Photon Cloud(五)
解決PUN資料庫問題

主講：紀曲峰

PUN的限制

- 無法撰寫Server端邏輯
- 無提供資料庫功能(就算有也不建議使用)

解決PUN資料庫方案

- 改用Photon Server
- 使用WWW或SOAP連結一般網頁資料庫後再導回PUN做後續處理

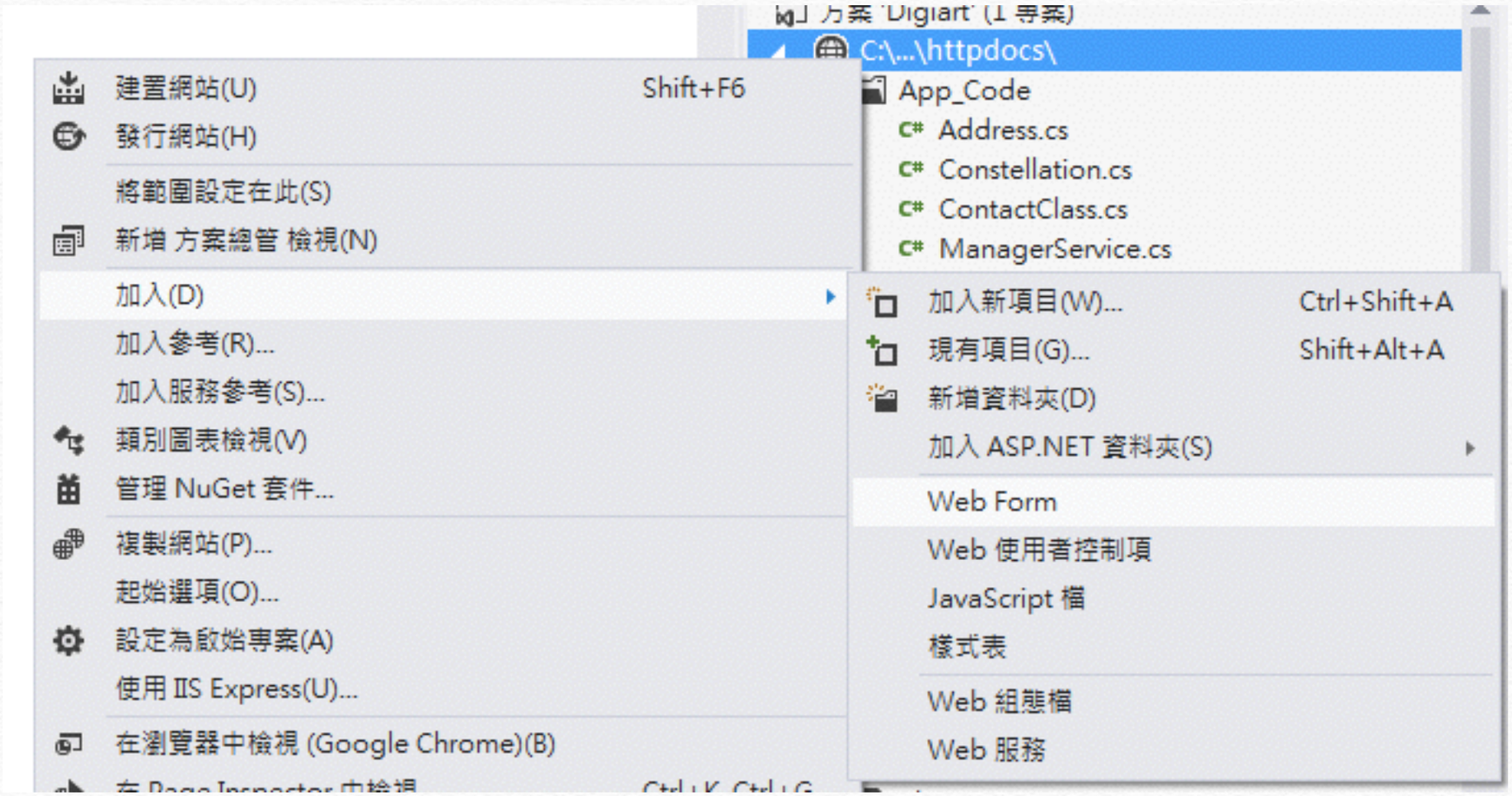
建立Server端網頁

網頁說明

- 此處使用aps.net建立模擬用的網頁
- 本範例開發 asp.net 使用 visual studio 2012，使用其他版本者請自行修改

建立Login網頁

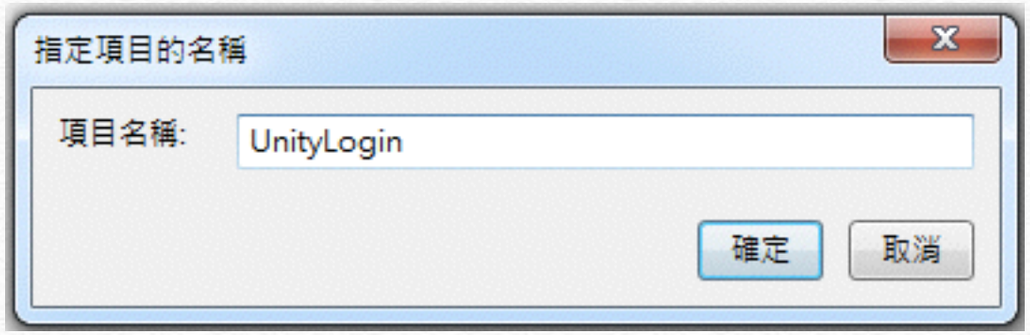
- 在VS的網頁專案下按滑鼠右鍵，建立新的Web Form



網頁名稱

- 此範例將網頁取名為UnityLogin，若使用其他名稱請自行修改後面範例讀取的網頁名稱

- 筆者習慣將程式碼與網頁分開，讀者可自行決定



前端頁面修改

- 開啟UnityLogin.aspx檔，只保留第一行，其他的內容均刪除

web.config　UnityLogin.aspx.cs　UnityLogin.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="UnityLogin.aspx.cs" Inherits="UnityLogin" %>
```

加入登入邏輯

- 加入簡單的登入邏輯，讀寫資料庫部份請自行撰寫，這裡先加入固定的abcd及qwer兩組帳號

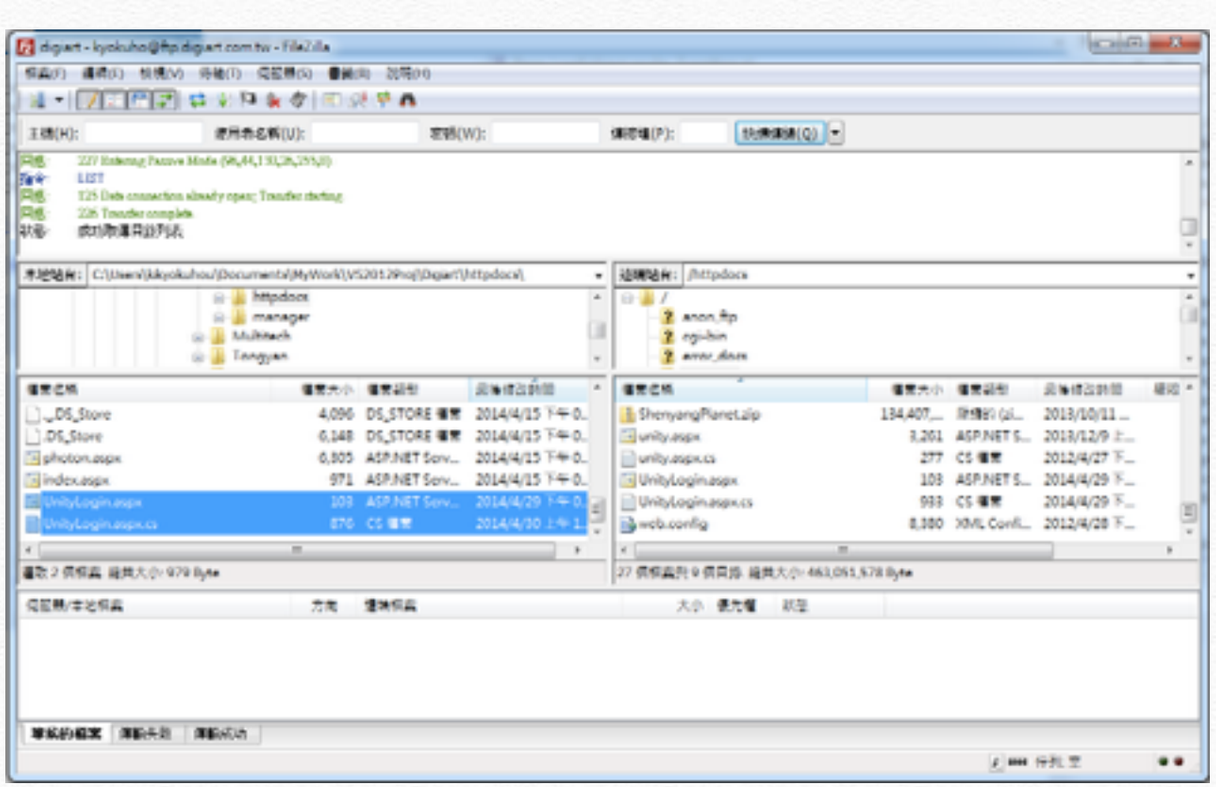
```
public partial class UnityLogin : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        try {
