

Google VR

愛迪斯科技 技術工程師 紀潔嵐 Kuma
kuma.ji@axis3d.com

虛擬實境 VR

- **虛擬實境**（virtual reality，簡稱VR），是利用電腦模擬產生一個三維空間的虛擬世界，提供使用者關於視覺等感官的模擬，讓使用者感覺仿佛身歷其境，可以及時、沒有限制地觀察三維空間內的事物。



透過視覺效果，讓使用者達到身歷其境的目的

Cardboard VR

- Google Cardboard 透過簡便又經濟實惠的方式獲得身歷其境的觀影體驗。
- 透過簡單的瓦楞紙組合加上智慧型手機，就可以輕鬆體驗VR的效果。



Google VR開發者官網：

<https://developers.google.com/vr/>

Cardboard SDK

The screenshot shows the Google VR Developer Overview page for Cardboard. The navigation bar includes 'Google VR', 'Daydream', 'Cardboard' (highlighted with a red box), and 'Documentation'. A search bar is on the right. The main content area is titled 'Developer Overview' and includes a sidebar with 'Cardboard', 'Developer Overview', and 'Designing for VR'. The main text describes the Google VR SDKs for Cardboard (Android and iOS). Below this, a section titled 'SDKs' lists four options: 'Google VR SDK for Android', 'Google VR SDK for Unity', 'Google VR SDK for Unreal', and 'Google VR SDK for iOS'. The first three SDK options are highlighted with a red box.

Google VR Daydream **Cardboard** Documentation 搜尋

Google VR Concepts

Cardboard
Developer Overview
Designing for VR

Developer Overview

☆☆☆☆

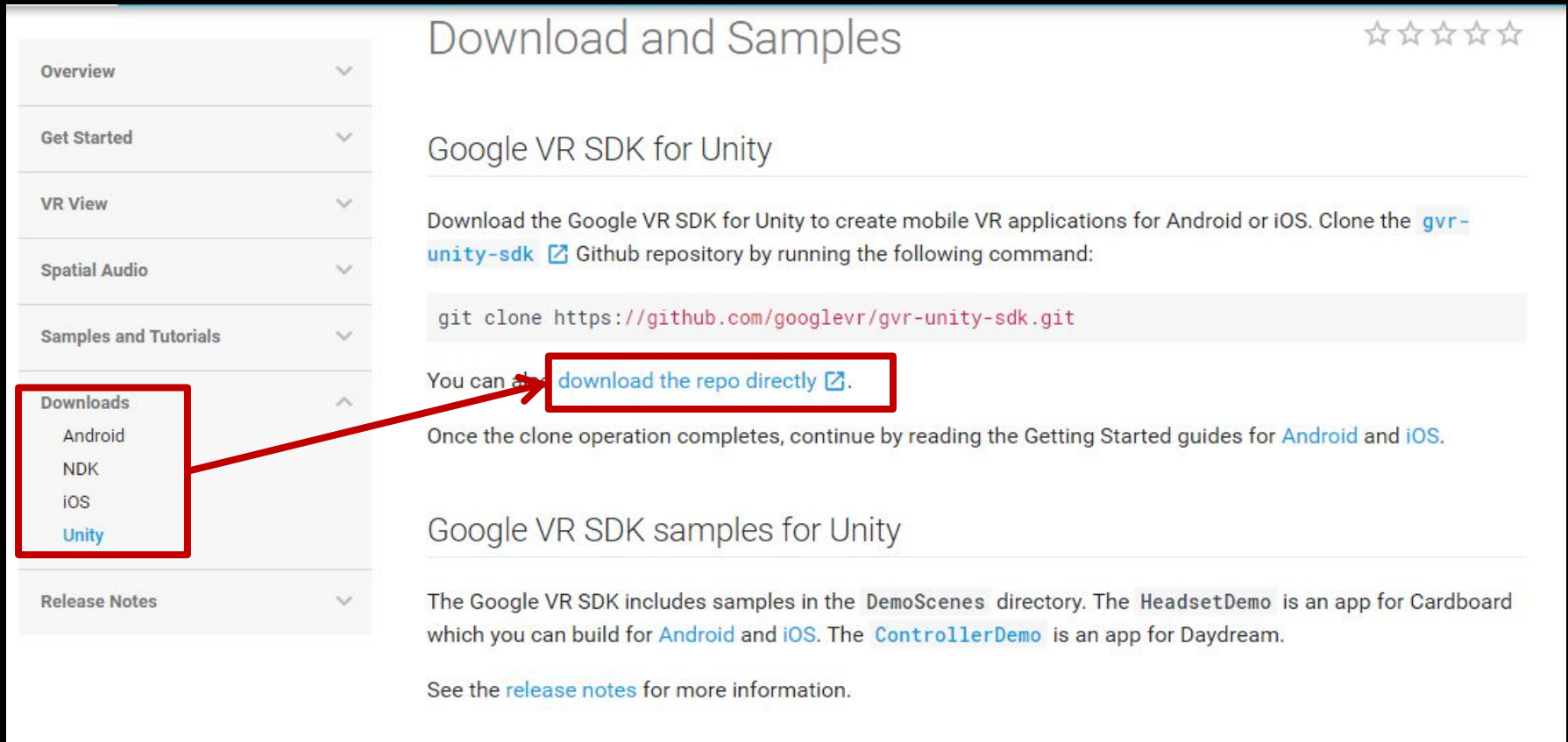
The Google VR SDKs let you build experiences for Cardboard (Android and iOS). Our SDKs simplify common VR development tasks so you can focus on building your new immersive experience.

SDKs

- Google VR SDK for Android**
Build apps that display 3D scenes with binocular rendering, render spatial audio, and track and react to head movements.
- Google VR SDK for Unity**
Easily adapt an existing Unity 3D app for virtual reality or build your own VR experience from scratch.
- Google VR SDK for Unreal**
Support for creating Cardboard applications is built into Unreal Engine 4.
- Google VR SDK for iOS**
Build a VR experience for your iOS devices.

依據開發的軟體，選擇下載的SDK

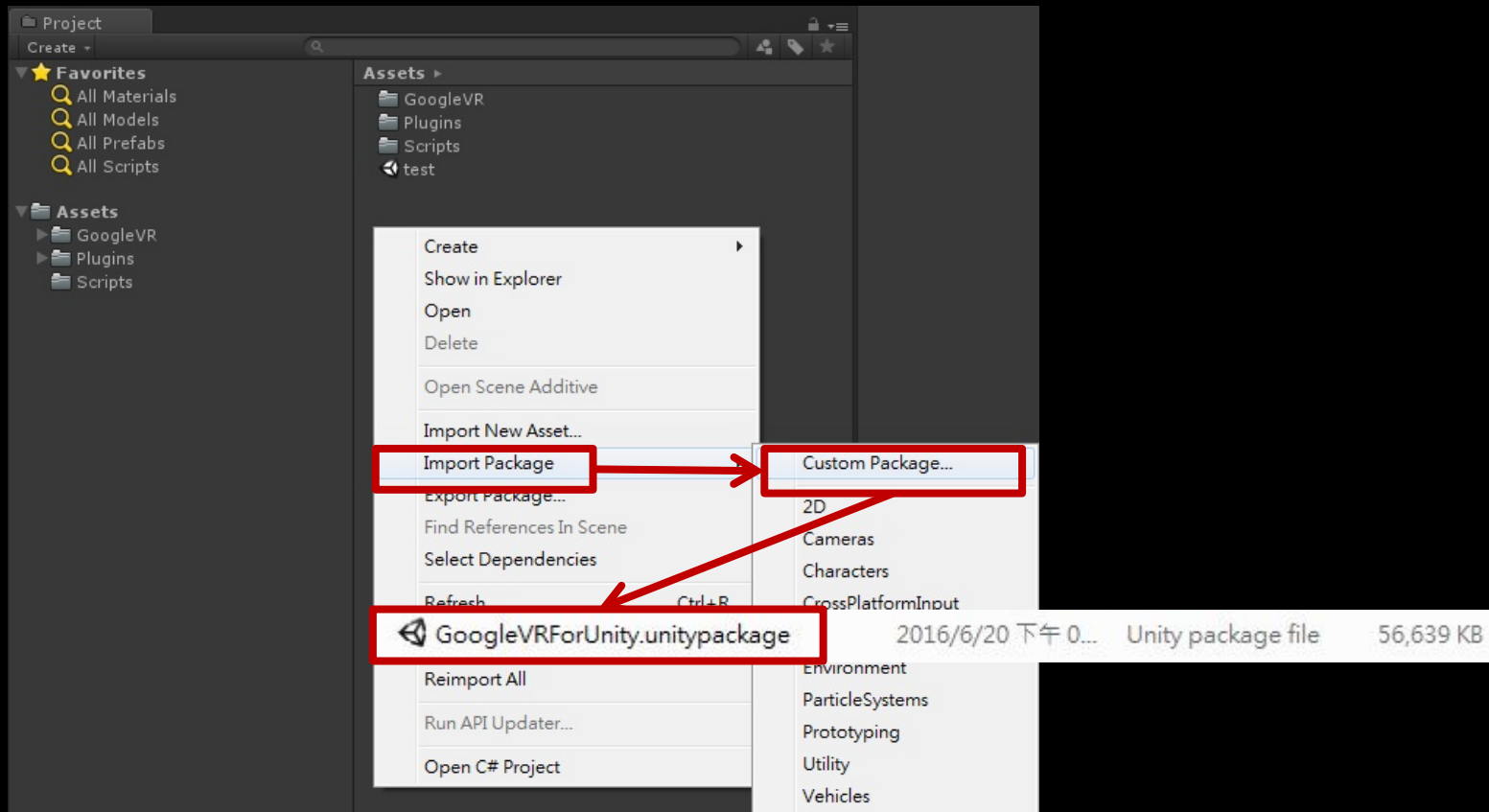
Cardboard SDK



The screenshot shows the 'Download and Samples' page for the Google VR SDK for Unity. On the left is a navigation menu with the following items: Overview, Get Started, VR View, Spatial Audio, Samples and Tutorials, Downloads, and Release Notes. The 'Downloads' menu is expanded, showing sub-items: Android, NDK, iOS, and Unity. A red box highlights the 'Downloads' menu and its sub-items, with a red arrow pointing from the 'Unity' sub-item to a red box around the 'download the repo directly' link in the main content area. The main content area has a title 'Download and Samples' with five stars, followed by the heading 'Google VR SDK for Unity'. Below this is a paragraph: 'Download the Google VR SDK for Unity to create mobile VR applications for Android or iOS. Clone the `gvr-unity-sdk` Github repository by running the following command:'. A code block contains the command: `git clone https://github.com/googlevr/gvr-unity-sdk.git`. Below the code block is the text: 'You can also [download the repo directly](#).' This link is highlighted with a red box. Further down is the text: 'Once the clone operation completes, continue by reading the Getting Started guides for [Android](#) and [iOS](#).' Below this is the heading 'Google VR SDK samples for Unity' and a paragraph: 'The Google VR SDK includes samples in the `DemoScenes` directory. The `HeadsetDemo` is an app for Cardboard which you can build for [Android](#) and [iOS](#). The `ControllerDemo` is an app for Daydream.' At the bottom of this section is the text: 'See the [release notes](#) for more information.'

