



Unity Engine Course

Introduction to Unity & C#



Syllabus & Grading

Attendance (10%)

9/25 (二)	18:00~21:00	<ul style="list-style-type: none">• Introduction to Unity & C#• AR/VR Application Development Using Unity	HW1 (20%)
9/27 (四)	18:00~21:00	<ul style="list-style-type: none">• UI & Framework & IO of Unity• Create your first 3D scene	
9/29 (六)	09:00~12:00	<ul style="list-style-type: none">• UI Development	HW2 (30%)
	14:00~17:00	<ul style="list-style-type: none">• 2D Game Development	
10/02 (二)	18:00~21:00	<ul style="list-style-type: none">• FPS Game Development	HW3 (40%)
10/14 (日)	14:00~17:00	<ul style="list-style-type: none">• Project Demo	

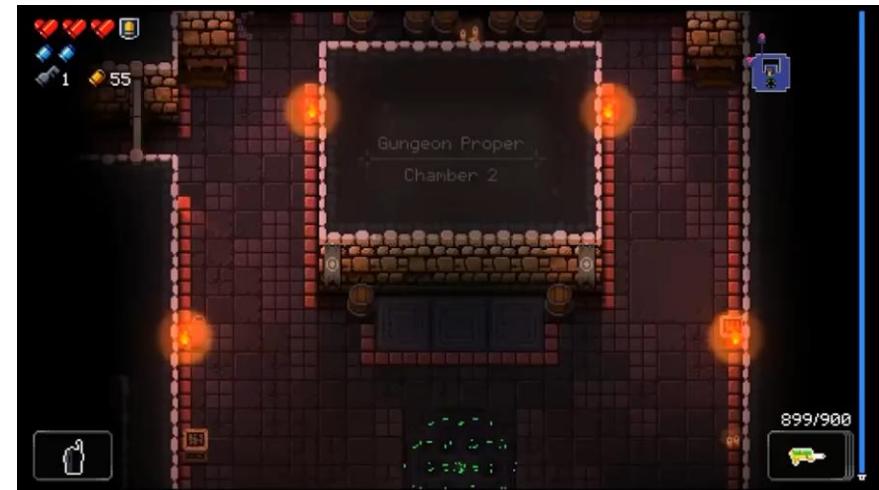
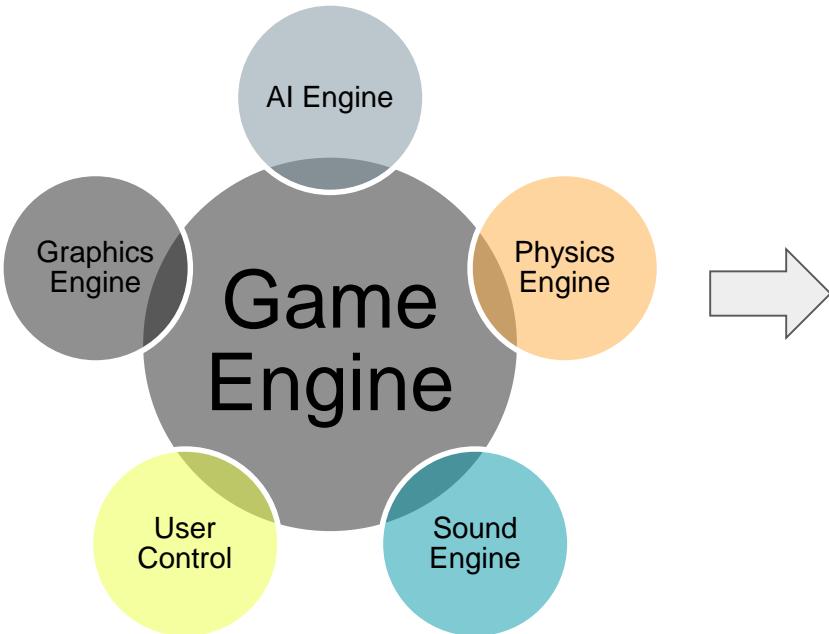
Course Info.

上課資料下載：<https://goo.gl/Y5VqeP>

助教：蔡宛倫 陳建文 彭建瑋 陳文正 張矽晶 蘇俐文
Email: unity@mislabs.csie.ncku.edu.tw

What is game engine & Why we need it?

A game engine is a software framework designed for the creation and development of games.

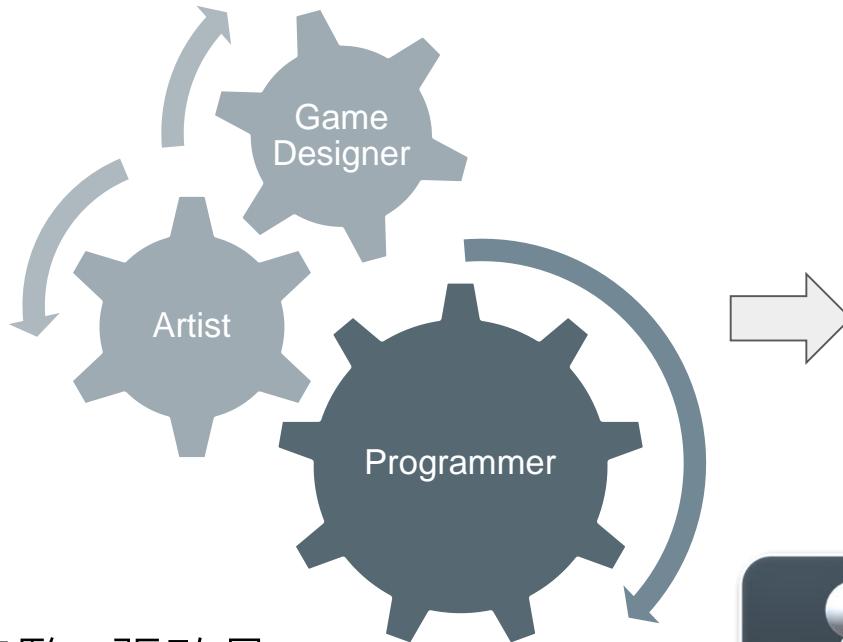


2D/3D Games

What Game Engine should I use?



Game Development is Interdisciplinary

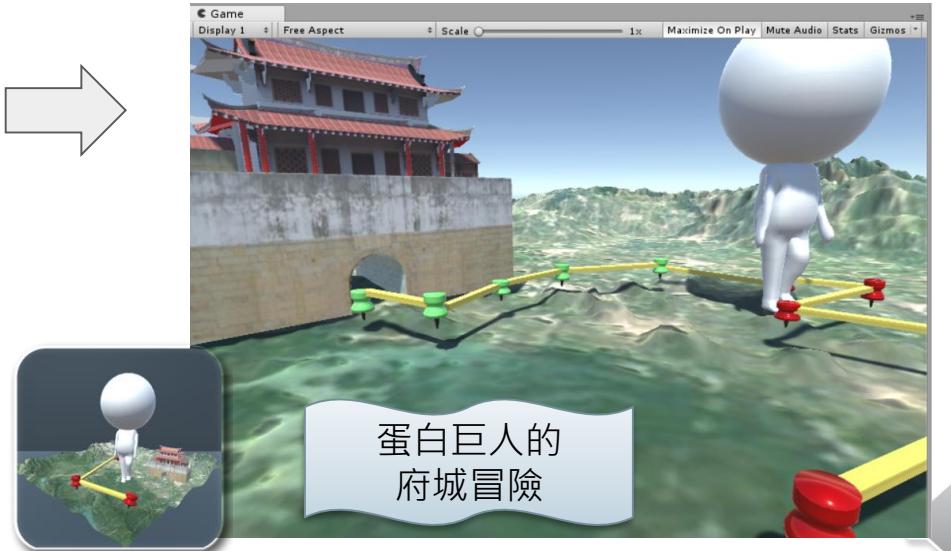


If you are an interdisciplinary talent

- Do the project on your own

Else, find your team member for the project ~

- Each team : no more than 2 people



美術總監：張矽晶
音樂總監：陳文正

Unity Learning Material

- Unity 官方教學 <https://unity3d.com/learn/tutorials>
- Unity Scripting API <https://docs.unity3d.com/ScriptReference/>
- Unity 聖典 <http://game.ceeger.com/>
- Unity 聖典論壇 <http://game.ceeger.com/forum/>
- u3DPro 論壇 <http://www.u3dpro.com/>
- Unity 3D 教程手冊 (遊戲蠻牛) <http://www.unitymanual.com/>
- 我愛Unity – EasyUnity <http://easyunity.blogspot.tw/>
- YouTube

What's programming?

- A way to compute and record data

hero

name	= LaiShen
speed	= 30
attackStrength	= 100
healthPoint	= 500
alive	= True

enemy 1

name1	= XiaoMa
speed1	= 50
attackStrength1	= 50
healthPoint1	= 500
alive1	= True

Variable

- 變數 (Variable)的宣告與使用
- 變數的資料型態
 - string : 文字與數字，儲存於引號之間
例如 "Hello Unity!"
 - int : 不含小數的整數值
 - float : 包含小數的數值(浮點數)
 - double: 包含小數但精度較高的數值(雙精度浮點數)
 - bool : 只有「true」或「false」，用於儲存某種狀態

What's programming?

- A way to compute and record data

hero

```
name      = "LaiSheng";  
speed     = 30.5f;  
attackStrength = 100;  
healthPoint = 500;  
alive     = true;
```

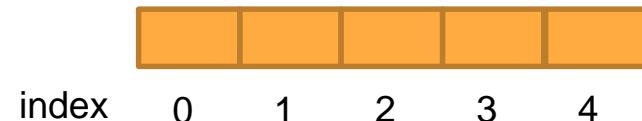
enemy 1

```
name1     = "XiaoMa";  
speed1    = 50.0f;  
attackStrength1 = 50;  
healthPoint1 = 500;  
alive1    = true;
```

Array

- An array contains multiple elements of the same data type

- `string[] nameArray = new string[5] ;`
 - `float[] speedArray = new float[5] ;`
 - `int[] attackStrengthArray = new int[5] ;`
 - `int[] healthPointArray = new int[5] ;`
 - `bool[] aliveArray = new bool[5];`



- Declare and assign values of an array

- `float[] floatArray = new int [5] ;`
`speedArray [0] = 10.0f; speedArray [1] = 20.0f; speedArray [2] = 30.0f; speedArray [3] = 40.0f; speedArray [4] = 50.0f;`
 - `float[] floatArray = new float[] { 10.0f, 20.0f, 30.0f, 40.0f, 50.0f};`

- How about printing the healthPoint of 100 enemies ?

Loop

- **for** Statement

```
for ( int i =0; i<maxValue; i++)  
{  
    // do something  
    Debug.Log ( healthPointArray[i] );  
}
```

- **foreach** Statement

```
foreach ( var item in intArray)  
{  
    // do something  
    Debug.Log ( item );  
}
```

- **while** Statement

```
int i=0;  
while (i< maxValue)  
{  
    // do something  
    Debug.Log ( healthPointArray[i] );  
    i=i+1; // i++; // i+=1;  
}
```

- **do while** Statement

```
int i=0;  
do{  
    // do something  
    Debug.Log ( healthPointArray[i] );  
    i=i+1;  
} while (i< maxValue);
```

Condition

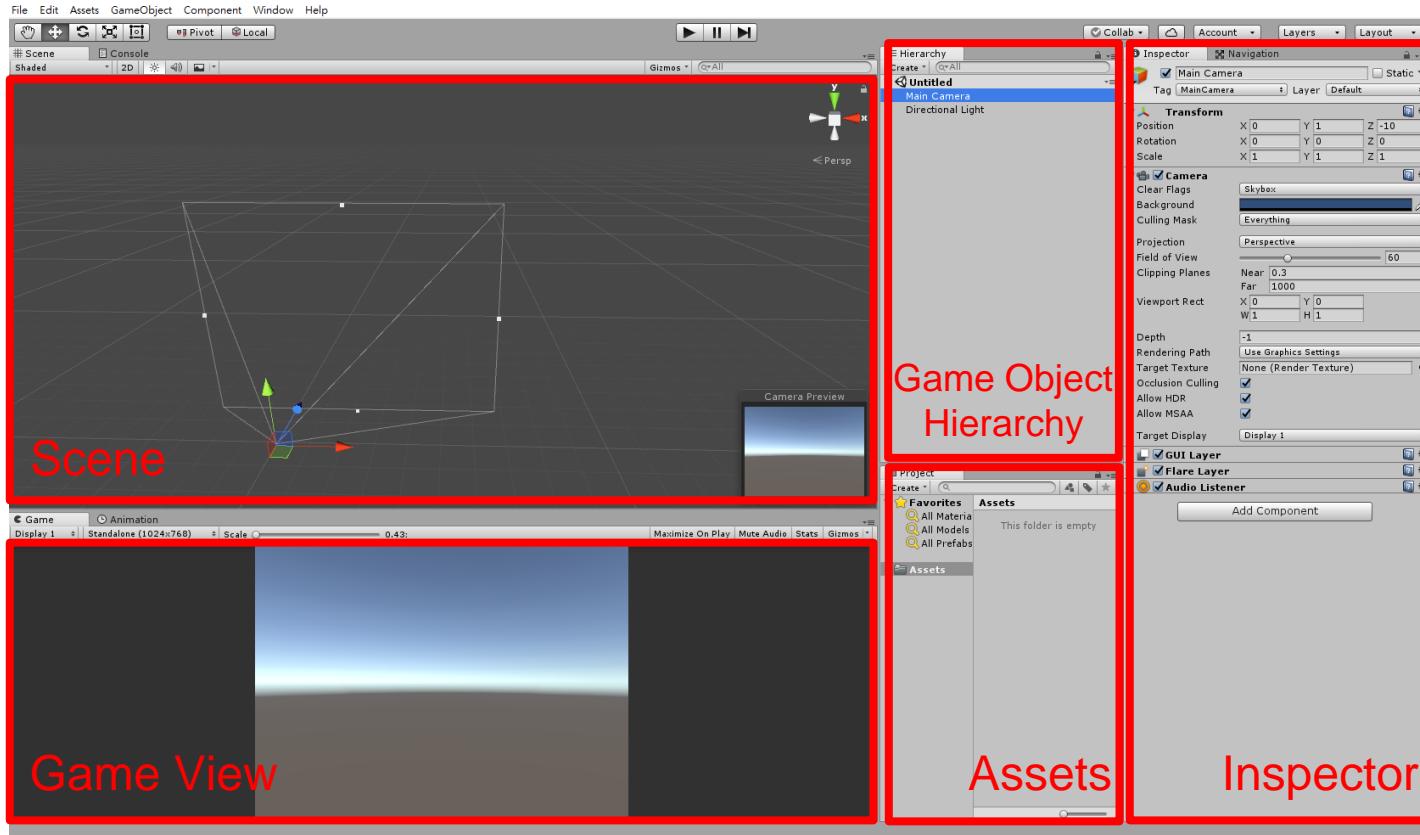
- **If Statement**

```
if ( healthPoint > 300)
{
    // do something
    Debug.Log ("I am alive !");
}
else if (healthPoint < 0)
{
    Debug.Log ("Oh no ~ I am dead @@ ");
}
else
{
    Debug.Log ("I am almost dead !!! ");
}
```

- **Switch Statement**

```
switch (variable)
{
    case 1:
        statement 1 ;
        break;
    case 2:
        statement 2 ;
        break;
    default:
        break;
}
```

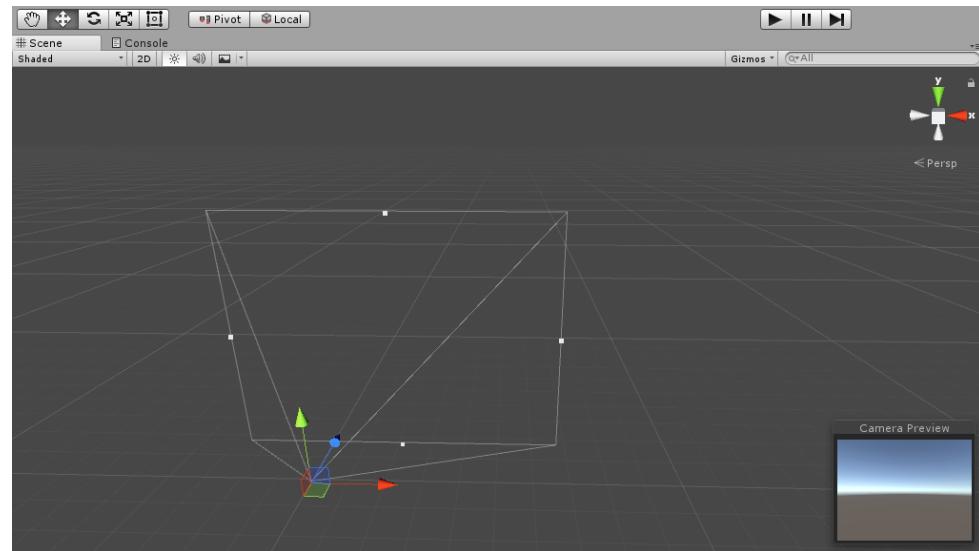
Unity Windows



Scene

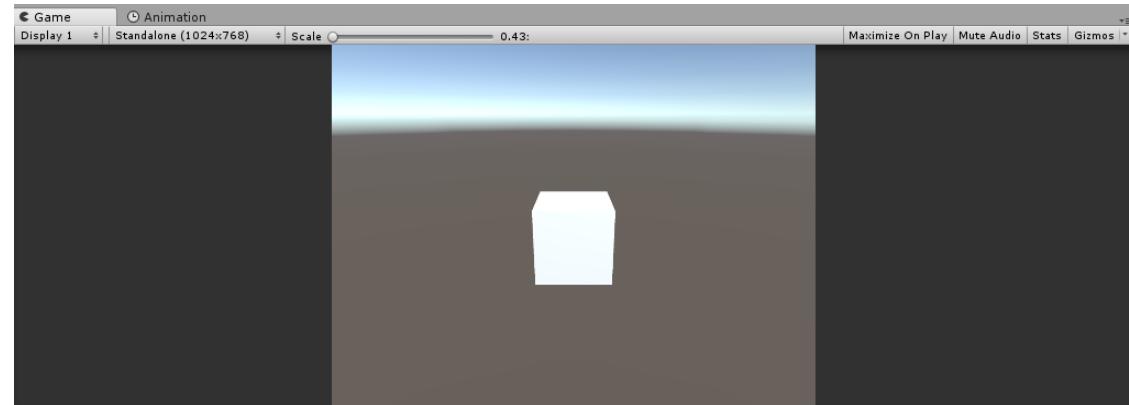
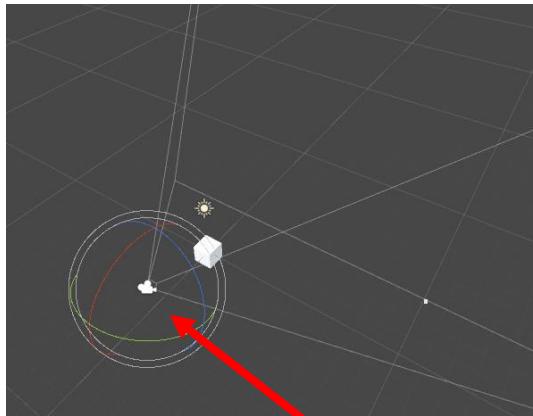
- You can scale, rotate, and move GameObjects in the scene
- You can also drag prefabs in your assets into the scene

