

Funglus



幾乎不需要寫程式，就能做出有分支的對話系統

適合製作**點擊式的 冒險遊戲** 或 **文字冒險遊戲**

詳細教學可以點我
(Unity新手利器 好用易上手的免費插件 Fungus)



那這次讓我們來做一個簡單的互動
吧！

你有辦法拔起石中劍嗎？



Why GitHub? Team Enterprise Explore Marketplace Pricing

Search

Sign in

Sign up

snozbob / fungus Public

Notifications

Star

941

Fork

212

<> Code

Issues 145

Pull requests 42

Actions

Wiki

Security

Insights

master

11 branches

66 tags

Go to file

Code



stevehalliwel Update Changelog and Version

2dbe6a4 on 27 Jun 2,625 commits

.github

Update feature_request.md

14 months ago

Assets

Update Changelog and Version

5 months ago

Docs/resources

Remove previous doco, now lives in the github wiki (#785)

2 years ago

Packages

Update Changelog and Version

5 months ago

ProjectSettings

Update Changelog and Version

5 months ago

.gitignore

Update changelog

15 months ago

CODE_OF_CONDUCT.md

Create CODE_OF_CONDUCT.md

2 years ago

CONTRIBUTING.md

Update CONTRIBUTING.md

2 years ago

Fungus-Playmaker Scripts.untypac...

Add files via upload

5 years ago

LICENSE

Update LICENSE

7 years ago

README.md

Update README.md

14 months ago

README.md

About

An easy to use Unity 3D library for creating illustrated Interactive Fiction games and more.

game

c-sharp

unity3d

fungus

Readme

MIT License

Code of conduct

Releases 61

Fungus v3.13.7 Latest

on 27 Jun

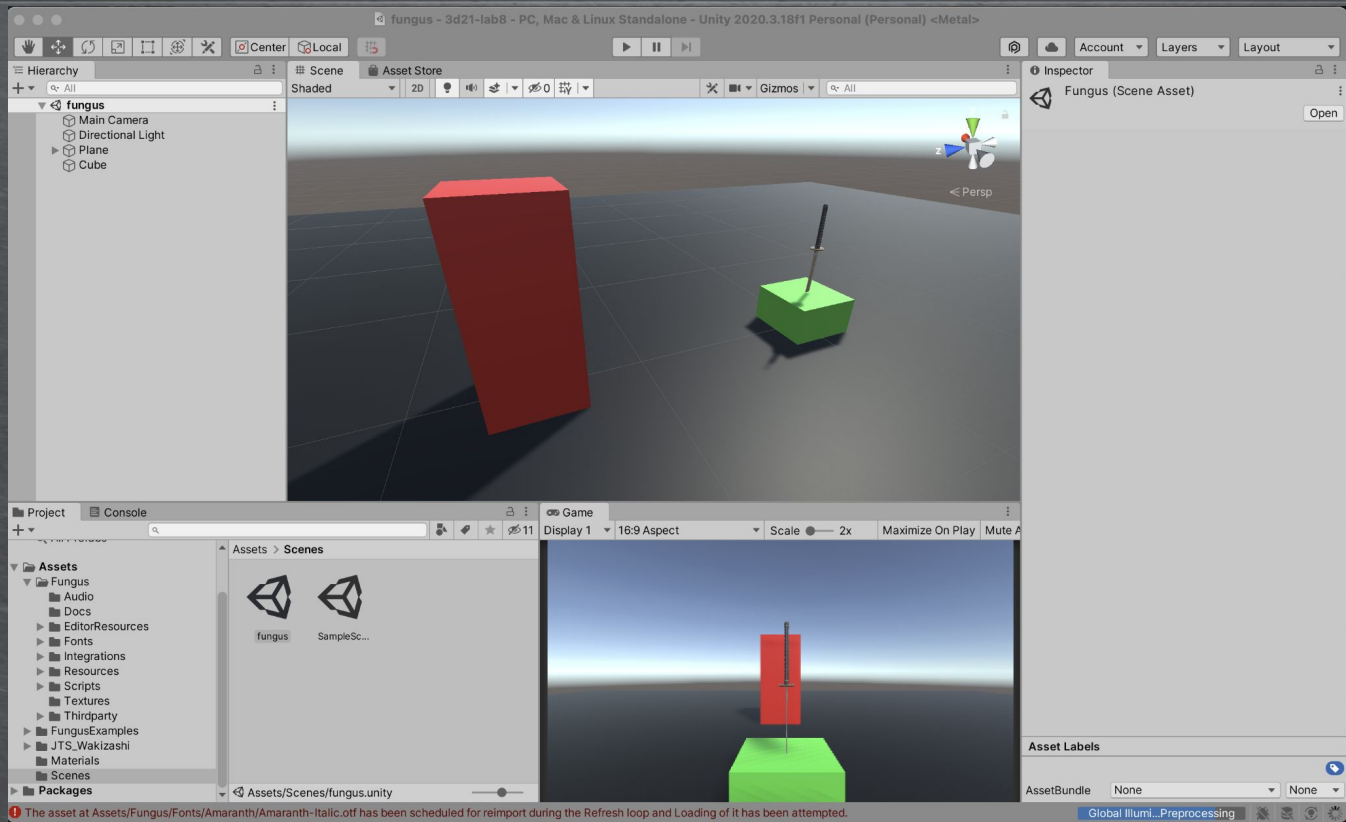
+ 60 releases

Contributors 37



請先下載 Fungus套件

or 匯入今天的素材包

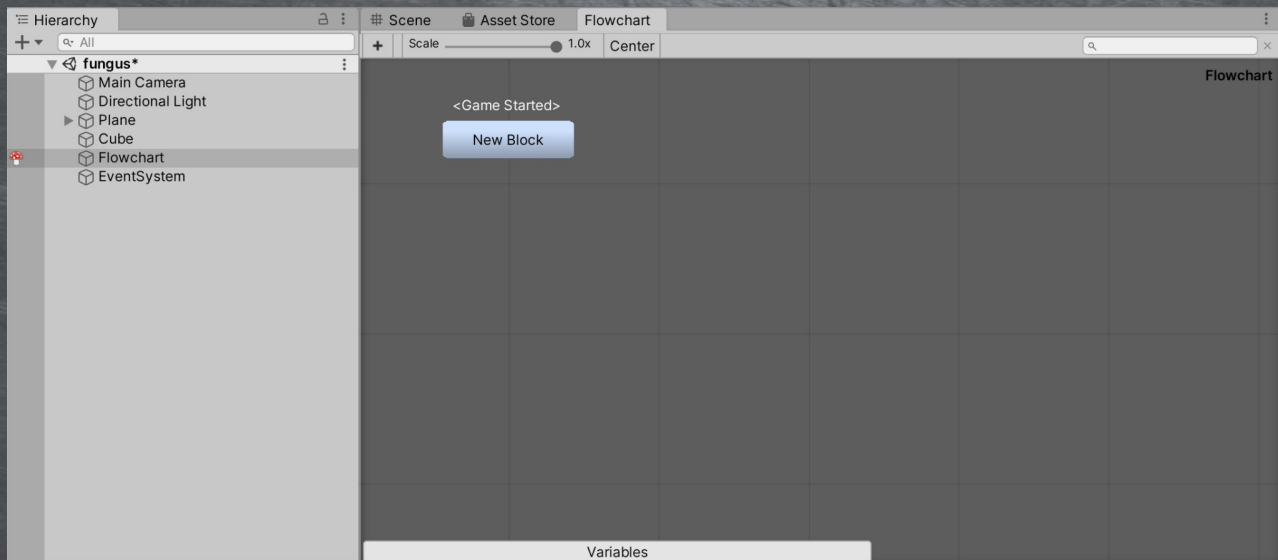
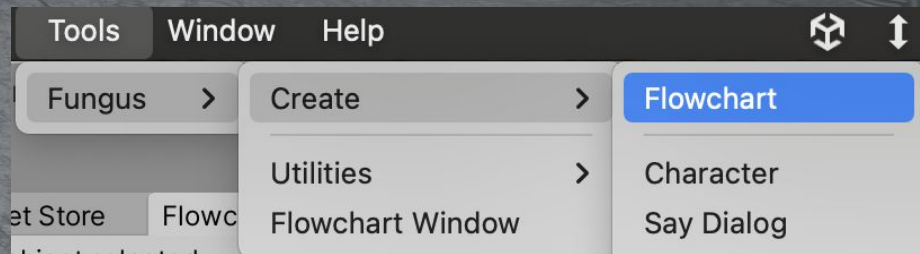
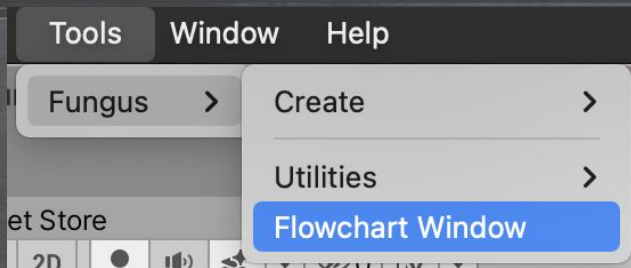