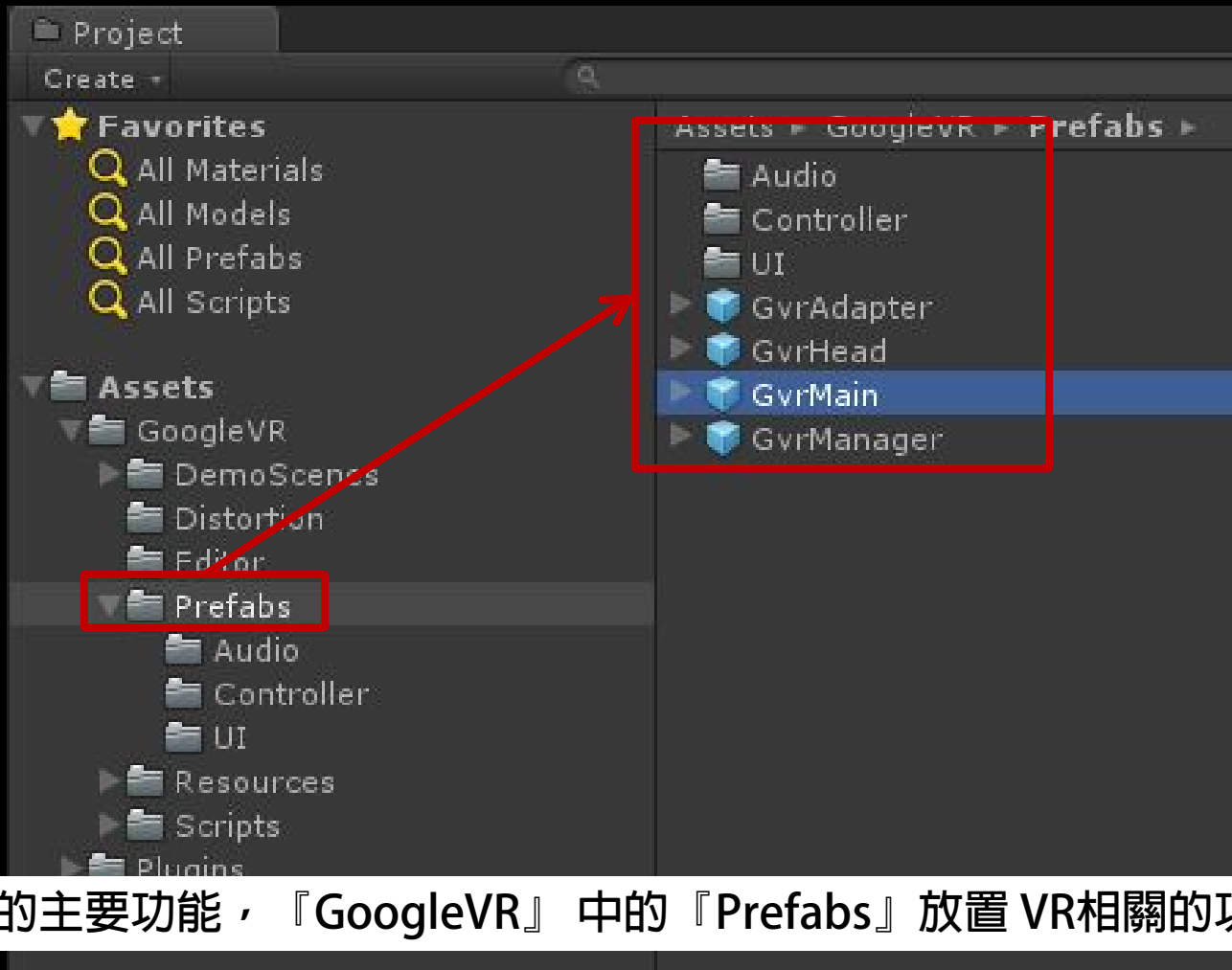
依據開發的軟體，選擇下載的SDK

Cardboard SDK



在Unity專案中，匯入所下載的SDK

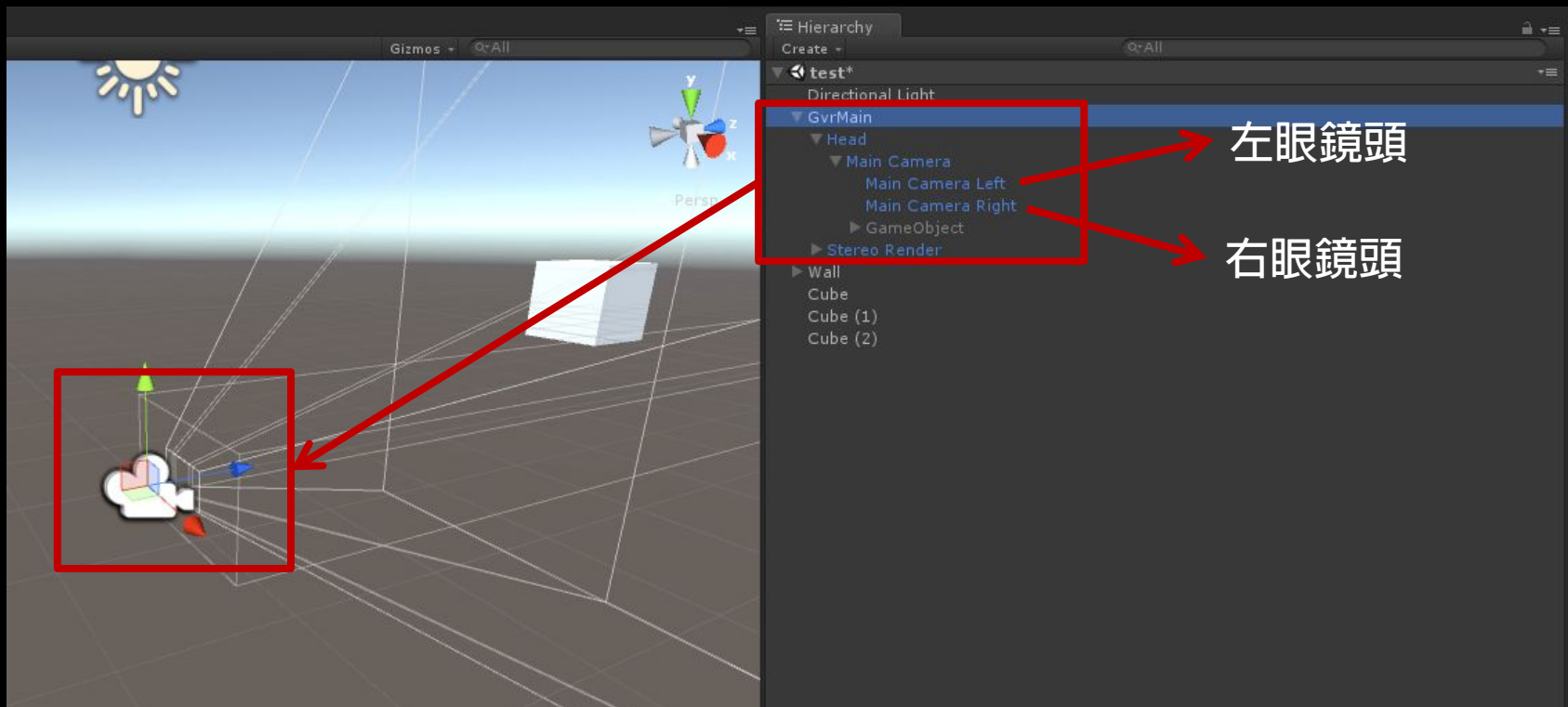
Cardboard SDK



SDK中的主要功能，『GoogleVR』中的『Prefabs』放置VR相關的功能模組

VR Camera

- 在VR Camera中，總共有三個攝影機，分別是左眼鏡頭、右眼鏡頭、整體鏡頭



放入GvrMain就可以達到觀看的VR效果

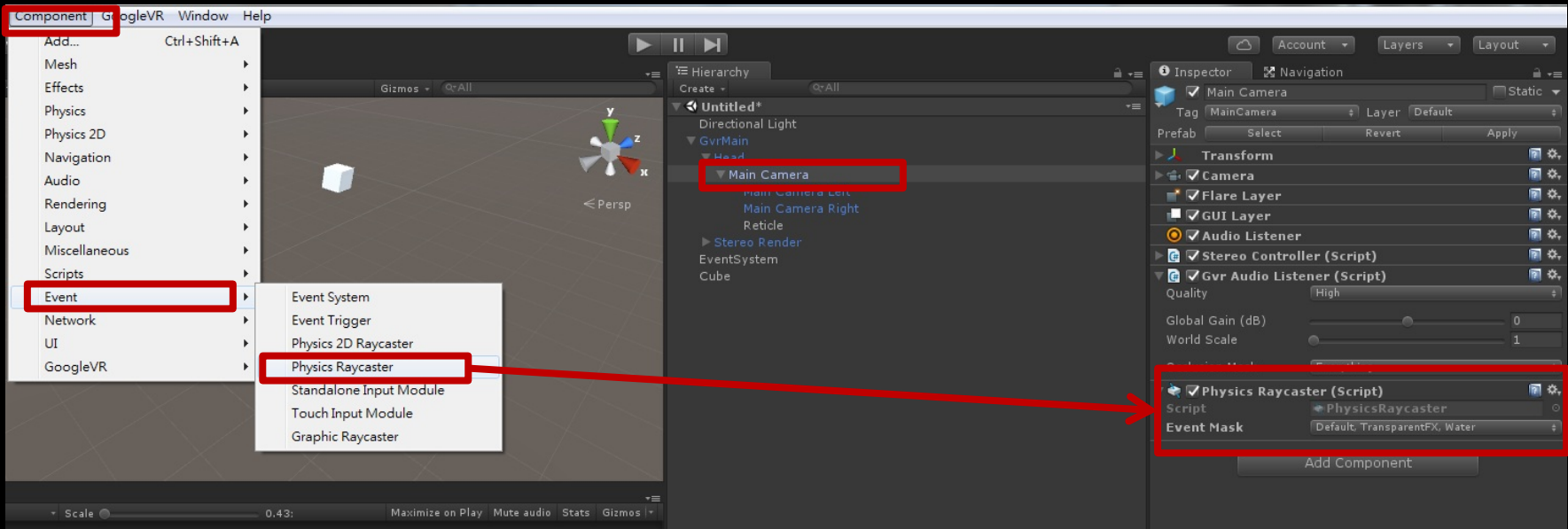
互動控制

- 沉浸式體驗 → Google VR + 3D環境
- 沉浸式體驗 + 外部硬體控制 → Google VR + 3D環境 + 外部控制硬體 (滑鼠、鍵盤、搖桿...)
- 沉浸式體驗 + 控制 → Google VR + 3D環境 + 視線控制

