

跨平台智慧手機開發工具－Cordova(1)

yehchitsai

Published
with GitBook



Table of Contents

| | |
|--------------|---|
| Introduction | 0 |
| 跨平台開發方式 | 1 |
| 安裝說明 | 2 |
| 使用說明 | 3 |
| Cordova 專案分析 | 4 |
| Cordova 範例 | 5 |
| 結論 | 6 |

前言

本課程主要是介紹智慧手持裝置的開發方法，目前市面上常見的智慧手持裝置的作業系統不外是Android，iOS與Windows Mobile，而各作業系統提供商所提供的開發工具則分別為Android Studio, Swift與Visual Studio。而本課程是提供一個可以使用網頁技術的開發方式，透過上述的開發工具來佈署在不同的智慧手持裝置，達到寫一次程式碼，執行在各式裝置(write once, run any devices)的目標。

修課須知

事先具備網頁開發技術如HTML, Javascript, jQuery等知識。

跨平台開發方式

目前市面有許多跨平台的開發方式，以下將逐一介紹

Unity

一個專業的遊戲開發引擎，並且開發效果最接近原生APP，能維持高性能。相對而言，也要付出較高的學習成本。可以各式的PC平台，手機平台甚至是PlayStation, Wii以及Xbox上執行。

參考資料 <http://unity3d.com/>

XAMARIN

2011年原本在Novell的Mono專案團隊，獨立成立一家公司命名為Xamarin，原本是在Linux上執行C#程式的一個開放原始碼專案，後來陸續發表支援iOS的Mono Touch framework以及Mono For Android Framework。可以使用C#撰寫出iOS, Android, Windows, Mac的原生程式碼。

參考資料 <http://xamarin.com/>

Apache Cordova

2009年在舊金山召開的iPhoneDevCamp大會，Nitobi的工程師Brock Whitten、Rob Ellis和Andre Charland提出了PhoneGap框架的概念：在iPhone，Android跟BlackBerry 系統內架設起Web介面和原生程式之間的橋樑，讓開發人員得以使用HTML5、JavaScript、CSS等web標準技術開發原生程式，實現一次編譯到處執行。Adobe在2011年10月4日收購了Nitobi軟體，將PhoneGap原始碼捐獻給Apache軟體基金會(Apache Software Foundation; ASF)，ASF重新命名為Apache Callstack，1.4版發布後，接著Apache Callstack的名稱變更為Apache Cordova，Cordova在2011年10月進入了Apache軟體基金會，並且在2012年10月正式成為Apache的頂級專案。

Cordova是一款開放原始碼的行動裝置開發框架，旨在讓開發者使用HTML、Javascript、CSS等Web APIs開發跨平臺的行動裝置應用程式。也可以利用JavaScript存取這些介面可以使用諸如攝影機、羅盤等硬體系統資源。Cordova之所以能讓網頁技術在原生環境中執行，其中主要兩個關鍵點為：

- 運用WebView元件，整合成瀏覽器環境

- 將一系列的相關資源都包附在網站應用程式資料夾中

因為現在市面上的手機有著各式各樣的作業系統，所以我們透過Cordova來幫我們完成跨平台的效果，並減少使用者因為使用平台不同而造成的問題。圖 1可以看出Cordova製作出的應用程式與原生的應用程式之間的差異。原生的應用程式是透過原生程式碼（如：Android與Java、iOS與Objective-C）去呼叫裝置的相關資源（如：相機、電池狀況、設備狀況等等）。但是Cordova不需要去撰寫原生的程式碼，只需要透過HTML、Javascript、CSS就能夠達到與原生程式碼相同的功能。