素材包內包含：fungus套件、劍的模型、名為 fungus的場景

若不使用素材包，請自行下載 fungus跟準備劍的模型
與場景建置 (一個平面、一把劍插在 cube上方、一個 cube)

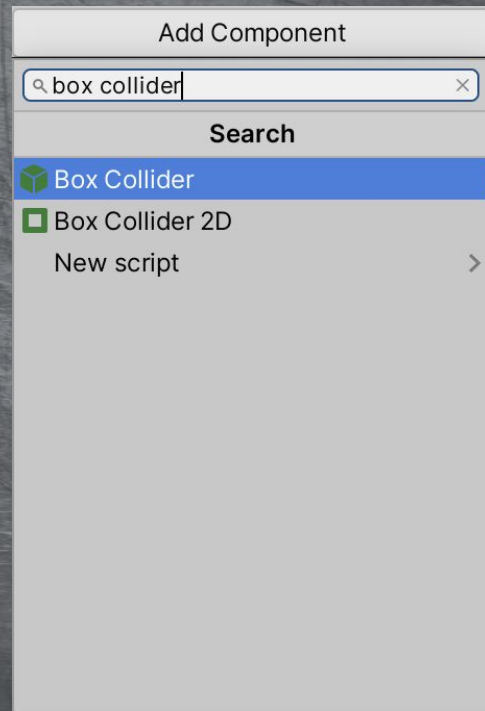
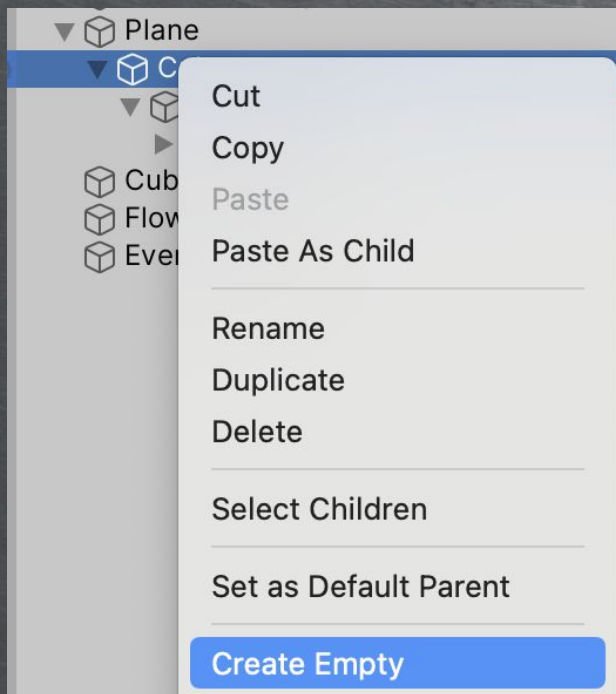

```
public class player_con : MonoBehaviour {  
    4 references  
    public float speed = 0.01f;  
  
    0 references  
    void Start() {  
  
    }  
  
    0 references  
    void Update() {  
        if (Input.GetKey("w")) { transform.Translate(0, 0, speed); }  
  
        if (Input.GetKey("s")) { transform.Translate(0, 0, -speed); }  
  
        if (Input.GetKey("a")) { transform.Translate(-speed, 0, 0); }  
  
        if (Input.GetKey("d")) { transform.Translate(speed, 0, 0); }  
    }  
}
```

在我們正式使用 Fungus前，我們先創建簡易的移動



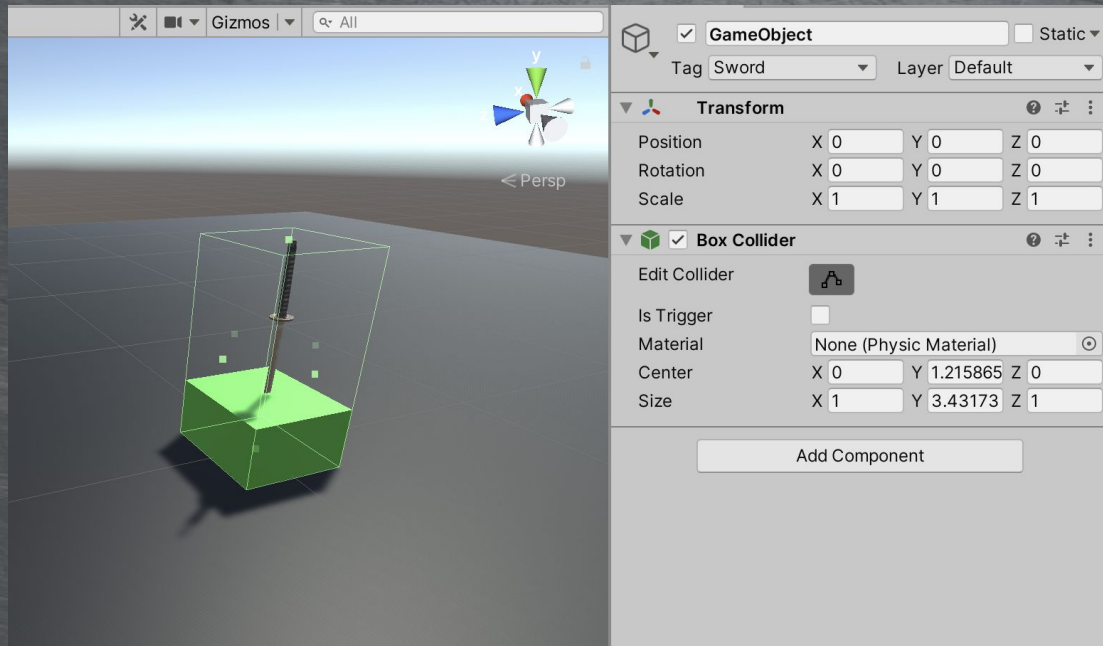
開始使用 Fungus, 開啟 Flowchart Window 和 創建 Flowchart
Fungus 的編輯區 (Flowchart) 很像之前教的動畫狀態機

