



Unity Engine Course

Introduction to Unity & C#



Syllabus & Grading

11/18	Morning	Introduction to Unity & C#	HW1 (25%)
	Afternoon	UI & Framework & IO of Unity	
11/25	Morning	2D Game Design	HW2 (25%)
	Afternoon	Create your first 3D scene	
12/02	Morning	FPS Game Development	Project (40%)
12/09	Morning	Show time !	

Attendance (10%)

TA

助教：彭建瑋 陳建文 陳文正 張矽晶 蘇俐文

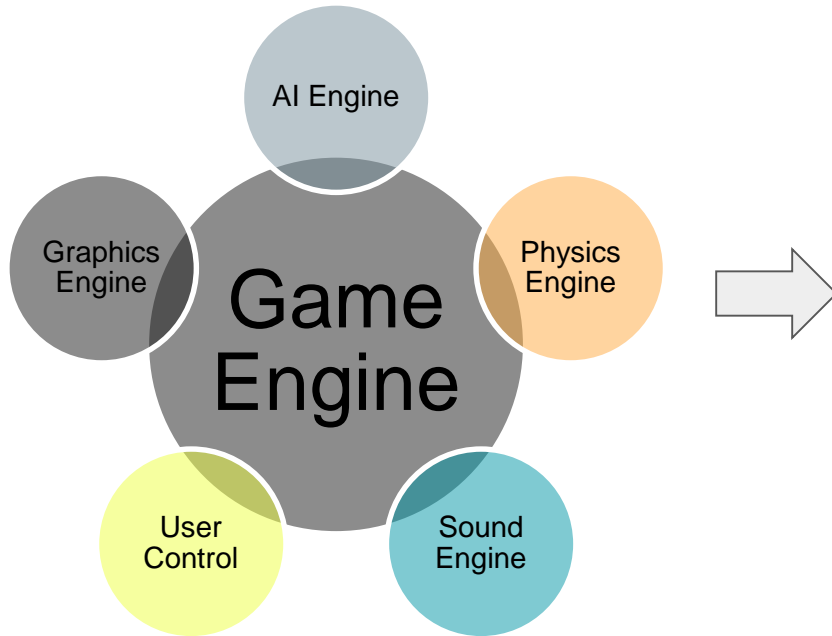
Email: unity@mislalab.csie.ncku.edu.tw

Office: 資工新館 6樓 R65601

Office Hours: 11/19, 11/26, 12/3 (Appointment by email)

What is game engine & Why we need it?

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

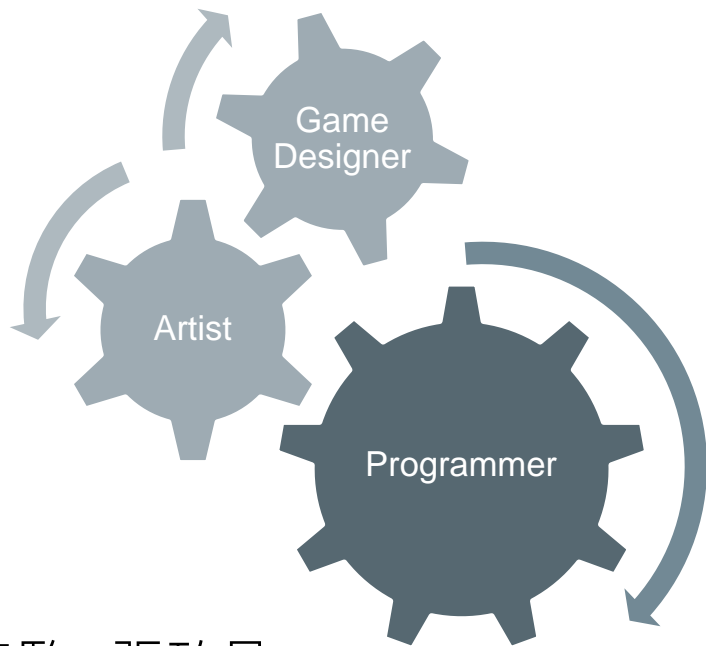


2D/3D Games

What Game Engine should I use?