            string memberID = Request["id"].ToLower();
            string memberPW = Request["pw"];

            if (memberID == "abcd" && memberPW == "1234") {
                Response.Write("1&" + "小時候胖不是胖");
            }
            else if (memberID == "qwer" && memberPW == "1234") {
                Response.Write("1&" + "Unity達人");
            }
            else {
                Response.Write("0&" + "帳號或密碼錯誤");
            }
        }
        catch {
            Response.Write("0&" + "帳號或密碼錯誤");
        }
    }
}
```

上傳網頁

將網頁上傳到空間

- 使用www的好處是一般的租用空間也可以使用，不需架設專用主機，省去昂貴的頻寬及機房費用
- 利用ftp上傳到空間主機上



新增Login場景

建立Login

- 請另外加入一個新場景，取名為Login

Unity中的GET方法

GET的網頁溝通法

- 網頁有GET和POST兩種溝通法
  - GET方法最為簡單，容易檢查錯誤，但安全性很低
  - 我們的網頁使用 Request[變數名稱] 指令取值，無論是GET或POST都可以取到

建立腳本

- 建立新C#腳本取名為Login
- 加入登入讀取用的變數

```
    private string receiveStr = "";
    private string memberID = "";
    private string memberPW = "";
```

建立GET登入方法

- 建立GET登入用的方法，取名為wwwGETLogin()，必須要回傳IEnumerator介面以利yeild運作
- 空的方法如下

```
IEnumerator wwwGETLogin(string MemberID, string MemberPW) {

}
```

實作GET登入內容

- 先利用WWW方法呼叫要執行的網頁內容，GET可直接將多個傳入值以&連接
- 接下來呼叫yeild將方法插斷回去執行Unity原本的工作，以免在這裡被www鎖住導致遊戲整個暫停

```
    IEnumerator wwwGETLogin(string MemberID, string MemberPW) {
        WWW www = new WWW(string.Format("http://www.digiart.com.tw/
UnityLogin.aspx?id={0}&pw={1}", MemberID, MemberPW));

        yield return www;

        if( www.error != null )
        {
            receiveStr = www.error;
            yield return null;
        }

        receiveStr = www.text;
    }
```

加入登入用的GUI

- 登入用的GUI用法和之前的類似，但在呼叫wwwGETLogin()時使用StartCoroutine進行呼叫

