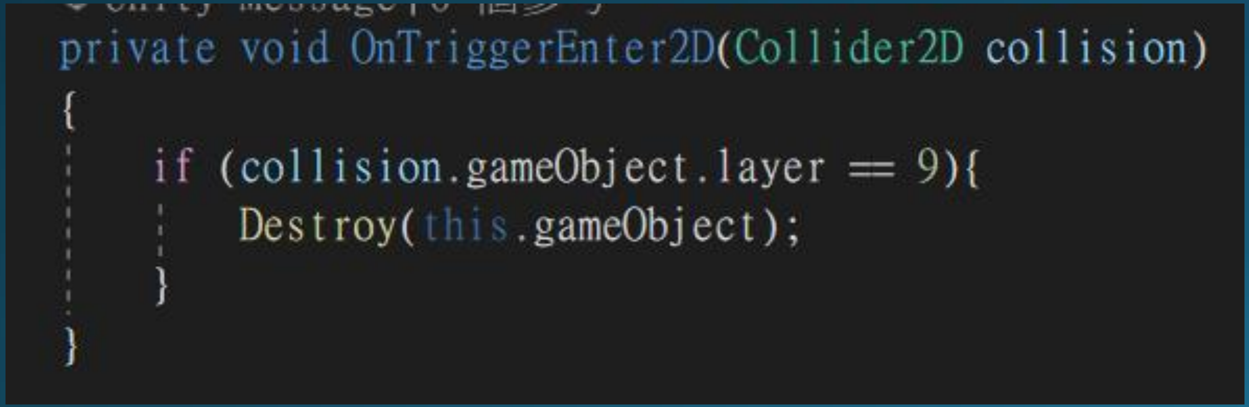


碰撞設定

如果撞到仙人掌，恐龍消失



新增仙人掌的圖層



```
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.gameObject.layer == 9){
        Destroy(this.gameObject);
    }
}
```

偵測碰撞

如果撞到的物件是屬於編號9(collider)
銷毀恐龍

控制恐龍完整code

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

public class playerJunp : MonoBehaviour
{
    private Rigidbody2D rb;

    public int jumpForce = 3;
    private bool isGrounded;
    public Transform feetPos;
    public float checkRadius;
    public LayerMask whatIsGround;

    private float jumpTimeCounter;
    public float jumpTime;
    private bool isJumping;

    // Start is called before the first frame update
    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
    }
```

控制恐龍完整code

```
void Update()
    {
        isGrounded = Physics2D.OverlapCircle(feetPos.position, checkRadius, whatIsGround);
        //OverlapCircle=>與圓圈重疊(要判斷重疊的是feetpos的這個位置 ， 需要判斷的半徑 ， 地板的layer)
        if (isGrounded == true && Input.GetButton("Jump")){
            isJumping = true;
            jumpTimeCounter = jumpTime;
            rb.velocity = Vector2.up * jumpForce;
        }
        if (Input.GetButton("Jump"))
        {
            if (jumpTimeCounter > 0 && isJumping == true)
            {
                rb.velocity = Vector2.up * jumpForce;
                jumpTimeCounter -= Time.deltaTime;
            }
            else
            {
                isJumping = false;
            }
        }

        if (Input.GetButtonUp("Jump")){
            isJumping = false;
        }
    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.layer == 9){
            Destroy(this.gameObject);
        }
    }
}
```