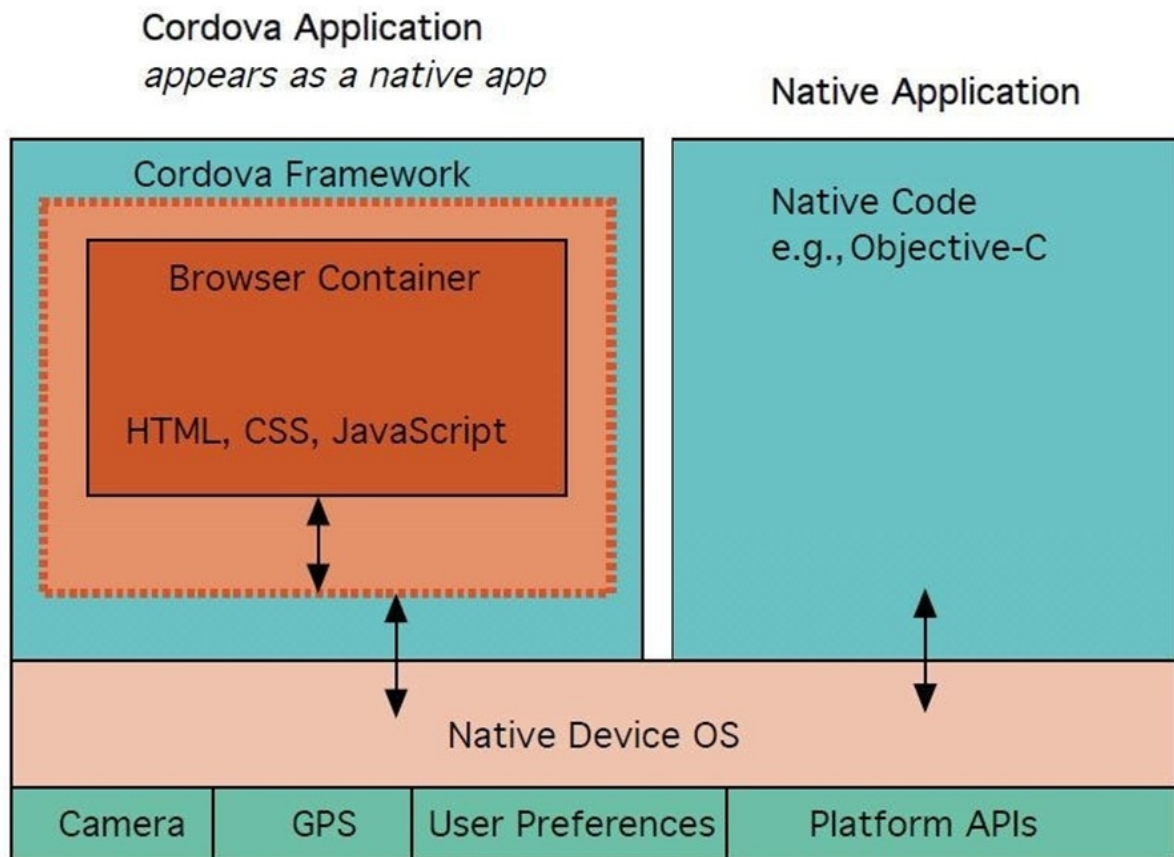


圖 1. Cordova結構圖

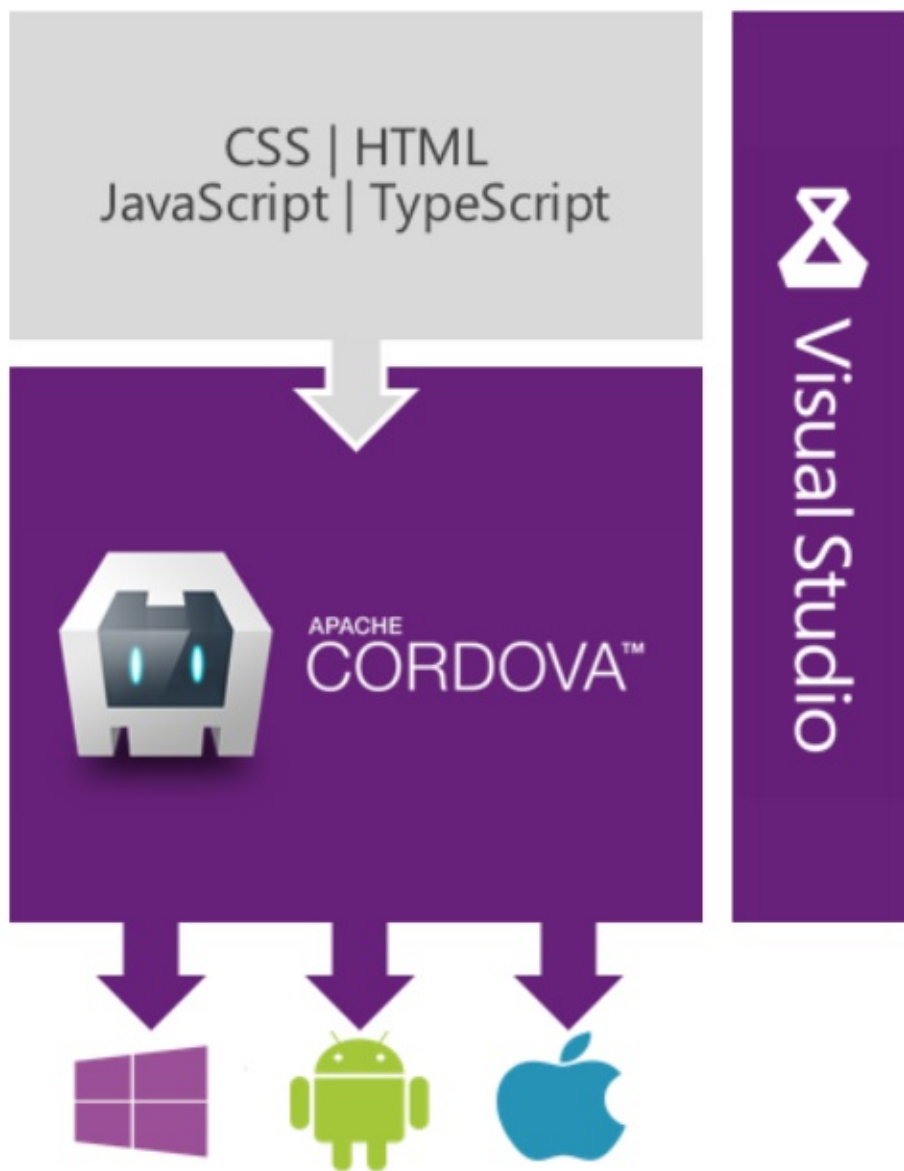


圖 1-1. Visual Studio結合Cordova

合Cordova

參考資料

<http://adon988.logdown.com/posts/249660-visual-studio-2015-and-apache-cordova>

<https://cordova.apache.org/>

安裝說明

1. 安裝的相關套件

要安裝Apache Cordova之前須安裝的工具程式

- Nodejs - v0.10.29 – 用來安裝cordova的套件管理程式
- Git client – 下載、更新位於github上的cordova plugin套件
- Apache ant - 將編譯、測試、部署等步驟自動化的一個工具，可以用來自動化生成iOS, Android等套件
- ADT Bundle – 編譯Android apk套件*

2. 開啟cmd並輸入以下指令，如圖2所示

```
$npm install -g cordova
```

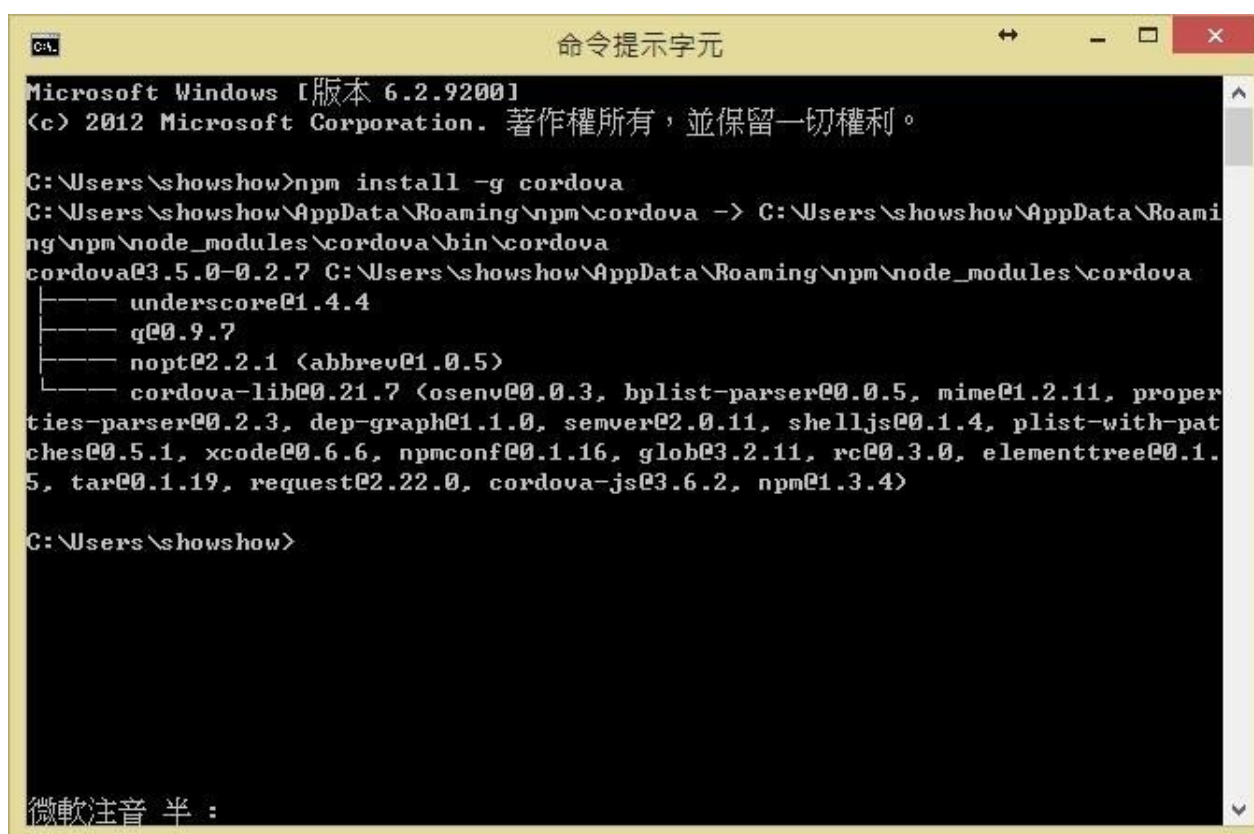


圖2. 使用npm(node package manager)安裝cordova

3. 設定環境變數

path=C:\ProgramFiles\nodejs\; C:\Users\master\AppData\Roaming\npm; D:\apache-ant\bin;
D:\adt-bundle-windows-x86_64-20140702\sdk\tools; D:\adt-bundle-windows-x86_64-
20140702\sdk\platform-tools; C:\Program Files\Java\jdk1.8.0_05\bin; C:\Program Files

(x86)\Git\bin;

下載jdk-><http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

新增 變數名稱：JAVA_HOME

變數值：C:\Program Files\Java\jdk1.8.0_05

4.重開cmd，產生一個新的Cordova App專案，如圖3所示

```
$cordova create gcm com.example.career HelloWorld
```

```
C:\Users\showshow>cordova create gcm com.example.hello HelloWorld
Creating a new cordova project with name "HelloWorld" and id "com.example.hello"
at location "C:\Users\showshow\gcm"
Downloading cordova library for www...
Download complete
```