VR的互動方式會依據外部所使用的硬體，而有不同的規劃

互動控制

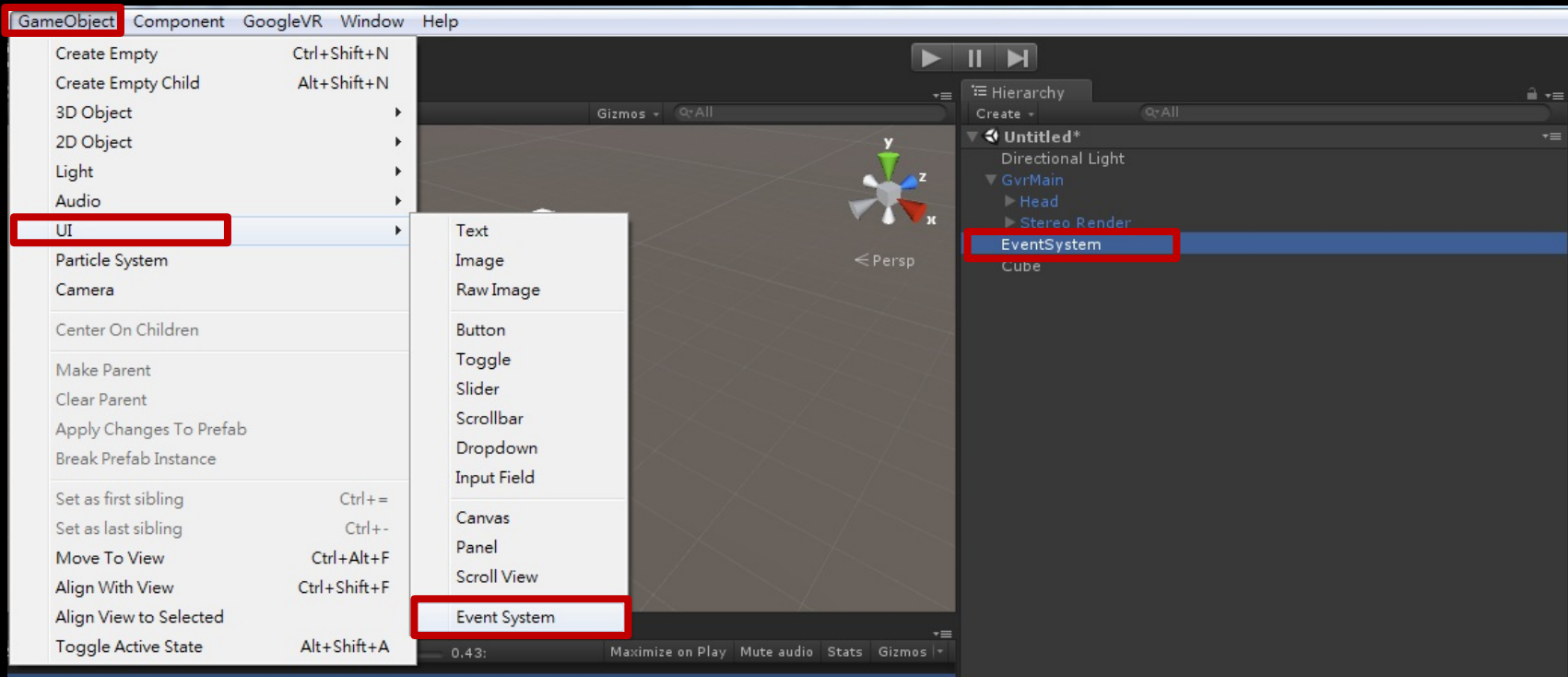
- 視線控制 → 在攝影機上增加Physics Raycaster，使射線具有物理效果