Game Development is Interdisciplinary

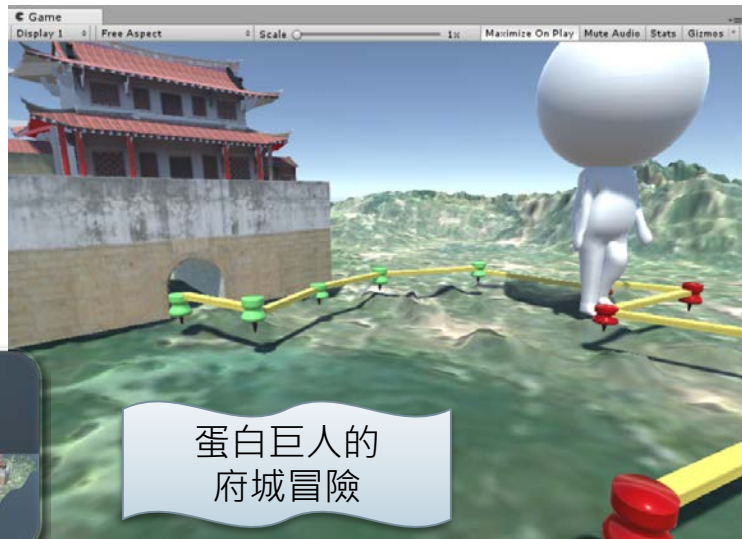
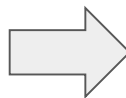