圖3. 產生一個新的CordovaApp專案，專案名稱為HelloWorld，id為com.example.career，專案目錄為gcm

5.至ADT Bundle安裝目錄執行SDK Manager.exe，如圖4所示

1、安裝對應的Android SDK，如Android 2.3.3(API10)、Android 4.4.2(API19)。

2、安裝Extra套件



圖4. 執行SDK Manager.exe

6.切換到新app的工作目錄下

```
$ cd gcm
```

7.產生對應的執行平台


```
# 移除既有平台
$ cordova platform remove wp8
$ cordova platform remove windows8
$ cordova platform remove android
# 新增平台
$ cordova platform add wp8
$ cordova platform add windows8
$ cordova platform add android
```

8.新增外掛模組及建立app

```
$ cordova plugin add https://github.com/phonegap-build/PushPlugin.git
$ cordova plugin add org.apache.cordova.device
$ cordova plugin add org.apache.cordova.media #optional
# 建立Android app
$ cordova build android
```

9.建立android模擬器，如圖5所示

```
$ android avd
```

- 1、AVD Name : 自訂模擬器名稱
- 2、Device : 使用的裝置 (測試中使用了Nexus5)
- 3、Target : 平台(測試中使用了Android2.3.3(API Level10))
- 4、CPU : ARM(armeabi)
- 5、skin : (測試中使用了HVGA)
- 6、Memory Options : RAM(768)

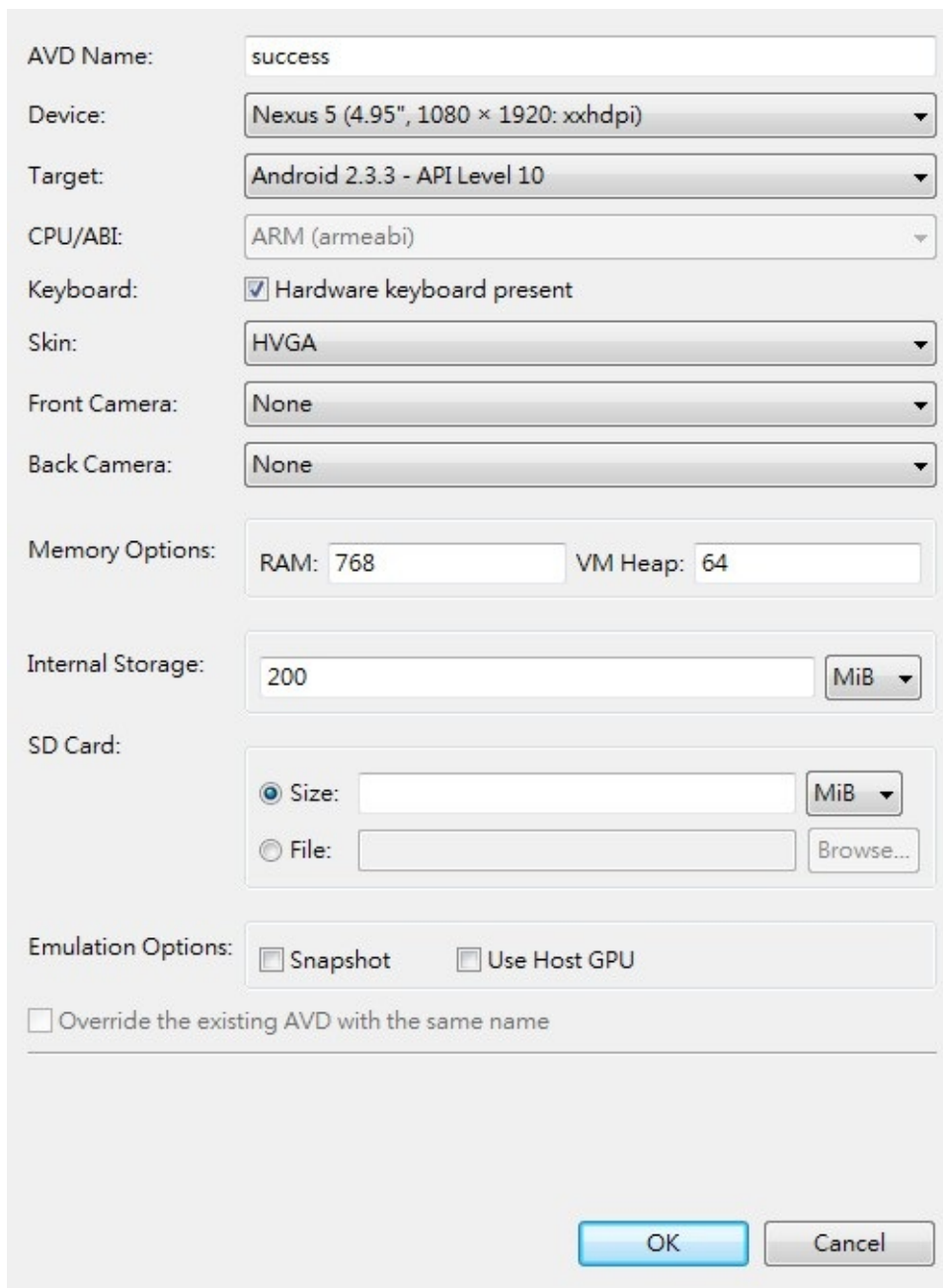


圖5. android模擬器

10. 啟動模擬器

點選剛剛建立好的模擬器 *start* (需要等一段時間才會跑好)