```
void OnGUI () {
     GUILayout.BeginHorizontal ();
     GUILayout.Label ("MemberID : ");
     memberID = GUILayout.TextField (memberID, GUILayout.Width (100));
     GUILayout.EndHorizontal ();

     GUILayout.BeginHorizontal ();
     GUILayout.Label ("Password : ");
     memberPW = GUILayout.TextField (memberPW, GUILayout.Width (100));
     GUILayout.EndHorizontal ();

     if( GUILayout.Button("Login") ) {
          StartCoroutine(wwwGETLogin(memberID, memberPW));
     }

     if( receiveStr.Length > 0 ) {
          GUILayout.Label(receiveStr);
     }
}
```

淺談yield

- yield return 與 return 的不同在於，使用return返回該函式後面的命令都不會被執行到，但使用yield return返回，系統會再回來將後面的程式碼執行完畢

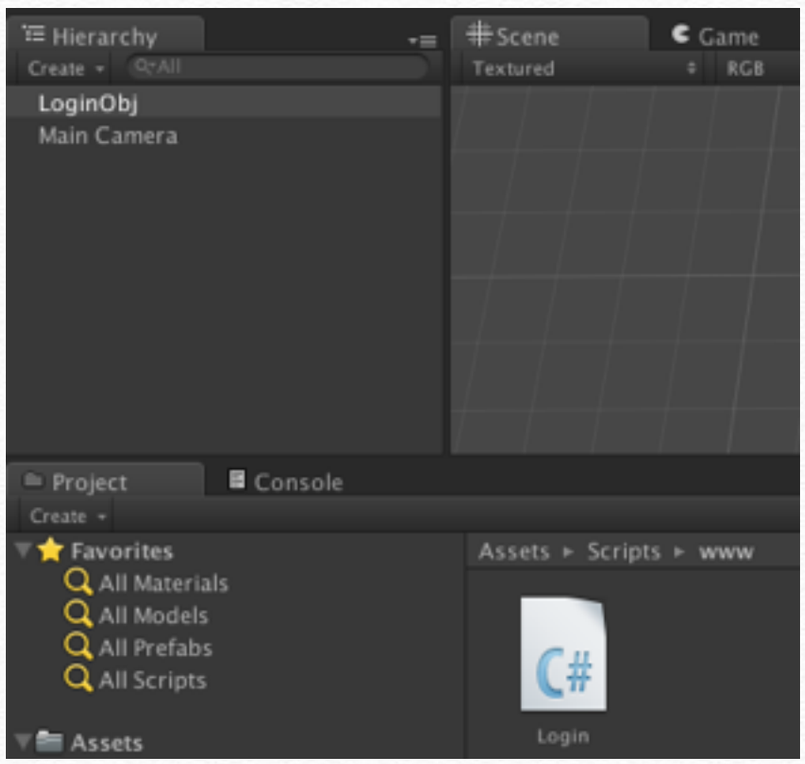
- 使用yield的方法必需宣告IEnumerator為回傳值

StartCoroutine

- 一般稱為協程
- 因為Unity3D沒有執行序，因此使用協程來處理多執行序的需求
- 在登入結束，轉換場景時，最好呼叫「StopAllCoroutines();」停止所有的協程

將腳本加入場景

- 加入一個Empty物件，取名為LoginObj
- 將腳本加到LoginObj上



測試

- 執行後只要輸入正確的帳號密碼即會傳回「 1&暱稱 」了

將密碼攻成密碼輸入欄

- 目前的密碼是明碼顯示，感覺上很不專業而且危險，改為 PasswordField 改以*號顯示輸入內容