Key shortcut	Function
Right Mouse Button	Rotation
Left Mouse Button	(Multiple) Selection
Left Mouse Button + WASD	Move
Mouse Wheel Button	Move
Mouse Wheel	Zoom in/out
Q	Drag Mode
W	Translation Mode
E	Rotation Mode
R	Scaling Mode
T	UI Transform Mode



Game View

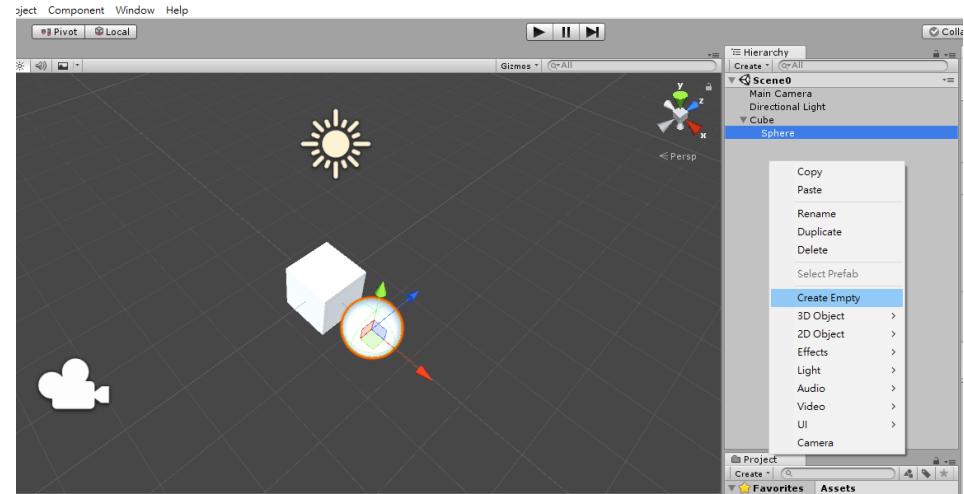
- The game view is rendered from the camera(s) in the scene



Current camera
in the scene

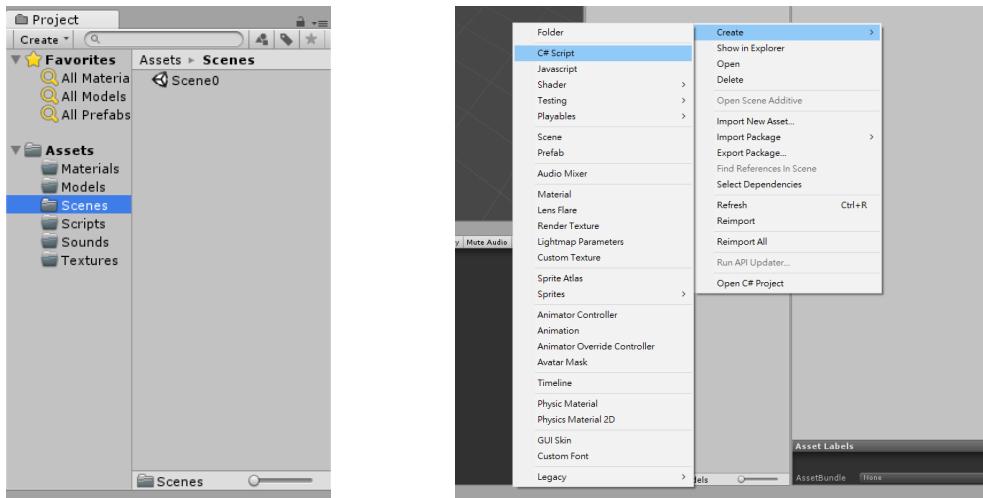
Game Object Hierarchy

- This window shows all your GameObjects in your scene
- You can add/delete a GameObject here
- The scene can be saved as an asset (*.unity file)



Assets

- Unity views all the files as assets, which can be imported to your game
- Put all your assets in the default “./Assets” folder under your project folder
- You also can create a new assets from the Unity asset template



Assets

- Prepare assets such as 3D models, sounds, and images

Project Folder: “./”

- Assets
- Library
- obj
- ProjectSettings
- Temp
- C# Assembly-CSharp.csproj
- Tuto1.sln

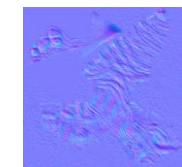


Asset Folder: “./Assets/”

- Materials
- Models
- Scenes
- Scripts
- Sounds
- Textures
- Materials.meta
- Models.meta
- Scenes.meta
- Scripts.meta
- Sounds.meta
- Textures.meta

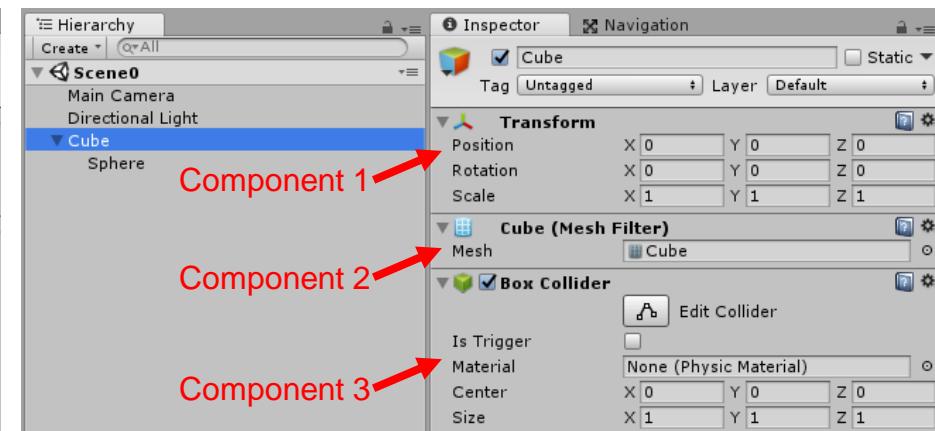
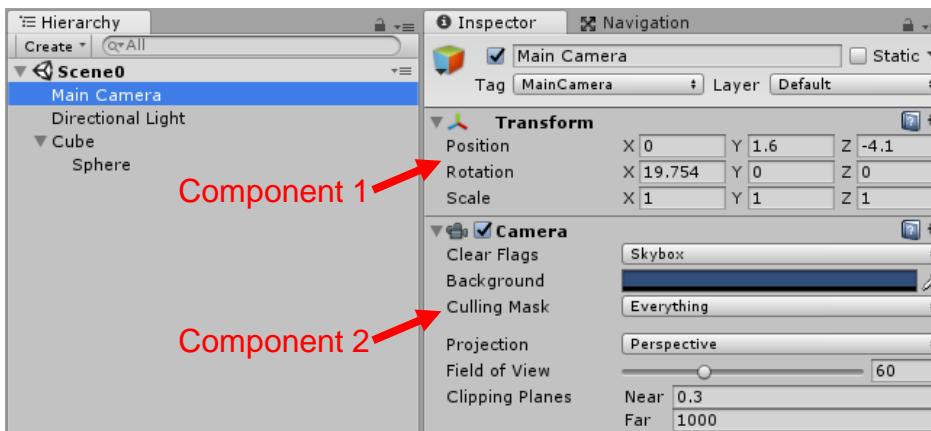


- *.obj
- *.fbx
- *.wav
- *.jpg
- *.png
- ...



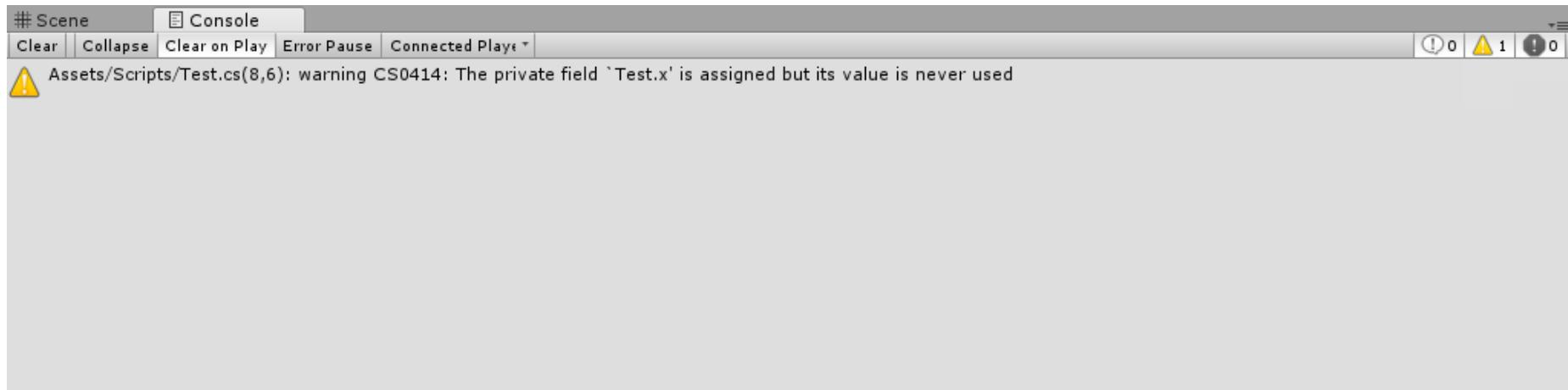
Inspector

- Inspector shows all the components attached to a GameObject
- A GameObject always has a **Transform** component



Console

- Console shows the message of the system information, compile information, and output logs
- System, compile error/warning & program output



C# Scripts

- 命名腳本檔

- 第一個字元不能是數字
 - 避免使用中文、空白與特殊符號
 - 使用容易理解功能的名稱

- 資料輸出

- Debug.Log()

- 註解

- //
 - /* */

- Start()

- 只在遊戲物件開始運作時執行一次的程式碼寫在這裡

- Update()