If you are an interdisciplinary talent

- Do the project on your own

Else, find your team member for the project ~

- Each team : no more than 2 people



蛋白巨人的
府城冒險

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;
```

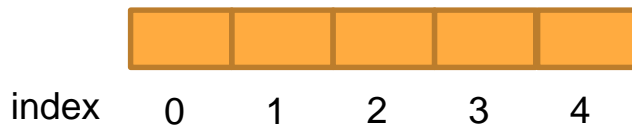
enemy 1

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

- 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

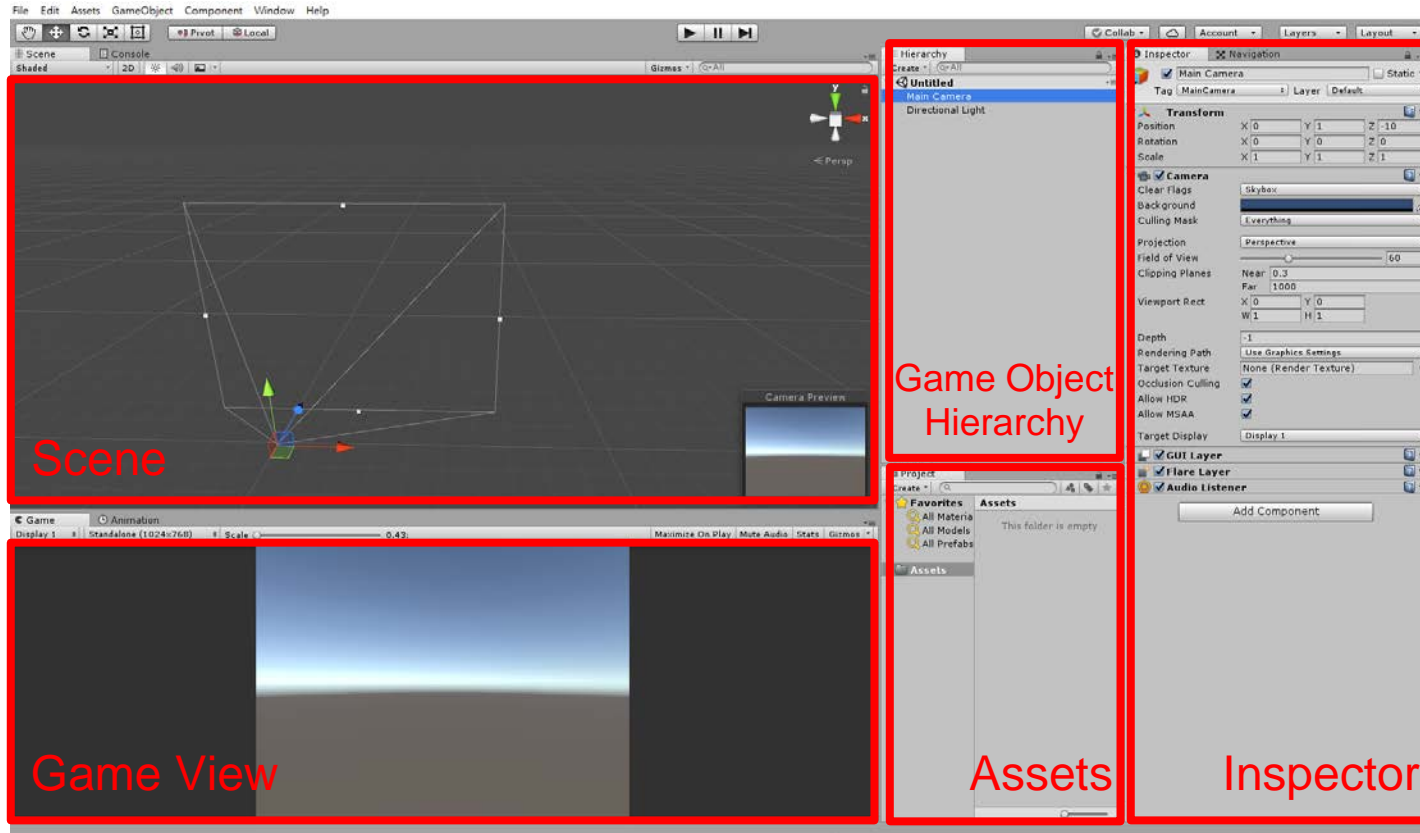
- If Statement