利用射線及UI來製作視線效果

互動控制

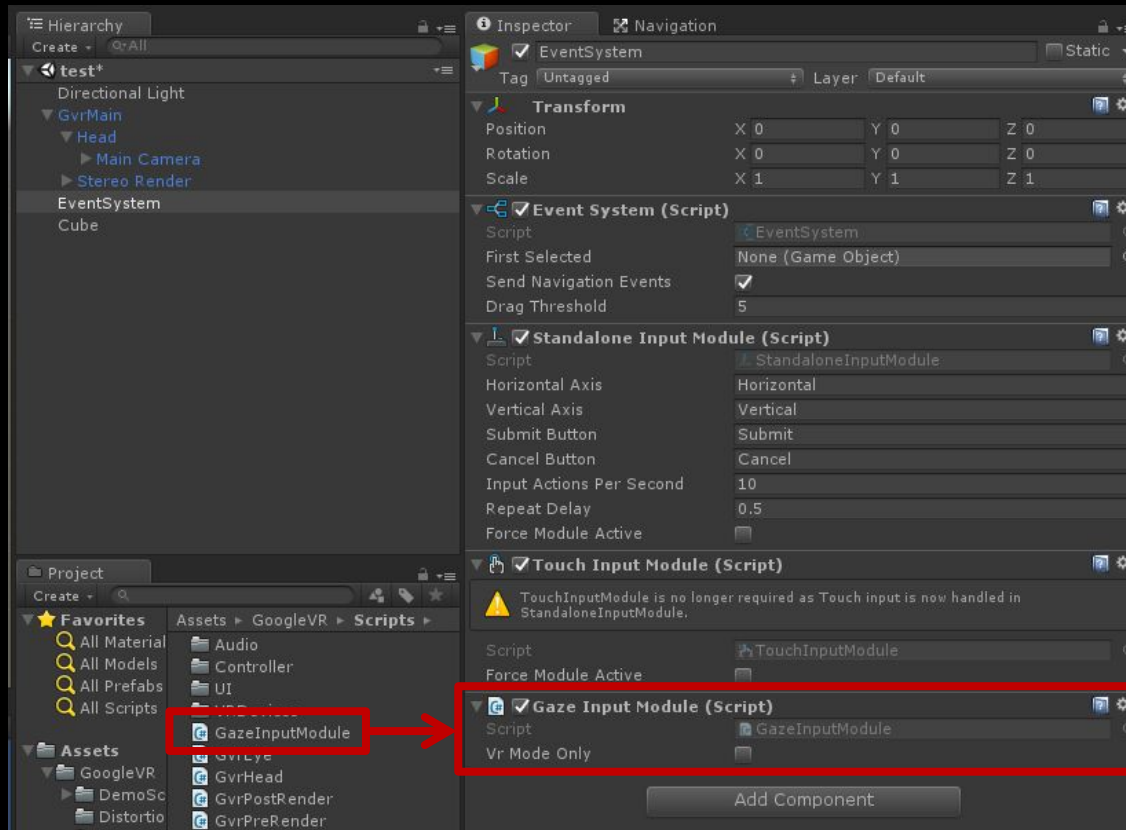
- 在場景中新增『Event System』，用來判斷UI上的基礎互動



利用射線及UI來製作視線效果

互動控制

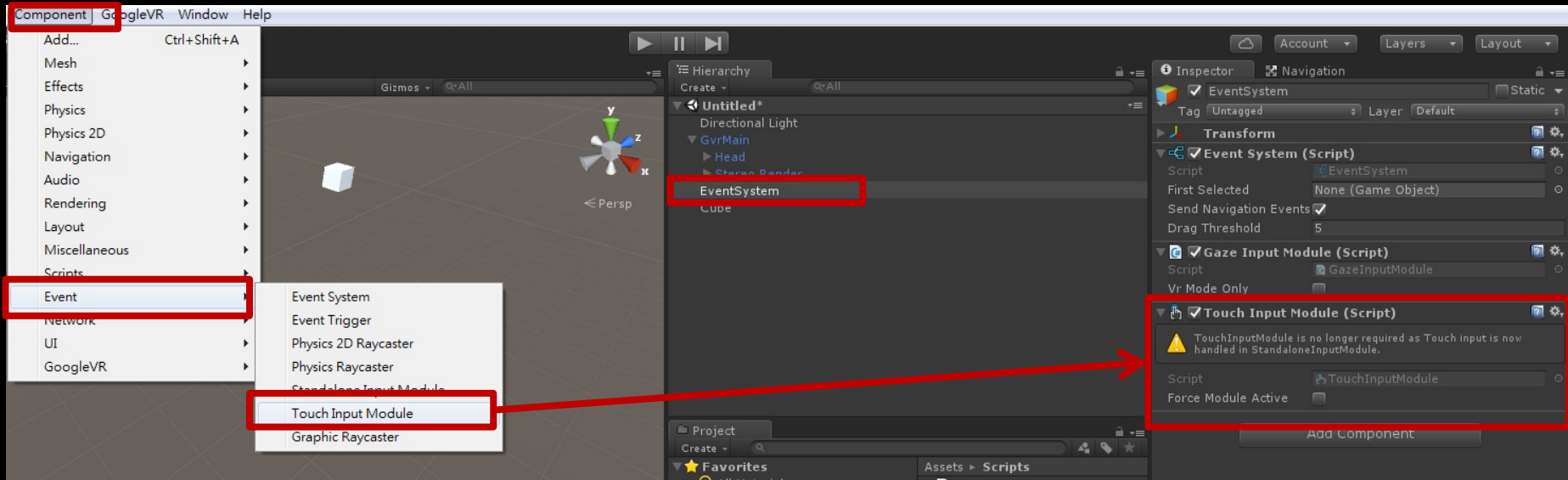
- 在『Event System』上新增『Gaze Input Module』



利用射線及UI來製作視線效果

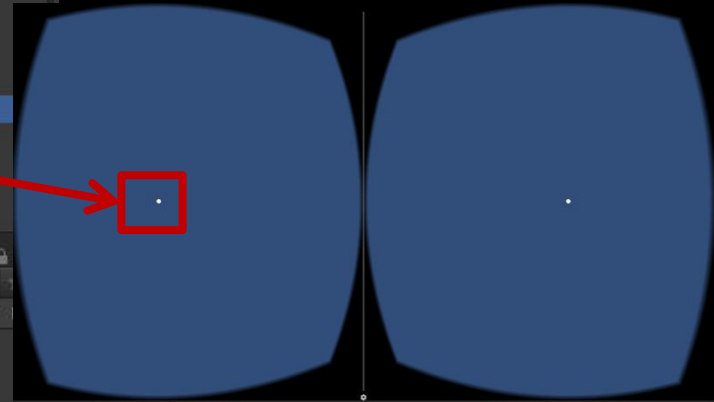
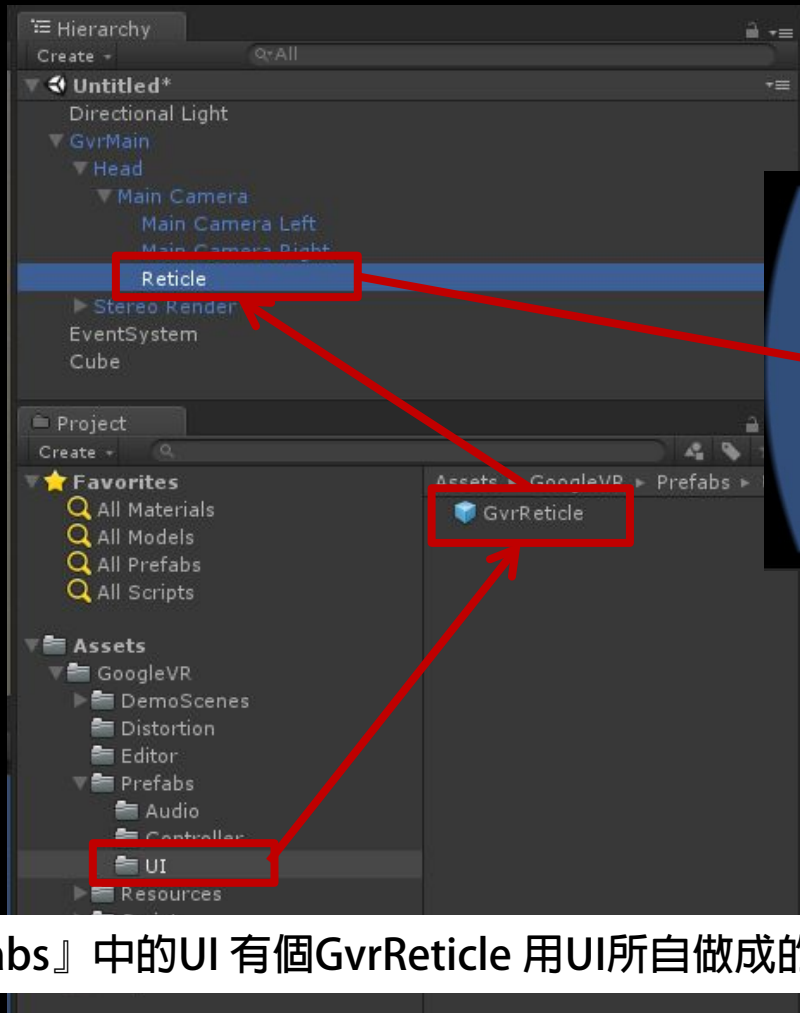
互動控制