使用碰撞器搭配 Fungus做出對話吧！



在石中劍處，創建一個空物件後在其物件上創建一個 **box collider**

使用碰撞器搭配 Fungus做出對話吧！



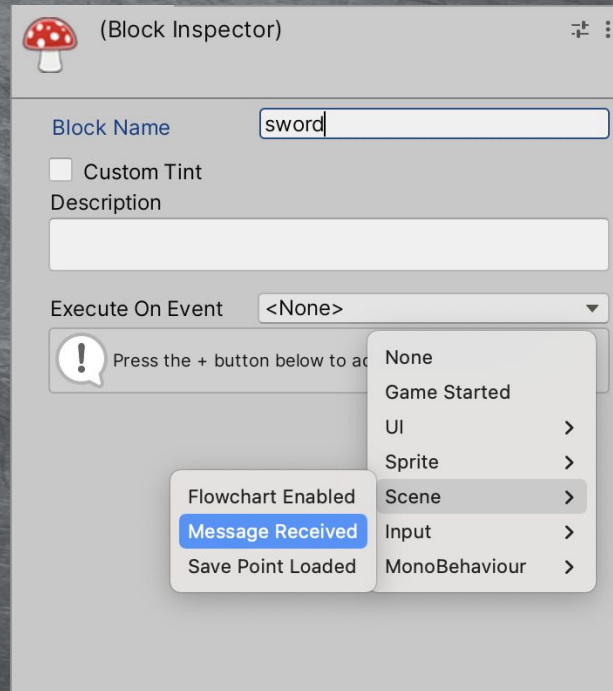
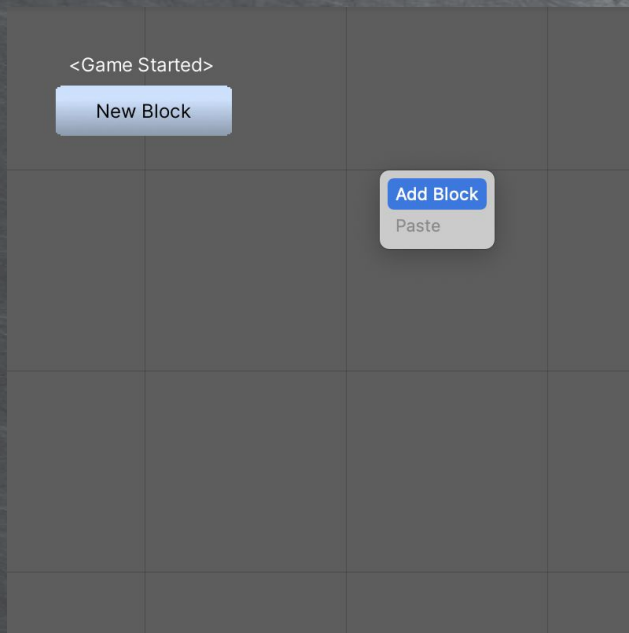
在石中劍處，創建一個空物件後在其物件上創建一個 **box collider**
記得**Tag**設定為**Sword**，或其他你想叫的，方便稍後程式碼判斷用
今天的實作會用 **OnCollisionEnter**，則**TriggerEnter**方式類似，就不另外 說明了