- 每個影格都執行一次的程式碼寫在這裡

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Script1 : 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 }
```

What's Object Oriented Programming?

hero

name = LaiShen
speed = 30
attackStrength = 100
healthPoint = 500
alive = True

enemy 1

name1 = XiaoMa
speed1 = 50
attackStrength1 = 50
healthPoint1 = 500
alive1 = True

Human
(class)

name
speed
attackStrength
healthPoint
alive

Attack();
Attacked();
IsDead();
IsEnemy();

Attribute

Method

Class & Object

- 類別 (class)

- 沒有實體 (房屋設計藍圖)
- 實作於程式碼中，
定義物件預設的屬性與方法
- 靜態

- Object (物件)

- 實體 (實際蓋好的房子)
 - ✓ public GameObject gameObject;
 - gameObject = new GameObject (" ");
 - ✓ GameObject obj = Instantiate(prefab) as GameObject;
- gameObject . Attribute
- gameObject . Method ()
- 動態



Qualcomm
vuforia



Unity + Vuforia

[Ref] https://otaru.tw/tutorial/unity-2017-2_vuforia_tutorial

Register



Go to <https://developer.vuforia.com/vui/auth/register>

The screenshot shows the 'vuforia™ Developer Portal' registration page. At the top, there is a navigation bar with links for Home, Pricing, Downloads, Library, Develop, and Support. On the right side of the navigation bar are 'Log In' and 'Register' buttons. The main title 'Register for a Vuforia Developer Account' is centered above a descriptive text: 'With an account you can download development tools, get license keys, and participate in the Vuforia community.' Below this, there are several input fields: 'First Name' and 'Last Name' in a row; 'Company' and 'Select Country of Residence' (with a dropdown arrow) in another row; 'Email Address' and 'Username' (each with a help icon) in a third row; and 'Password' and 'Confirm Password' in a fourth row. At the bottom, there is a 'Captcha Code' field next to a CAPTCHA image showing the text 'PVJTA'.

Create License (1)



Go to <https://developer.vuforia.com/license->

vuforia™ Developer Portal

Hello trimy | Log Out

Home Pricing Downloads Library Develop Support

License Manager Target Manager

License Manager

Create a license key for your application.

[Get Development Key](#) [Buy Deployment Key](#)

Name	Type	Status	Date Modified
------	------	--------	---------------

 Create your first license key

Create License (2)

The screenshot shows the Vuforia™ Developer Portal interface. At the top, there's a green header bar with the Vuforia logo and the text "vuforia™ Developer Portal". On the right side of the header, it says "Hello trimy" with a dropdown arrow and "Log Out". Below the header, there's a navigation menu with tabs: Home, Pricing, Downloads, Library, Develop (which is highlighted in grey), and Support. Under the Develop tab, there are two sub-tabs: License Manager (which is selected and highlighted in grey) and Target Manager. Below the navigation, there's a link "Back To License Manager". The main content area has a title "Add a free Development License Key". There's a section for "App Name" with an input field containing a placeholder " ". To the right of the input field, a note says "You can change this later". Below this, there's a section for "License Key" with the status "Develop". It also lists some usage details: "Price: No Charge", "Reco Usage: 1,000 per month", "Cloud Targets: 1,000", "VuMark Templates: 1 active", and "VuMarks: 100". At the bottom of the form, there's a checkbox with the text "By checking this box, I acknowledge that this license key is subject to the terms and conditions of the [Vuforia Developer Agreement](#)". At the very bottom, there are two buttons: "Cancel" and "Confirm".

Create License (3)

The screenshot shows the Vuforia™ Developer Portal interface. At the top, there is a green header bar with the Vuforia logo and the text "vuforia™ Developer Portal". On the right side of the header, it says "Hello trimy" with a dropdown arrow and "Log Out". Below the header, there is a navigation menu with tabs: Home, Pricing, Downloads, Library, Develop (which is highlighted in dark grey), and Support. Under the Develop tab, there are two sub-tabs: License Manager (which is selected and highlighted in light grey) and Target Manager.

The main content area is titled "License Manager" and contains the sub-instruction "Create a license key for your application." Below this, there are two buttons: "Get Development Key" and "Buy Deployment Key".

A table is displayed below these buttons, showing a list of existing license keys. The table has columns for Name, Type, Status, and Date Modified. One row is visible in the table:

Name	Type	Status	Date Modified
AR01	Develop	Active	May 01, 2018 18:16

At the bottom of the page, there is a footer bar with the text "Last updated: Today 6:16 PM" and a "Refresh" button.

Create License (4)

