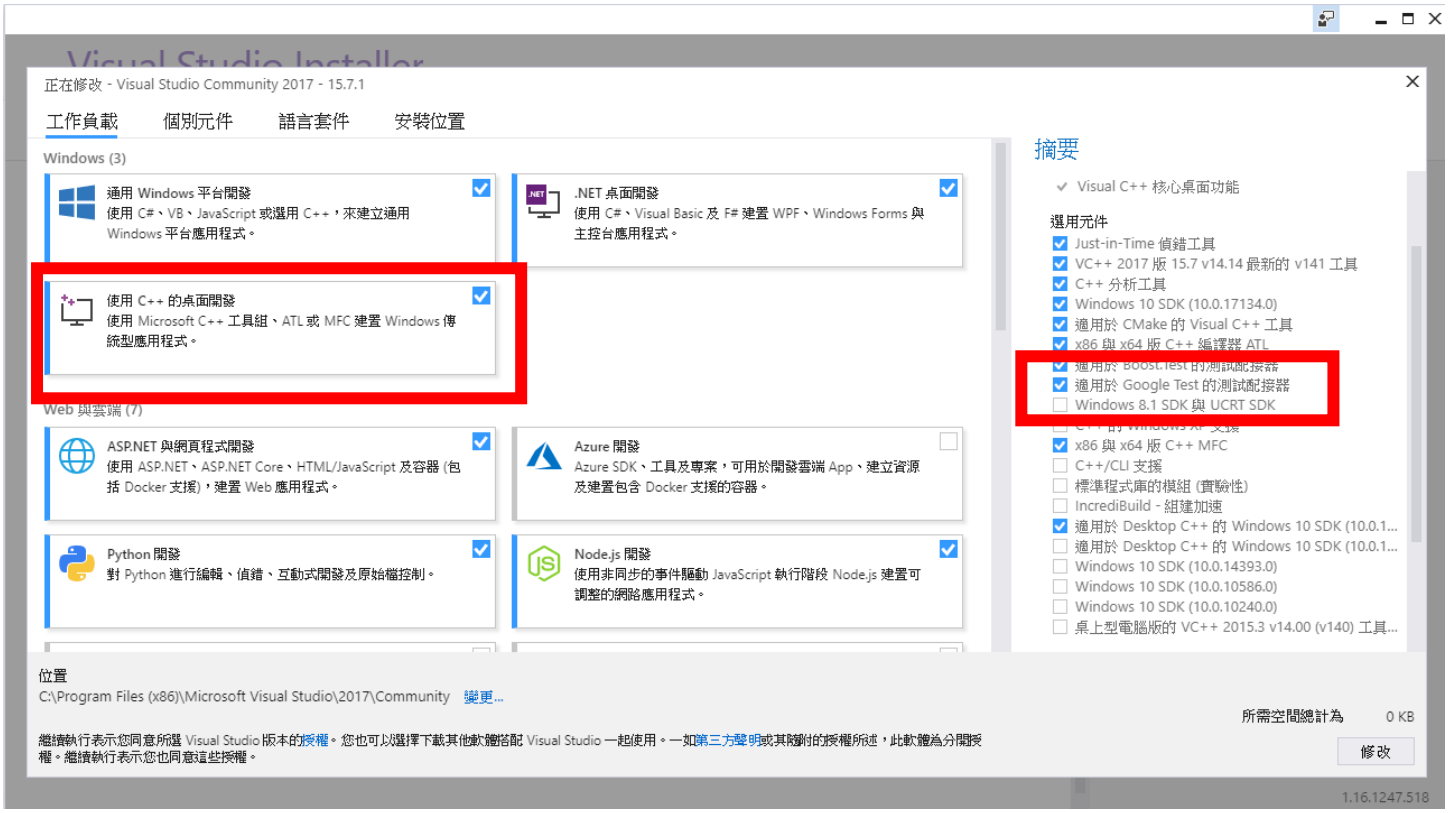
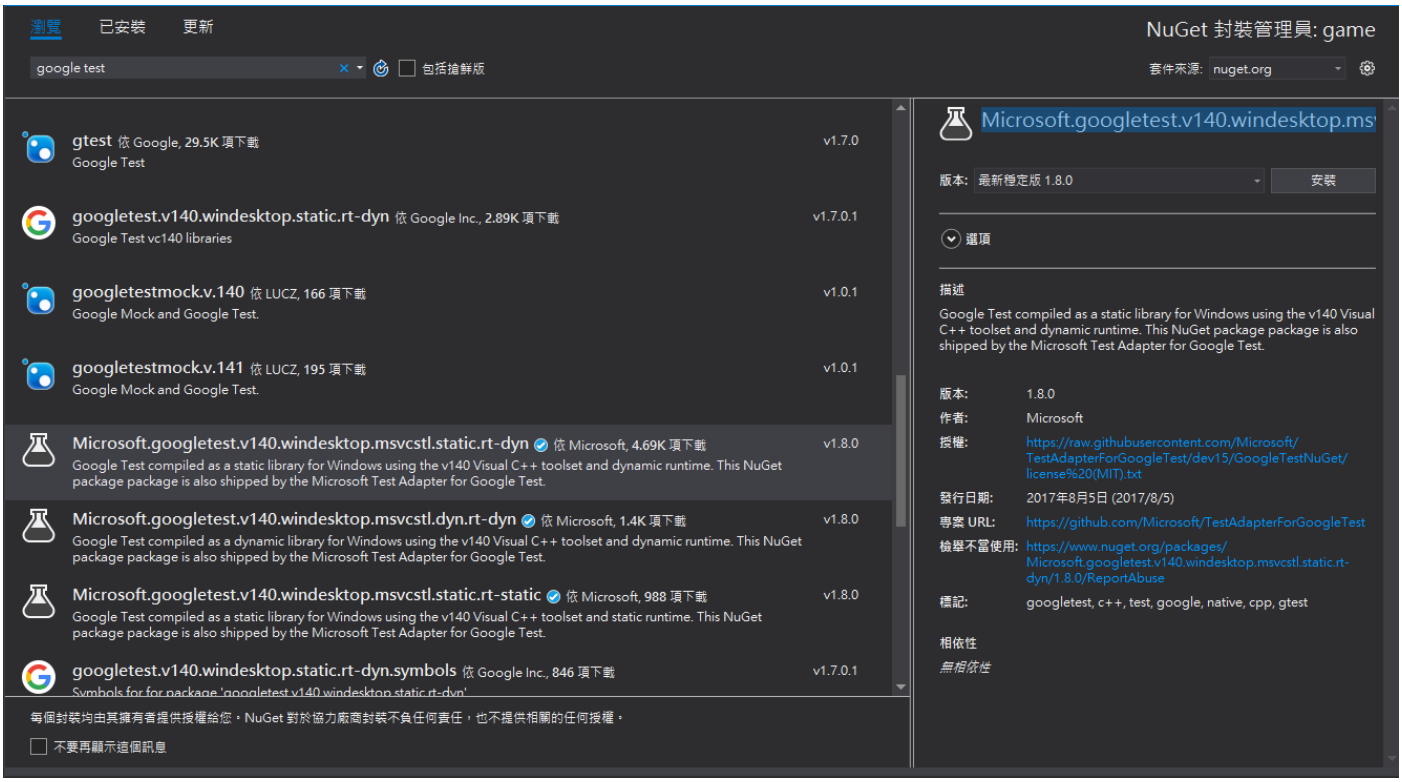
GameFramework with Google Test (gtest)