使用碰撞器搭配 Fungus做出對話吧！

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using Fungus;
4  using UnityEngine;
5
6  0 references
7  public class playerTriggerFungus : MonoBehaviour {
8      // Start is called before the first frame update
9      0 references
10     void Start() {
11
12     }
13
14     // Update is called once per frame
15     0 references
16     void Update() {
17
18     }
19
20     0 references
21     private void OnCollisionEnter(UnityEngine.Collision collision) {
22         if (collision.gameObject.tag == "Sword") {
23             Flowchart.BroadcastFungusMessage("hitSword");
24         }
25     }
26 }
```

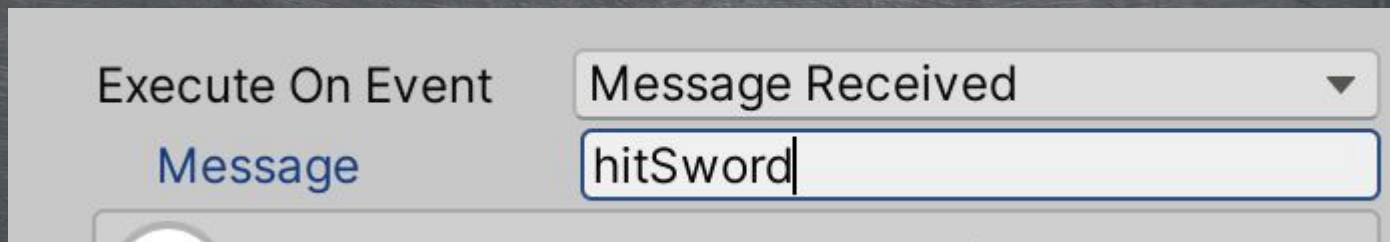
創建一個新的 Script,用它來執行 fungus相關指令
碰撞到碰撞器後,告知 FlowChart要做事了

使用碰撞器搭配 Fungus做出對話吧！



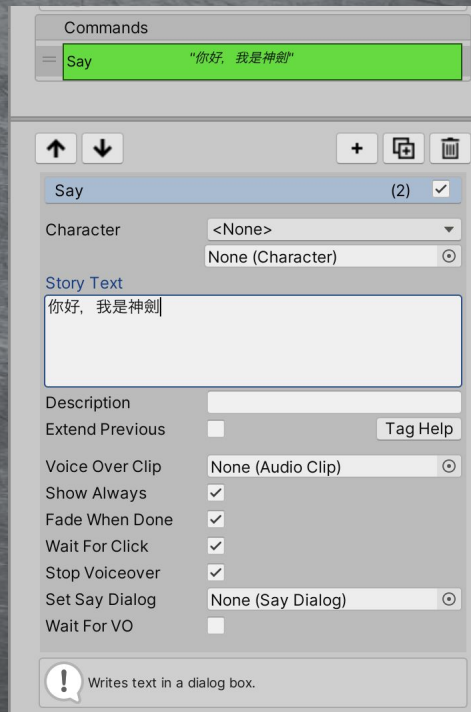