```
memberPW =
GUILayout.PasswordField (memberPW, "*"[0], 10, GUILayout.Width (100));
```

Unity中的POST方法

建立POST的登入方法

- GET是危險性比較高，且容易發生URL編碼錯誤的傳輸方法，一般來說都不建議使用GET去存取網路資料

- 使用 POST存取不旦安全性較高，也沒有傳輸的容量限制，但為了網路效能讀者最好還是儘量減小傳輸內容

```
IEnumerator wwwPOSTLogin(string MemberID, string MemberPW) {

}
```

加入POST的內容

- POST和GET方式差不多，但多加了標頭的部份

```
IEnumerator wwwPOSTLogin(string MemberID, string MemberPW) {
    System.Collections.Hashtable headers = new Hashtable ();
    headers.Add ("Content-Type", "application/x-www-form-urlencoded");

    string loginData = string.Format("id={0}&pw={1}", MemberID, MemberPW);
    byte[] binStr = System.Text.UTF8Encoding.UTF8.GetBytes (loginData);

    WWW www = new WWW ("http://www.digiart.com.tw/UnityLogin.aspx", binStr,
headers);

    yield return www;

    if( www.error != null )
    {
        receiveStr = www.error;
        yield return null;
    }

    receiveStr = www.text;
}
```

修改OnGUI

- 修改OnGUI的登入指令，改成呼叫wwwPOSTLogin()

```
        if( GUILayout.Button("Login") ) {
            //StartCoroutine(wwwGETLogin(memberID, memberPW));
            StartCoroutine(wwwPOSTLogin(memberID, memberPW));
        }
```

修改Server端只讀取POST

- 目前的Server端使用Request[]讀取參數，這種方法可同時讀取GET和POST，但GET很容易被玩家利用try主機漏洞，因此最好改成只讀取POST的內容

- 修改網頁，將Request改成Request.Form強制讀取POST內容

```
string memberID = Request.Form[“id"].ToLower();
string memberPW = Request.Form["pw"];
```

測試

- 重新用GET方式登入會傳回失敗，只有用POST登入可以成功了

解析回傳値

解析回傳字串

- 利用string.Split切割字串，比對格式是否正確，若正確表示是網頁的回傳內容