2018.5 By Cheng Hsin

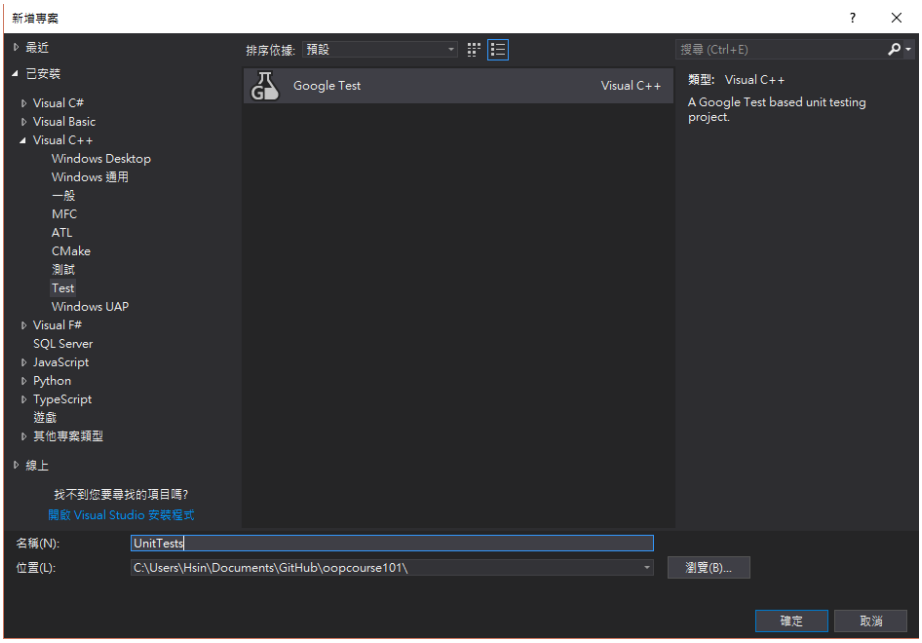
Step.1 在 Visual Studio 2017 中，開啟 Visual Studio Installer，安裝「使用 C++的桌面開發」選取「適用於 Google Test 的測試配接器」