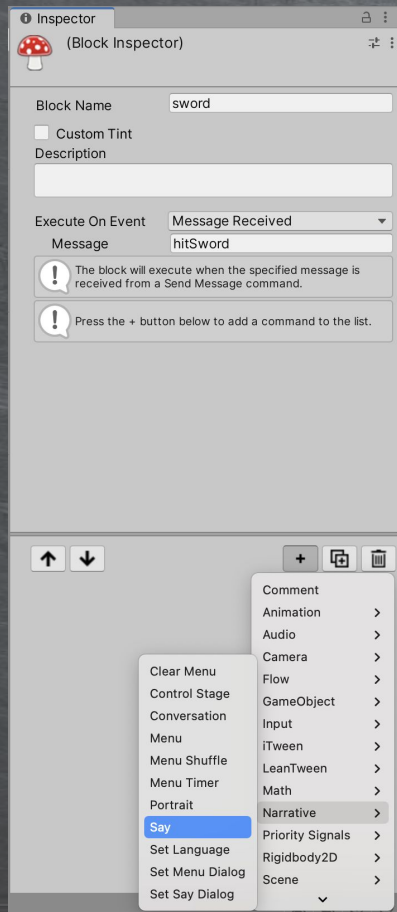
創建一個 Block，用來執行 Fungus 功能
並把右邊參數欄位區，Execute on Event 改成 Message Received

使用碰撞器搭配 Fungus做出對話吧！



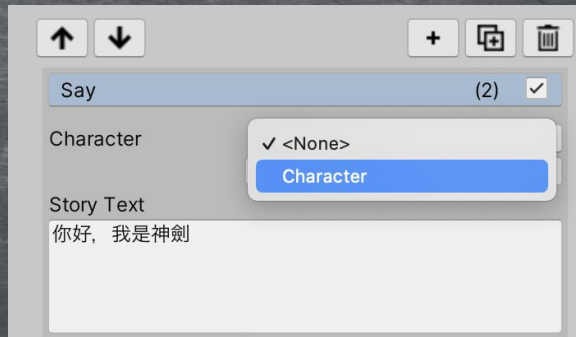
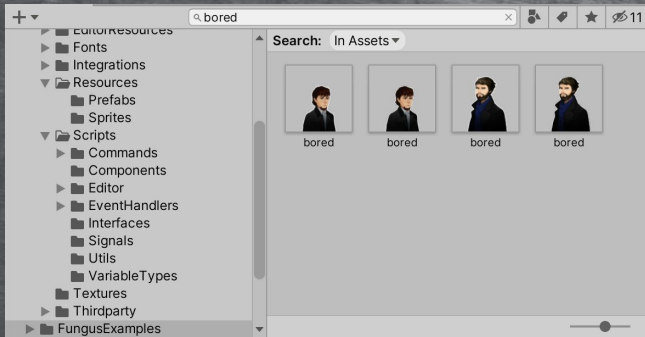
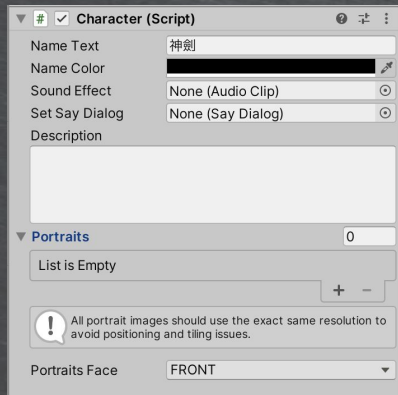
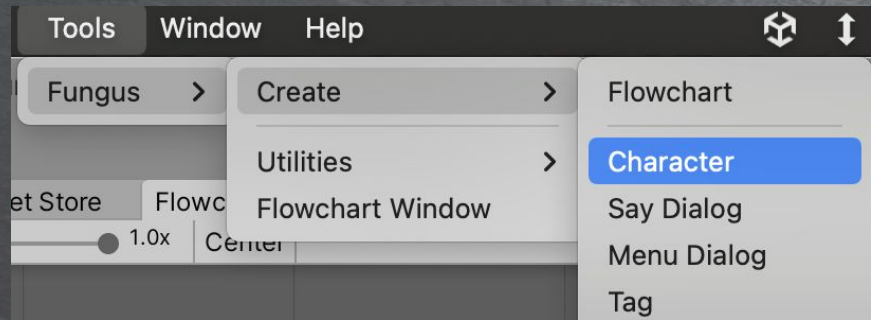
它的Message正好就是你剛剛程式碼所打的 內容

使用碰撞器搭配 Fungus 做出對話吧！



創建對話

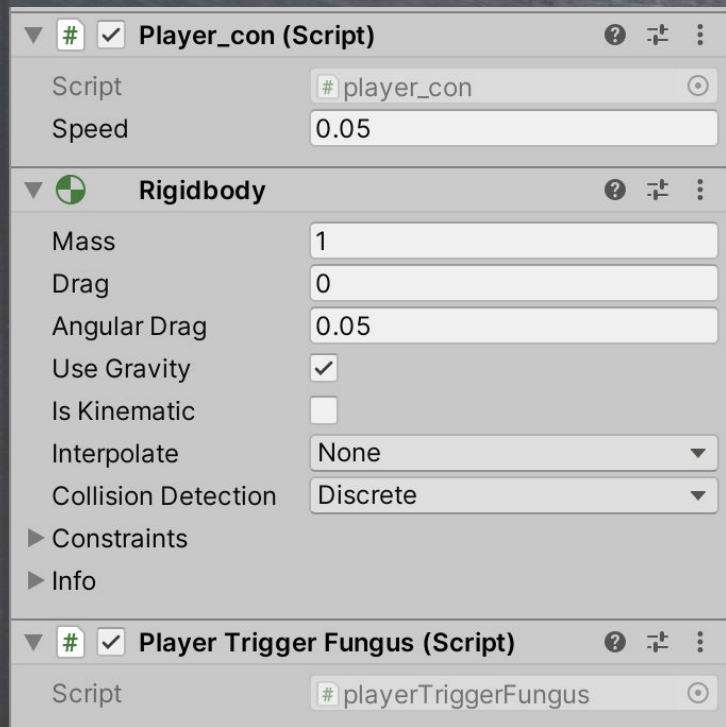
使用碰撞器搭配 Fungus做出對話吧！



創建角
色
圖

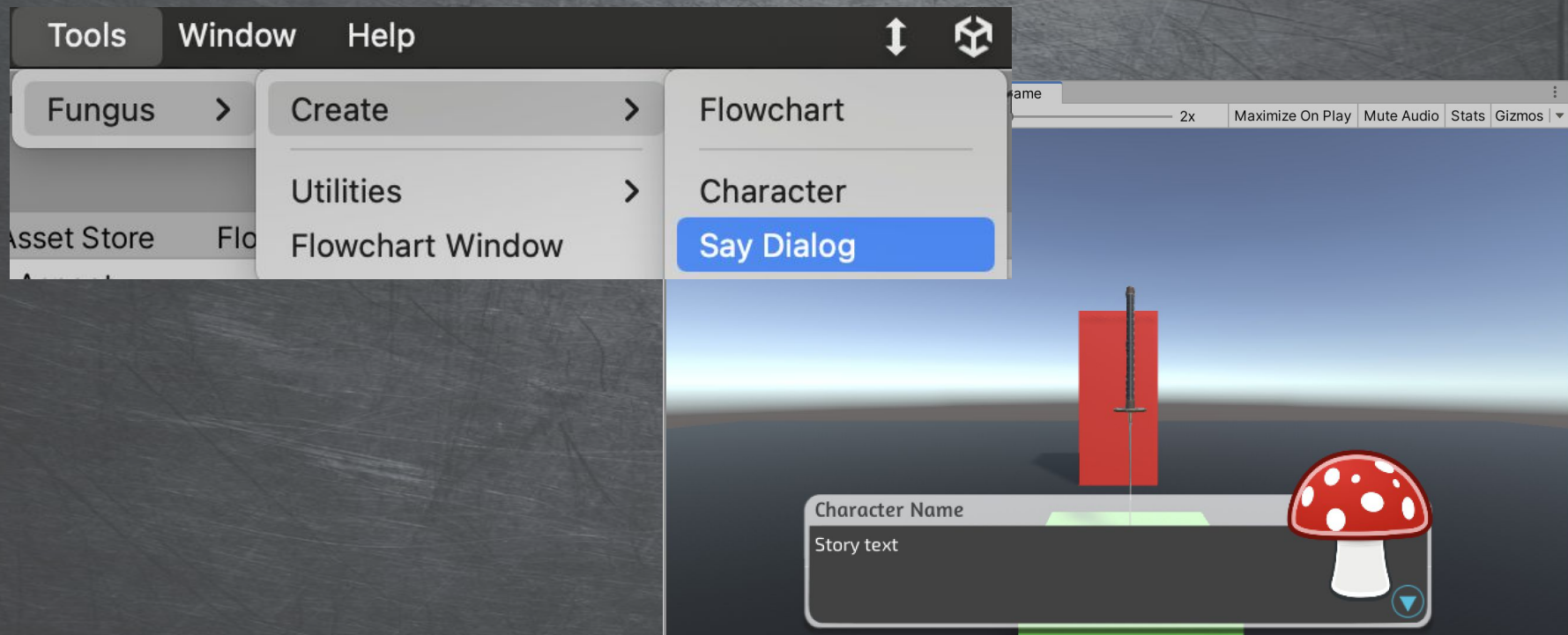
可以設定名字、名字顏色、登場音效、下方則是
可以儲存角色的貼圖

使用碰撞器搭配 Fungus做出對話吧！



讓我們來測試說話吧！
對話框前也可以插入音效
唱：

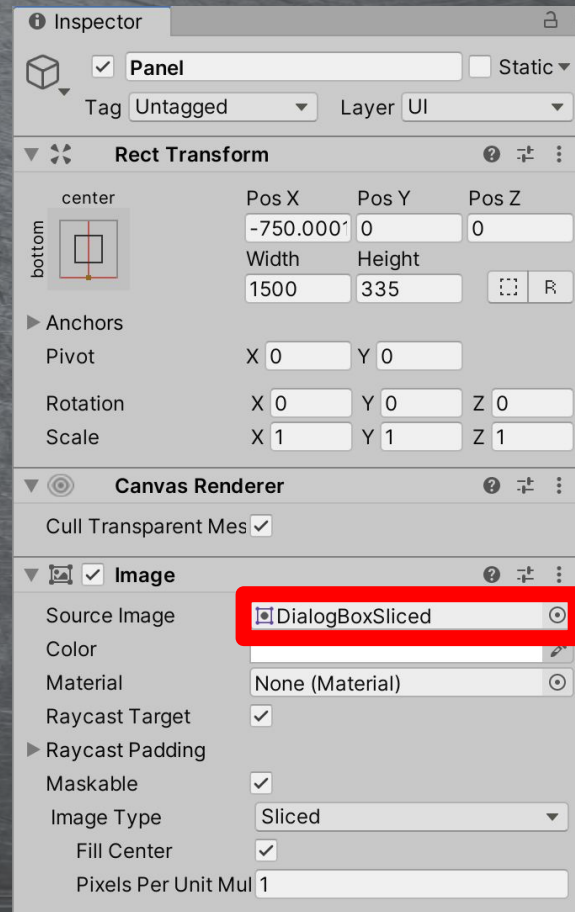
自定義對話框吧！



創建Say Dialog

它會創建一個對話框樣板

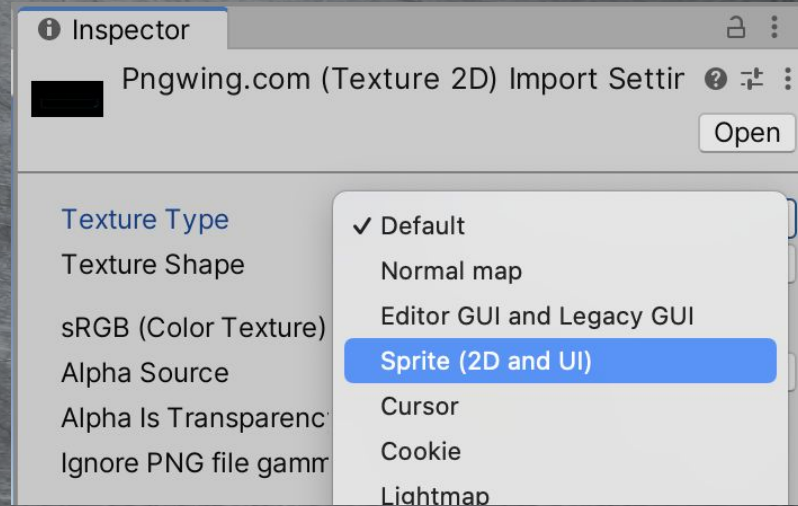
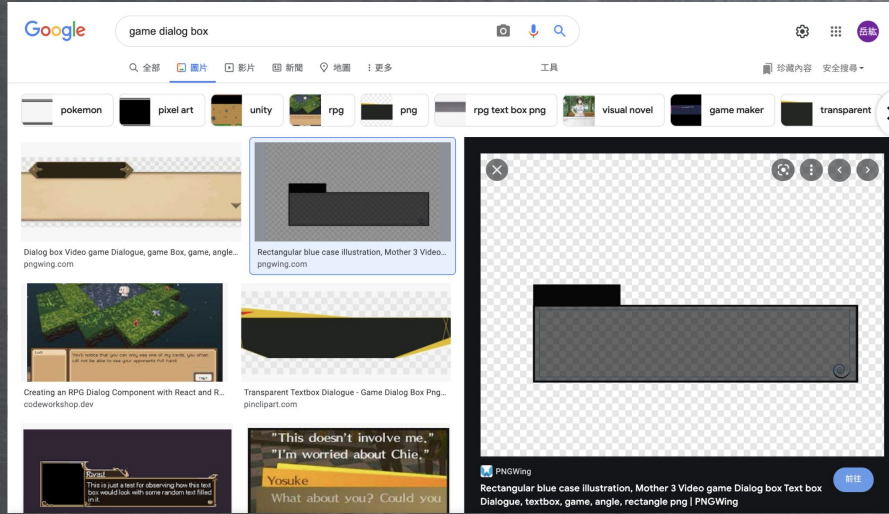
自定義對話框吧！– 調整對話框圖片



我們要更換底



自定義對話框吧！— 調整對話框圖片



找到圖片素材，去背調整後