11. 將建立好的app上傳到模擬器執行

開一個新的cmd

```
$ cd gcm
$ cordova emulate android
```

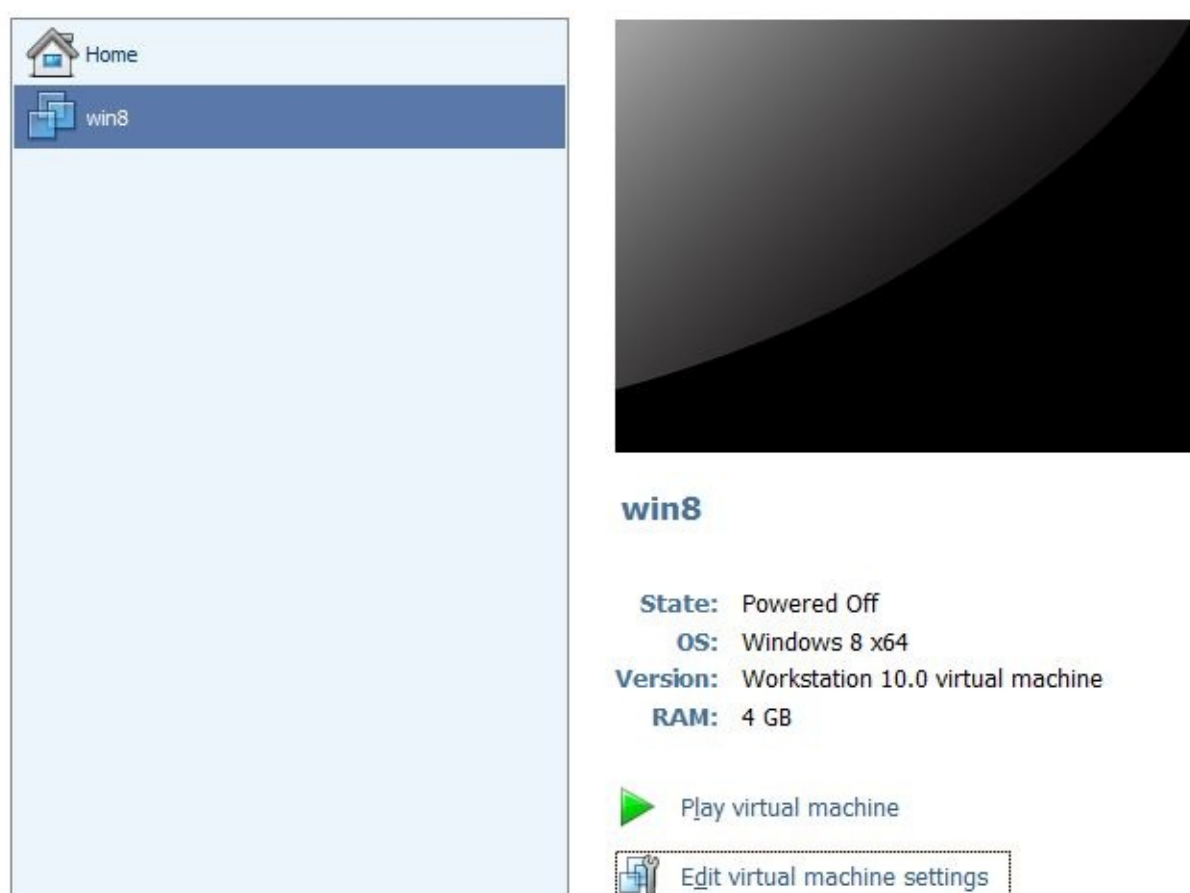
模擬器中的功能頁就會新增app 並且自動執行該app

使用說明

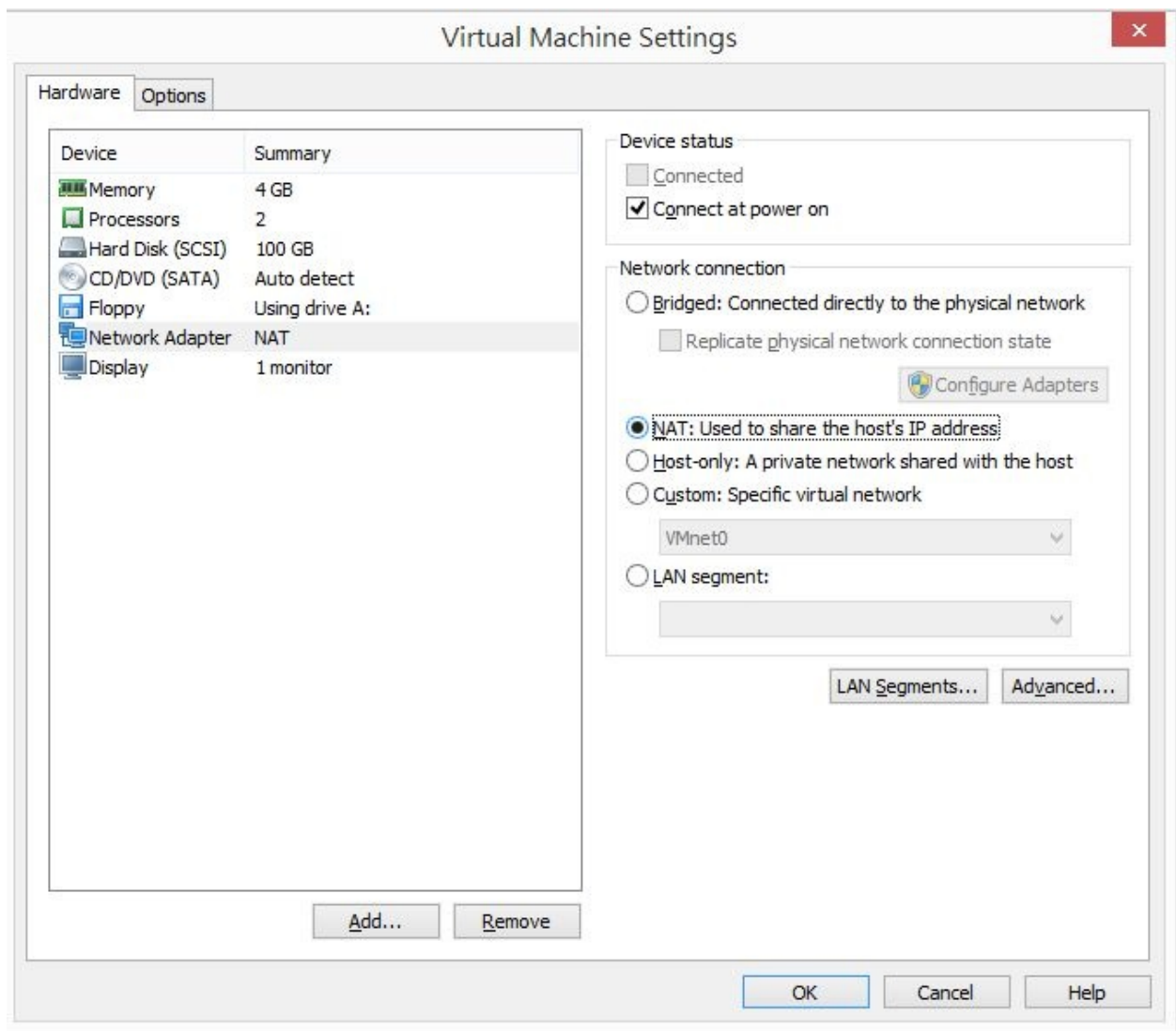
建立一個新專案

考慮cordova安裝耗時，所以授課以VMware虛擬機來進行實際演練，提供win8.vmx與win8-disk1.vmdk等虛擬機裝置，安裝vmware player後，直接選擇win8.vmx組態檔執行，因虛擬機原先設定問題，請先修改虛擬機設定與客端主機設定。

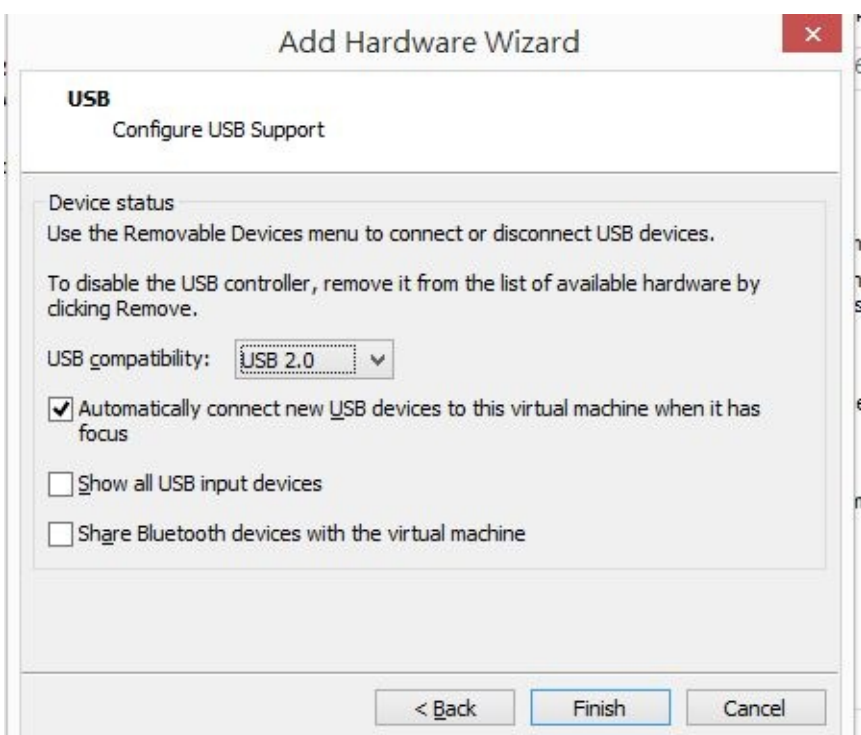
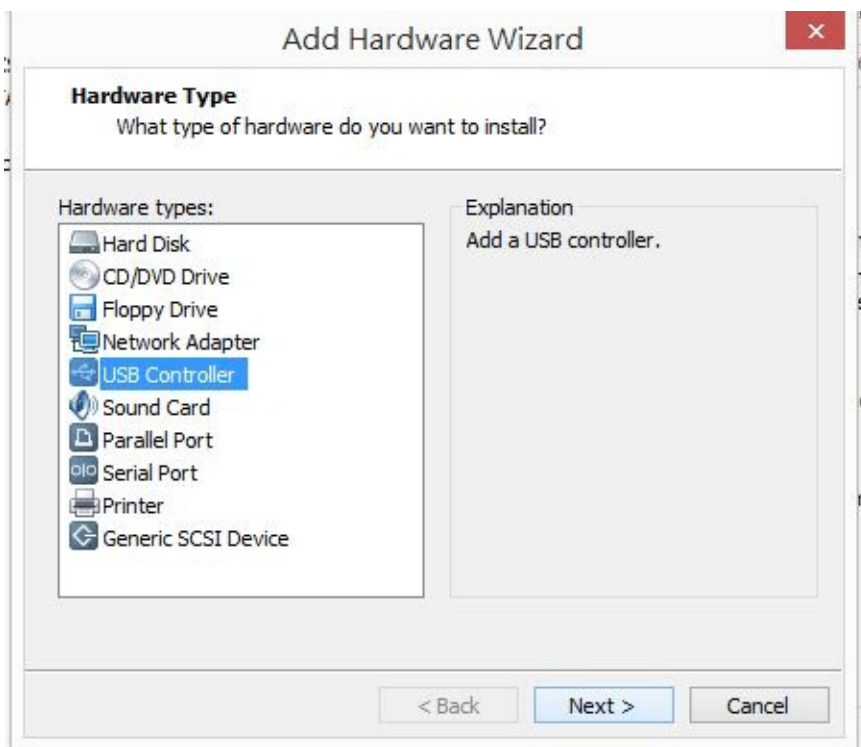
1.選擇win8後，點擊Edit virtual machine settings