```
if( receiveStr.Length > 0 ) {
    string[] splitStr = receiveStr.Split(new char[] {'&'} ); // 不正確的回傳值
    if( splitStr.Length != 2 ) {
        GUILayout.Label(receiveStr);
    }
    else if ( splitStr[0] != "0" && splitStr[0] != "1" ) { // 不正確的回傳值
        GUILayout.Label(receiveStr);
    }
    else {
        if( splitStr[0] == "0" )
            GUILayout.Label(splitStr[1]);
        else
            GUILayout.Label("登入成功，暱稱 : " + splitStr[1]);
    }
}
```

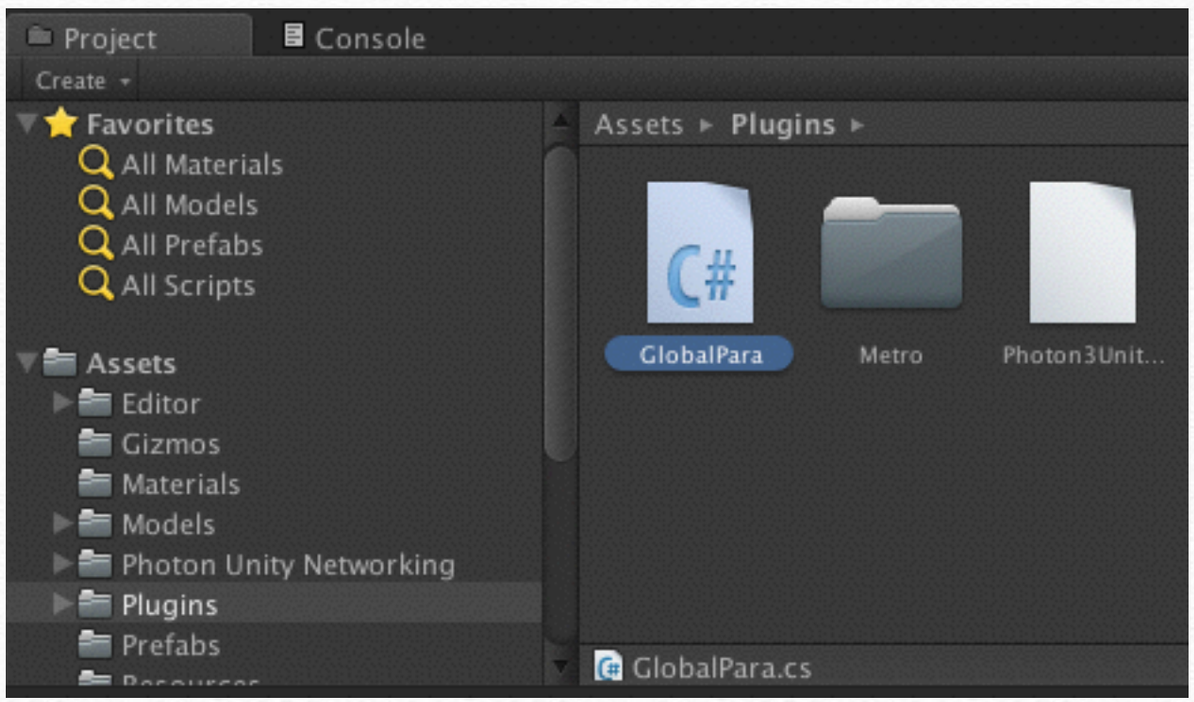
執行測試

- 重新執行並測試結果

將登入資訊回傳PUN

建立全域變數

- 跨場景傳值最簡單的方法就是利用全域變數
- 全域變數最好放在Plugins資料夾以確保所有語言的腳本都讀得到
- 在Plugins資料夾內新增一個C#腳本取名為GlobalPara



加入靜態變數

- 在腳本內加入存放暱稱的靜態變數
- 因為是純變數因此不需要繼承自MonoBehaviour

```
using UnityEngine;
using System.Collections;

public static class GlobalPara {
    public static string NickName = "";
}
```

在登入時清空暱稱

- 回到登入頁的Login腳本，加入Awake ()方法，將全域變數的NickName清空