匯入unity

※ 記得要把圖片 type 改成 Sprite

自定義對話框吧！– 調整對話框圖片

把剛剛找的圖片放到剛剛的屬性欄
位

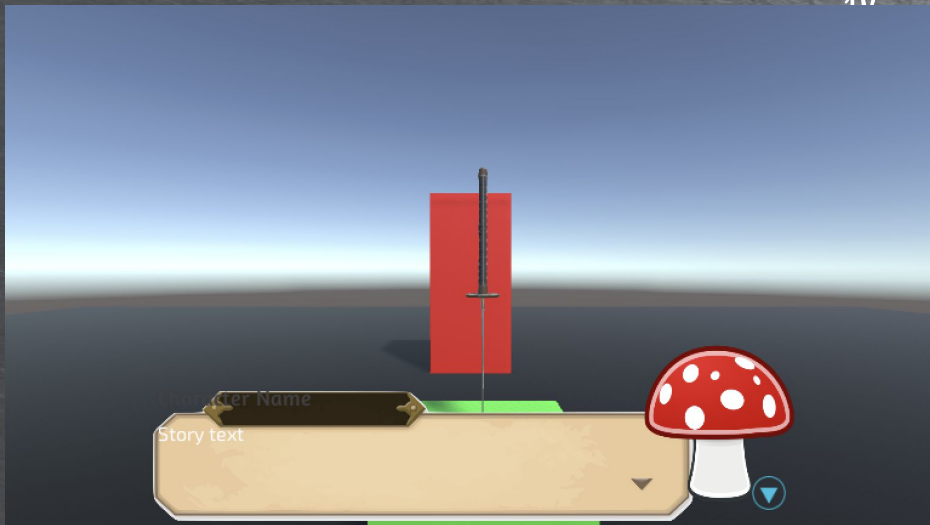
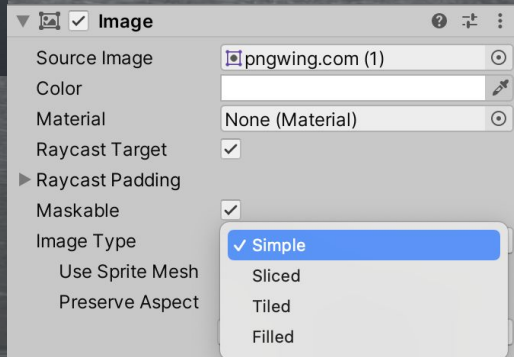


image type 可以調整填滿方法，找到適合的吧！

※大家記得有參數都可以多玩玩看，看看變化



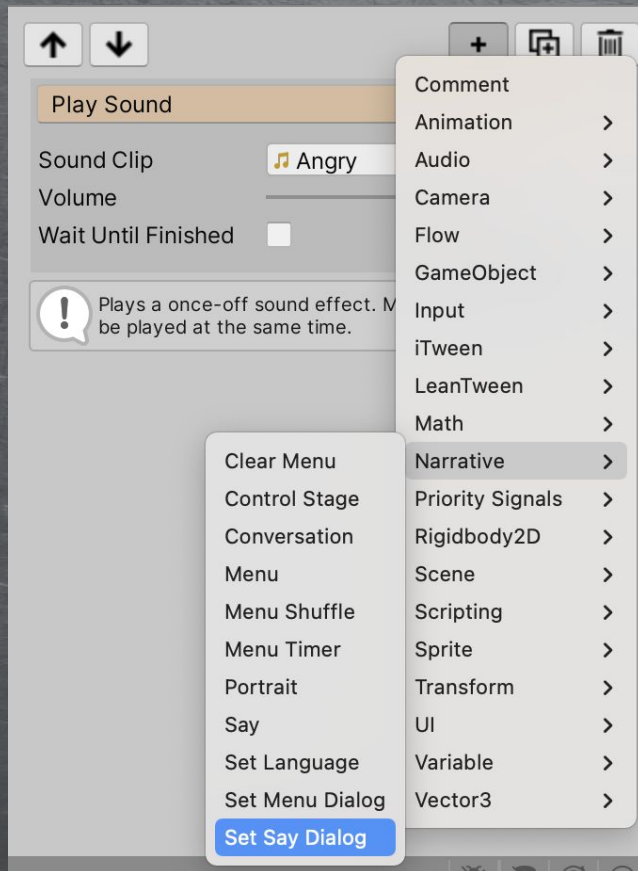
自定義對話框吧！－ 調整對話框圖片

調整文字等物件位 置

- Main Camera
- ▼ SayDialog
 - ▼ Panel
 - NameText
 - Image**
 - ▶ StoryText
 - Directional Light
 - ▼ Plane
 - ▼ Cube
 - Wakizashi
 - ▶ _Colliders
 - GameObject
 - Cube
 - Flowchart
 - EventSystem
 - Character



自定義對話框吧！— 見證調整效果



自定義對話框吧！— 見證調整效果



使用Fungus呼叫程式碼，拔出石中劍吧！



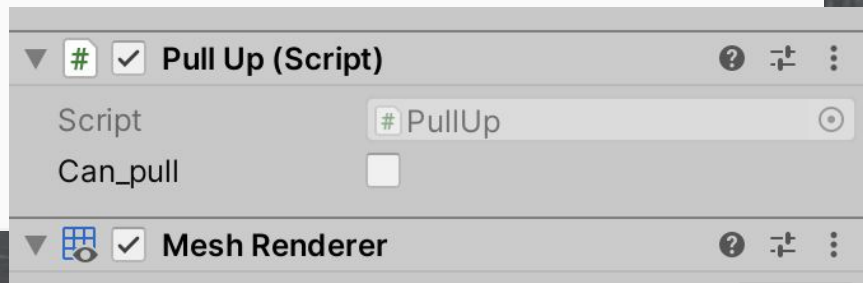
我們在要拔除的 劍上加上程式碼

pull 是等等要給 Fungus呼叫的

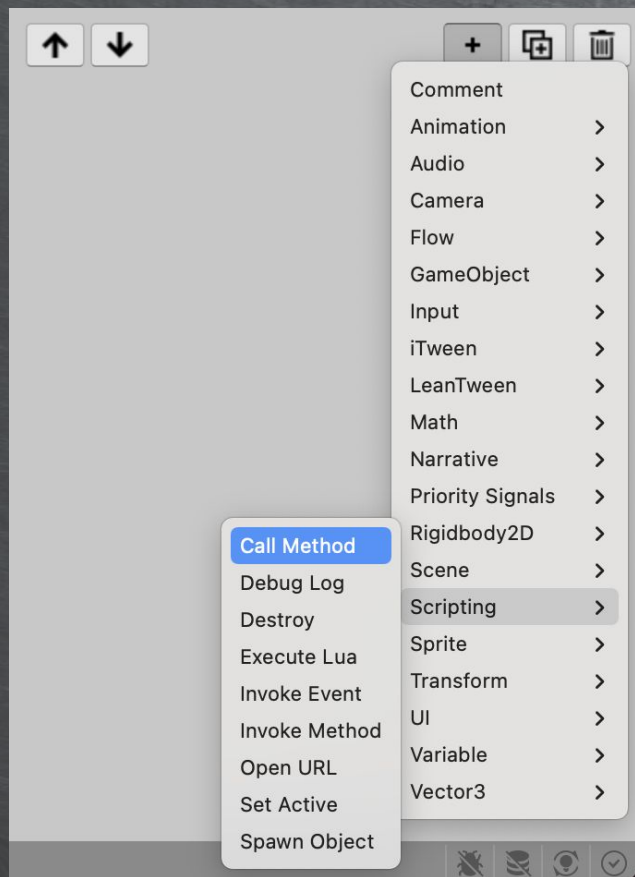
然後設置一個 bool阻擋拔起
當can_pull是true時

物件才會自己升起

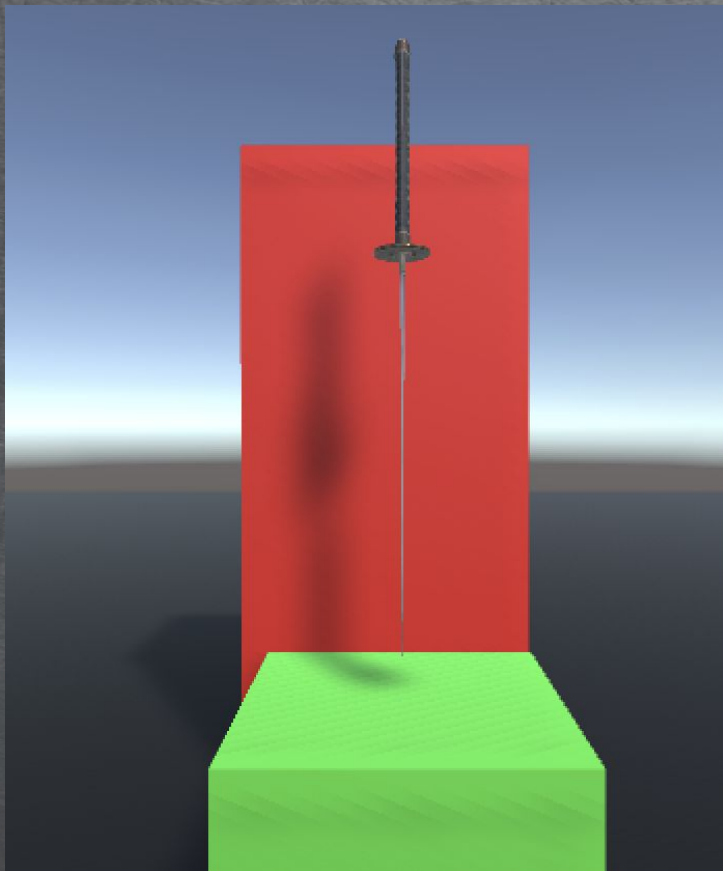
```
0 references