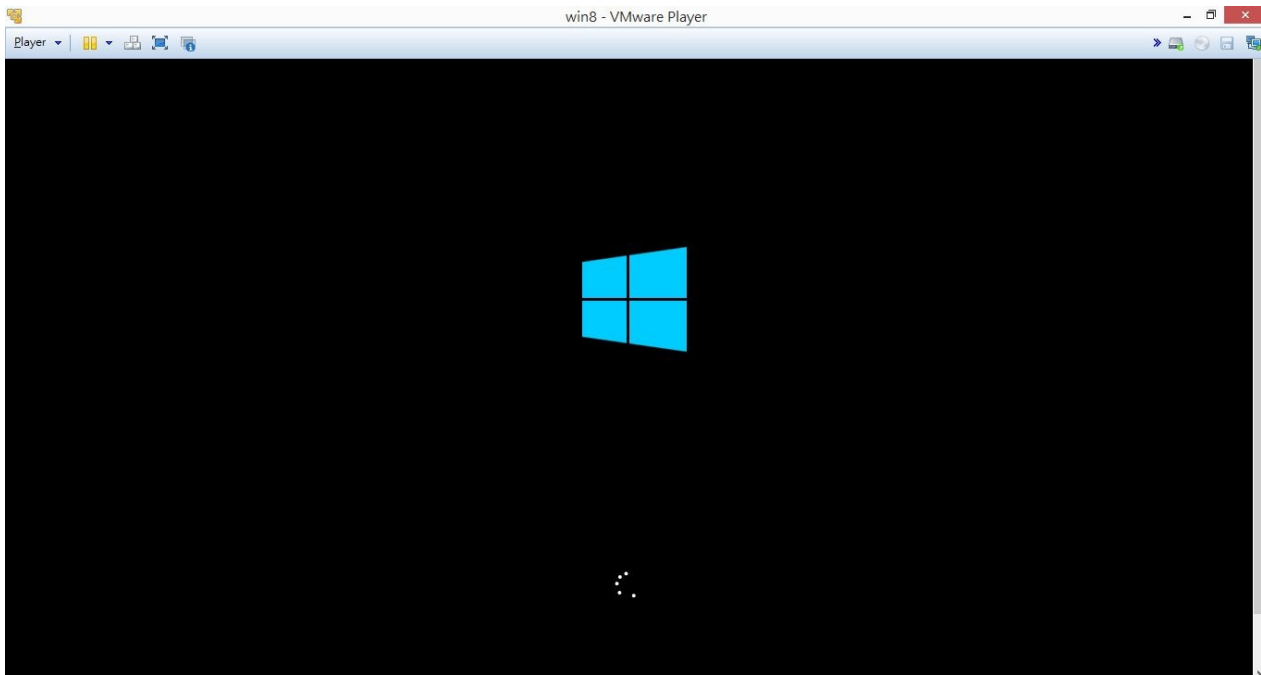
2.設定網路連線，讓虛擬機(guest OS)可以透過主機(host OS)連上網路：在Hardware頁籤中選擇Network Adapter，在右手邊的選項請選擇NAT: Used to share the hosts's IP address



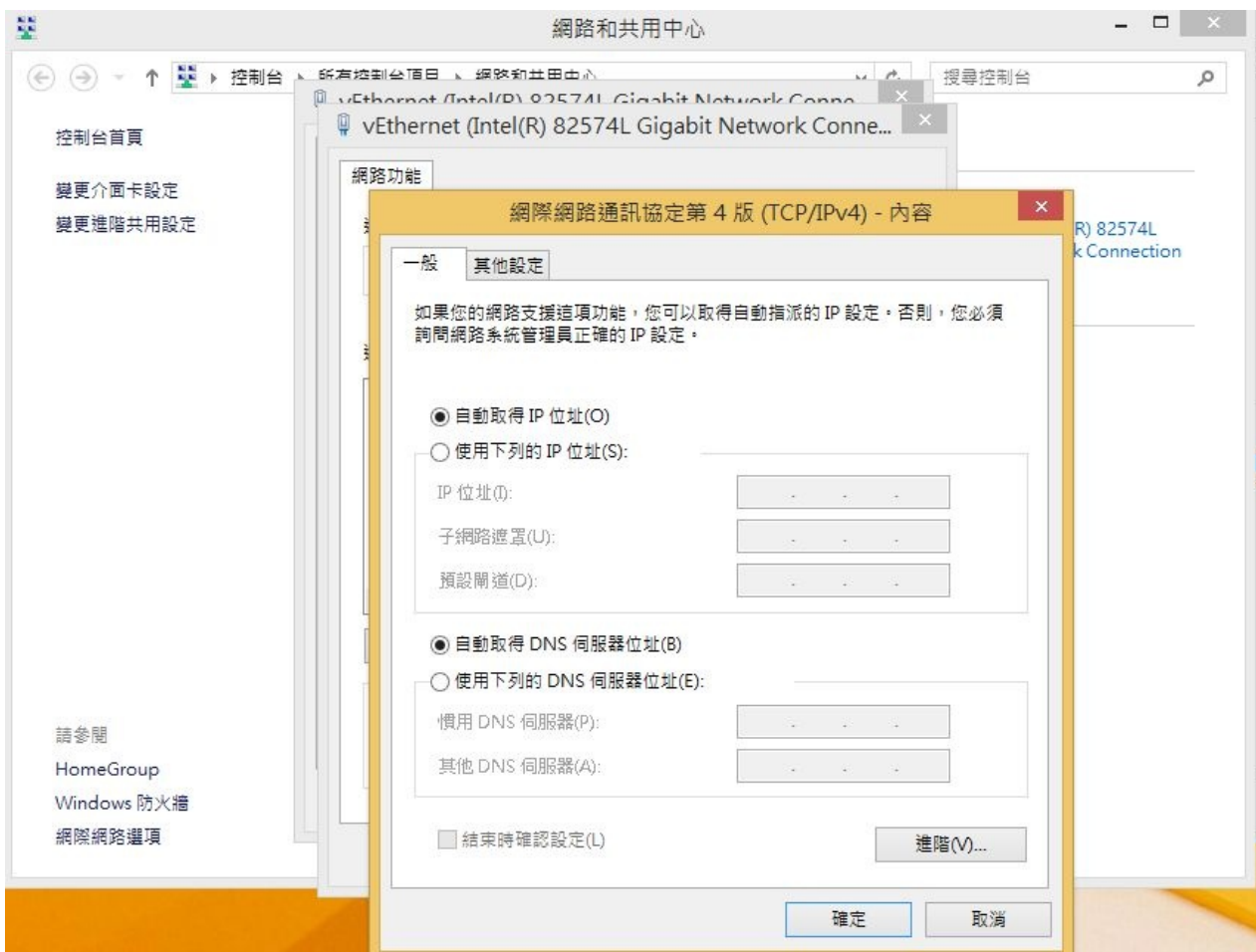
3.新增USB裝置，讓虛擬機用來存取手機裝置：在Hardware頁籤中選擇下方的Add...按鈕，跳出Add Hardware Wizard對話視窗，請選擇*USB Controller*後按*Next*，下一視窗若無需異動請按下*Finish*完成新增。