```
void Awake () {
    GlobalPara.NickName = "";
}
```

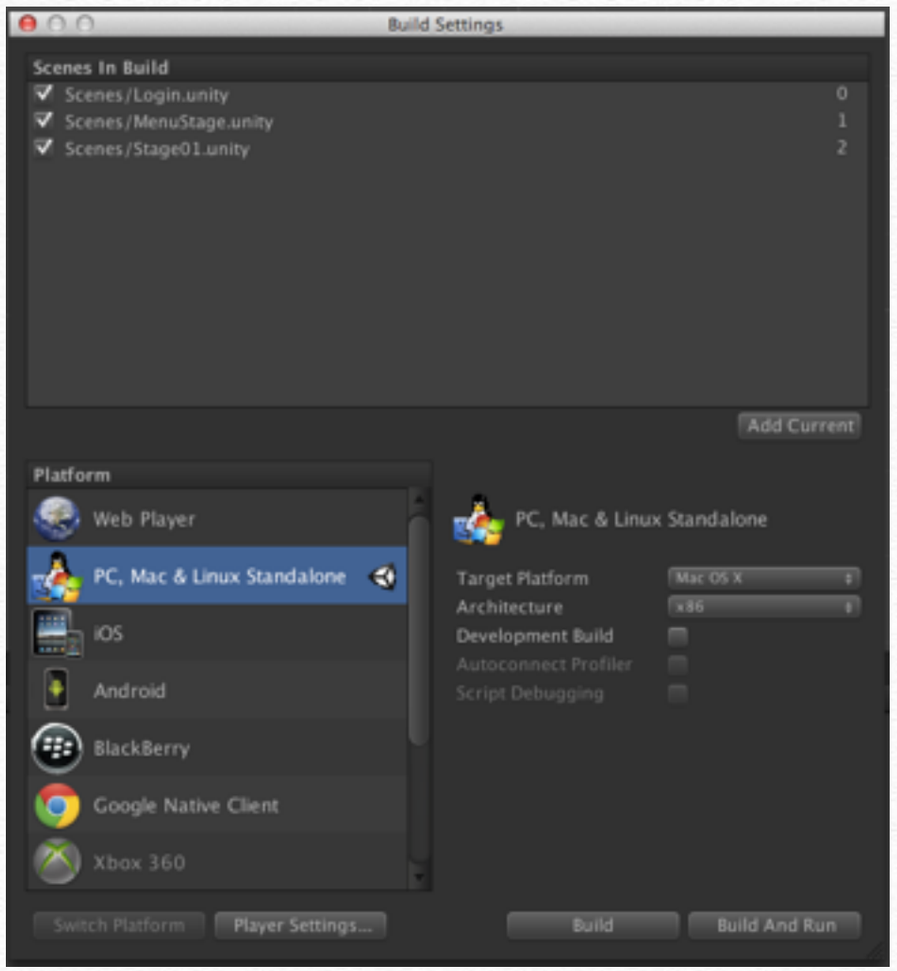
修改GUI登入成功部份

- GUI登入成功部份進行修改，若是成功則將暱稱存放於GlobalPara.NickName，然後跳轉到MenuStage場景

```
        else {
            if( splitStr[0] == "0" )
                GUILayout.Label(splitStr[1]);
            else {
                GUILayout.Label("登入成功，暱稱 : " + splitStr[1]);
                GlobalPara.NickName = splitStr[1];
                Application.LoadLevel("MenuStage");
            }
        }
```

將Login加入編譯列表

- 開啟Build Settings視窗，將Login場景加入列表，並排在第一位



MenuStage登入判斷

- 回到MenuStage場景的PhotonRoomMenu腳本，在Awake()裡加上若全域變數的暱稱為空字串則跳回登入頁重新登入，否則就將暱稱存放於playerName變數內

```
void Awake()
{
    if( GlobalPara.NickName == "" ){
        Application.LoadLevel("Login");
        return;
    }
    playerName = GlobalPara.NickName;
    PhotonNetwork.ConnectUsingSettings("1.0");
}
```

MenuStage暱稱處理

- 因為玩家不再需要取暱稱了，在取暱稱處直接改成顯示目前登入暱稱

```
// 玩家取暱稱
GUILayout.Label("Player name :");
//playerName = GUILayout.TextField(playerName, GUILayout.Width(200));
GUILayout.Label(playerName);
```

測試遊戲

- 重新以Login方式執行遊戲，玩家不再需要取暱稱了

網頁空間注意事項

適合的網路空間

- 網路速度
- 價格及收費方式
- 主機位置
- 支援的網頁語法及資料庫
  asp.net, PHP, JSP, MSSQL, MySQL . . .

不要使用sa作為帳號

- sa權限過高
- sa一旦被破解易造成嚴重傷害
- 應建立存取資料庫專用帳號，並設該帳號只能存取授權的資料庫

限制輸入內容及長度

- 登入頁的帳號密碼請限制輸入長度
- 輸入的內容要做安全判斷，只允許認可的字元內容
- 認識常見的隱碼攻擊技巧，試著攻擊自己的服務

資料的安全

- 就算一般資料沒有加密，至少玩家的密碼務必要加密
- 若沒有自行維護加密主機的能力和預算，金流部份採用介接銀行或平台商的金流
- 除非有過人的自信，不要存放信用卡資料，最好個資也盡量不要存

加密方式

- Base64並非加密，請勿用Base64當作加密
- 雜湊
- 金鑰加密

End