5 public class PullUp : MonoBehaviour {
    2 references
6     public bool can_pull = false;
7     // Start is called before the first frame update
    0 references
8     void Start() {
9     }
10
11    // Update is called once per frame
    0 references
12    void Update() {
13        if (can_pull) {
14            float height = Mathf.Clamp(transform.localPosition.y + 0.01f, 0, 3);
15            transform.localPosition = new Vector3(transform.position.x, height, transform.position.z);
16        }
17    }
18
    0 references
19    public void pull() {
20        can_pull = true;
21    }
22 }
```



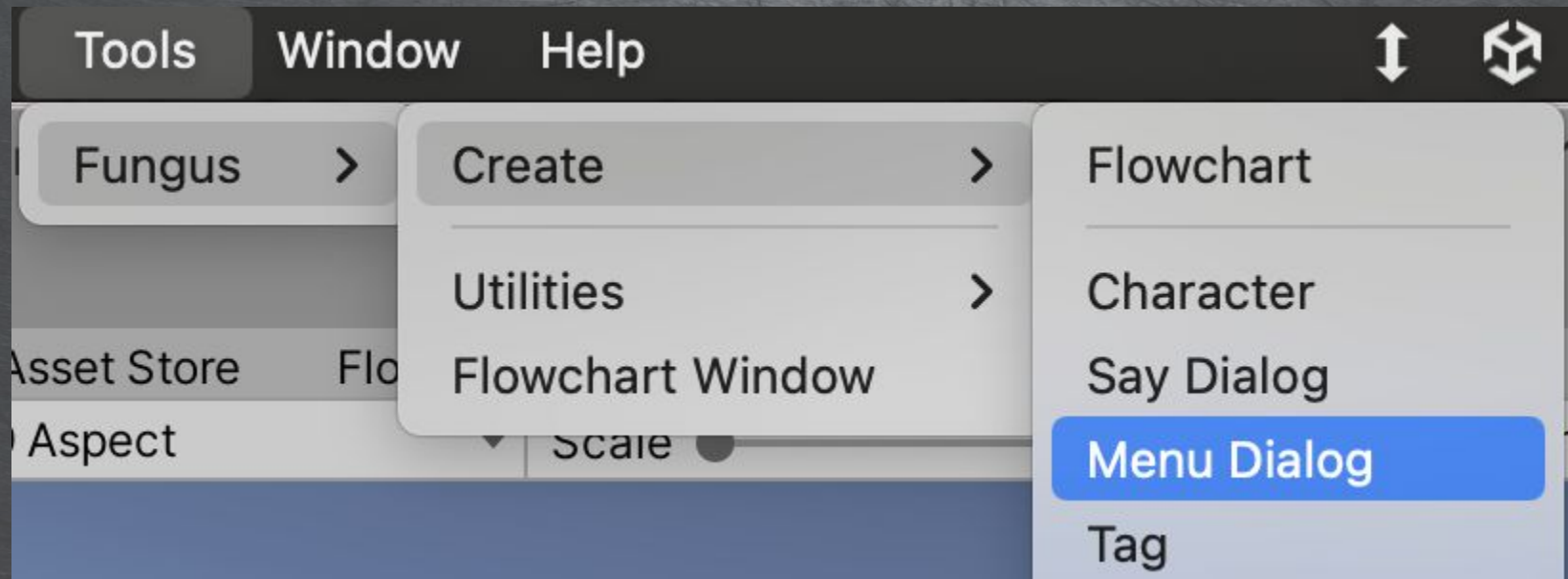
使用Fungus呼叫程式碼，拔出石中劍吧！– 回到FlowChart，創建新行為



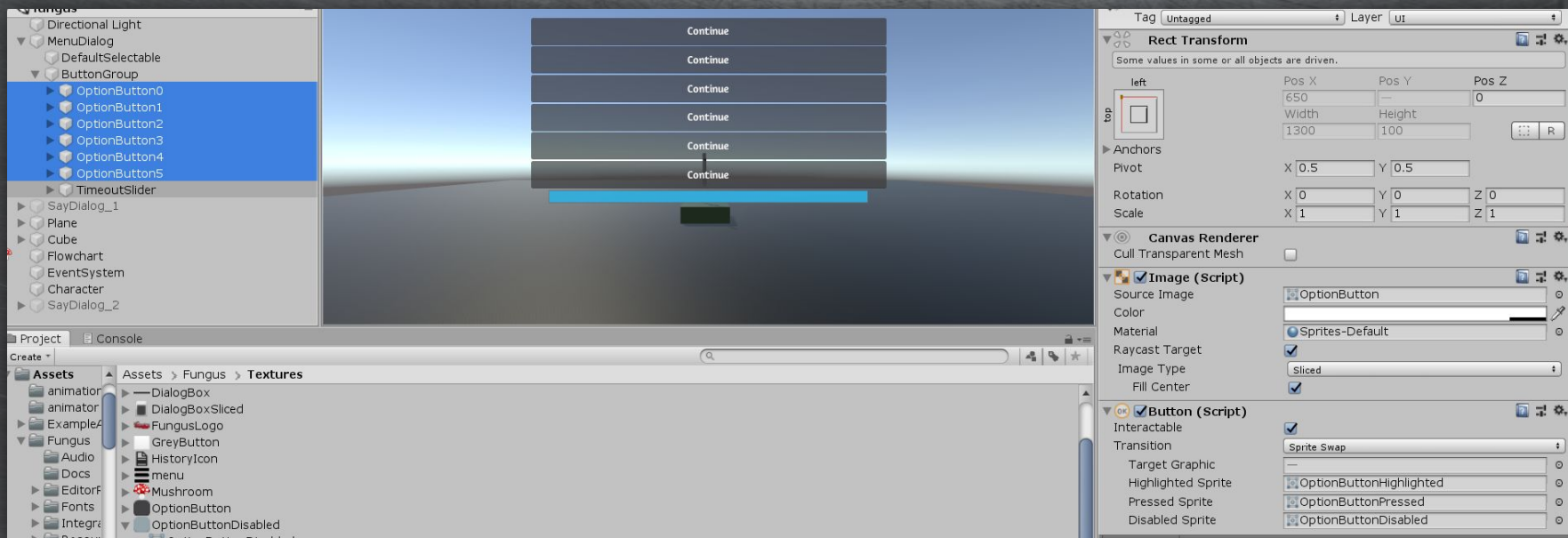
使用Fungus呼叫程式碼，拔出石中劍吧！– 再次測試，是否拔起來了



使用Fungus呼叫程式碼，拔出石中劍吧！– 新增對話選項 (Menu dialog)