4. 按下 Play virtual machine 來啟動虛擬機

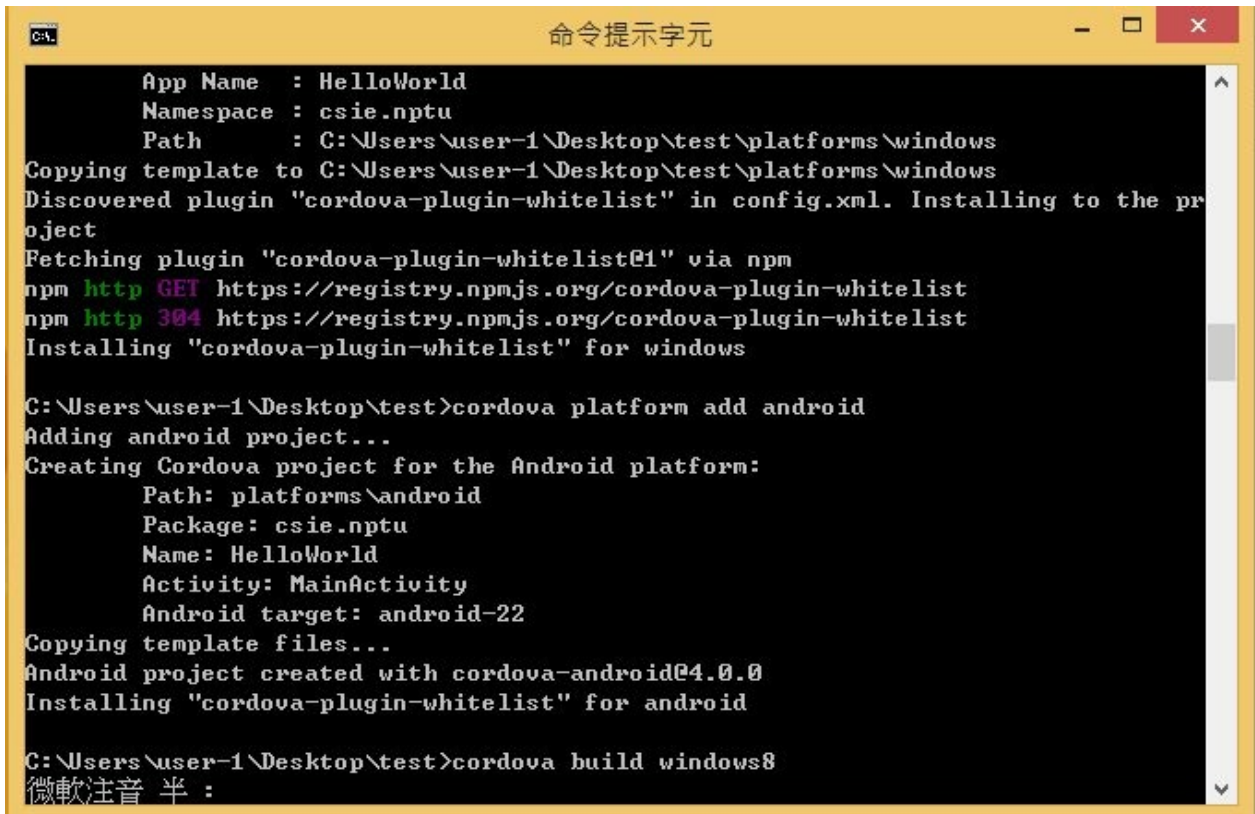


5.設定虛擬機內的網路：請將客端作業系統的網路改成自動取得IP位址與自動取得DNS伺服器位址，確定後開啟Chrome瀏覽器確認網路是否可用。



6.啟動後新增一個Cordova專案：開啟下方工作列的命令提示字元，輸入以下指令：

```
cd Desktop
cordova create test csie.nptu HelloWorld
cd test
# 新增windows 8與Android平台
cordova platform add windows8
cordova platform add android
# 建立windows 8 app
cordova build android
```



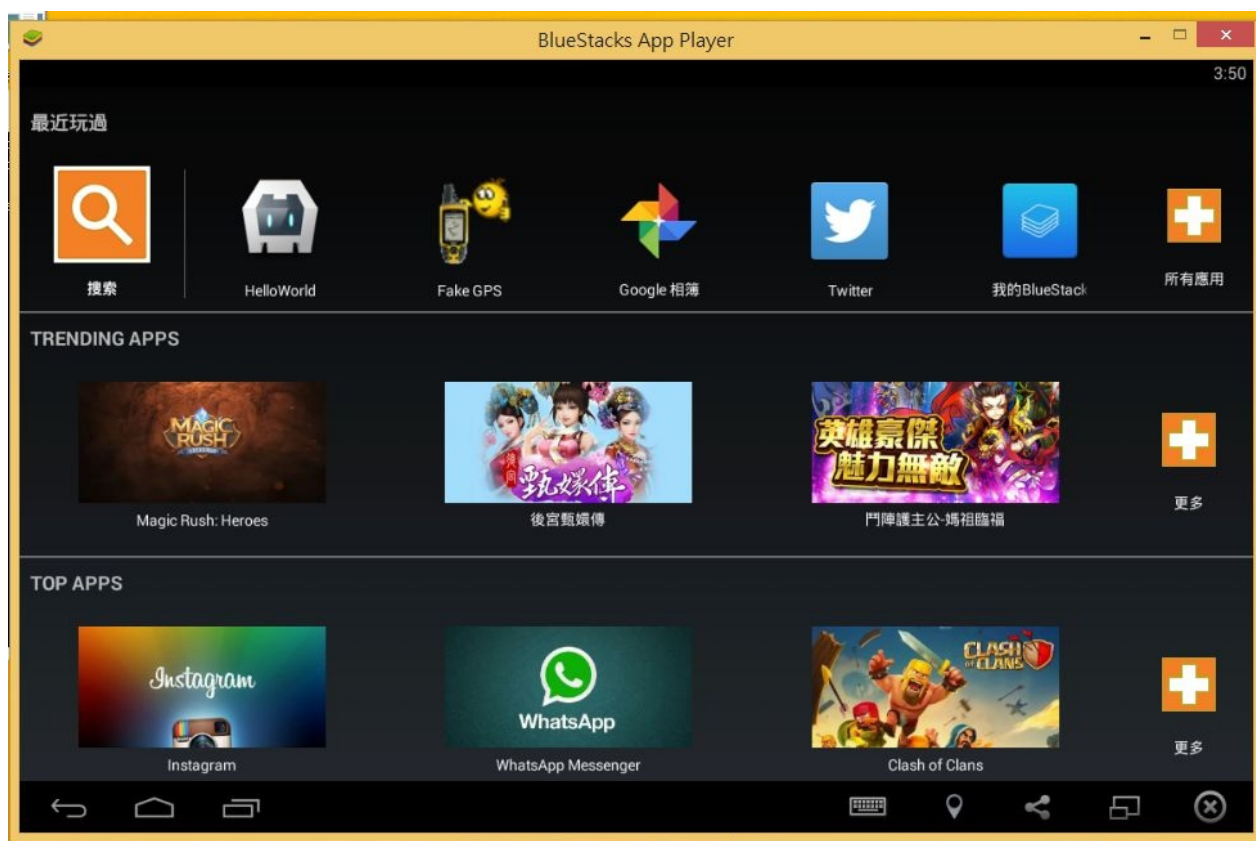
```
App Name : HelloWorld
Namespace : csie.nptu
Path : C:\Users\user-1\Desktop\test\platforms\windows
Copying template to C:\Users\user-1\Desktop\test\platforms\windows
Discovered plugin "cordova-plugin-whitelist" in config.xml. Installing to the project
Fetching plugin "cordova-plugin-whitelist@1" via npm
npm http GET https://registry.npmjs.org/cordova-plugin-whitelist
npm http 304 https://registry.npmjs.org/cordova-plugin-whitelist
Installing "cordova-plugin-whitelist" for windows

C:\Users\user-1\Desktop\test>cordova platform add android
Adding android project...
Creating Cordova project for the Android platform:
  Path: platforms\android
  Package: csie.nptu
  Name: HelloWorld
  Activity: MainActivity
  Android target: android-22
Copying template files...
Android project created with cordova-android@4.0.0
Installing "cordova-plugin-whitelist" for android

C:\Users\user-1\Desktop\test>cordova build windows8
微軟注音 半 :
```