The screenshot shows the Vuforia Developer Portal interface. At the top, there's a green header bar with the Vuforia logo and "Developer Portal". On the right side of the header, it says "Hello trimy" with a dropdown arrow and "Log Out". Below the header, there's a navigation menu with tabs: Home, Pricing, Downloads, Library, Develop (which is highlighted in green), and Support. Under the Develop tab, there are two sub-tabs: License Manager (which is selected and highlighted in grey) and Target Manager.

In the main content area, the path "License Manager > AR01" is displayed. The title "AR01" is shown in bold, with "Edit Name" and "Delete License Key" links next to it. There are two tabs below the title: "License Key" (selected) and "Usage".

The "License Key" tab contains a large text box with the following content:

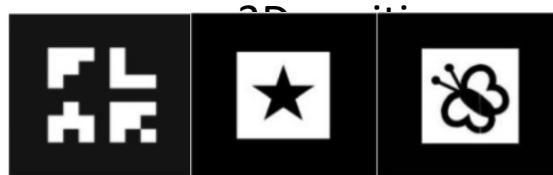
```
AaeAnZX////AADm+ebLRVuubeiBQbKtkauzs0Jtimy3Xbkxky
zr5C12V3RUd4wA6YYX+KCJS4hmnh3qhpUbdgk3TYmI+d/mmoLuc
z3XivLdgA16x+biEryfLJtA9J2+aM5m8Ao+rV00sHn9h9UwzUbb
JaJo9eRpyF7C3ijekg1Xbmte+iaa9aHbL3OloCHUqavuk3AVJPDd
ueaY7kFwUOEExurnFH+fxfa8EP+Dx31FZ6yVmqlMIKNsnlmLosB
Sq6o4HsWsgMaE3JKDOSNnWiUae5+c81QPShFu6SPqXXN6Oic3jN
xfmaTfGacLFHn9pZ5vvXFx8NderuRgOs+GugRh4f7CZF241G7o04
DzdfAIa+U2y8FA61
```

Below this text box, there are three status labels: "Type: Develop", "Status: Active", and "Created: May 01, 2018 18:16".

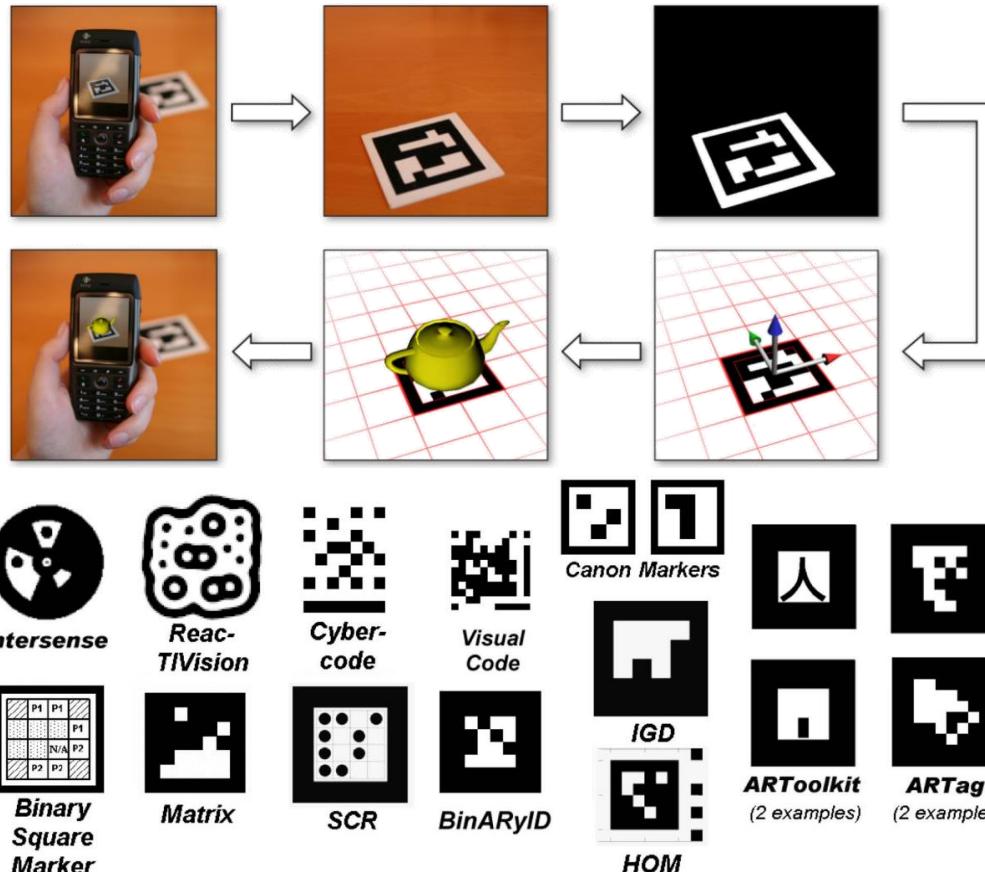
At the bottom left, there's a section labeled "History:" with the entry "License Created - Today 18:16".

Marker-based AR

- Uses black and white geometric markers
- Recognizes pre-registered patterns in a video
- Provide the **3D position relative to the camera** for each identified marker
- Play interactive 3D animations in real-time according to



ARToolkit (Marker-based)



Markerless AR



Feature Points



Build Image Database (1)

- 💡 Vuforia AR uses **Marker-less Recognition Technology**
- 💡 [Optimizing Target Detection and Tracking Stability](#)
 - Attributes of an Ideal Image Target
 - Natural Features and Image Ratings
 - Local Contrast Enhancement
 - How To Evaluate a Target Image in Grayscale
 - How To Use the Feature Exclusion Buffer
 - How To Create Non-Rectangular Image Targets
 - How To Optimize the Physical Properties of Image Targets

Build Image Database (2)

vuforia™ Developer Portal

Hello trimy | Log Out

Home Pricing Downloads Library **Develop** Support

License Manager **Target Manager**

Target Manager

Use the Target Manager to create and manage databases and targets.

Add Database

Database	Type	Targets	Date Modified
----------	------	---------	---------------

Create your first Database

Build Image Database (3)

Create Database

Name:

Type:

- Device
- Cloud
- VuMark

Build Image Database (4)

The screenshot shows the Vuforia Developer Portal interface. At the top, there's a green header bar with the Vuforia logo and "vuforia™ Developer Portal". On the right side of the header, it says "Hello trimy" with a dropdown arrow and "Log Out". Below the header, there's a navigation menu with links: Home, Pricing, Downloads, Library, Develop (which is highlighted in blue), and Support. Under the Develop menu, there are two sub-links: License Manager and Target Manager, with "Target Manager" being the active one. In the main content area, there's a breadcrumb navigation: "Target Manager > ARImages". The title "ARImages" is displayed in bold, with a link to "Edit Name". Below the title, it says "Type: Device". A section titled "Targets (0)" is shown, with a red box highlighting the "Add Target" button. To the right of the "Add Target" button is a "Download Database (All)" button. Below these buttons is a table header with columns: "Target Name" (with a checkbox icon), "Type", "Rating", "Status", and "Date Modified". At the bottom of the page, there's a footer bar with the text "Last updated: Today 06:51 PM" and a "Refresh" button.

Build Image Database (5)

Add Target

Type:

Single Image Cuboid Cylinder 3D Object

File:

.jpg or .png (max file 2mb)

Width:

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Build Image Database (6)

The screenshot shows the Vuforia Developer Portal interface. At the top, there's a green header bar with the Vuforia logo and the text "vuforia™ Developer Portal". On the right side of the header, it says "Hello trimy" followed by a dropdown arrow and "Log Out". Below the header, there's a navigation menu with links: Home, Pricing, Downloads, Library, Develop (which is highlighted in blue), and Support. Under the Develop menu, there are two sub-links: License Manager and Target Manager, with "Target Manager" being the active one. In the main content area, there's a breadcrumb navigation: "Target Manager > ARImages". The main title is "ARImages" with a link to "Edit Name". Below the title, it says "Type: Device". There's a button labeled "Targets (1)". Underneath, there are two buttons: "Add Target" and "Download Database (All)". A table lists the target details:

Target Name	Type	Rating	Status	Date Modified
google	Single Image	★★★★★	Active	May 01, 2018 18:58

At the bottom of the page, there's a note "Last updated: Today 06:58 PM" and a "Refresh" button.

Build Image Database (7)

Download Database

1 of 1 active targets will be downloaded

Name:
ARImages

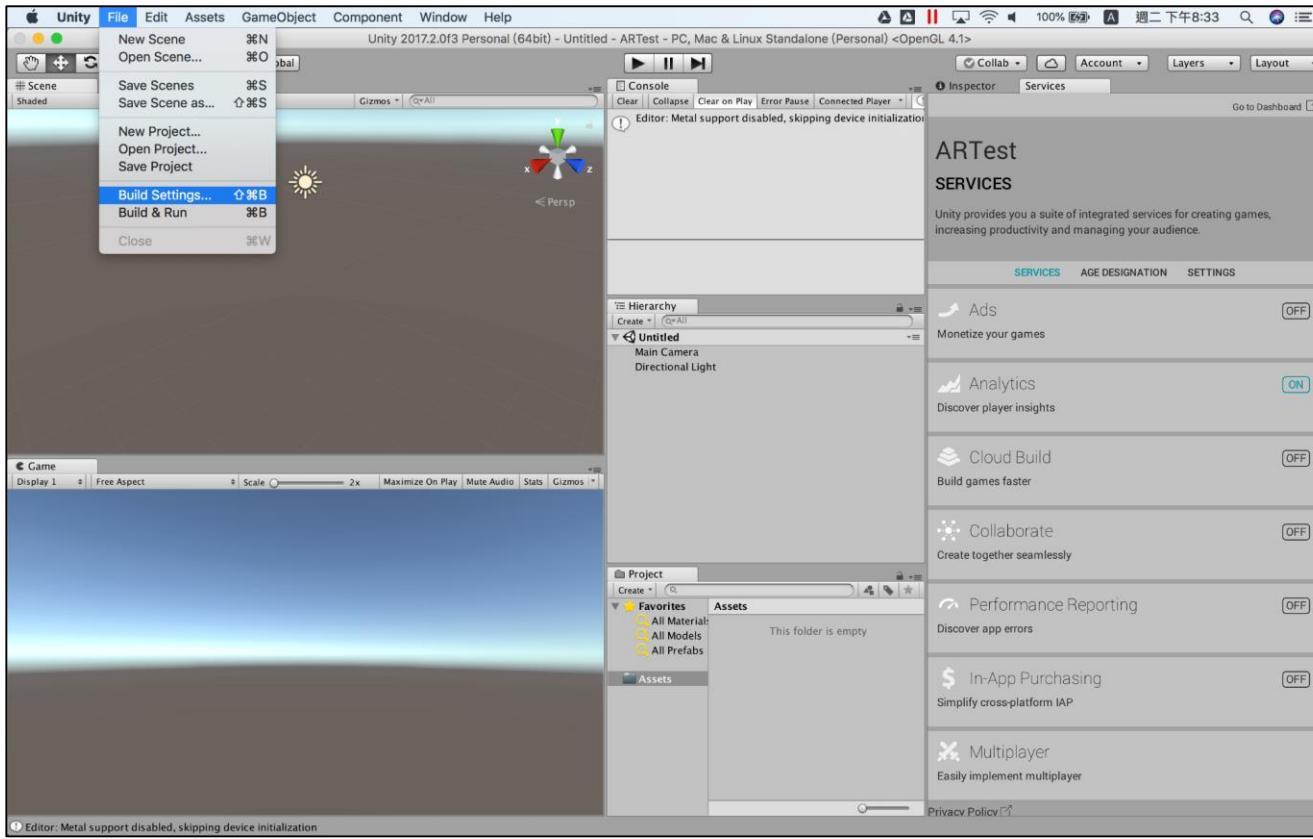
Select a development platform:

Android Studio, Xcode or Visual Studio

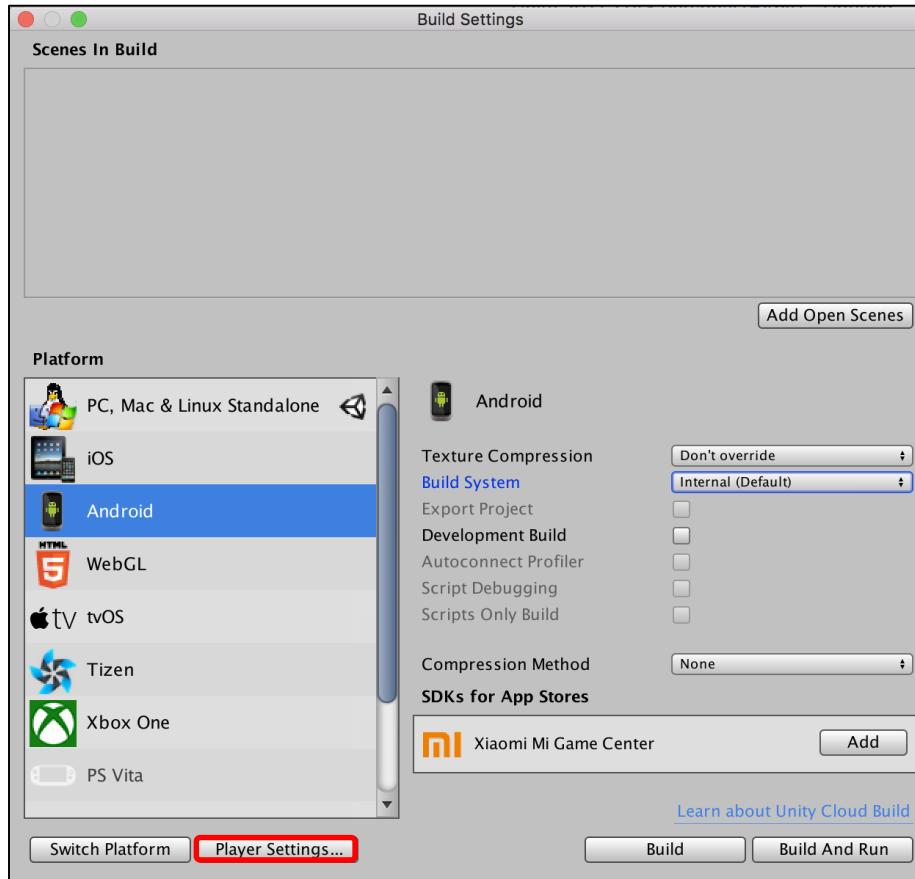
Unity Editor

Cancel **Download**

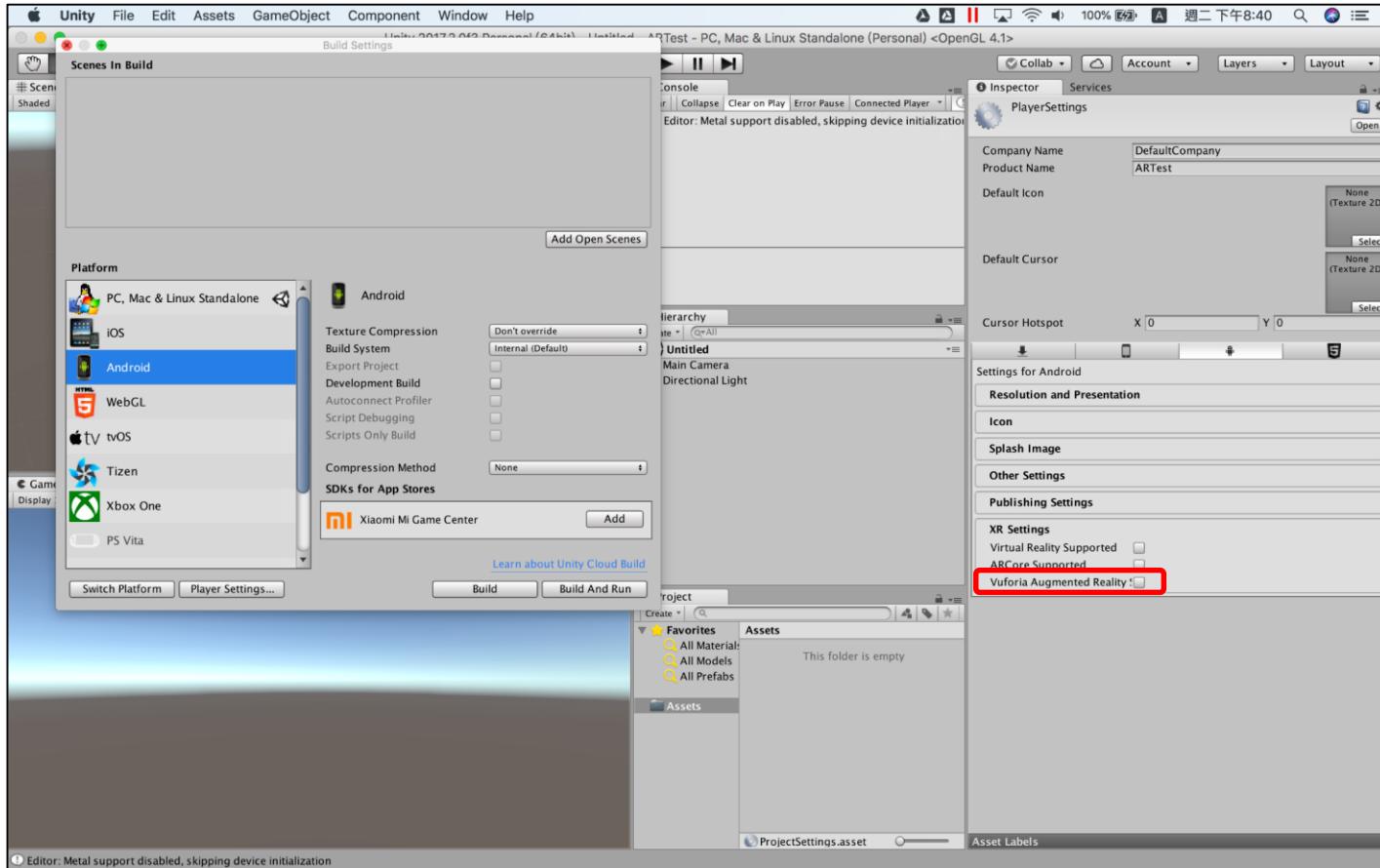
Install Vuforia in Unity (1)



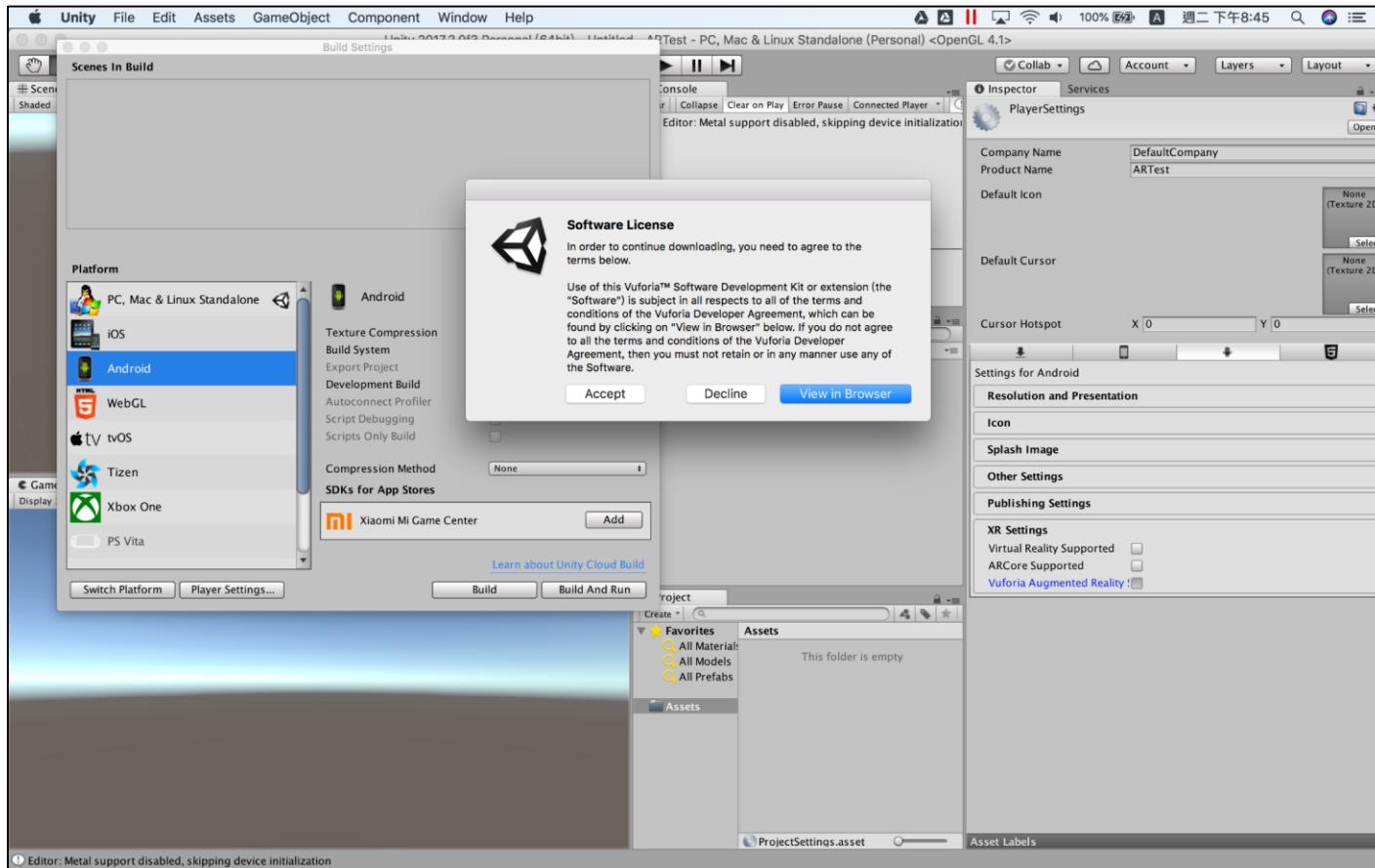
Install Vuforia in Unity (2)



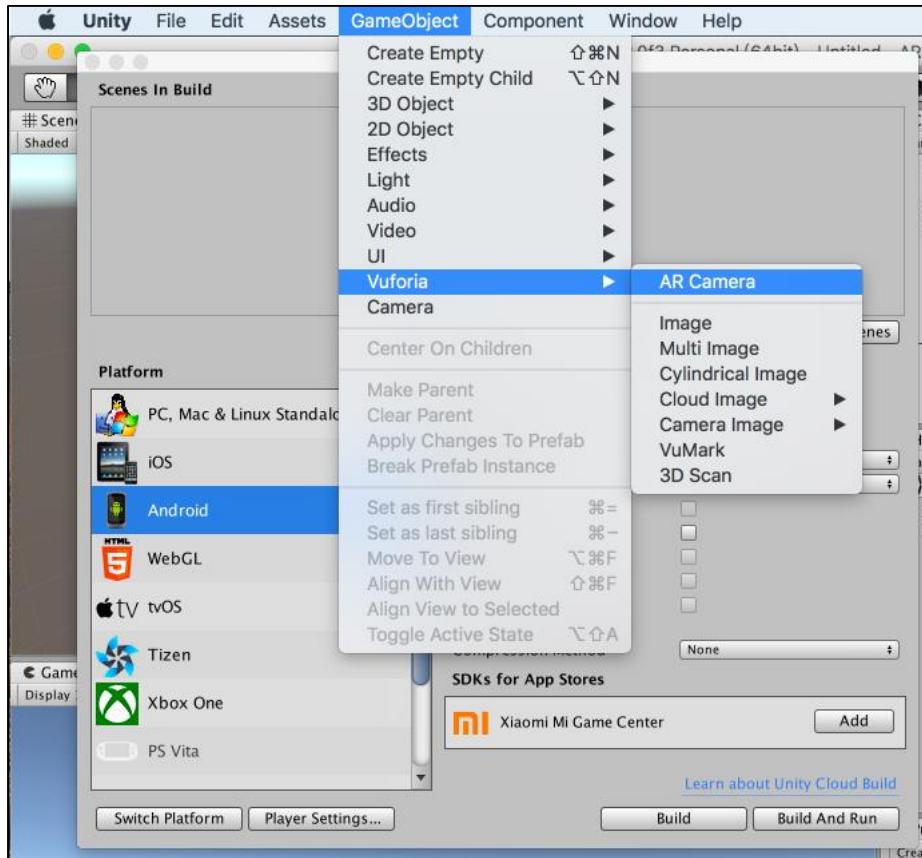
Install Vuforia in Unity (3)



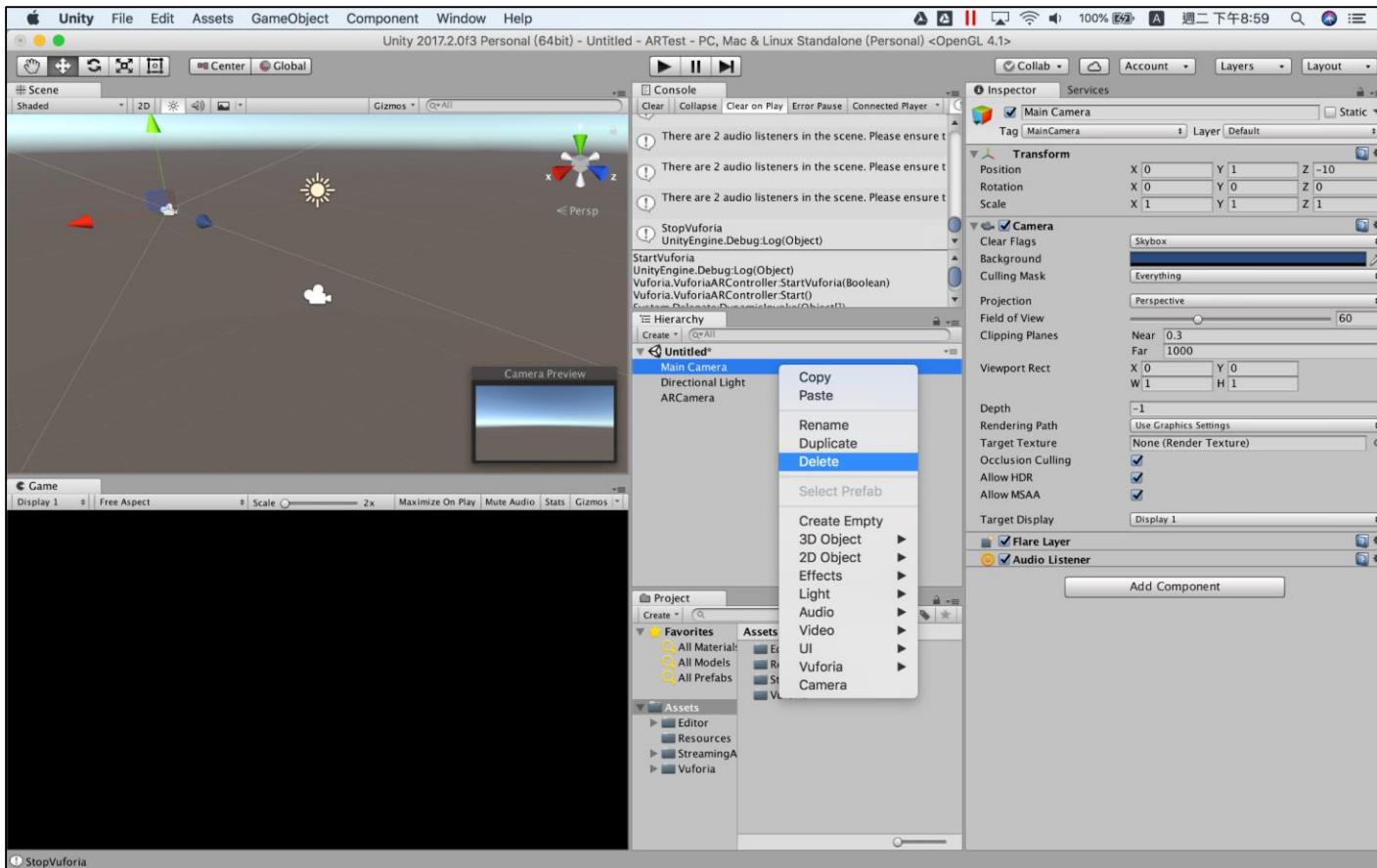
Install Vuforia in Unity (4)



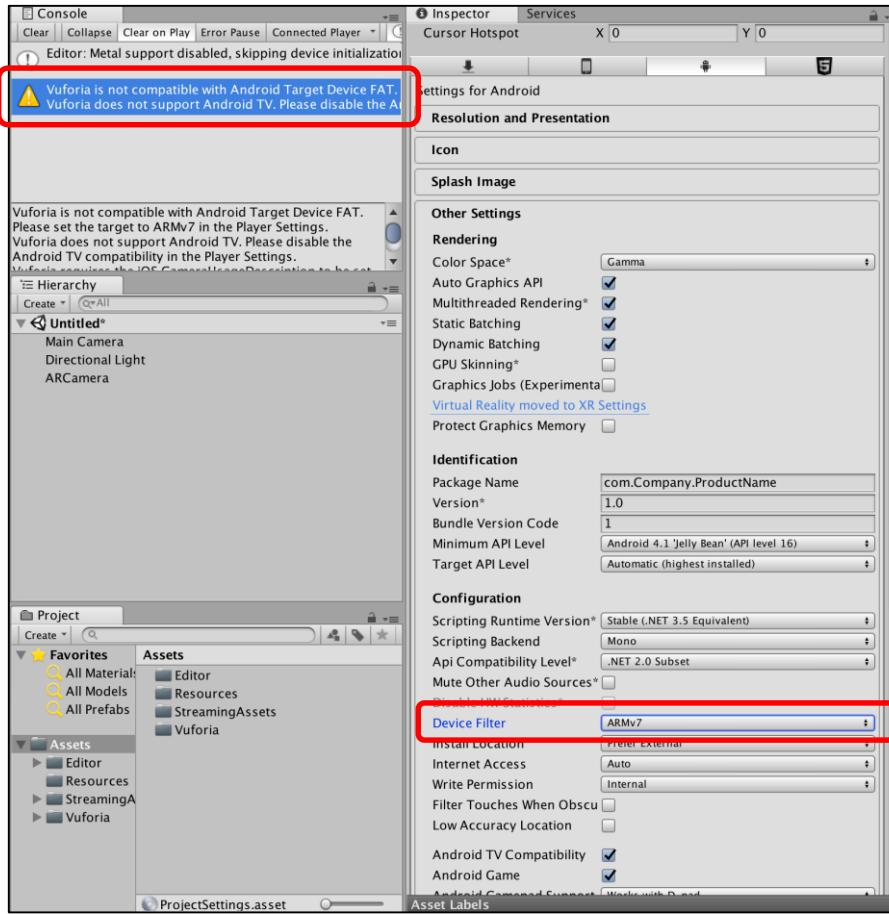
Add ARCamera



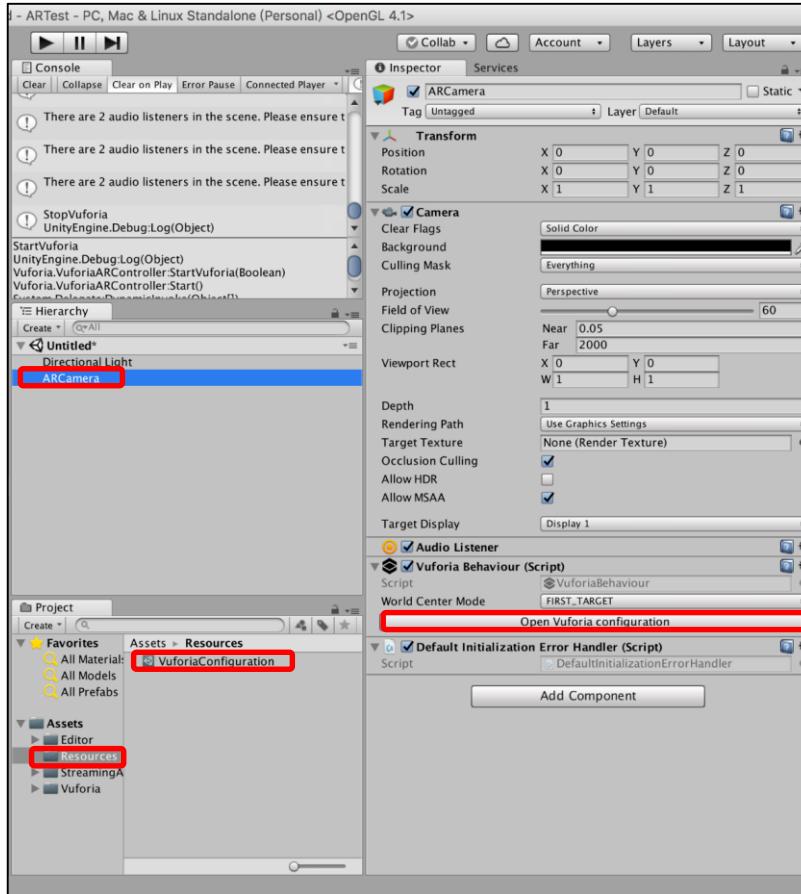
Delete Main Camera



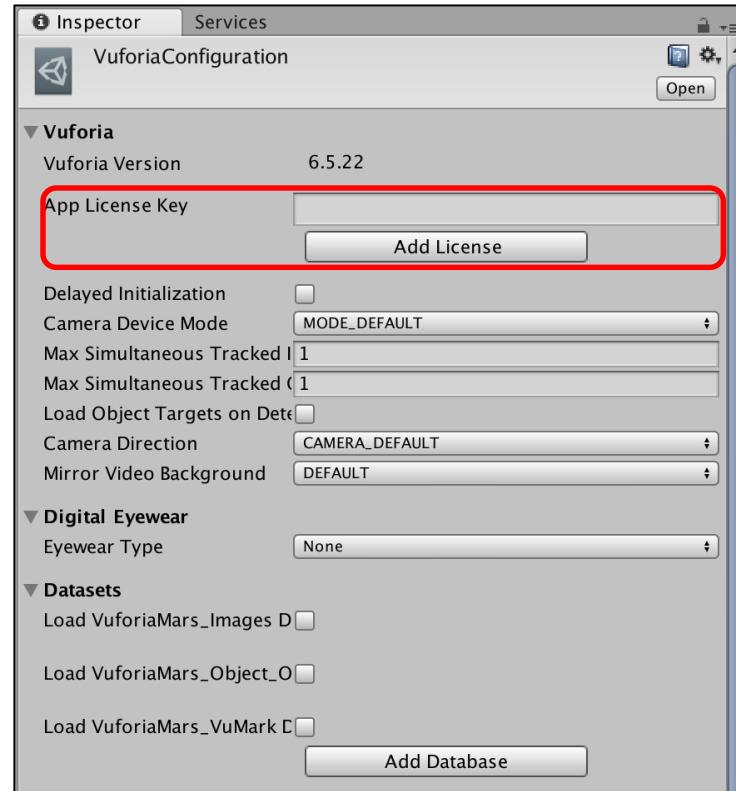
Change Device Target/Filter If Needed



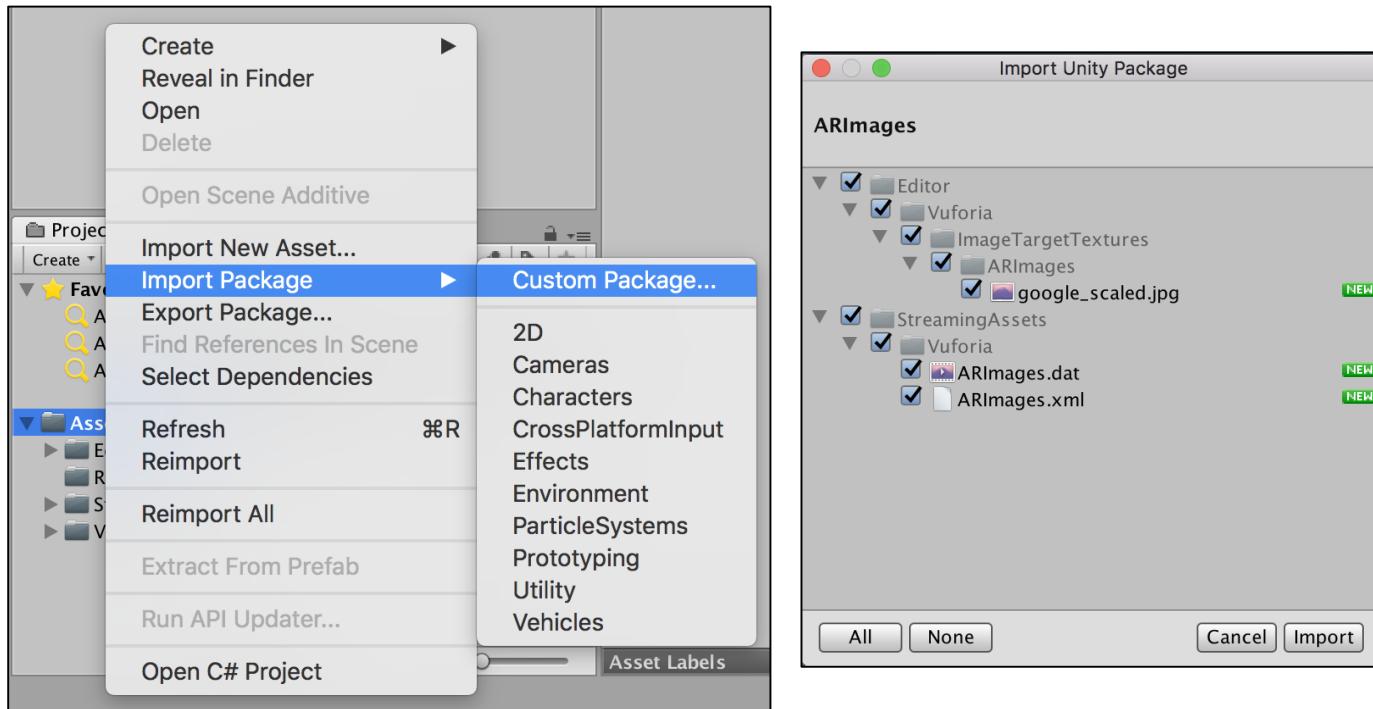
Open Vuforia Configuration



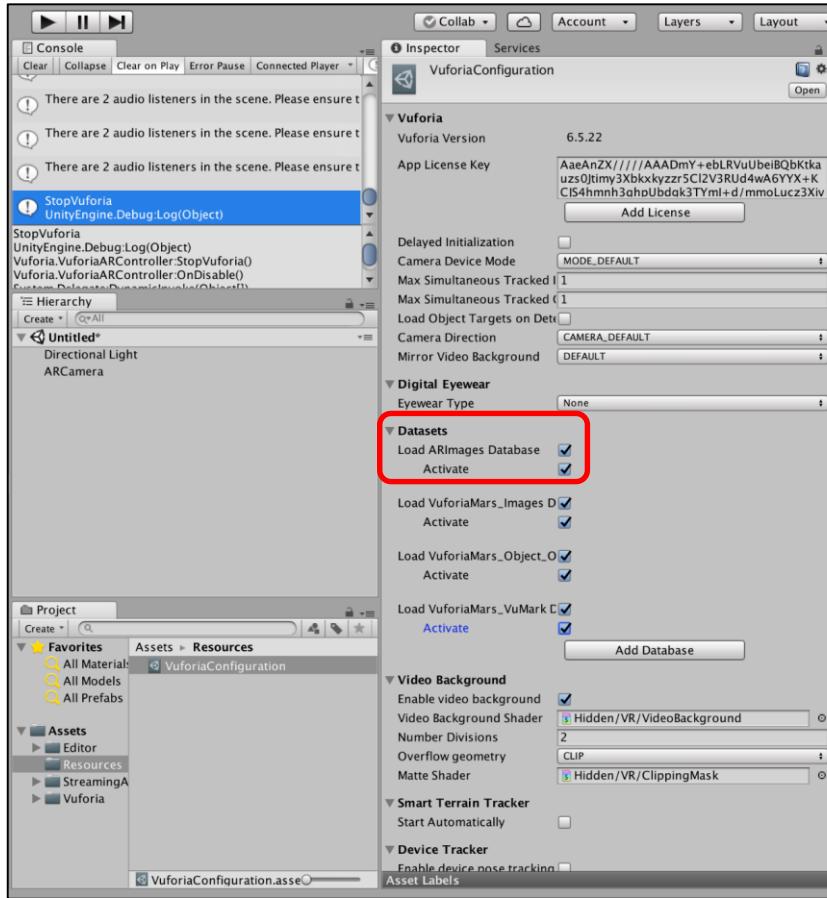
Add License Key



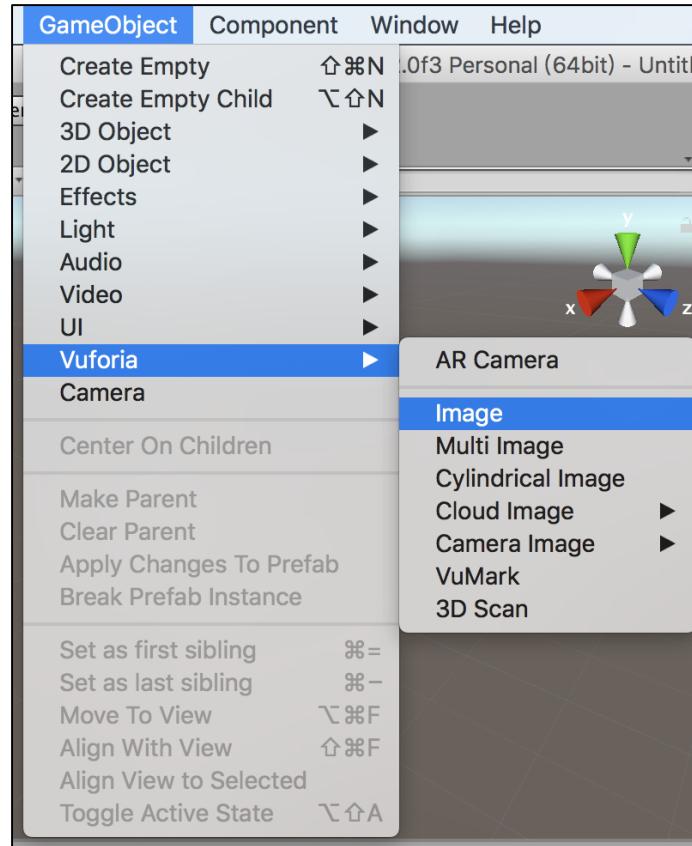
Import Vuforia Image Database (1)



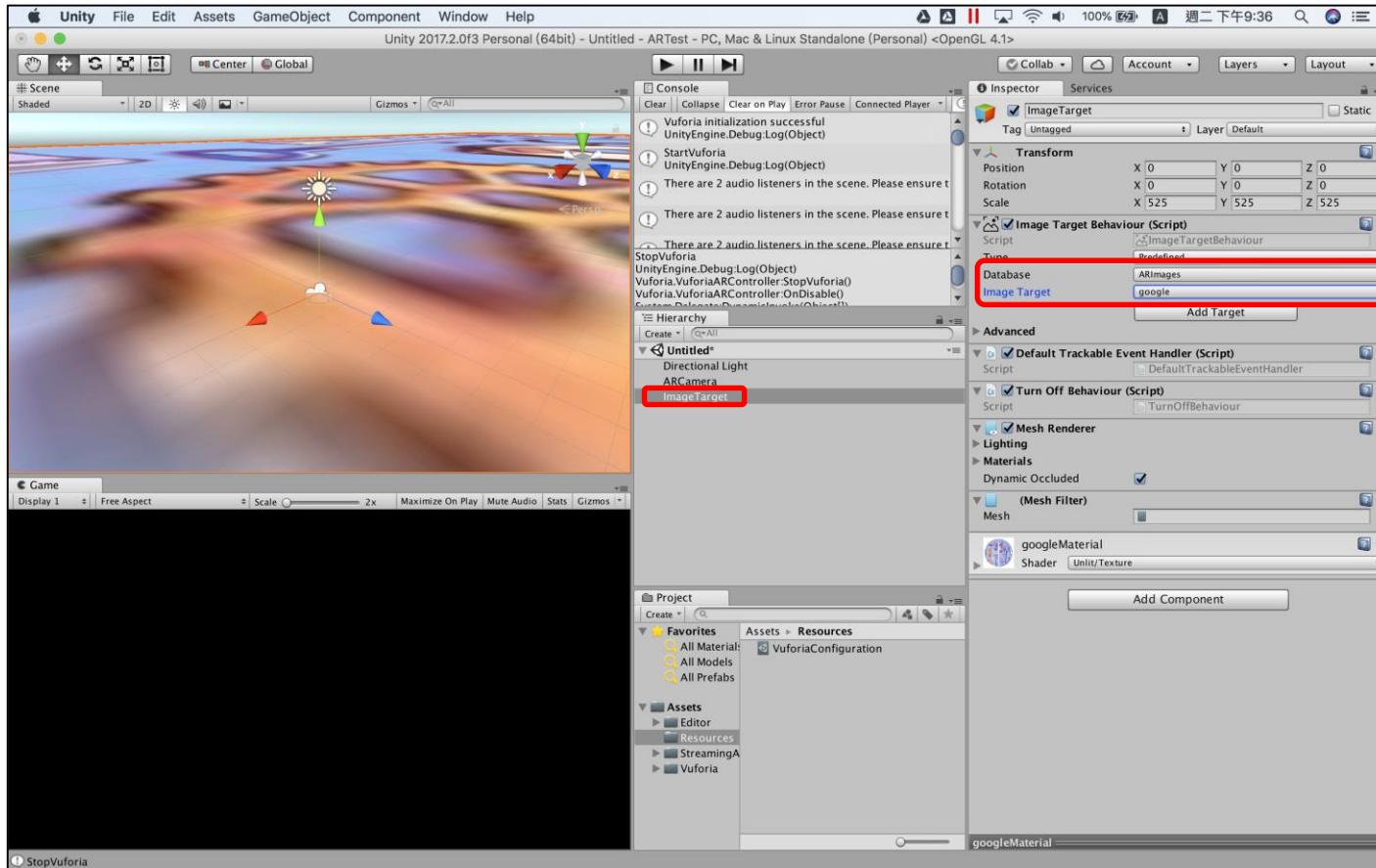
Import Vuforia Image Database (2)



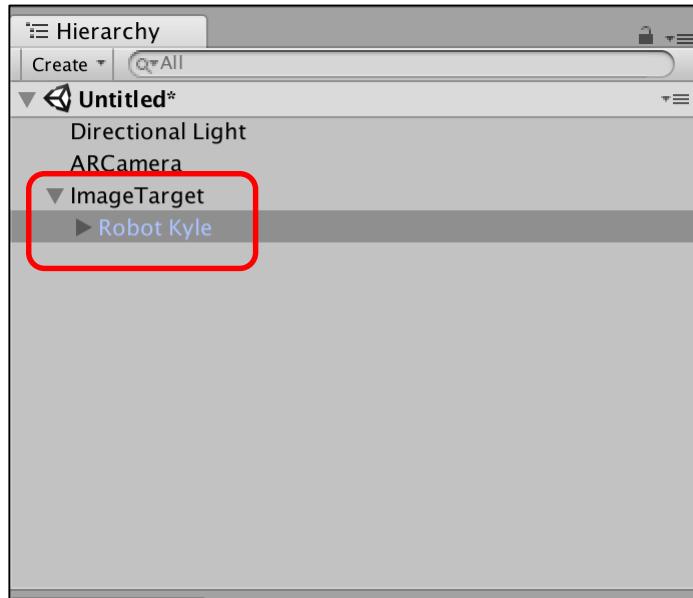
Create an Image Target (1)



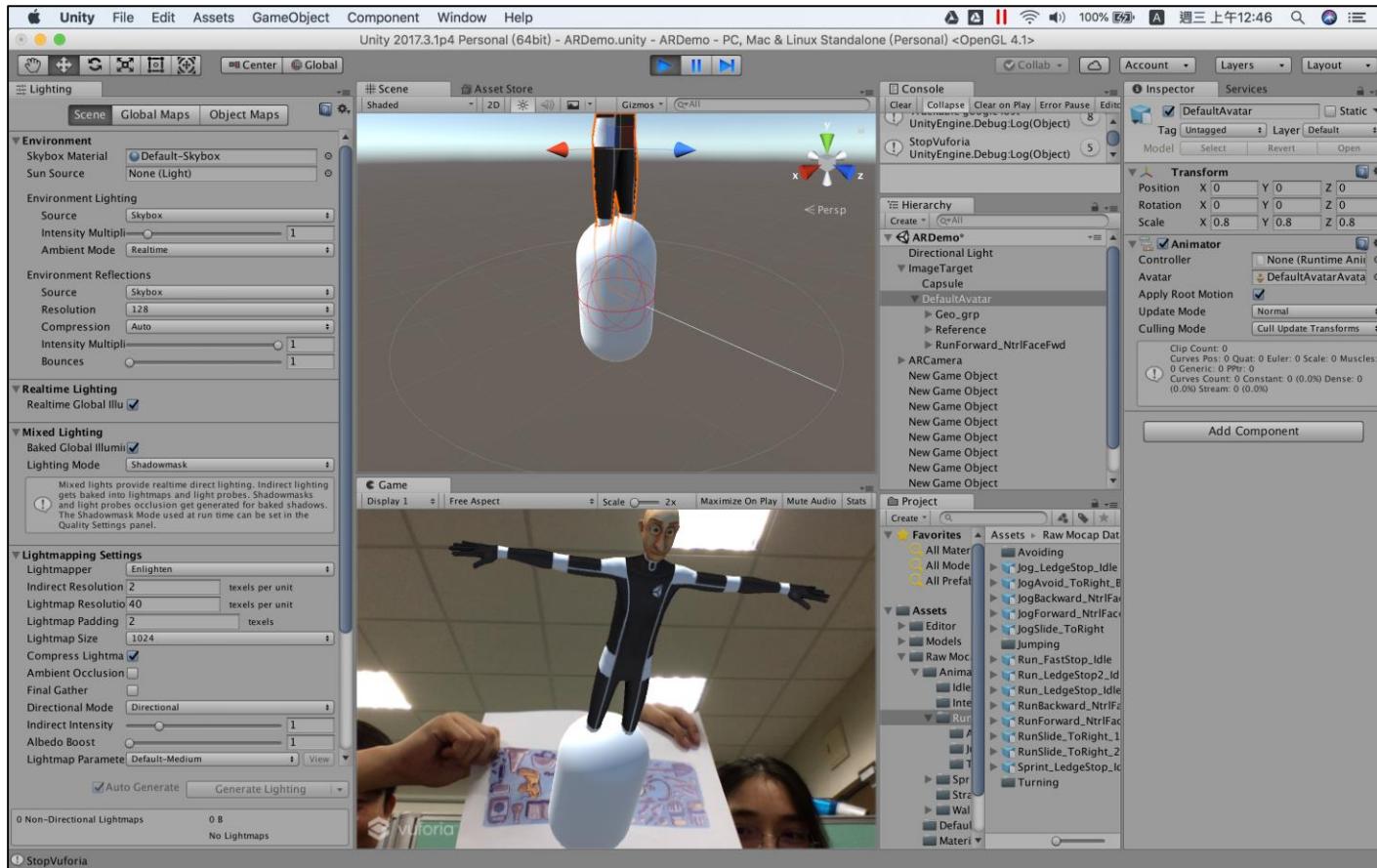
Create Image Target (2)



Import a 2D or 3D Model



Try it !



HW1

- HW1.1:

找出並在console視窗印出1000以下的質數

- HW1.2:

定義一個FPS遊戲的Enemy類別，並建立100個Enemy物件

- HW1.3:

使用Unity + Vuforia 實作AR

Appendix

ARKit (Markerless for iOS)

- ARCamera

- Image Capturing



- ARSession

- Pose Tracking

- Scene Understanding

- Lighting Estimation



- ARSCNView

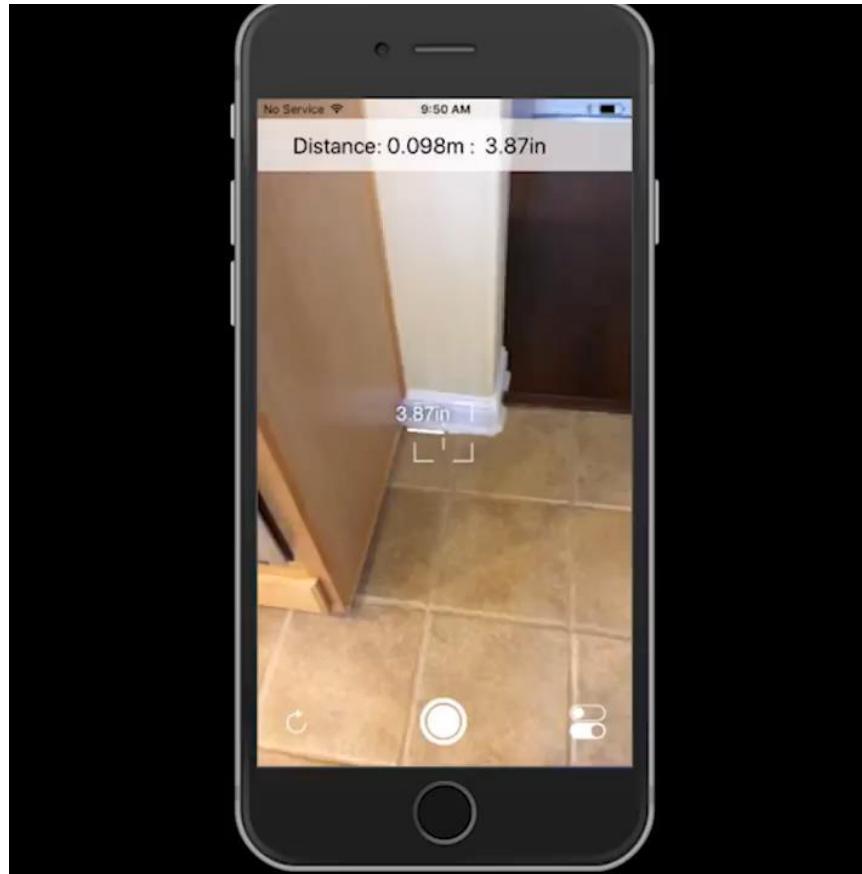
- ARAnchor for Virtual Object Rendering



Virtual Object Rendering

- SCNNode
- SCNScene
- SCNMaterial

Room Measurement Using Apple's ARKit



ARCore (Markless for Android)

- 📍 Session
- 📍 OpenGL ES
- 📍 GLSurfaceView



HTC VIVE

<https://www.vive.com/tw/>

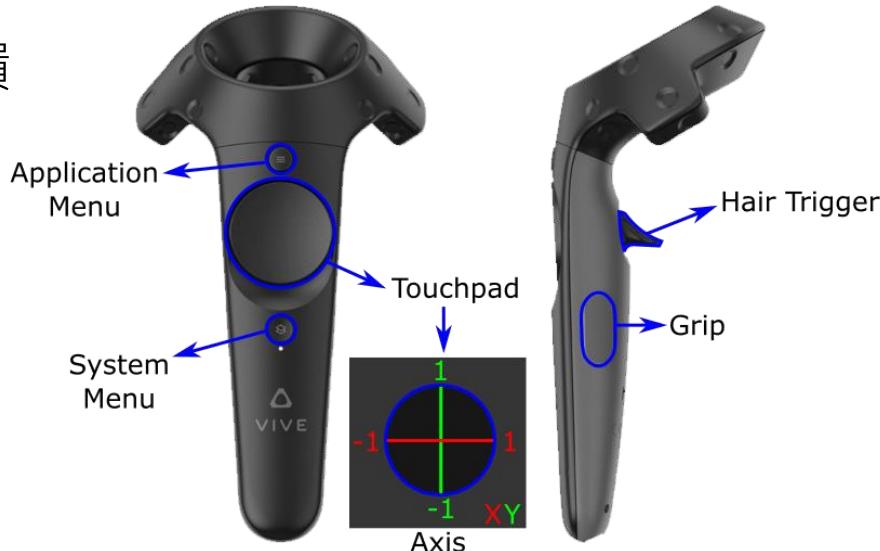
VIVE Head Mounted Display

- 可調整的VIVE標準型頭戴與可替換的VIVE鼻部靠墊
- 前置相機
- 眼睛焦距調整裝置
- 適合大多數眼鏡配戴



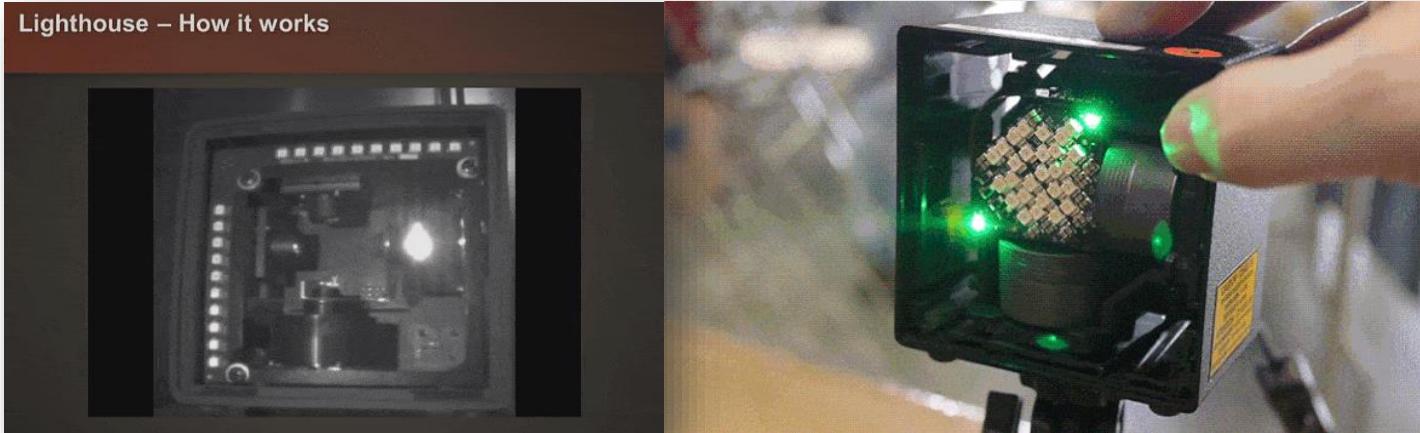
VIVE Controller

- 專為VR設計
- 直覺性的操作與手勢偵測
- 真實高解析的互動回饋
- 具震動回饋



VIVE Lighthouse

- 360°全區追蹤定位覆蓋
- 有線同步 A-B
- 無線同步 B-C
- 使用標準規格螺紋，安裝簡易



VIVE Light Sensors



VIVE 移動定位器

- VIVE 移動定位器使用無線、無縫的方式連接VIVE系統與附加工具



SteamVR Tracking HDK

SteamVR Tracking HDK



The SteamVR HDK is a kit of components that can be used to quickly prototype and evaluate custom SteamVR Tracked objects. The HDK is a modular design that can evolve with your product development.

Each Kit Contains:

- One Watchman Core Module
- One EVM Application Board
- 40 TS4231 Chiclet Sensor Modules
- One Sensor Breakout Board
- One Steam Wireless Dongle
- Four packs of 8in Flex Cables (32 cables total)
- One 2.4 GHz Antenna with u.FL Cable

Pricing/Quantity

Item	1	10	50	100
Complete HDK Kit	\$595.00	\$579.00	\$549.00	\$499.00
Watchman Core Module	\$145.55	\$141.63	\$134.29	\$122.06
EVM Application Board	\$171.22	\$170.50	\$161.67	\$146.95
TS4231 Chiclet Sensor (pack of 40)	\$278.00	\$270.80	\$268.08	\$263.92
Sensor Breakout Board	\$89.00	\$86.00	\$83.13	\$74.65
USB Dongle	\$11.99	\$11.99	\$11.99	\$11.99
4in Flex Cable (pack of 8)	\$16.00	\$14.00	\$12.00	\$10.00

Design Guidelines:

[PCB Layout Design Guidelines for the TS4231](#)

Demo of VIVE



VIVE vs VIVE Pro

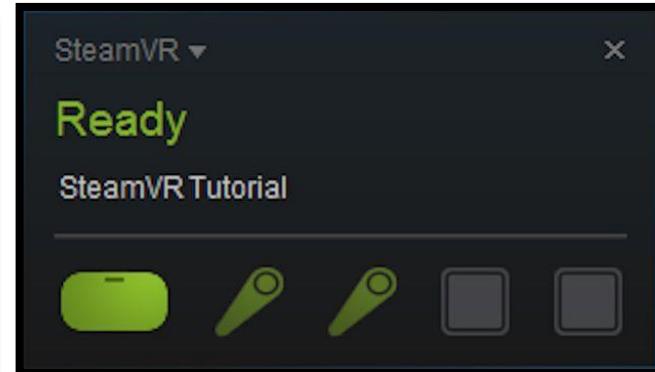
	VIVE	VIVE Pro
螢幕	3.6吋雙AMOLED	3.5吋雙AMOLED
螢幕更新率	90 Hz(赫茲)	90 Hz(赫茲)
視野	110度	110度
解析度	單眼1080 x 1200 像素 (雙眼2160 x 1200 像素)	單眼1440 x 1600 像素 (雙眼2880 x 1600 像素)
音訊	3.5mm 外接音源 (可買專屬耳機 NT\$3,499)	內建耳機，支援hi-Res音效、 3D音效、主動式降躁
感應器	SteamVR追蹤技術、 重力感測器、 陀螺儀感測、 距離偵測裝置	SteamVR追蹤技術、 重力感測器、 陀螺儀感測裝置、 趨近感測器、 瞳距感測器
裝置連接埠	HDMI、USB 2.0、藍芽	USB-C 3.0、DP 1.2、藍芽
房間規模	最小: 2m x 1.5m 最大:兩對角線距離5m	最小: 2m x 1.5m 最大:6m x 6m(需使用2.0基地台)



推薦的電腦規格

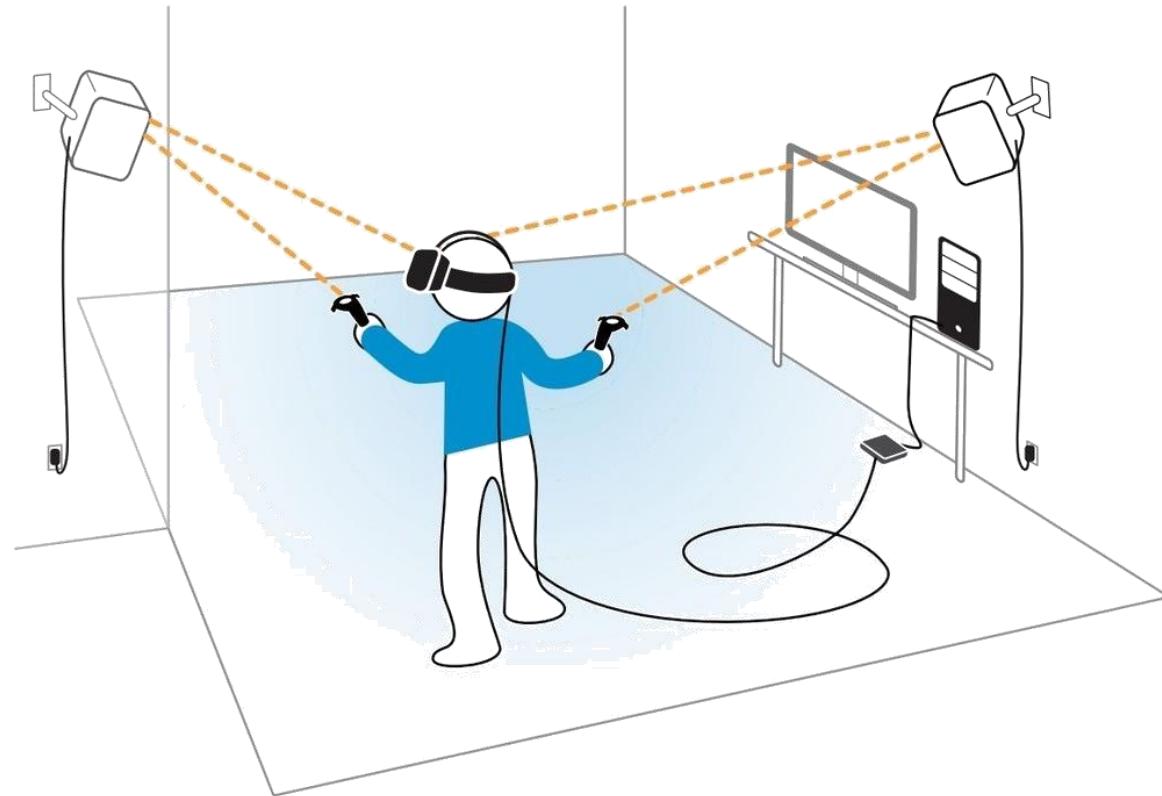
- ⦿ 顯卡：NVIDIA GeForce™ GTX 1060或AMD Radeon™ RX 480
- ⦿ 處理器：Intel™ Core™ i5-4590或AMD FX™ 8350
- ⦿ 4 GB RAM
- ⦿ 視訊輸出：HDMI 1.4連接埠，或者DisplayPort 1.2
- ⦿ USB：1個USB 2.0或更新版本
- ⦿ 作業系統：Windows™ 7 SP1、Windows™ 8.1或Windows™ 10

Install Steam VR SDK



<https://store.steampowered.com/steamvr>

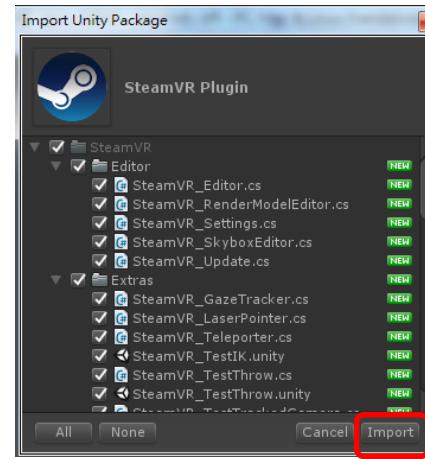
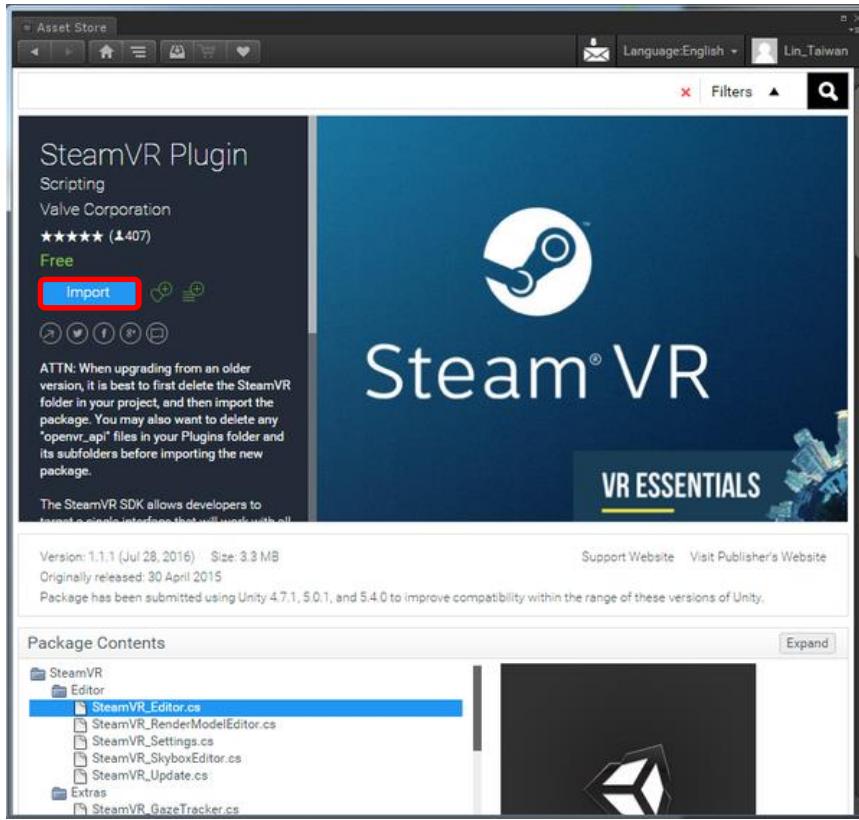
設定互動空間



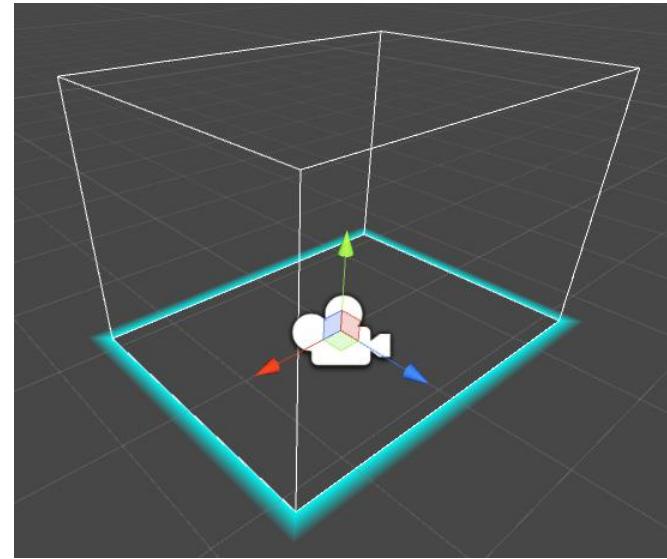
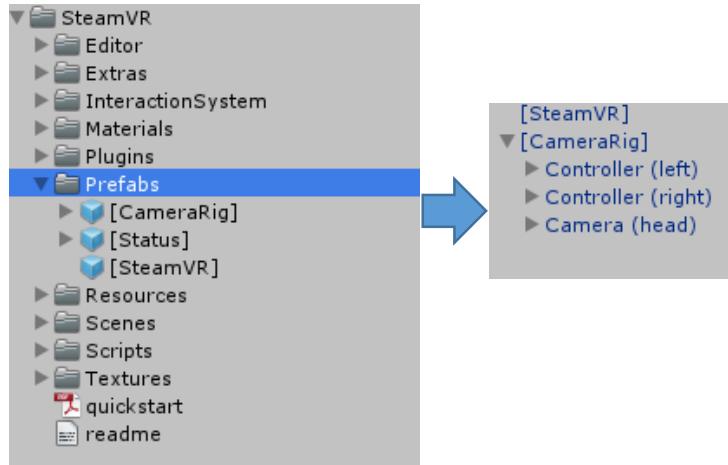
設定互動空間



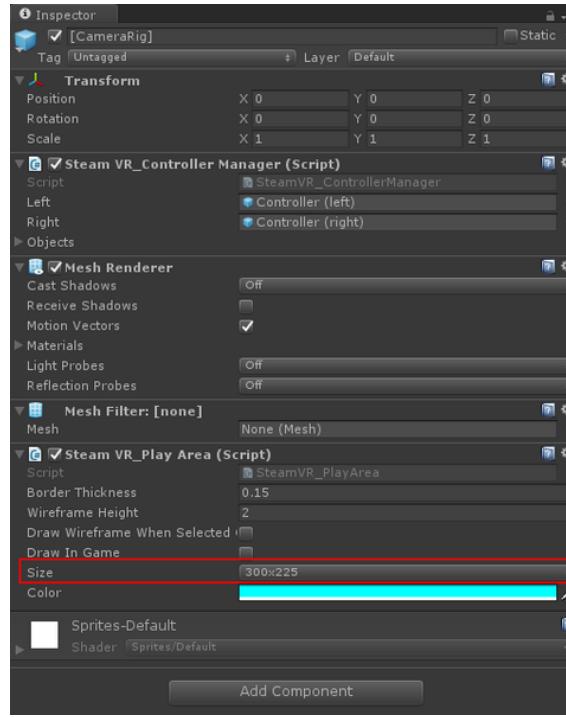
Steam VR Plugin for Unity



Steam VR Plugin for Unity



Steam VR Plugin for Unity

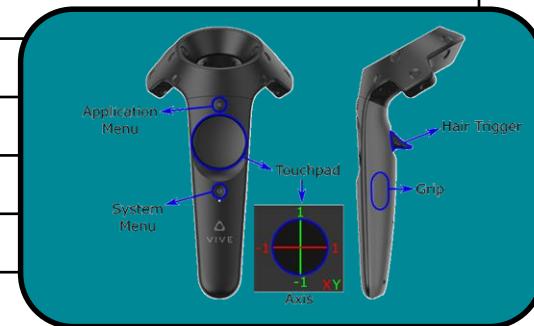


Access VIVE Controllers in Unity