使用Fungus呼叫程式碼，拔出石中劍吧！– 新增對話選項 (Menu dialog)

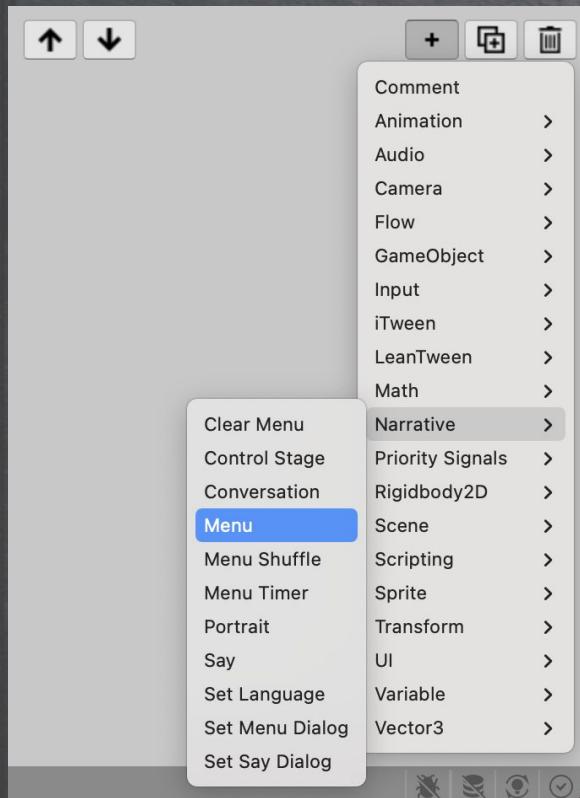


預設會給你 6個選項格，代表若你等等使用這個 menu dialog你只能創建 6個選項

若你想要更多，就再新增 optionButton物件，並自己調整位置

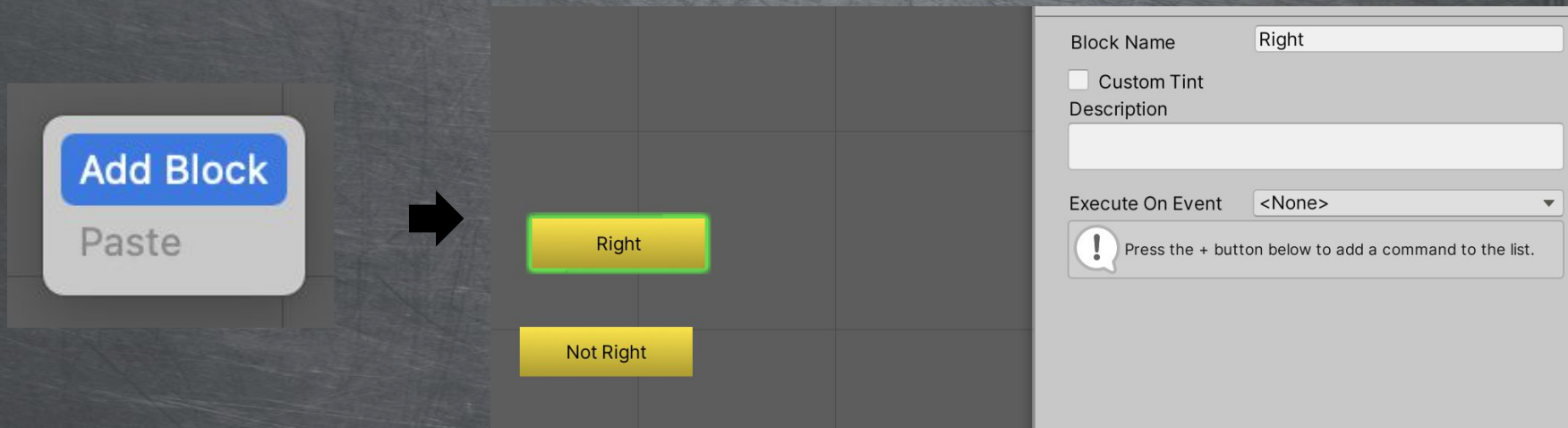
它跟say dialog一樣，可以自行調整風格，右方屬性欄位可以自己嘗試看看

使用Fungus呼叫程式碼，拔出石中劍吧！– Menu dialog



一個menu等同於一個選項
創建多個，玩家就可以選擇了

使用Fungus呼叫程式碼，拔出石中劍吧！– Menu dialog



一個menu等同於一個選項
創建多個，玩家就可以選擇了

使用Fungus呼叫程式碼，拔出石中劍吧！– Menu dialog



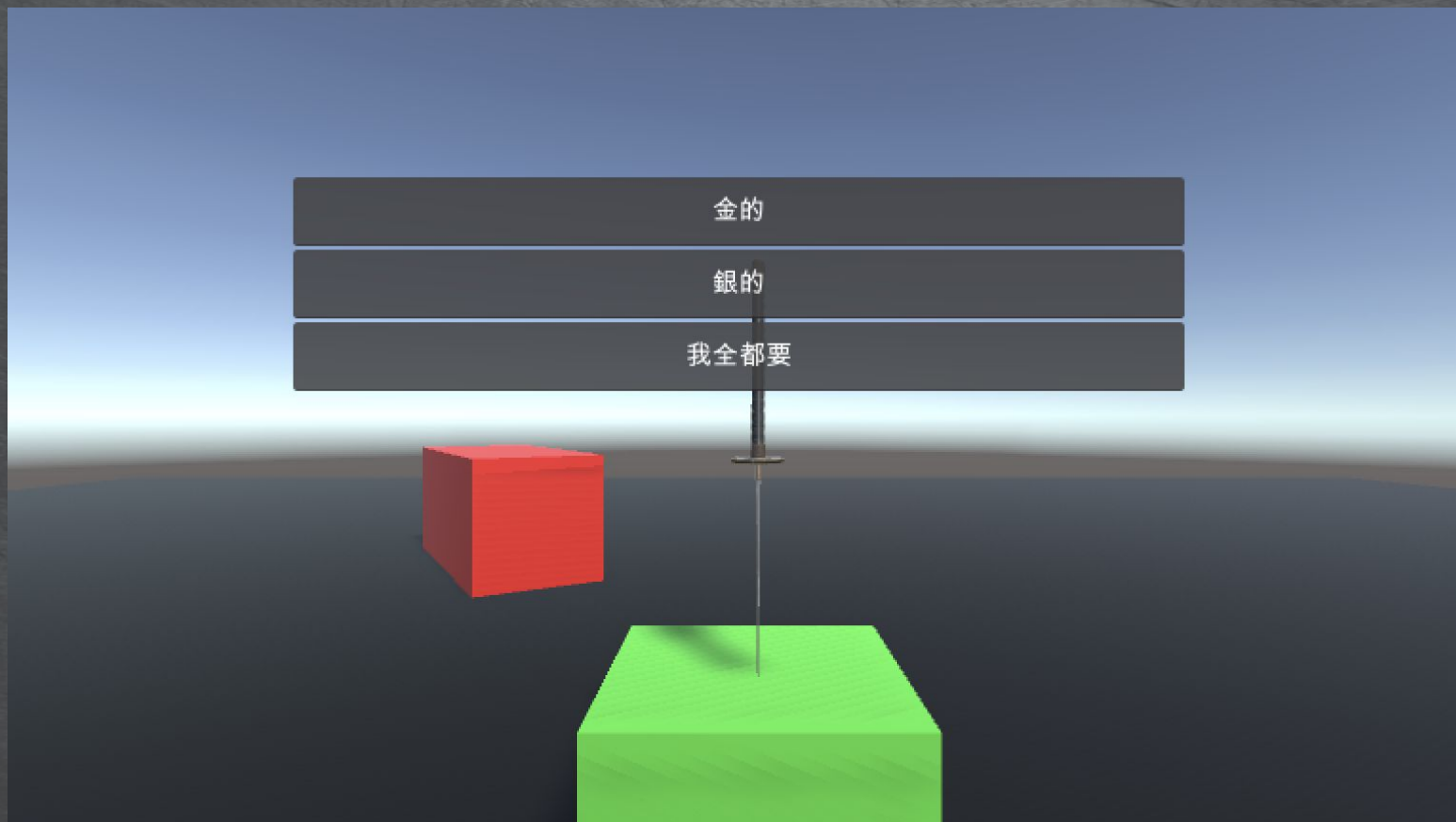
選項內

容

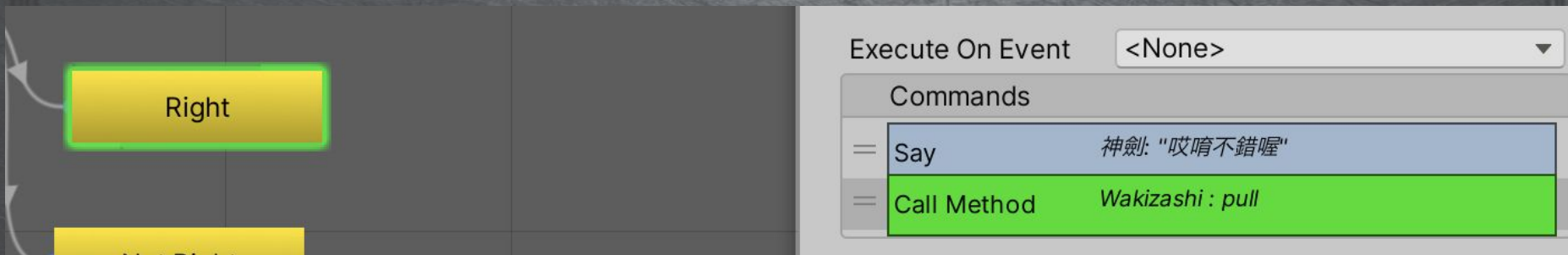
選了前往的 block

用哪個 menuDialog
(選我們剛剛創建的)

使用Fungus呼叫程式碼，拔出石中劍吧！– Menu dialog



使用Fungus呼叫程式碼, 拔出石中劍吧 ! – Menu dialog



那我們就可以自定義一個正確回答, 然後正確回答的 block 呼叫剛剛拔劍的程式碼

※看你要在原本的 block ctrl+v, 在正確的 block ctrl + v
或者你直接把原本的刪掉, 在正確的 block 創建 call method 也可以
(實作的方法百百種的)

Funglus

謝謝大
家