佈署到Android手機

執行Android app(apk)的方式有兩種，一個是在模擬器，另一個則是直接佈署在手機上測試，本次主要介紹如何佈署到主機測試，模擬器可參考[這裡](#)



安裝BlueStacks，並將HelloWorld在上面執行

1.開啟手機測試模式：一般來說，Android手機僅能接受來自Google play市集的安裝，所以要開啟手機的測試模式。進入設定選擇安全性，勾選不明的來源來允許手機安裝不明來源的apk檔；接著跳回設定主選單，選擇開發人員選項，勾選USB偵錯好允許cordova將apk檔透過USB傳送到手機。



選擇安全性

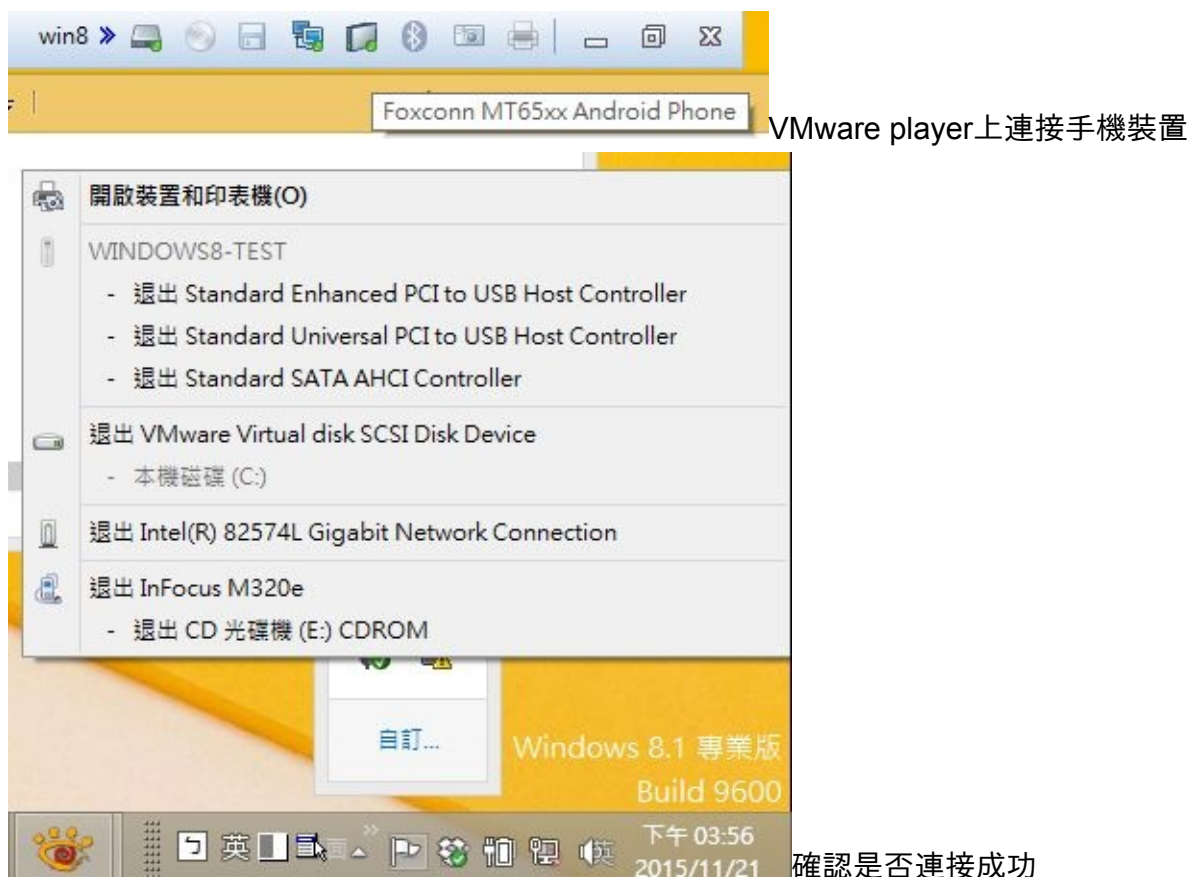


勾選不明的來源



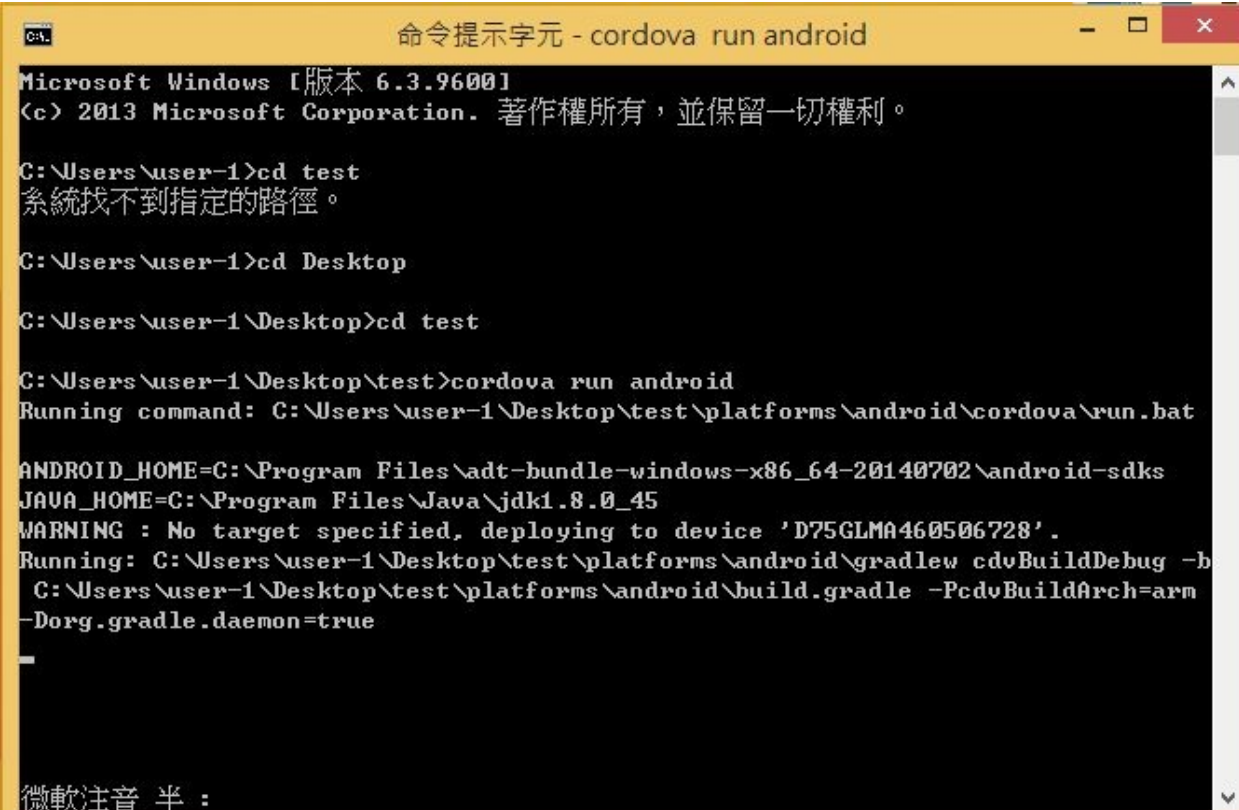
勾選USB偵錯

2.將手機透過USB連線至電腦，在VMware player上連接手機裝置，並確認是否連接成功



3.接著在命令列上輸入，應該會看到手機裝置編號，並且成功在手機上執行

```
cordova run android
```



```
Microsoft Windows [版本 6.3.9600]
(c) 2013 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\user-1>cd test
系統找不到指定的路徑。

C:\Users\user-1>cd Desktop

C:\Users\user-1\Desktop>cd test

C:\Users\user-1\Desktop\test>cordova run android
Running command: C:\Users\user-1\Desktop\test\platforms\android\cordova\run.bat

ANDROID_HOME=C:\Program Files\adt-bundle-windows-x86_64-20140702\android-sdk
JAVA_HOME=C:\Program Files\Java\jdk1.8.0_45
WARNING : No target specified, deploying to device 'D75GLMA460506728'.
Running: C:\Users\user-1\Desktop\test\platforms\android\gradlew cdvBuildDebug -b