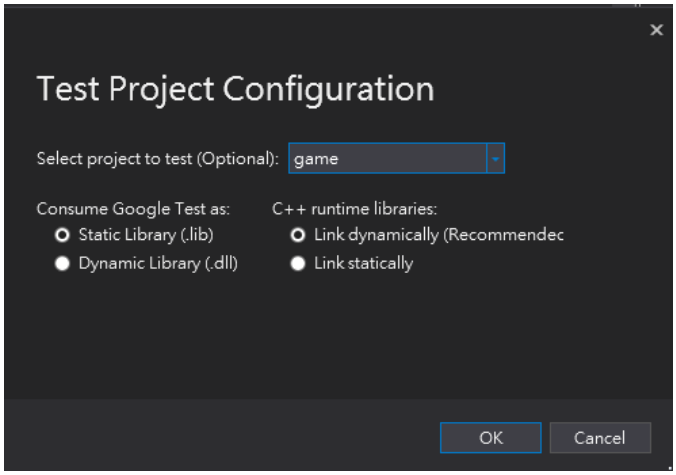


Step.2 對方案按右鍵，點選「管理方案的 NuGet 套件」，安裝「Microsoft.googletest.v140.windesktop.msvcstl.static.rt-dyn」

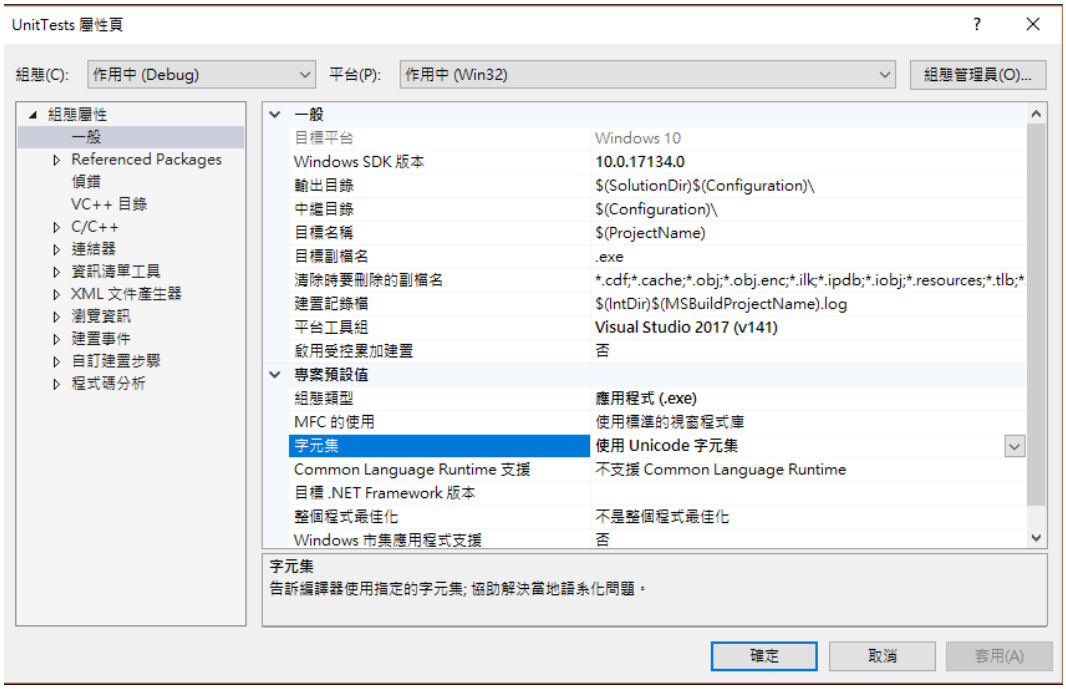


Step.3 加入新測試專案至方案(必須先完成 Step1&2 才會出現 Google Test)





Step.4 對測試專案按右鍵，確認屬性設定



Step.5 因原遊戲程式中有許多圖形介面及系統環境之函式庫，我們在 UnitTest 中並不需要去測試，然而若缺少其一或是環境不符，測試就無法被編譯，此時最好的做法，就需要利用 Fake Object 來騙編譯器，簡單來說就是沒有功能只有介面的 Class 和 function，讓我們不須 include 一堆不用測試的東西，亦可減少測試與環境之 dependency。

1. 新增檔案放 Fake Object，並加入會用到之空介面

```
class LPDIRECTDRAWCLIPPER {};
class HRESULT {};
class LPDIRECTDRAWSURFACE {};
class DWORD {};
class CRect {};
class CDC {
public:
    CPen * SelectObject(CPen* pPen) {};
    bool Ellipse(int x1, int y1, int x2, int y2) {};
};
class CPen {
```

如圖，Fake Object 就是長這樣，遊戲會用到之介面通常可以在 gamelib.h 中找到。

2. 那原本的程式要如何知道現在是不是在跑測試，要 include 假的物件還是正常時的有功能物件，此時可利用#ifdef 和 #ifndef 指示詞來讓程式知道

(1) 在測試中(可放在 pch.h 內)

```
#define TESTING
```

(2) 在原物件中

```
#ifndef TESTING
#include "stdafx.h"
#else
#include "../UnitTests/fakeObject.h"
#endif // !TESTING
```

如此，如果不是在測試時(TESTING 被定義過)，原物件就會 include 到原本需要的東西

Step.6 寫一個測試吧，在工具列 > 測試 > 測試總管 內就可以簡單的執行測試並確認結果了喔

Google Test 語法 參考

https://github.com/google/googletest/blob/master/googletest/docs/Primer.md

PS

1.測試中別忘惹

```
using namespace game_framework;
```

2.如果覺得 setup 每次被執行很煩，可到方案屬性內的組態取消建置