- 在『Event System』上新增『Touch Input Module』



利用射線及UI來製作視線效果

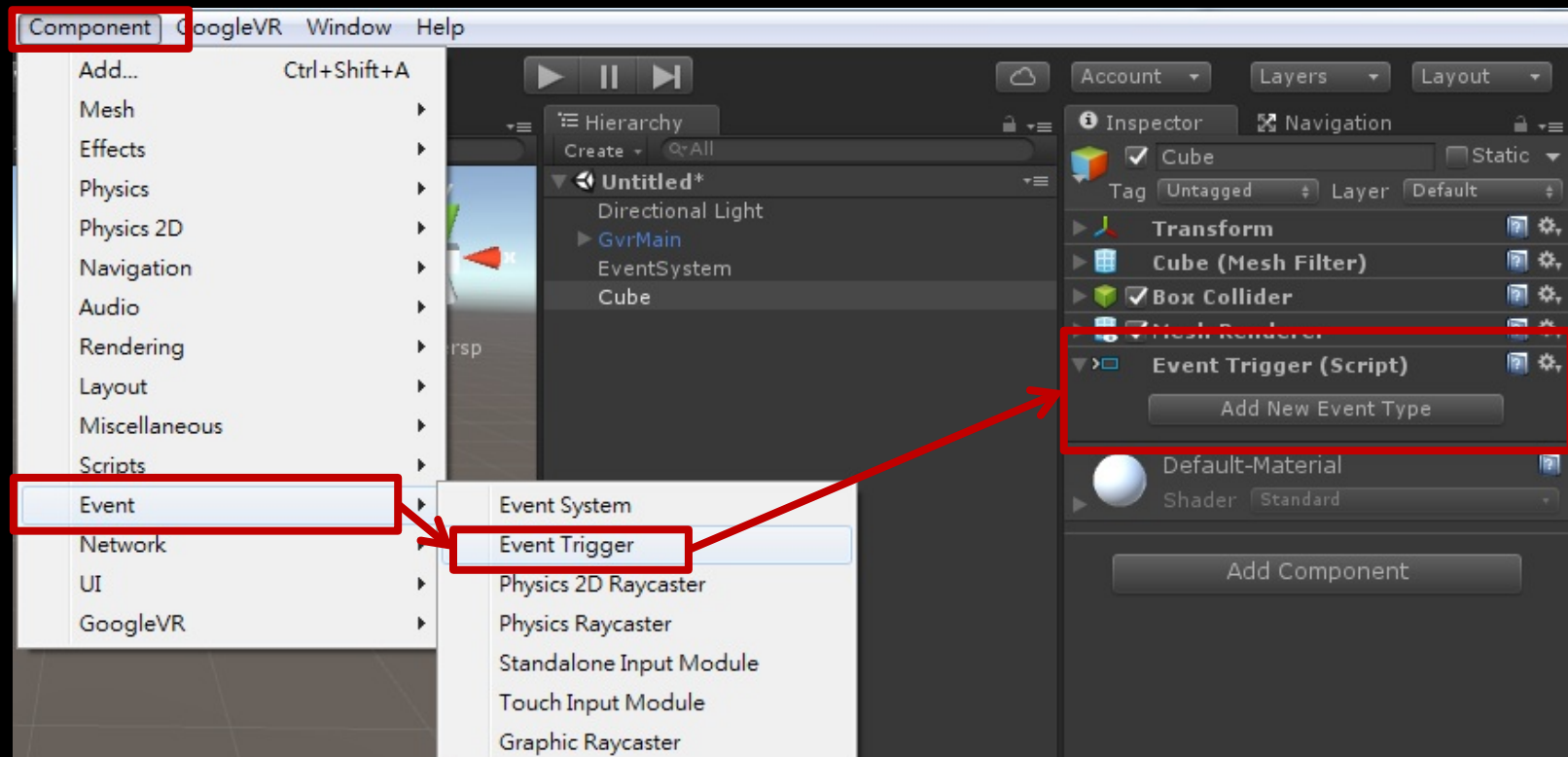
互動控制



『Prefabs』中的UI有個GvrReticle用UI所自做成的視線互動

互動物件

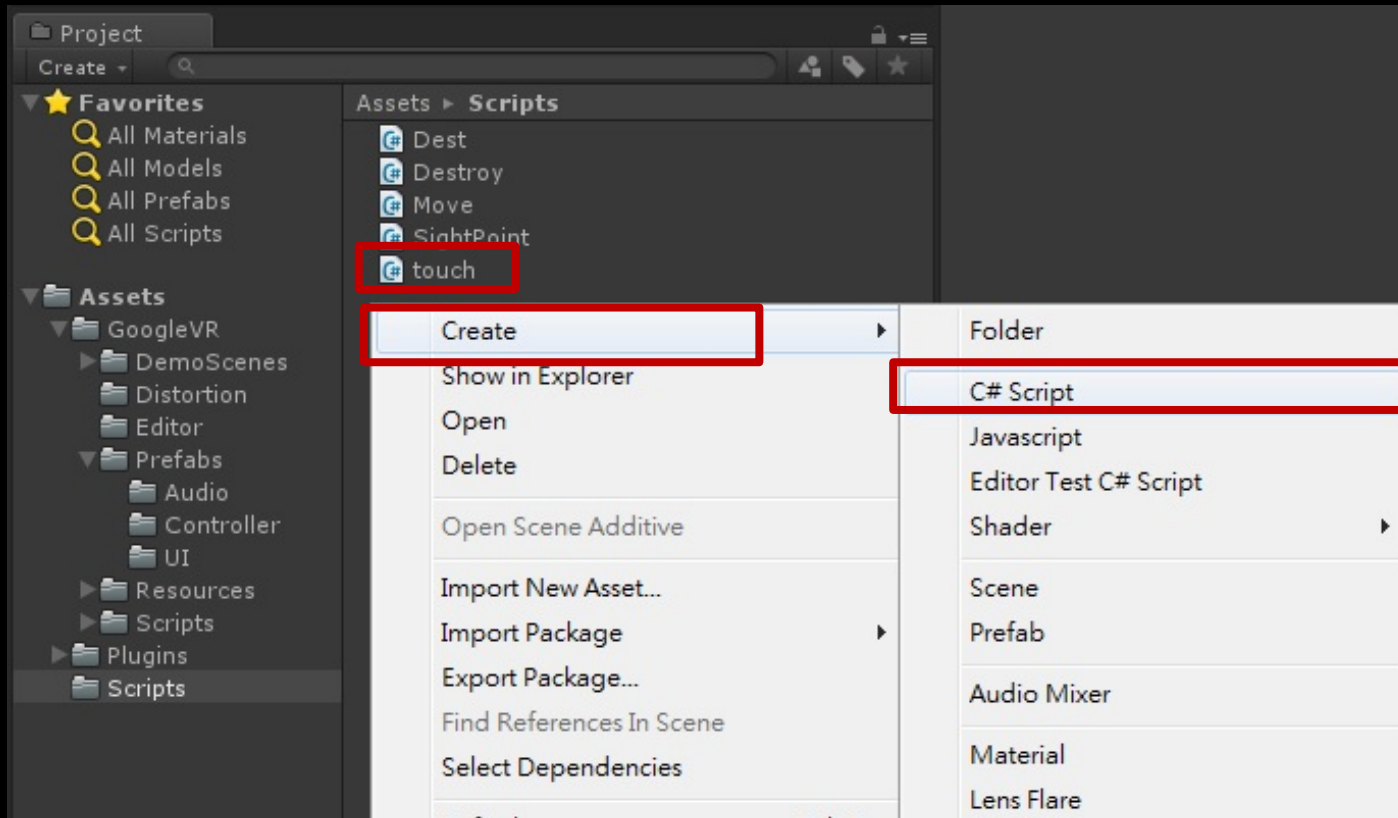
- 在互動物件上增加『Event Trigger』



利用觸發來產生互動

互動物件

- 建立新腳本



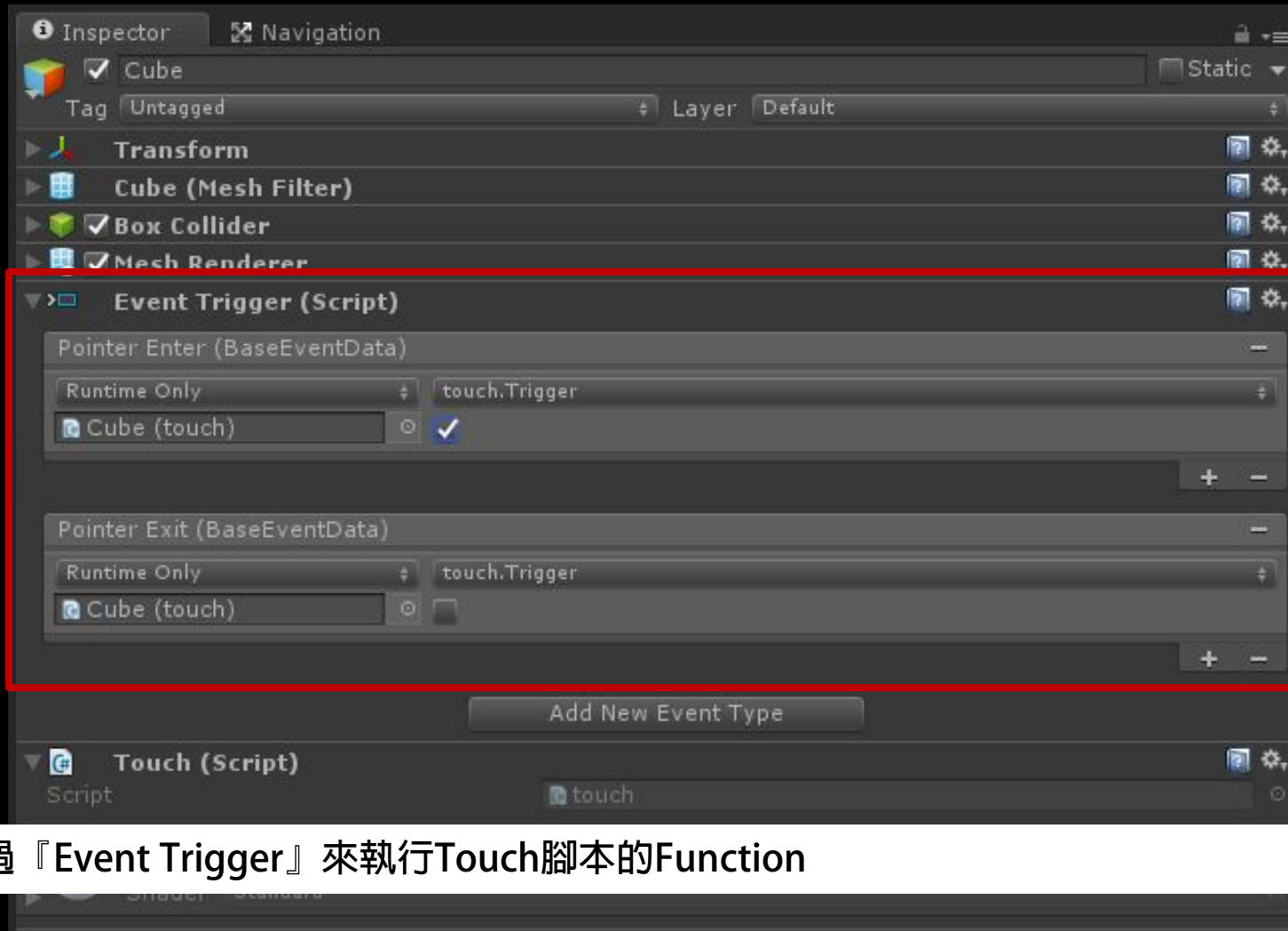
透過腳本來設定互動結果

互動物件

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class touch : MonoBehaviour
5 {
6
7     public void Trigger(bool Trigger)
8     {
9         GetComponent<Renderer>().material.color = Trigger ? Color.green : Color.red;
10    }
11
12 }
13 |
```

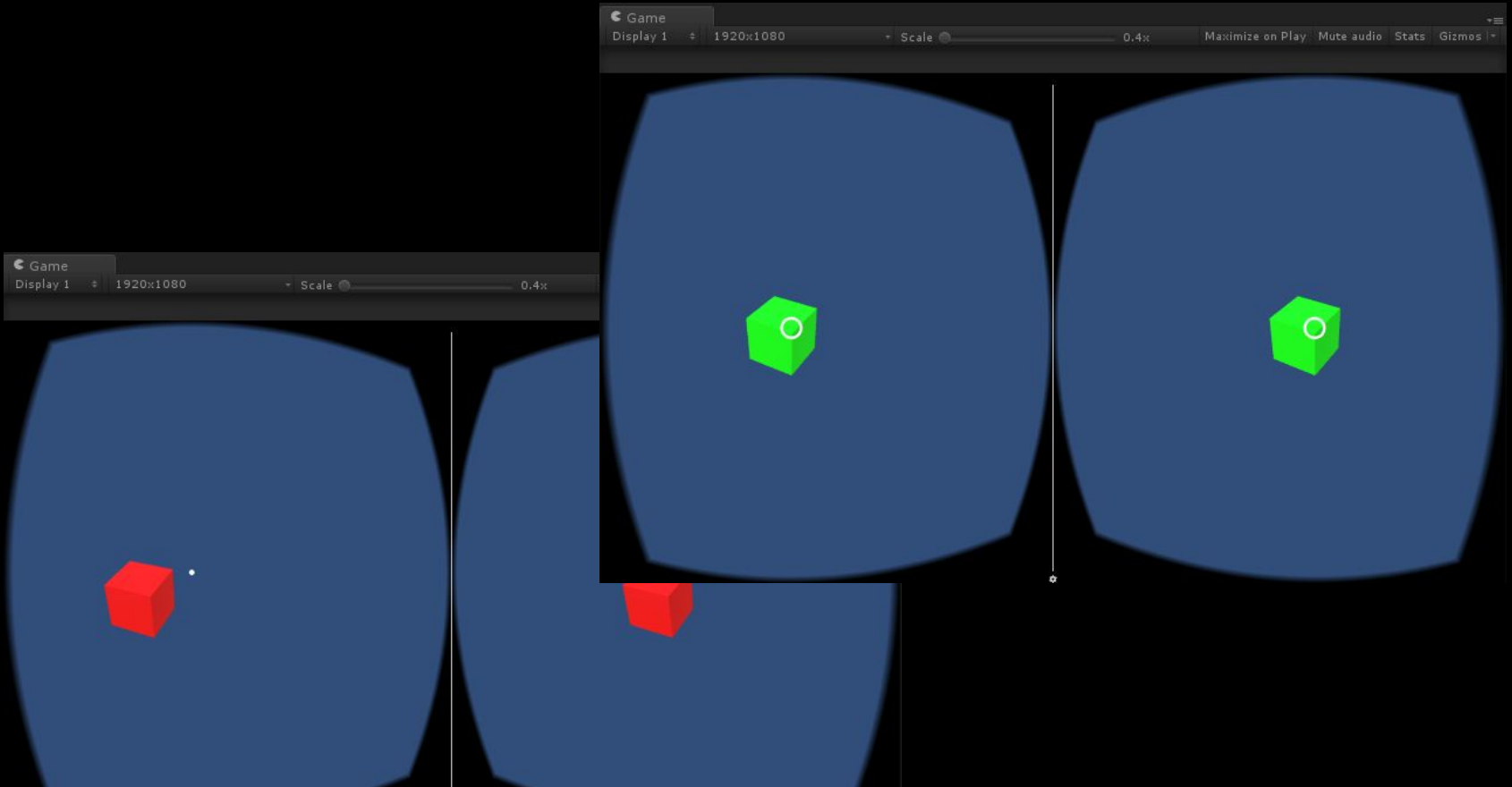
撰寫物件變色的Function

互動物件



透過『Event Trigger』來執行Touch腳本的Function

實際測試



使用「滑鼠+Alt鍵」可以在PC上模仿『陀螺儀』來做視角的轉動

簡易範例

- 透過剛剛做的互動功能，做一個簡易的互動範例。
- 利用視線觀看，刪除場景中的互動物件獲取分數。
- 在特定的時間內計算所獲取的分數。

簡易範例

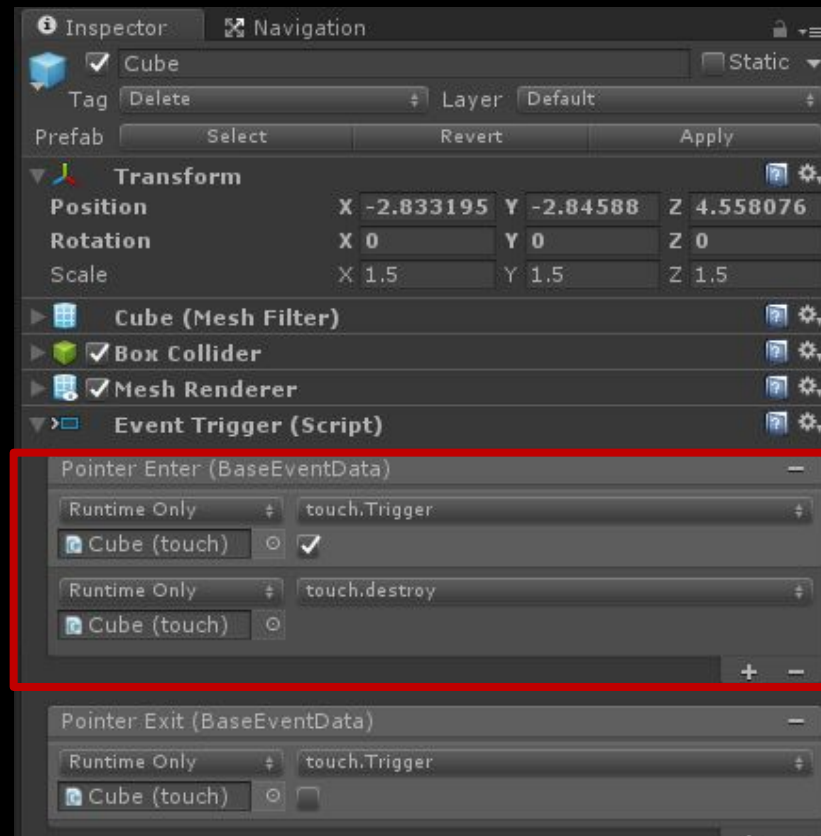
- 在touch腳本中增加刪除事件

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class touch : MonoBehaviour
5 {
6
7     public void Trigger(bool Trigger)
8     {
9         GetComponent<Renderer>().material.color = Trigger ? Color.green : Color.red;
10    }
11
12    public void destroy()
13    {
14        Destroy(this.gameObject);
15    }
16 }
17
18 }
```