```
1 private SteamVR_TrackedObject trackedObjectRight;
2 private SteamVR_TrackedObject trackedObjectLeft;
3
4 private SteamVR_Controller.Device controllerRight
5 {
6     get{
7         return SteamVR_Controller.Input((int)trackedObjectRight.index);
8     }
9 }
10
11 private SteamVR_Controller.Device controllerLeft
12 {
13     get{
14         return SteamVR_Controller.Input((int)trackedObjectLeft.index);
15     }
16 }
17
18
19 void Awake()
20 {
21     trackedObjectRight = controllerObjectRight.GetComponent<SteamVR_TrackedObject>();
22     trackedObjectLeft = controllerObjectLeft.GetComponent<SteamVR_TrackedObject>();
23 }
```

Access VIVE Controller Buttons

Button Name	ID
System	EVRButtonId.k_EButton_System
ApplicationMenu	EVRButtonId.k_EButton_ApplicationMenu
Grip	EVRButtonId.k_EButton_Grip
Axis0	EVRButtonId.k_EButton_Axis0
Axis1	EVRButtonId.k_EButton_Axis1
Axis2	EVRButtonId.k_EButton_Axis2
Axis3	EVRButtonId.k_EButton_Axis3
Axis4	EVRButtonId.k_EButton_Axis4
Touch Pad	EVRButtonId.k_EButton_SteamVR_Touchpad
Trigger	EVRButtonId.k_EButton_SteamVR_Trigger

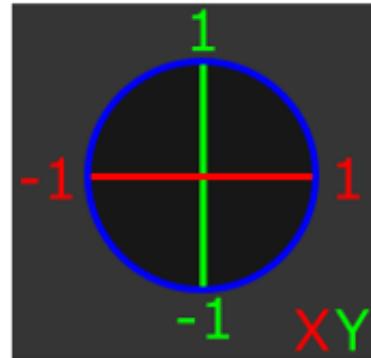


Access VIVE Controller Buttons



Touch Pad

1. Press
2. Touch
3. Touch coordinate (x, y)



