**Part0介紹**

**Part01編輯 Component**

**Part02編輯 GUILayer**

(彈跳視窗 參考http://blog.csdn.net/u011416077/article/details/50359878)

**Part03 Custom Inspector 1**

<file: PlayerInspector>

* Custom Inspector 自訂檢視器

(參考http://www.morningfungame.com/2016/09/unity-c-custom-inspector.html)

* 隱藏前面Component的性質，只顯示要顯示的特性

(EditorGUILayout.LabelField("Player id : " + **player.id**);)

* 換GUI顏色 (GUI.color = Color.green)，但會改變這行以後的GUI都變色!
* 矩形的進度條!!!

**Part04 Custom Inspector 2**

<file: PlayerInspector>

* 下拉式選單 GUILayout.Foldout()
* 警告提示框 HelpBox
* 放多個GUI在同一列

**Part05 Custom Inspector 3**

<file: PlayerInspector2>

SerializedProperty 將編輯器序列化(從小排到大之類的)

[CustomEditor(typeof(*class name*))] // 這個Editor是哪個腳本的界面

[CanEditMultipleObjects] // 可以一起編輯多個物體

**Part06 Custom Inspector 4**

<file: PlayerInspector、RandomNameGenerator>

隨機生成名字。

**Part07 Editor Windows 1**

<file: MyFirstEditorWindow>

建立彈跳視窗。

關閉視窗功能。

**Part08 Editor Windows 2**

using System.IO;// 儲存BUG資料

Application.CaptureScreenshot// 系統內對場景截圖

**Part09 Handles 1**

Handles.Label //名字標籤



Handles.RadiusHandle //半徑控制

Mathf.Repeat(t, length) //循環數值t，0到length之間。t值永遠不會大於length，也永遠不會小於0。

**Part10 Handles 2**

在場景加字加按鈕。

**Part11 Gizmos**

有時候，我們會想要在場景中繪製一些符號、圖案或區塊，來標記一些特殊的區域。例如：重生點、死亡區或生怪區等。這些區塊我們只想要在編輯時看到，**執行遊戲時是不能讓玩家看到的**。這種情形很適合利用 Gizmos 來繪製物件。

(<https://play2gethergamenote.blogspot.tw/2016/09/unity-gizmos.html>)

#為了測試Gizmos可以達到跟Handles一樣的效果，有在AIInspector註解掉一些程式碼。

**Part12 Property Drawers**

<file: Shoe、ShoePropertyDrawer>

Property Drawers are scripts that control how a particular class will look inside the default Unity Inspector.

[System.Serializable]//可以植入一個類作為子屬性顯示在Inspector（檢視面板）。

(參考網站: http://godstamps.blogspot.tw/2015/06/unity-propertyattribute-propertydrawer.html)

**Part13 Property Attributes**

<file: ReadOnlyAttribute、ReadOnlyAttributeDrawer>

# ReadOnlyAttribute要跟Player在同一個資料夾，這樣[ReadOnly]才會有作用!

(參考網站: http://godstamps.blogspot.tw/2015/06/unity-propertyattribute-propertydrawer.html)

**Part14 Decorator Drawers**