使物件在被視線對上後，進行自身刪除

簡易範例

- 在視線觸碰到物件時執行touch腳本中的刪除事件



使物件在被視線對上後，進行自身刪除

簡易範例

- 使互動物件可朝一個隨機方向移動

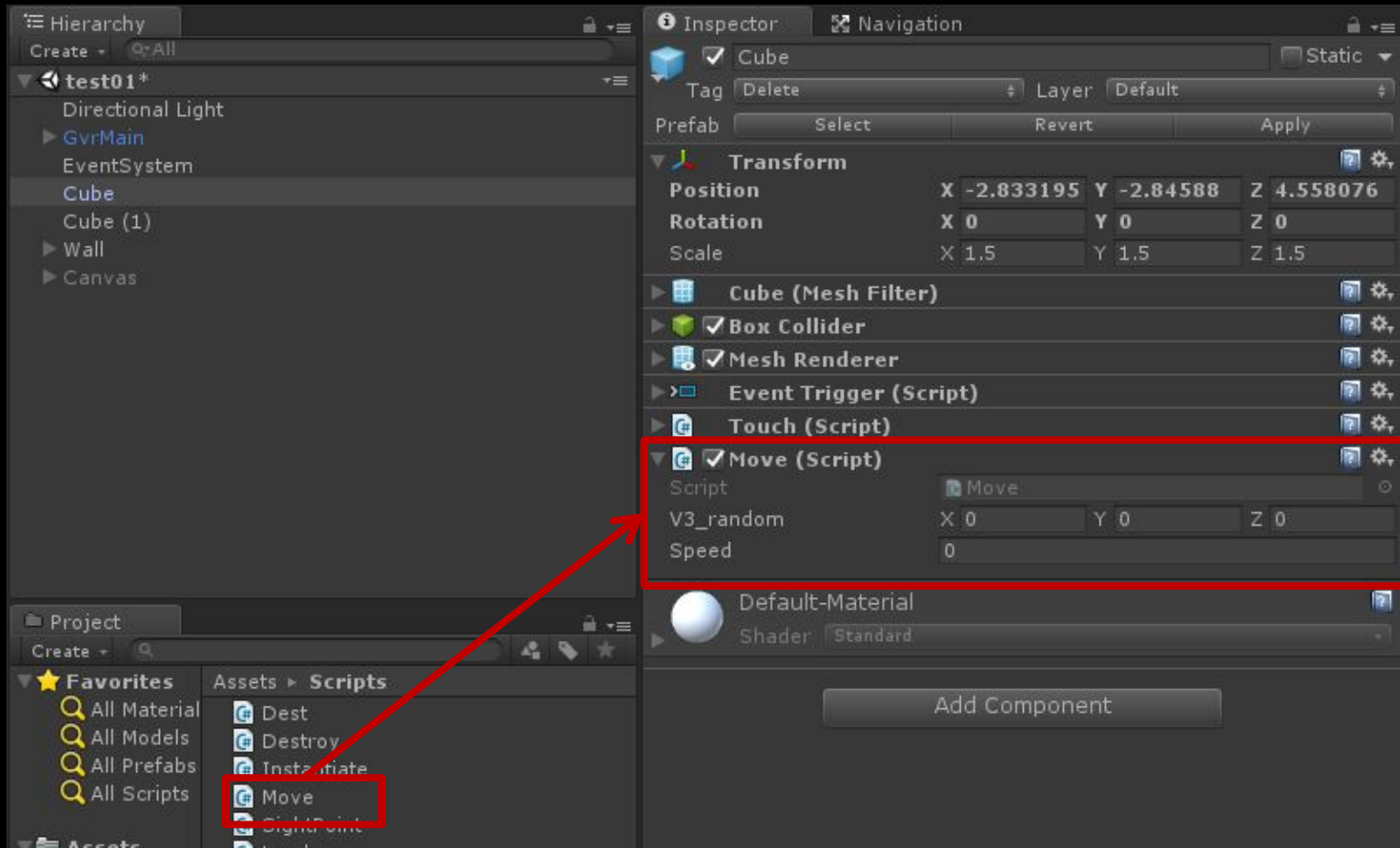
```
1 using UnityEngine;
2 using System.Collections;
3
4 public class Move : MonoBehaviour
5 {
6     public Vector3 V3_random;
7     public int speed;
8
9     void Start()
10    {
11        V3_random = new Vector3(Random.Range(-1.5f,1.5f),Random.Range(-1.5f,1.5f),Random.Range(-10f,0f));
12        speed = Random.Range(1,5);
13
14    }
15
16
17    void Update ()
18    {
19        transform.Rotate (new Vector3 (45, 45, 45) * Time.deltaTime);
20        transform.position += (V3_random * speed * Time.deltaTime);
21    }
22 }
```

隨機座標及速度

使物件累加隨機座標移動

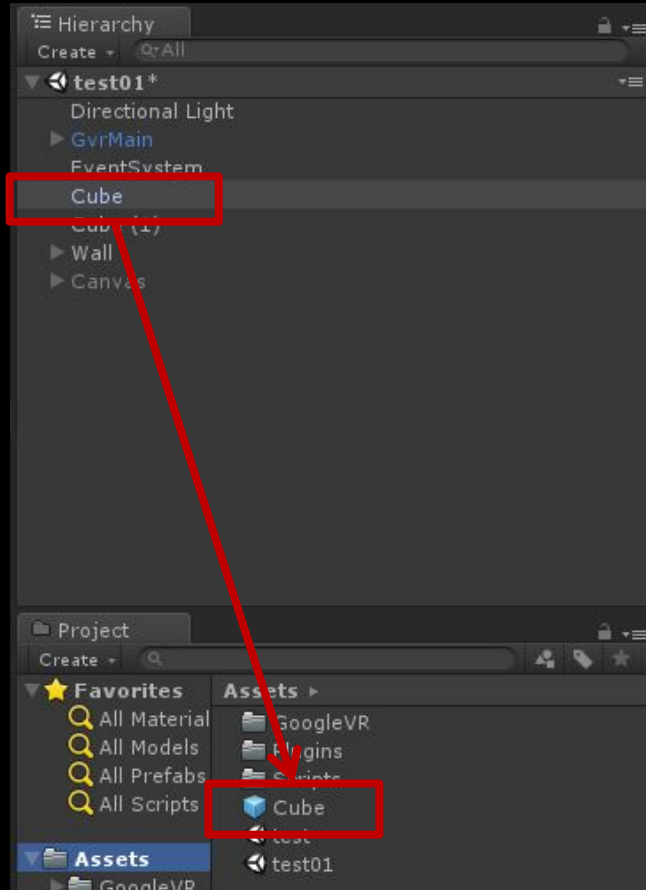
使互動物件移動增加刪除難度

簡易範例



將腳本放置互動物件身上

簡易範例



將製作好的互動物件建立成預製物

簡易範例

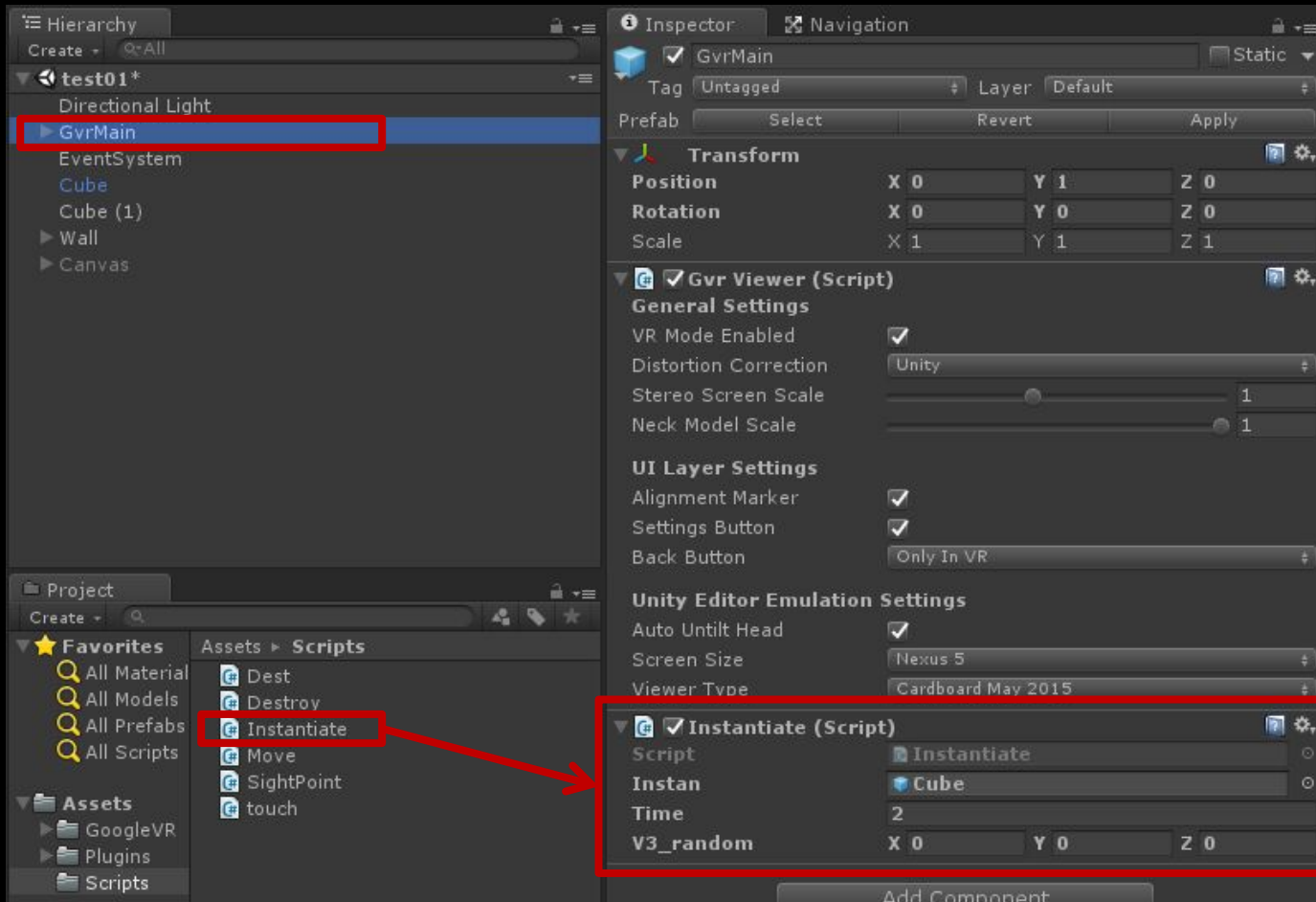
- 建立生成腳本

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class Instantiate : MonoBehaviour
5 {
6     public GameObject instan; //生成物
7     public float time = 1.0f; //間隔時間
8     public Vector3 V3_random; //生成位置
9
10    // Use this for initialization
11    void Start () {
12
13    }
14
15    // Update is called once per frame
16    void Update ()
17    {
18        V3_random = new Vector3(Random.Range(-15f,15f),Random.Range(-15f,15f),Random.Range(-15f,15f));
19
20        if(time > 0)
21        {
22            time -= Time.deltaTime;
23        }
24        else
25        {
26            GameObject Obj = Instantiate ( instan, V3_random, Quaternion.identity)as GameObject;
27            time = 1.0f;
28        }
29
30    }
```

倒數時間歸零後，產生物件

使互動物件可以隨機生成

簡易範例



將腳本放置攝影機之上

簡易範例