```
controllerRight.GetTouch(Valve.VR.EVRButtonId.k_EButton_Axis0);
controllerRight.GetTouchDown(Valve.VR.EVRButtonId.k_EButton_Axis0);
controllerRight.GetTouchUp(Valve.VR.EVRButtonId.k_EButton_Axis0);
controllerRight.GetPress(Valve.VR.EVRButtonId.k_EButton_Axis0);
controllerRight.GetPressDown(Valve.VR.EVRButtonId.k_EButton_Axis0);
controllerRight.GetPressUp(Valve.VR.EVRButtonId.k_EButton_Axis0);
controllerRight.GetAxis(Valve.VR.EVRButtonId.k_EButton_Axis0);
```

Access VIVE Controller Buttons

2

Hair Trigger

1. Press
2. Press value x

```
controllerRight.GetHairTriggerDown();
controllerRight.GetHairTriggerUp();
controllerRight.GetPress(Valve.VR.EVRButtonId.k_EButton_SteamVR_Trigger);
controllerRight.GetPressDown(Valve.VR.EVRButtonId.k_EButton_SteamVR_Trigger);
controllerRight.GetPressUp(Valve.VR.EVRButtonId.k_EButton_SteamVR_Trigger);
controllerRight.GetAxis(Valve.VR.EVRButtonId.k_EButton_SteamVR_Trigger).x;
```

3

Grip

1. Press