```
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;
}
```

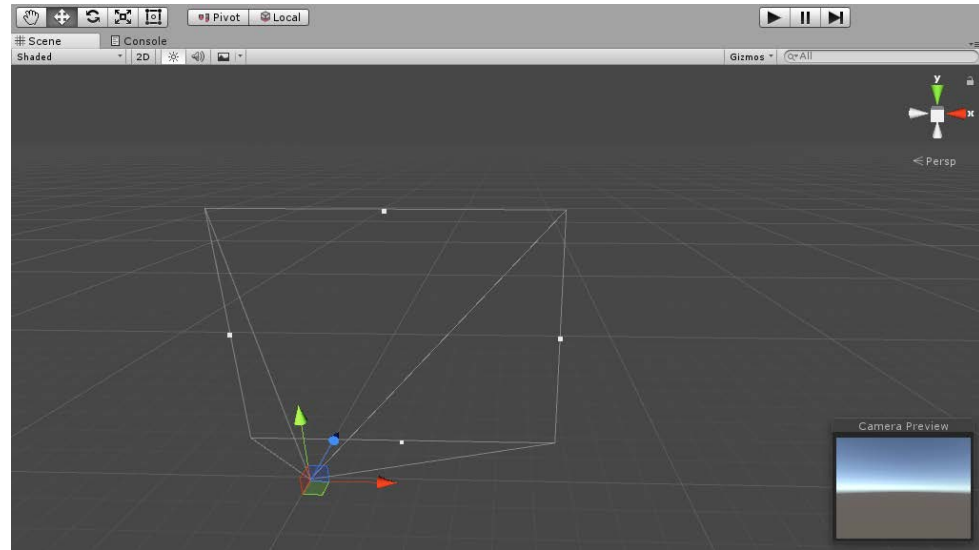
Unity Windows



Scene

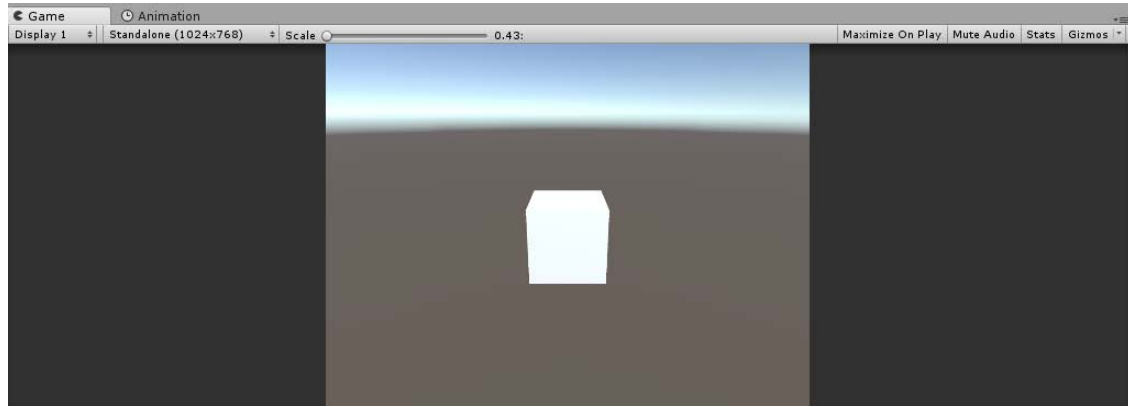
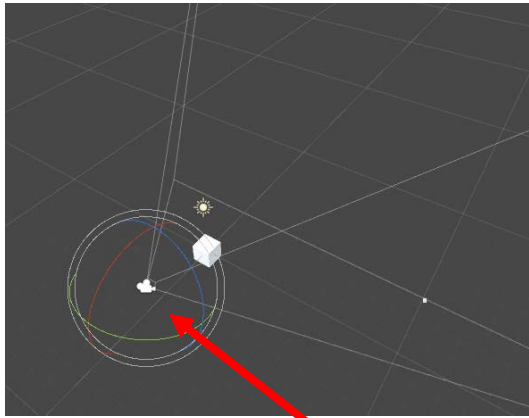
- You can scale, rotate, and move GameObjects in the scene
- You can also drag prefabs in your assets into the scene

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

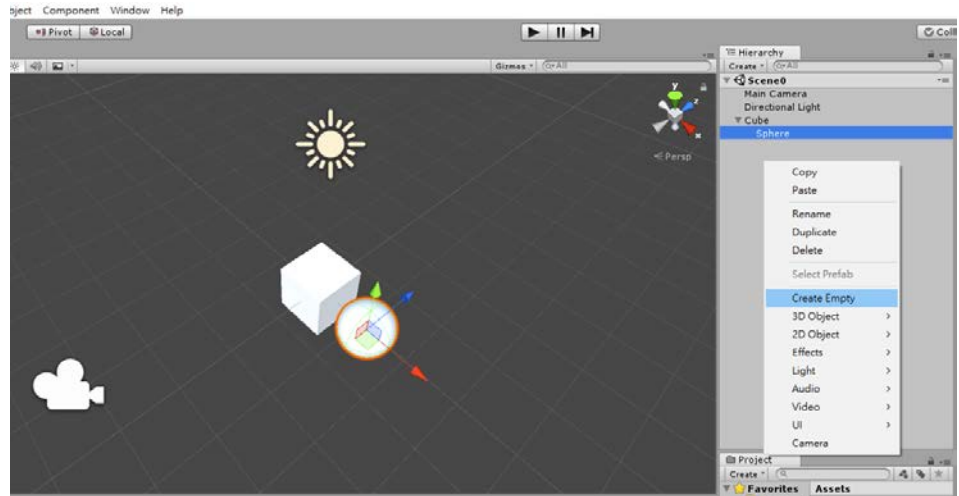
- The game view is rendered from the camera(s) in the scene



Current camera
in the scene

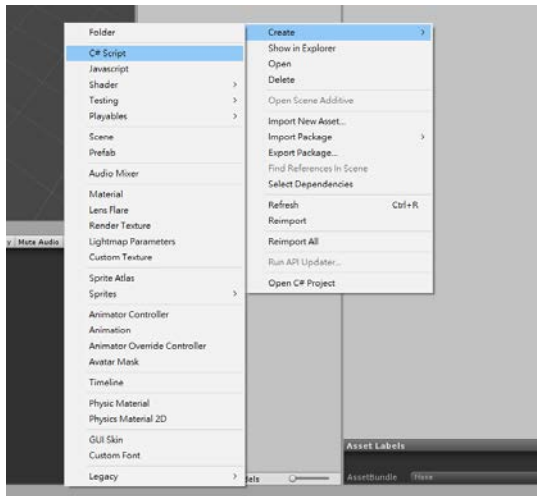