```
1 using UnityEngine;
2 using UnityEngine.UI;
3 using System.Collections;
4
5 public class ins : MonoBehaviour
6 {
7     public GameObject instan; //生成物
8     public float time = 1.0f; //間隔時間
9     public Vector3 V3_random; //生成位置
10
11     public Text countText;
12     public Text OverText;
13     public int Overcount;
14
15     public static int count;
16
17     bool generate = true;
18
19     // Use this for initialization
20     void Start () {
21
22     }
23
```

UI顯示資訊

結束遊戲的總分

目前分數

是否繼續生成物件

計算得分以及勝利

簡易範例

```
25 void Update ()
26 {
27     V3_random = new Vector3(Random.Range(-15f,15f),Random.Range(-15f,15f),Random.Range(-15f,15f));
28
29     if(generate)
30     {
31         if(time > 0 )
32         {
33             time -= Time.deltaTime;
34         }
35         else
36         {
37             GameObject Obj = Instantiate ( instan, V3_random, Quaternion.identity)as GameObject;
38             time = 1.0f;
39         }
40     }
41     SetCountText();
42 }
```

生成成立時

UI顯示資訊

計算得分以及勝利

簡易範例

```
43
44 void SetCountText()
45 {
46     countText.text = "Count: "+ count.ToString();
47     if(count >= Overcount)
48     {
49         OverText.text = "You Win!!";
50         generate = false;
51     }
52
53 }
```

顯示得分

判斷是否過關、過關後則生成不成立

計算得分以及勝利

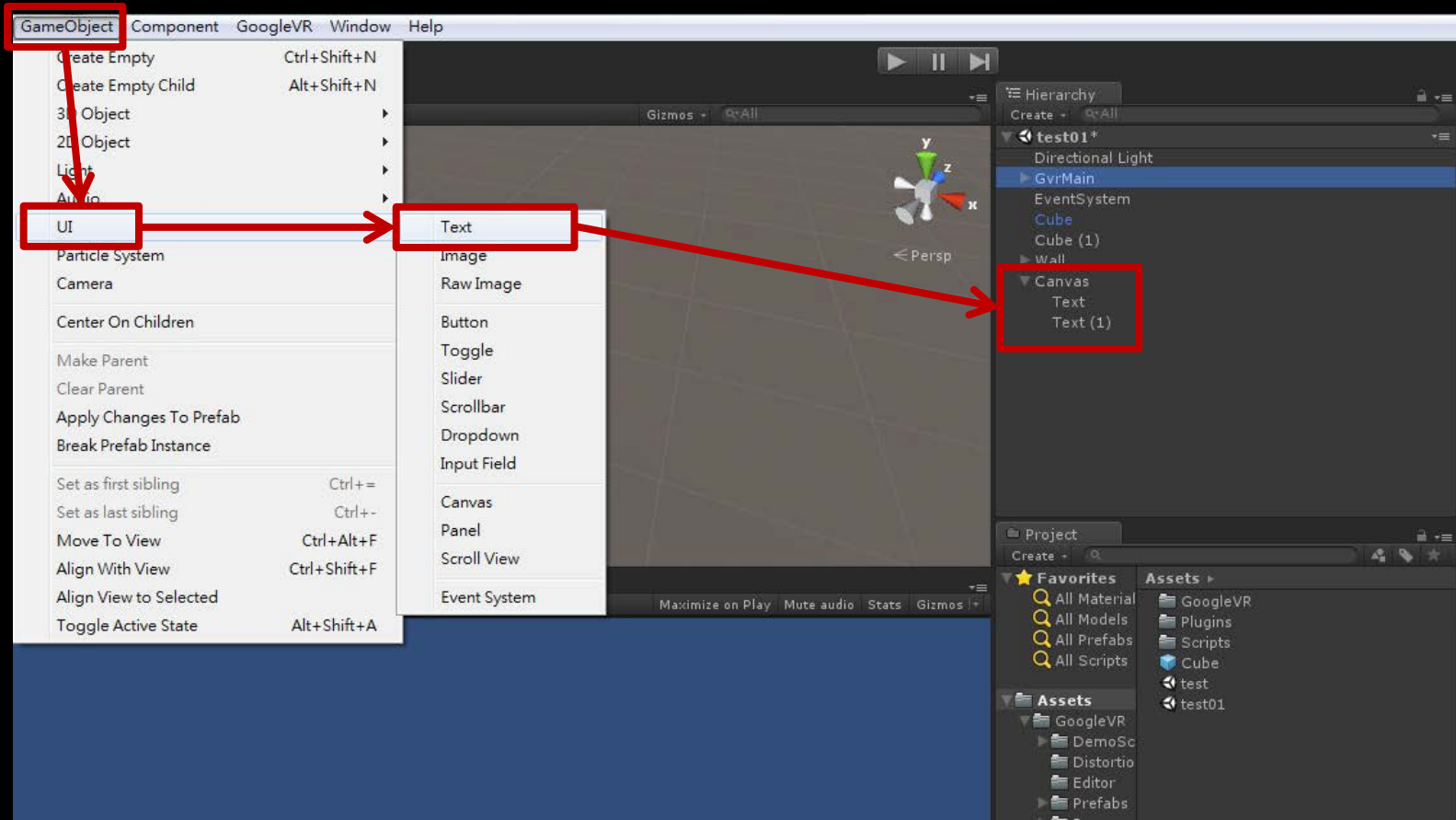
簡易範例

- 在刪除事件中增加分的程式

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class touch : MonoBehaviour
5 {
6
7     public void Trigger(bool Trigger)
8     {
9         GetComponent<Renderer>().material.color = Trigger ? Color.green : Color.red;
10    }
11
12    public void destroy()
13    {
14        Destroy(this.gameObject,1);
15        ins.count = ins.count + 1;
16    }
17
18 }
19
```