```
controllerRight.GetPress(Valve.VR.EVRButtonId.k_EButton_Grip);
controllerRight.GetPressDown(Valve.VR.EVRButtonId.k_EButton_Grip);
controllerRight.GetPressUp(Valve.VR.EVRButtonId.k_EButton_Grip);
```

References

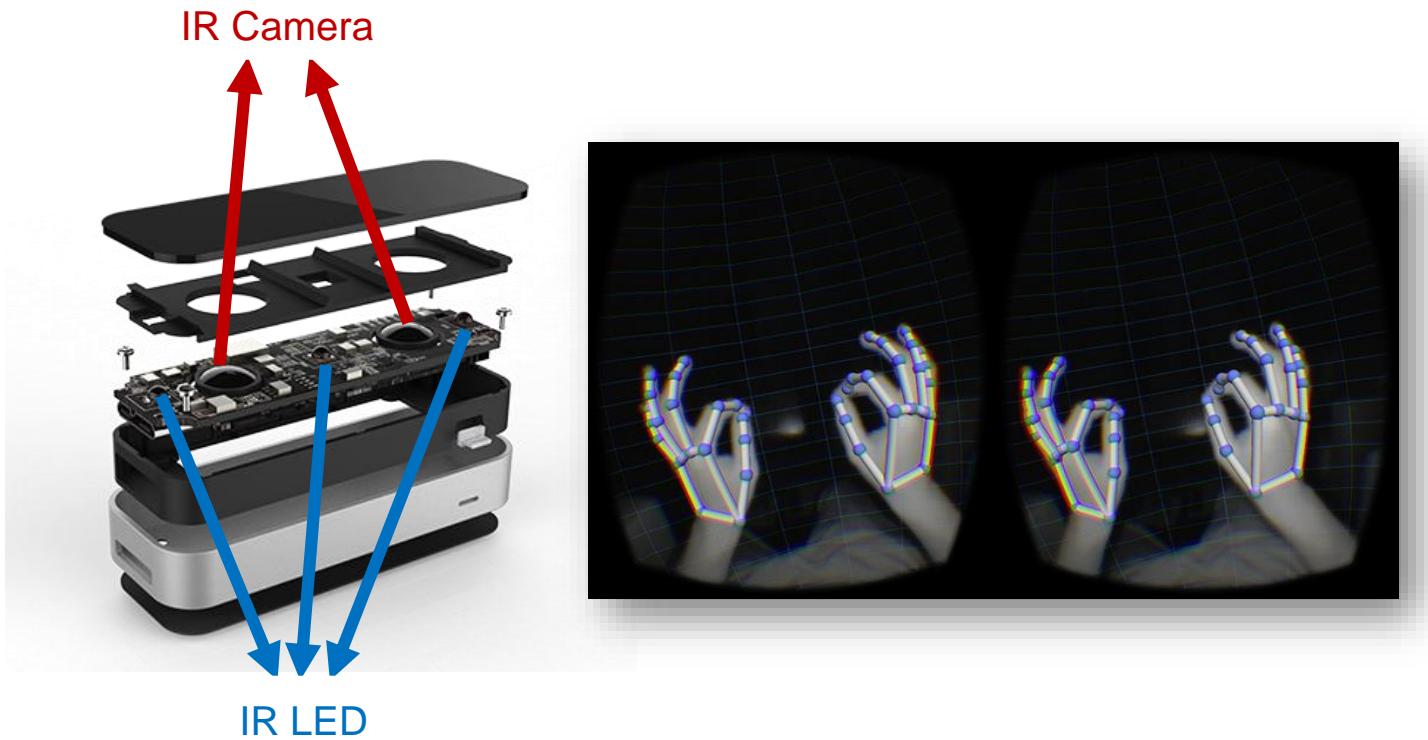
- 🔊 <https://www.raywenderlich.com/149239/htc-vive-tutorial-unity>
- 🔊 <https://docs.unity3d.com/Manual/OpenVRControllers.html>

LEAP
MOTION



Leap Motion

Decompose Leap Motion



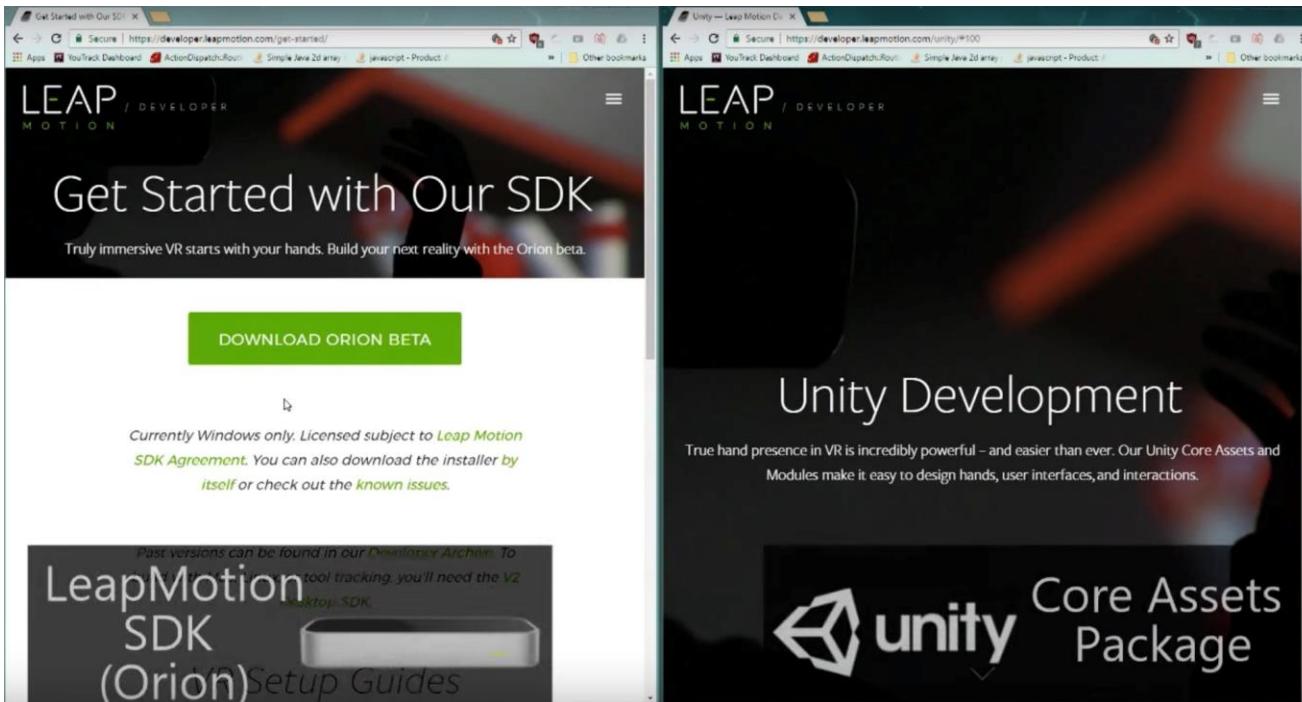
Demo of Leap Motion



The logo consists of the word "LEAP" in a large, white, sans-serif font. The letter "E" has a small green horizontal bar extending from its top right side. Below "LEAP", the word "MOTION" is written in a smaller, green, sans-serif font.

Installation & Setup

- Install LeapMotion SDK and download the LeapMotion core Unity assets



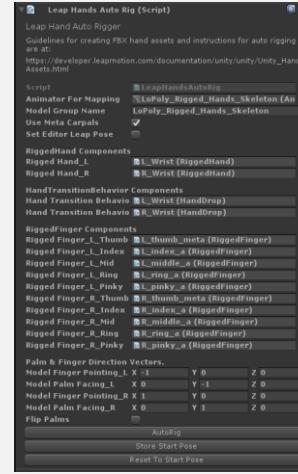
<https://www.youtube.com/watch?v=Wh3YK2Ocl4M>

Hand Models



Build in hand models

- ### Leap Mesh Hands Rigging Process
- 1 Assign RiggedHands Scripts
 - 2 Determine & Assign Handedness
 - 3 Find & Assign Palms
 - 4 Find Finger Base Joints
 - 5 Assign RiggedFinger Scripts to Base Joints
 - 6 Assign FingerType in Each RiggedFinger Script
 - 7 Reference RiggedFinger's in RiggedHand's
 - 8 Find & Assign Finger Joints in Each RiggedFinger Script
 - 9 Calculate Palm Facing Vector for Each Hand
 - 10 Calculate Finger Pointing for Each RiggedHand
 - 11 Populate Vector Values to RiggedFingers
 - 12 Save Start Pose
 - 13 Add, Name and Populate a New ModelGroup in HandPool

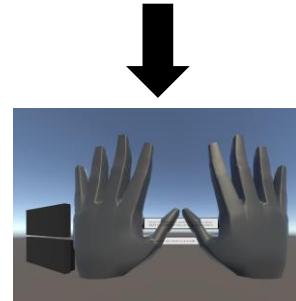
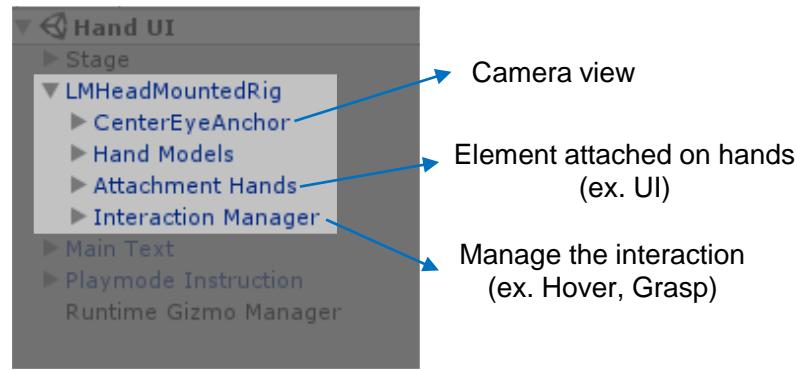
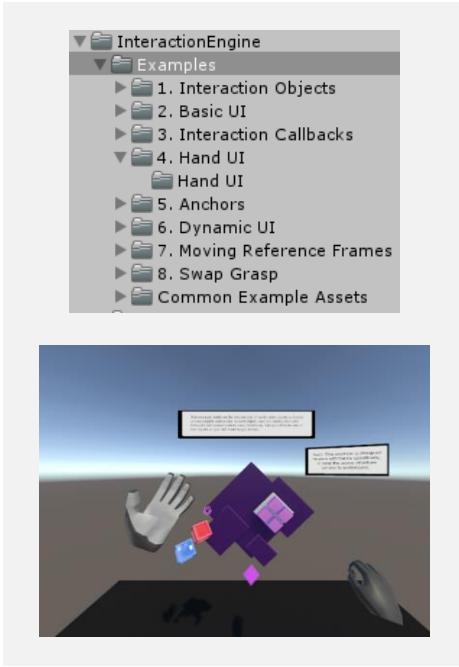


Customized your own hands

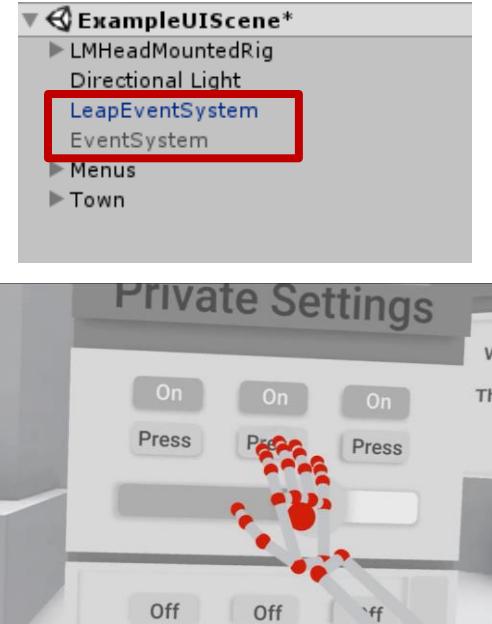
<https://github.com/leapmotion/UnityModules/wiki/Hands-Module>

<http://blog.leapmotion.com/hands-module-2-0-bring-your-hand-designs-to-life-in-two-minutes-or-less/>

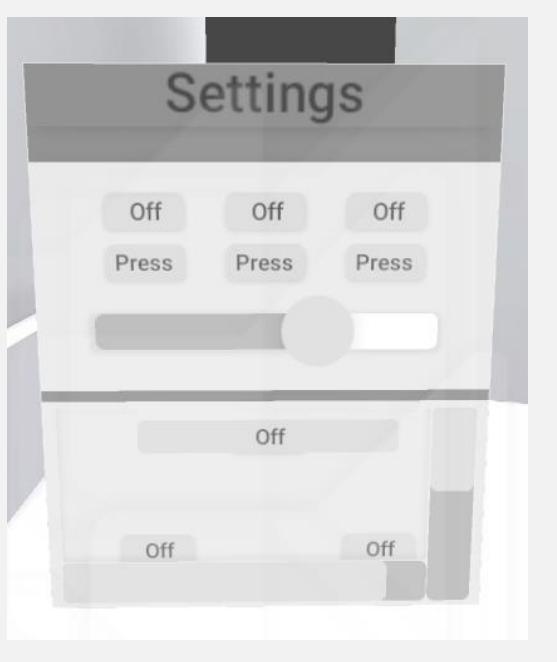
Interaction Engine



UI Input



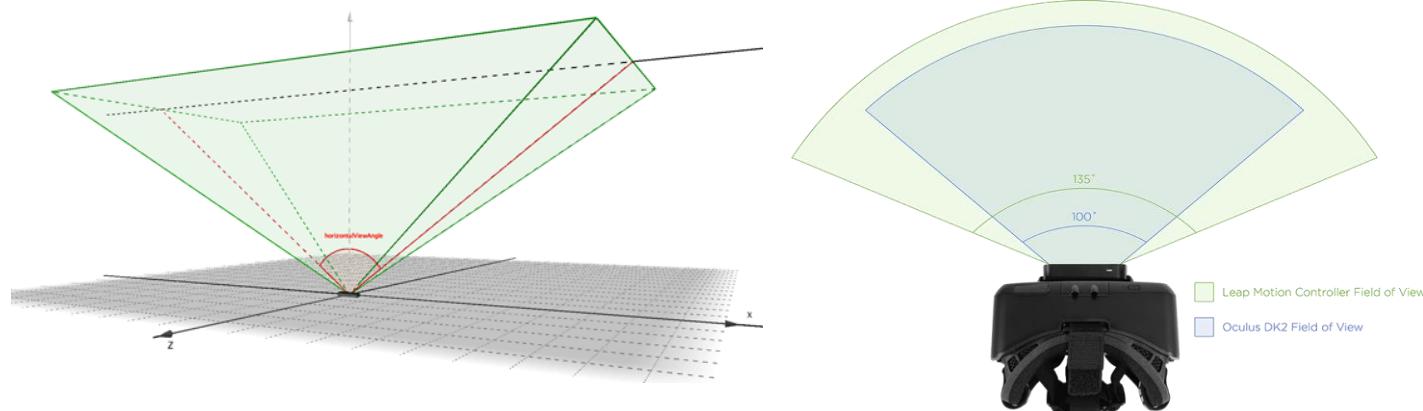
1. Toggle
2. Button
3. Slider
4. Scroll View



Limitation of Leap Motion

- Optical-based hand tracking

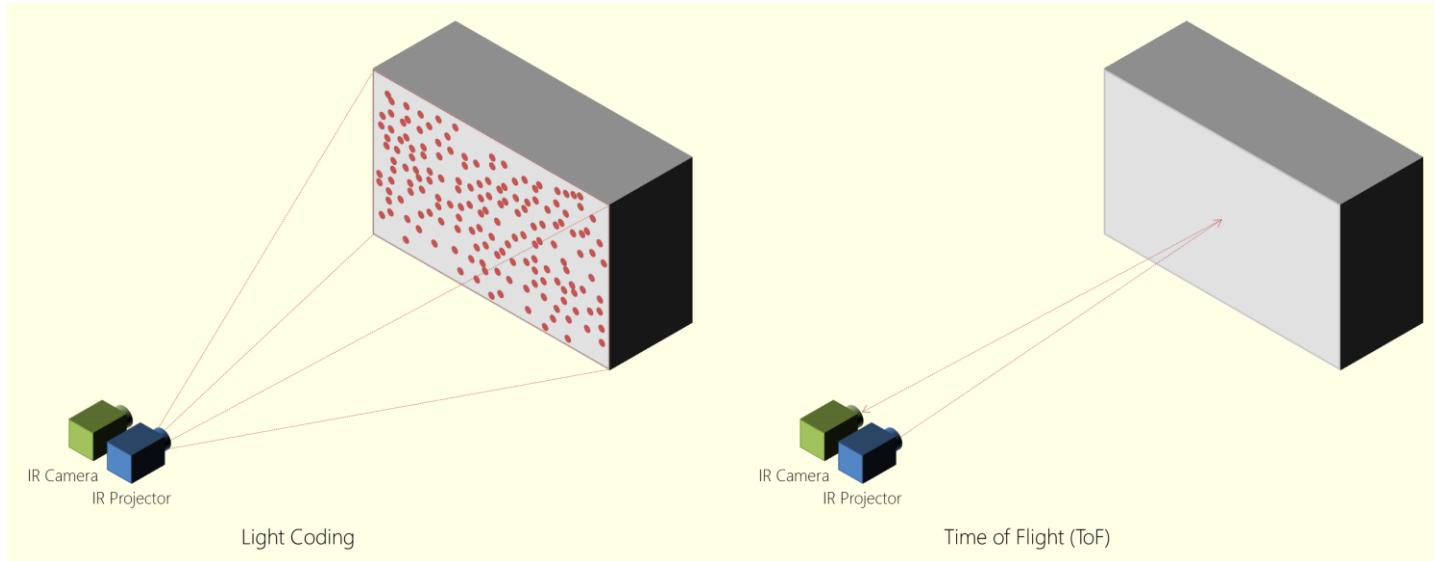
→ The interaction is limited by the camera view



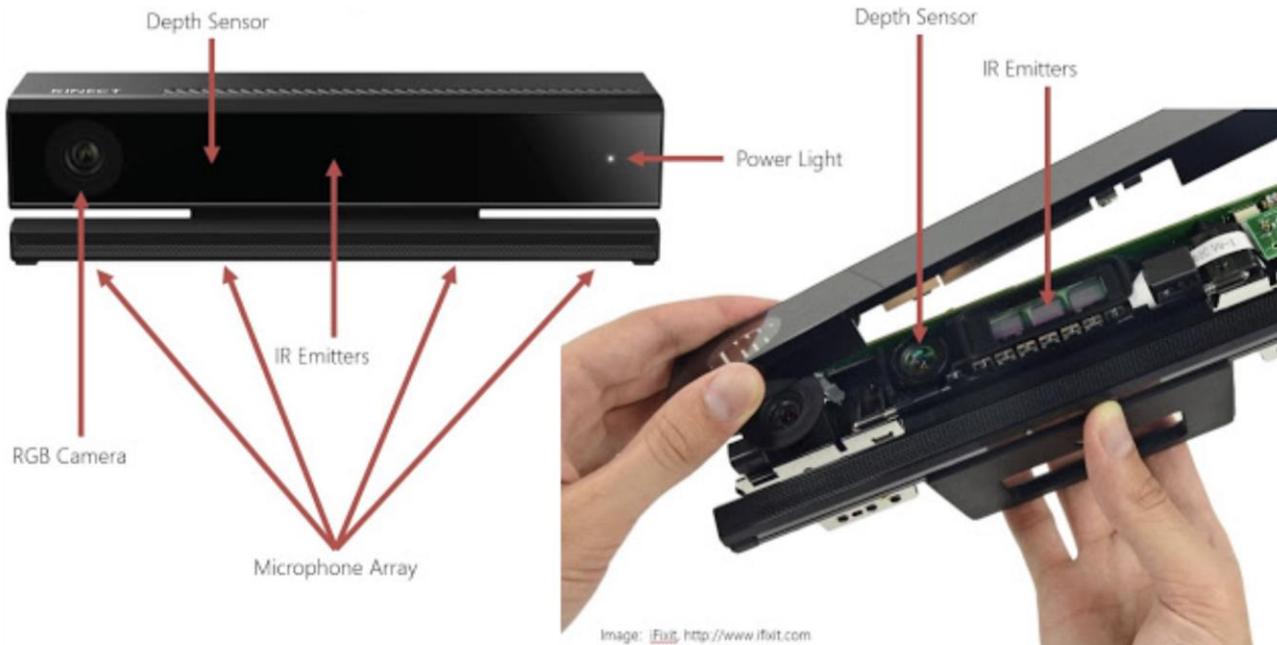


Kinect

How does it work?



Sensors in Kinect



- RGB Image : 1920 x 1080 (30 / 15 fps)
- Depth Image : 512 x 424 (30 fps)(0.5 ~ 8m distance)
- Infrared Image : 512 x 424 (30 fps)
- Microphone Array

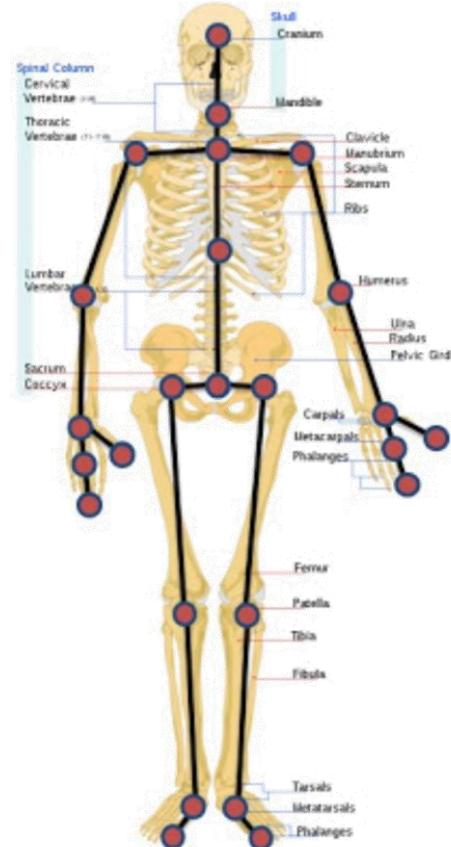
Get Sensor Data from API

IKinectSensor

- get_DepthFrameSource() -> `IDepthFrameSource` (Depth Image)
- get_ColorFrameSource() -> `IColorFrameSource` (RGB Image)
- get_InfraredFrameSource() -> `IInfraredFrameSource` (Infraed Image)
- getIAudioSource (Audio)
- get_BodyFrameSource() -> `IBodyFrameSource` (Skeleton)
- get_CoordinateMapper() -> `ICoordinateMapper` (Coordinate)

Skeleton Detection and Tracking

- At most 6 People
- 25 Joints
- Position & Orientation
- Thumb information



IBodyFrameSource -> get_BodyCount()

IBodyFrame -> GetAndRefreshBodyData(BodyCount, Ibody[])

Skeleton Detection and Tracking



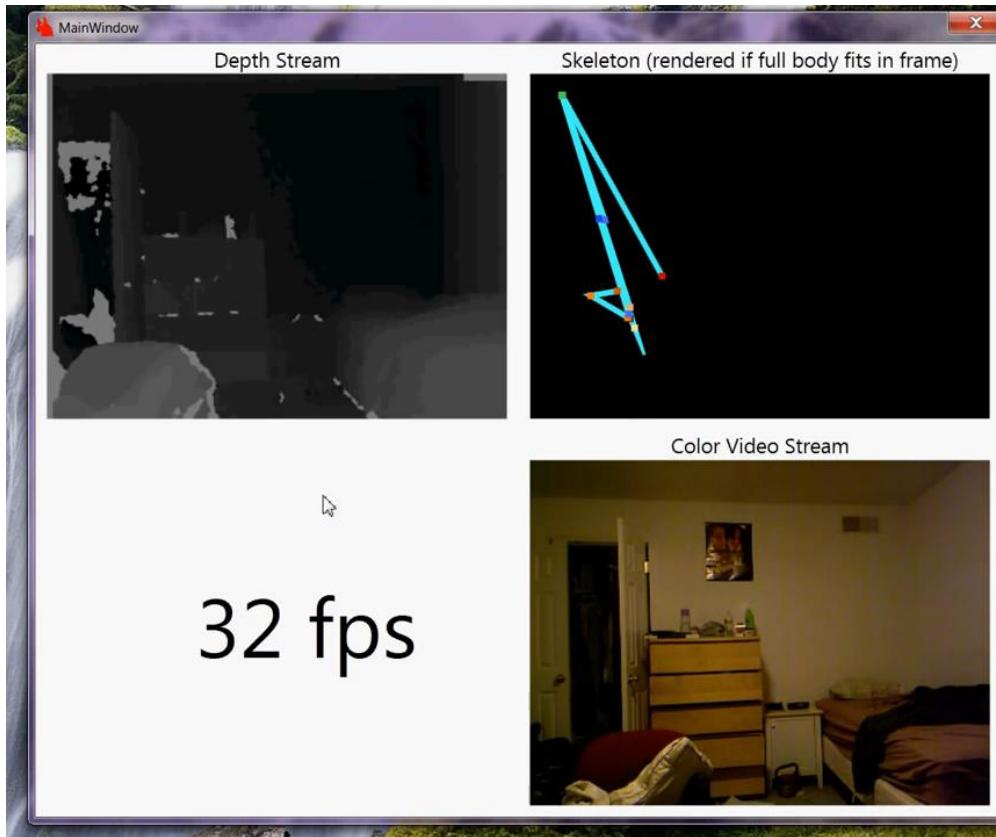
IBody

- GetJoints() -> Get joints coordinate
- GetJointOrientations() -> Get joints orientation
- get_IsTracked() -> Check if this skeleton is being tracked

Comprehensive Kinect Tutorial on C++ API :

<https://kheresy.wordpress.com/kinect-for-windows-v2-cpp-index/comment-page-1/>

Demo of Kinect Skeleton Tracking



Fun Kinect Applications





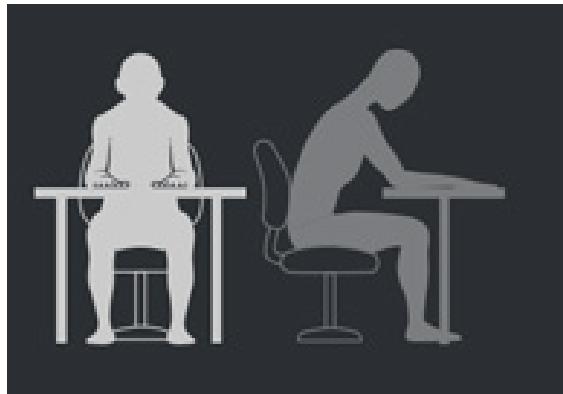
Noitom

<https://www.noitom.com/>

Demo of Noitom



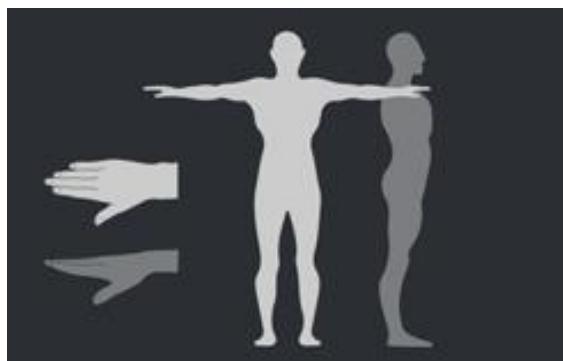
Calibration Procedure



Steady pose



A-pose

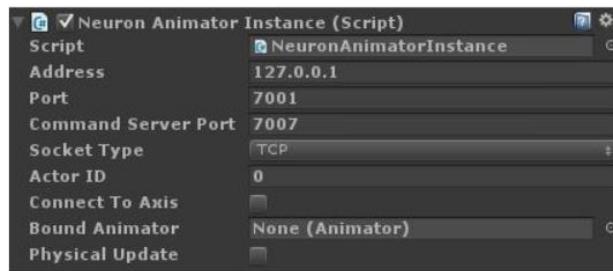
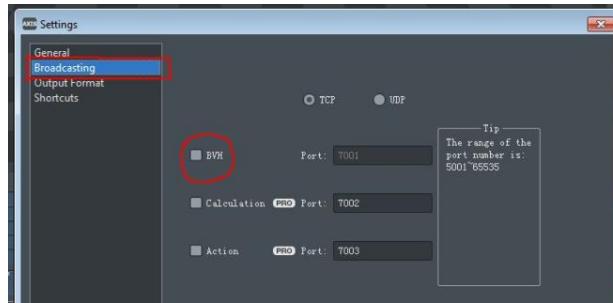
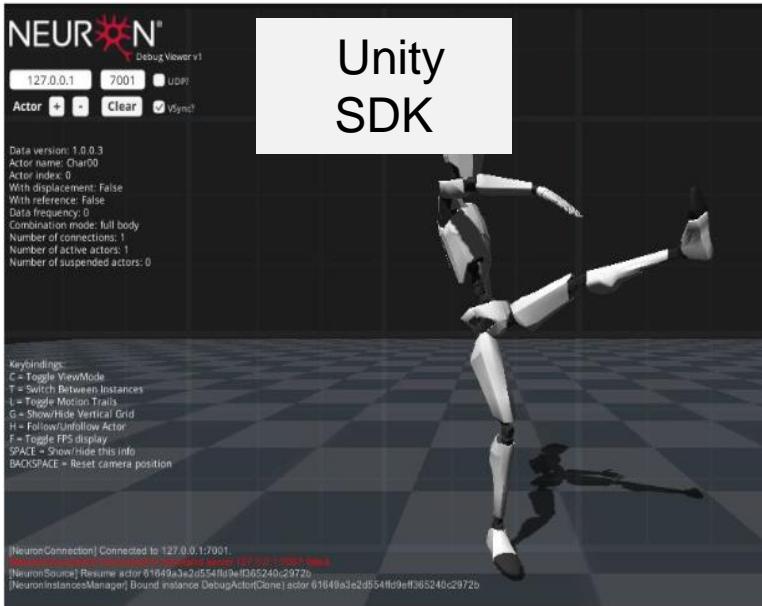


T-pose



S-pose

Unity + Noitom in Real Time

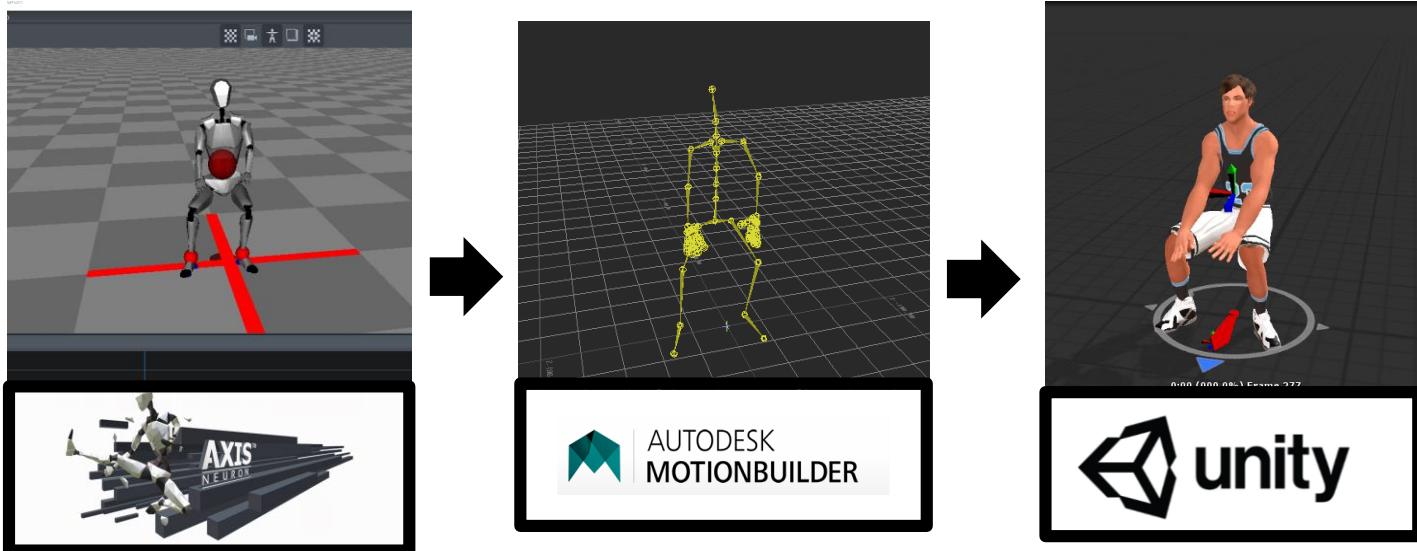


<https://www.youtube.com/watch?v=UOinLfGGJyl>

<https://neuronmocap.com/software/unity-sdk>

https://neuronmocap.com/sites/default/files/downloads/unity_integration.pdf

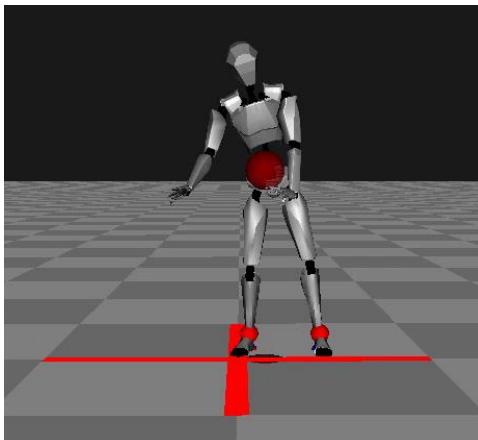
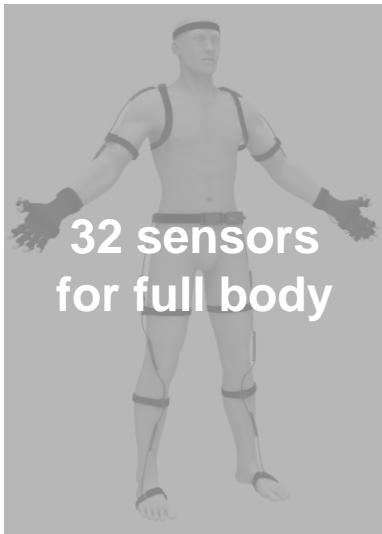
Unity + Noitom Skeleton Data



<https://hackmd.io/s/BJdpVKWpG#>

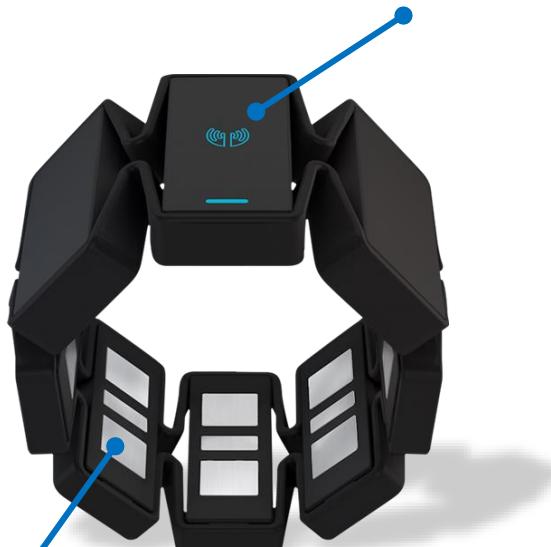
Limitation of Noitom

- ⌚ Wearing full body suit is time-consuming
- ⌚ Drifting problem
- ⌚ Need to properly set the size of every body part



Female150	
Head	15.43
Neck	7.71
Body	50.39
Shoulder Width	29.00
Upper Arm	19.50
Forearm	20.00
Palm	15.50
Hip Width	20.00
Upper Leg	36.51
Lower Leg	34.96
Heel Height	6.62
Foot Length	21.50
Female145	

Inertial Measurement Unit (IMU)



Electromyography (EMG)

Myo

Decompose Myo



Gestures Provided by Myo SDK



DOUBLE TAP



WAVE LEFT



WAVE RIGHT

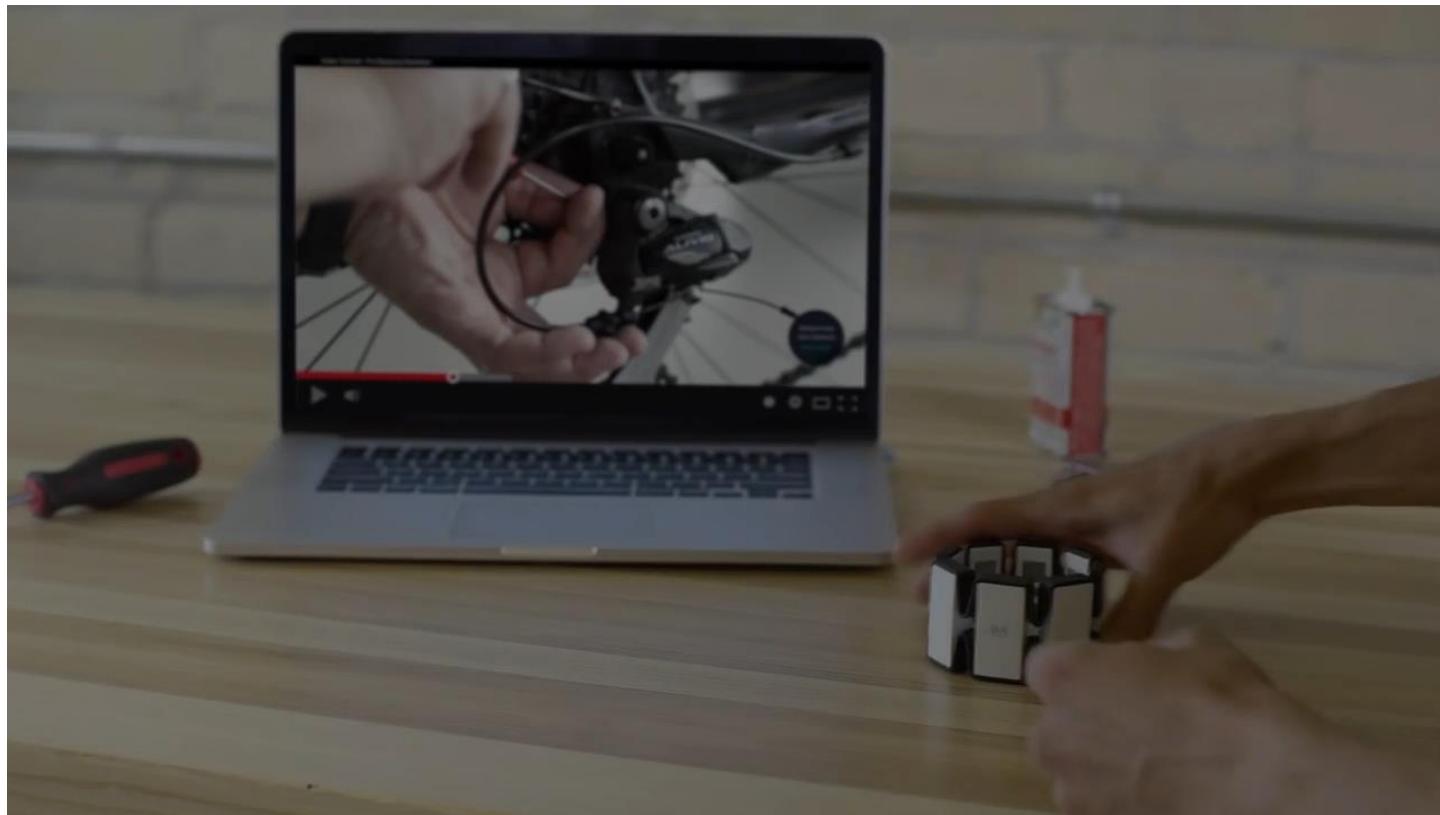


SPREAD FINGERS



MAKE FIST

Demo Video of Myo

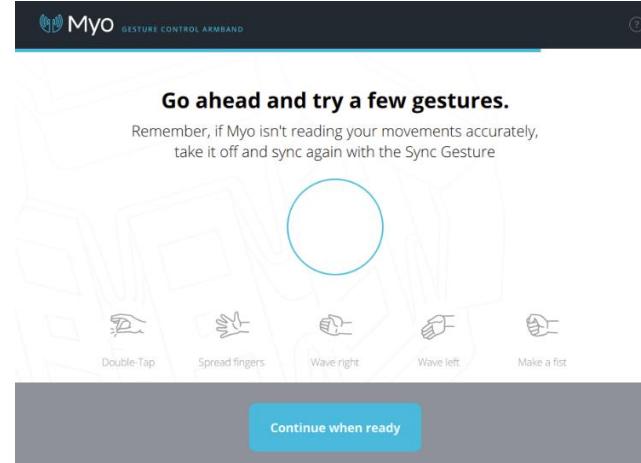


SDK Installation (1)

- ➊ Download Unity and Visual Studio
- ➋ Download Myo Connect from
<https://developer.thalmic.com/start/>



要按照規定插好USB和Cable，讓系統更新韌體，更新韌體時不要動到線，否則硬體很容易壞。



如果沒辦法正確偵測這些手勢，請把Myo關掉重開。

SDK Installation (2)

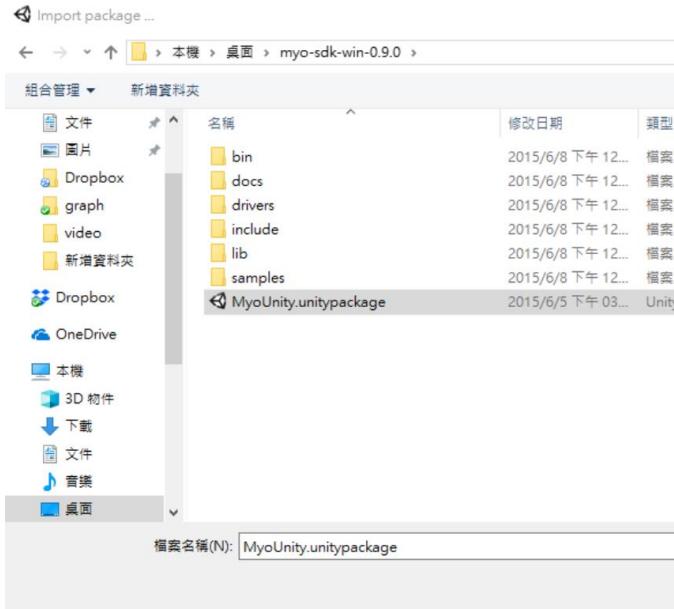
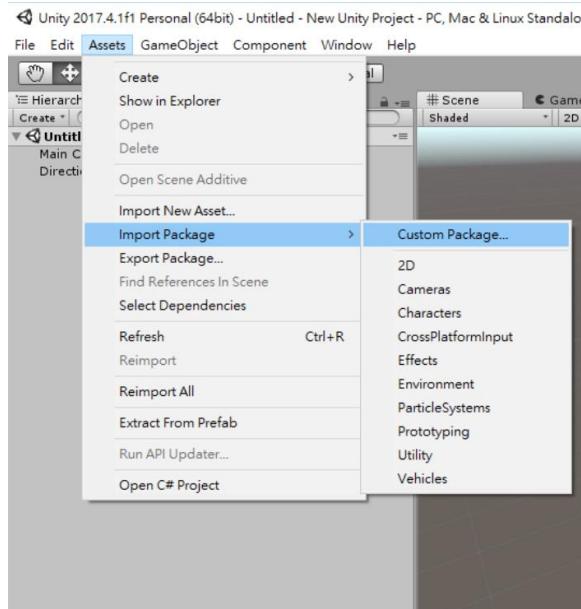
- Download Windows SDK or Mac SDK from
<https://developer.thalmic.com/start/>
- Extract the *.zip file.



SDK Installation (3)



Open Unity



Assets → Import Package → Custom Package

找到剛剛解壓縮的SDK檔案，選取 MyoUnity.unitypackage

Tutorial of Myo

- ➊ Setting up the Myo Package in Unity
 - <http://developerblog.myo.com/setting-myo-package-unity/>

- ➋ Myo Project With Unity3D
 - <https://www.youtube.com/watch?v=3VLoGSVORjY>

Limitation of Myo

- 換使用者時，最好重新Connect和Calibration，因為每個人的肌電訊號強度不同。
- 如果是拿來控制Animation，沒辦法做太精確的手指動作。如果是拿來與虛擬物件互動，建議就使用內建手勢。
- 同台電腦無法同時收兩個Myo的訊號。