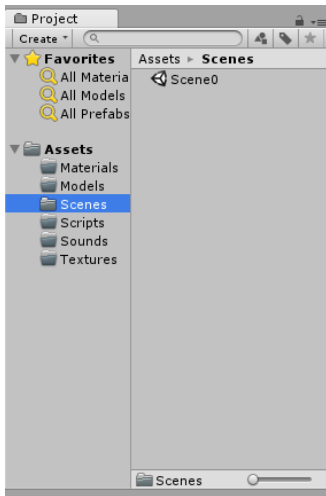
Game Object Hierarchy

- 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

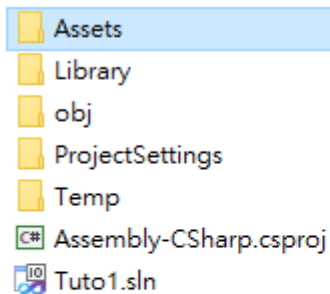
- Unity views all the files as assets, which can be imported to your game
- Put all your assets in the default “./Assets” folder under your project folder
- You also can create a new assets from the Unity asset template



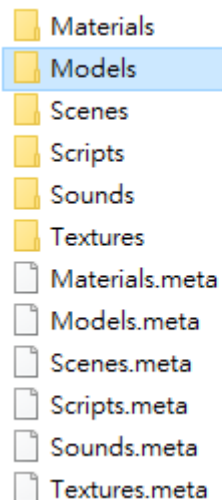
Assets

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

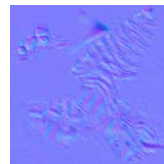