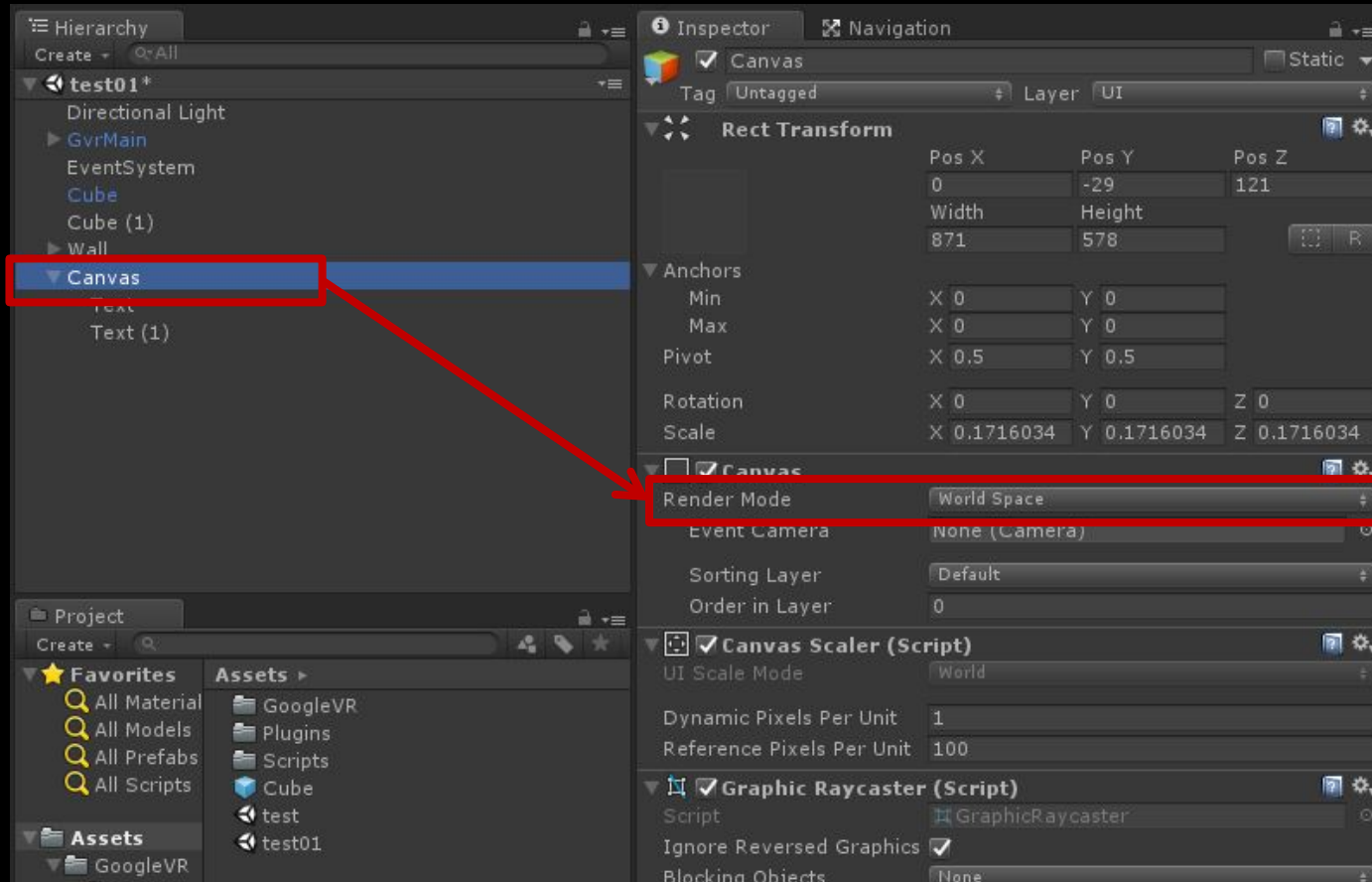
物件刪除後，就會增加得分

簡易範例



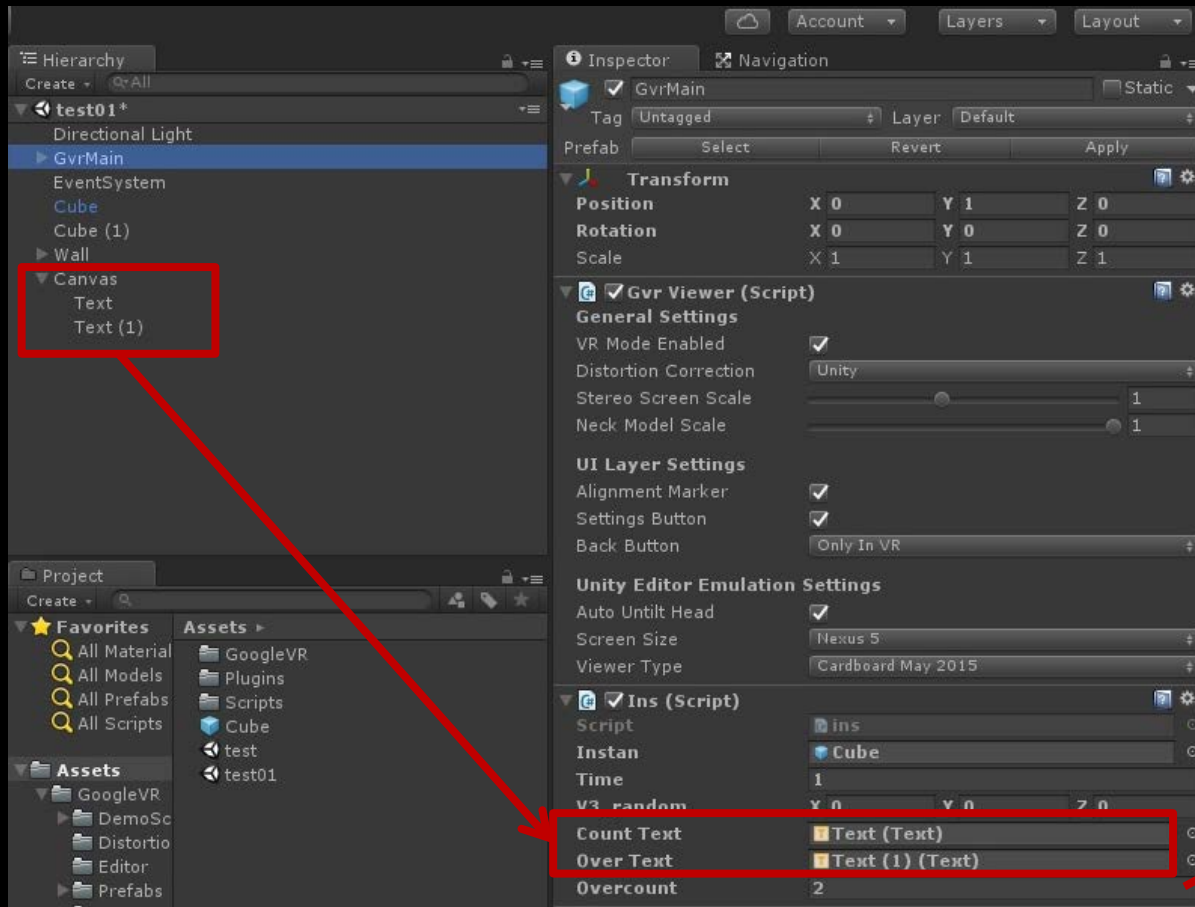
建立UI中的Text來顯示分數

簡易範例



建立UI中的Canvas修改成World Space 方便在VR中觀看

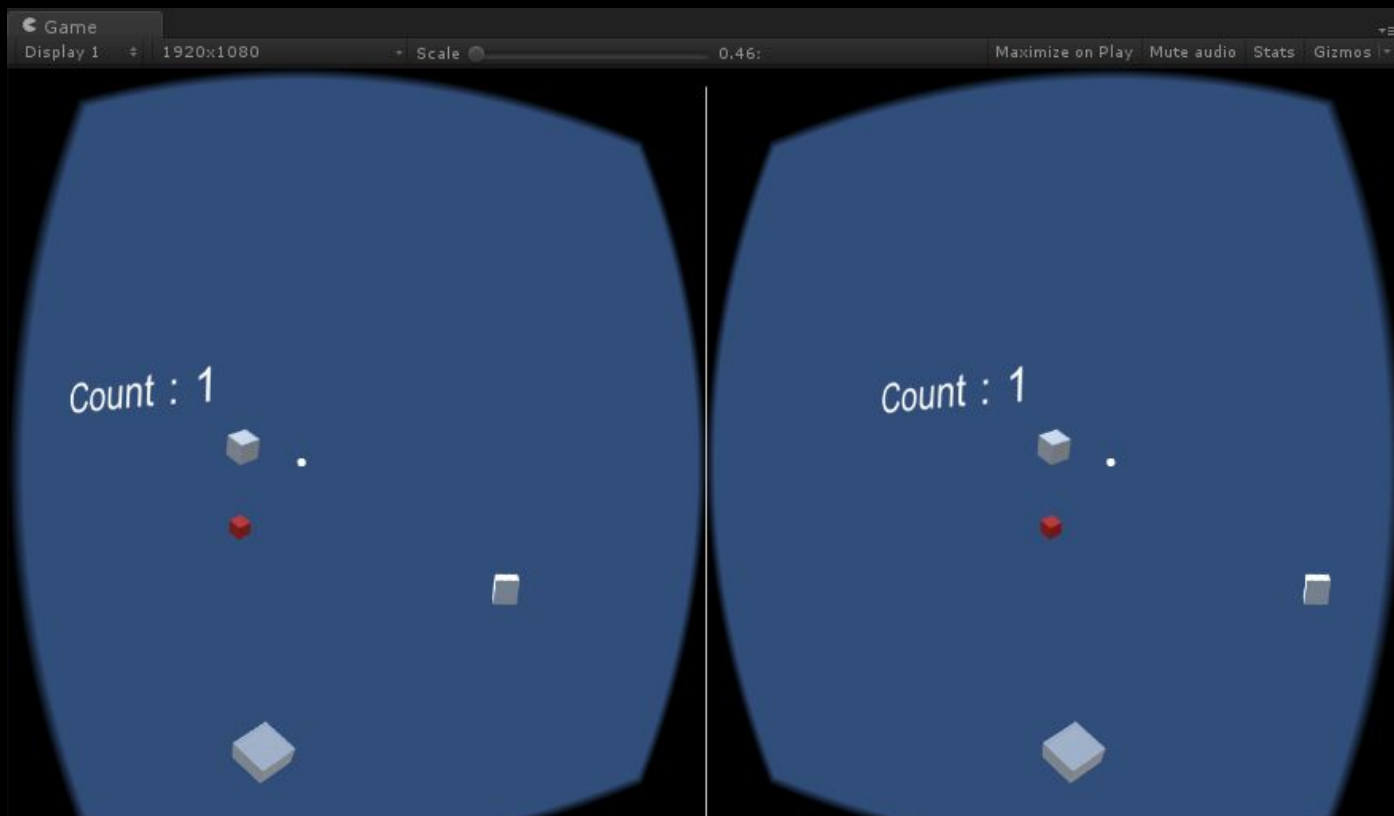
簡易範例



設定結束遊戲的分數

建立UI中的Text來顯示分數

測試



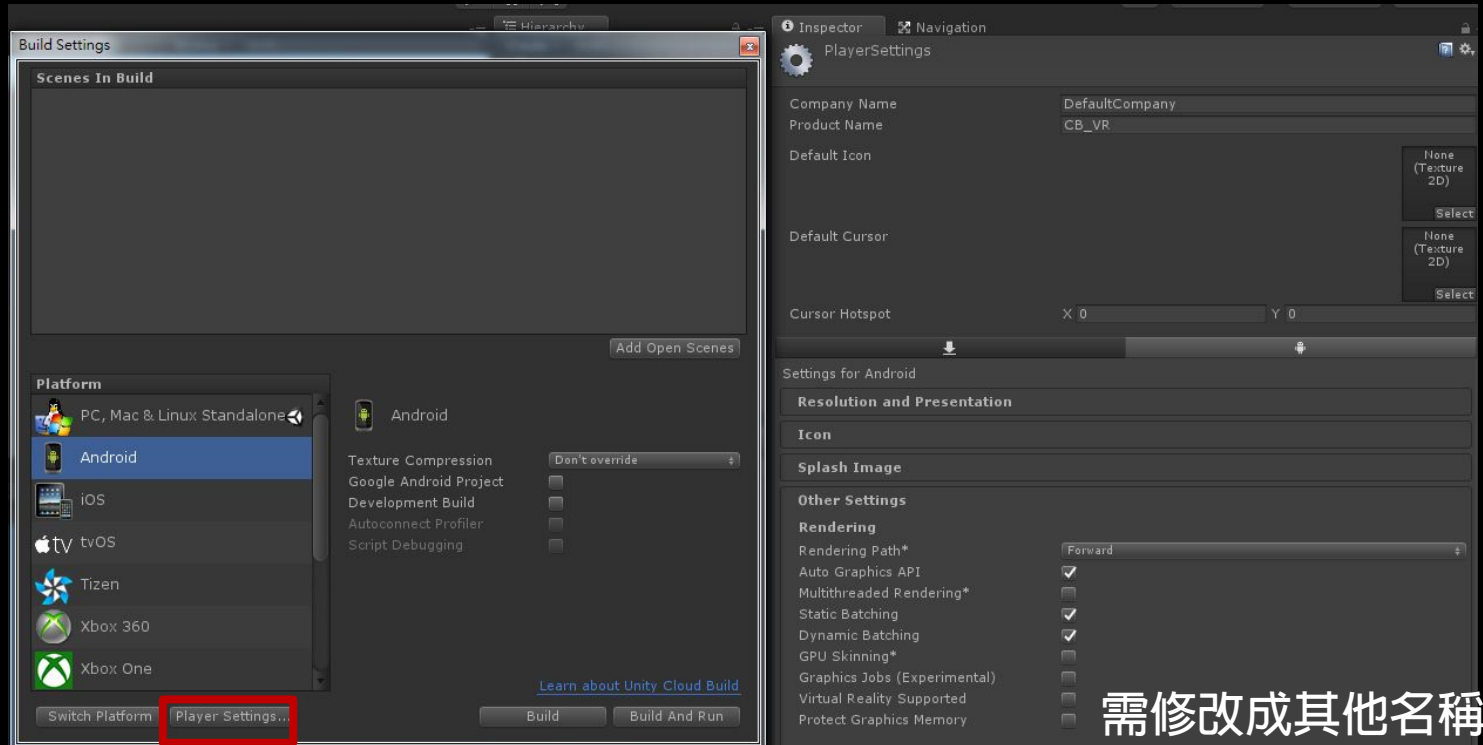
建立UI中的Text來顯示分數

輸出環境

- Java SE JDK
<http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>
- Android SDK
<https://developer.android.com/studio/index.html>

輸出apk時，需在電腦中設定輸出環境

輸出環境



Bundle Identifier 需要另外設定名稱

Thanks!!