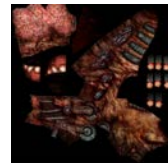
Project Folder: “./”



Asset Folder: “./Assets/”

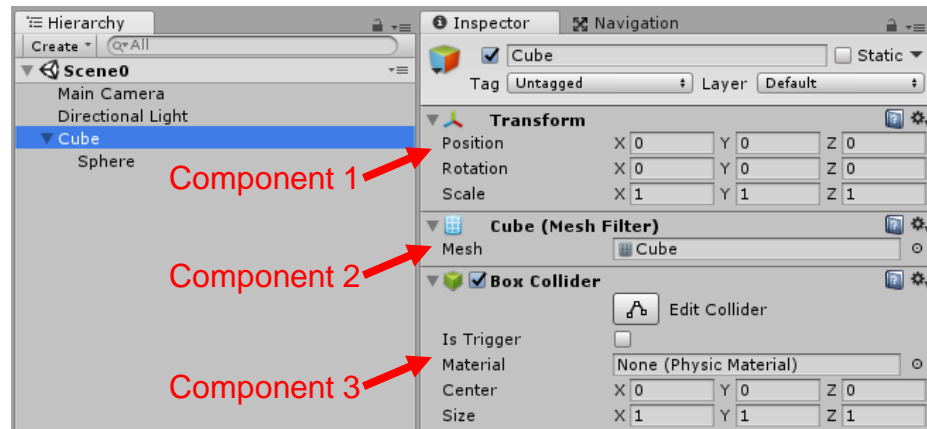
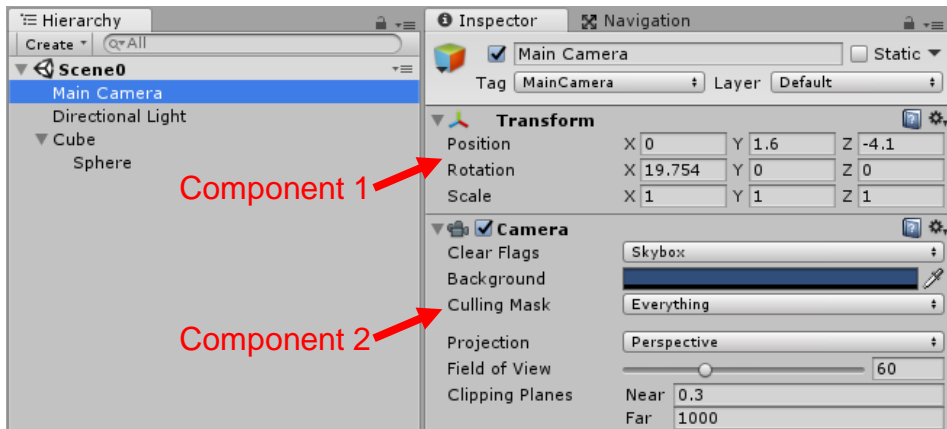


*.obj
*.fbx
*.wav
*.jpg
*.png
...



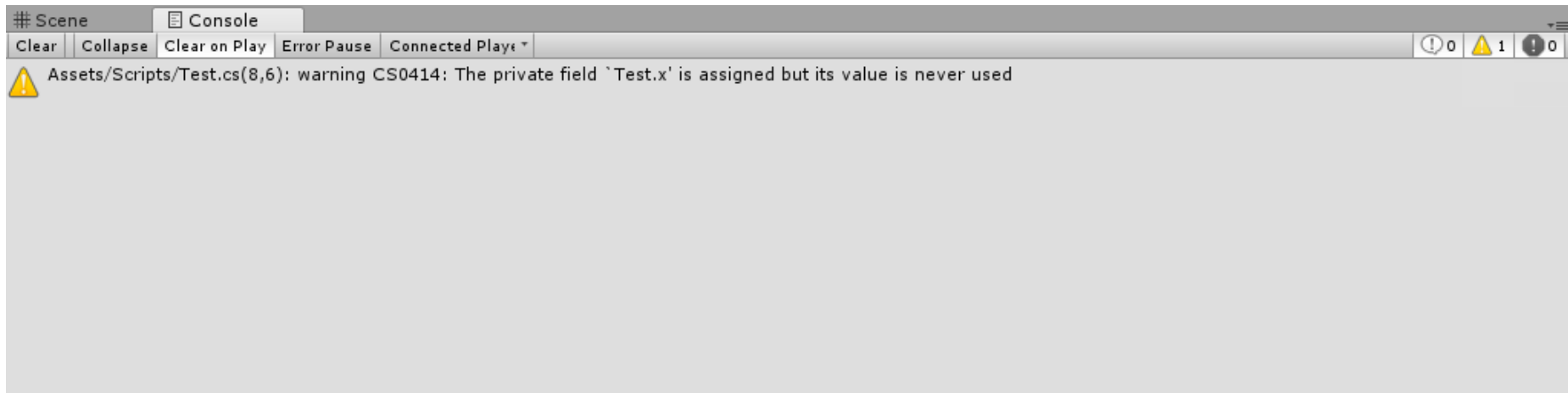
Inspector

- 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

- Update()

- 每個影格都執行一次的程式碼寫在這

```
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

Attribute

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

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