C:\Users\user-1\Desktop\test\platforms\android\build.gradle -PcdvBuildArch=arm
-Dorg.gradle.daemon=true

微軟注音 半：
```

看到手機裝置編號



成功在手機上執行

Cordova 專案分析

接下來介紹一個Cordova專案的細部說明

目錄說明

建好一個Cordova專案，基本上包含以下四個目錄及一個檔案

- hooks/：提供編譯時的一些動作設定，比方說after_build時加入一些動作。
- platforms/：建立的平台檔案會放在這個目錄，如android, ios等。
- plugins/：放置外掛檔案的目錄。
- www/：主要網頁設計檔案的目錄。
- config.xml：組態設定檔，可根據不同的平台來進行個別的設定。

第一個專屬APP

修改config.xml來建立一個有自己名稱跟圖示的Android App，修改如下

```
<name>HelloWorld</name>
```

改成

```
<name>我的第一個APP</name>
```

接下來針對Android平台，修改APP圖示，找到以下文字

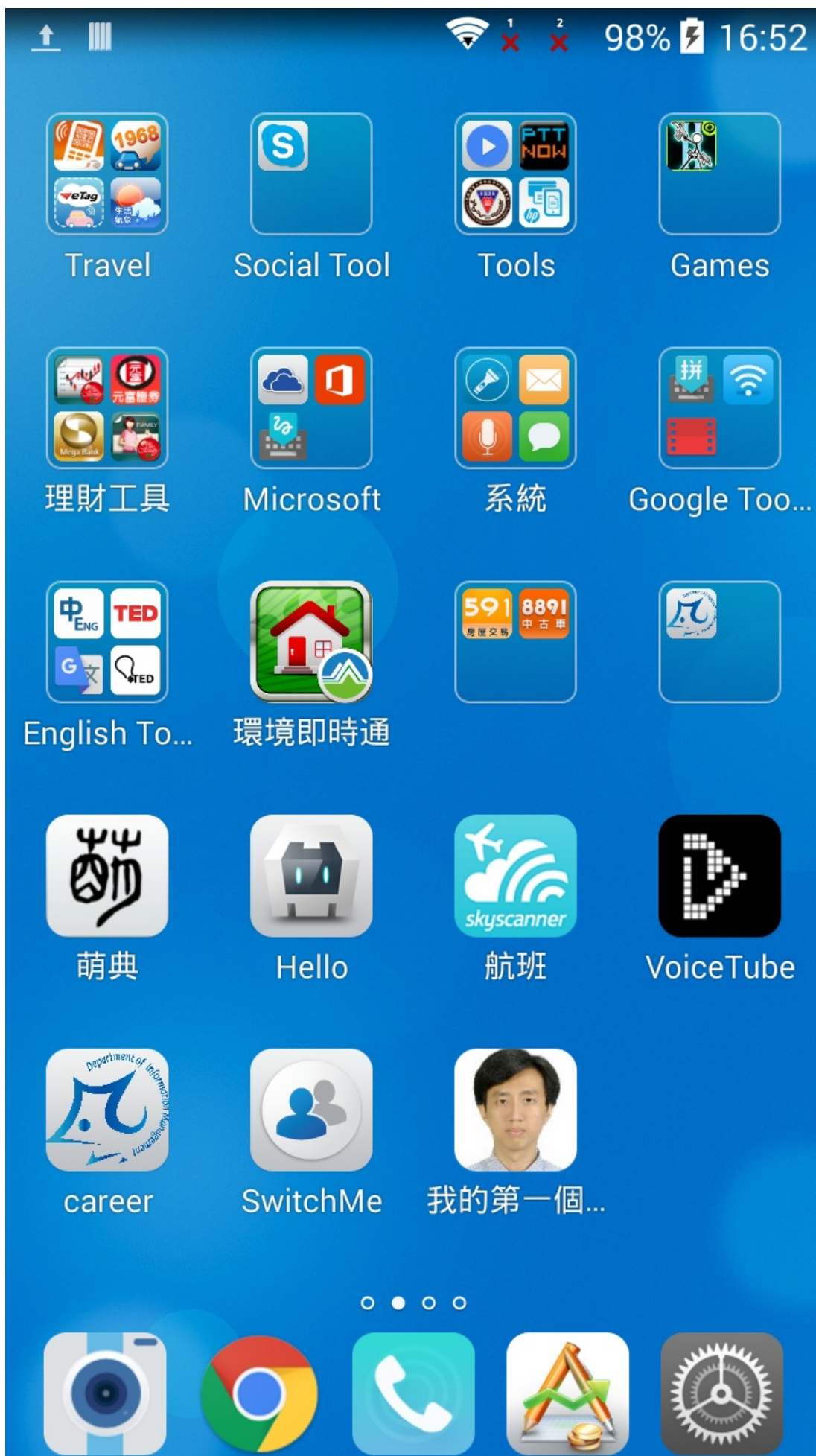
```
<platform name="android">  
  ...  
</platform>
```

修改成針對不同解析度的Android手機提供不同圖示，目錄的起始點為config.xml(注意：圖示僅能為png格式)

```
<platform name="android">
  <icon src="res/android/android.png" density="ldpi" />
  <icon src="res/android/android.png" density="mdpi" />
  <icon src="res/android/android.png" density="hdpi" />
  <icon src="res/android/android.png" density="xhdpi" />
</platform>
```

修改完畢後，在命令提示字元中輸入，即可在手機桌面上看到該圖示。

```
cordova run android
```

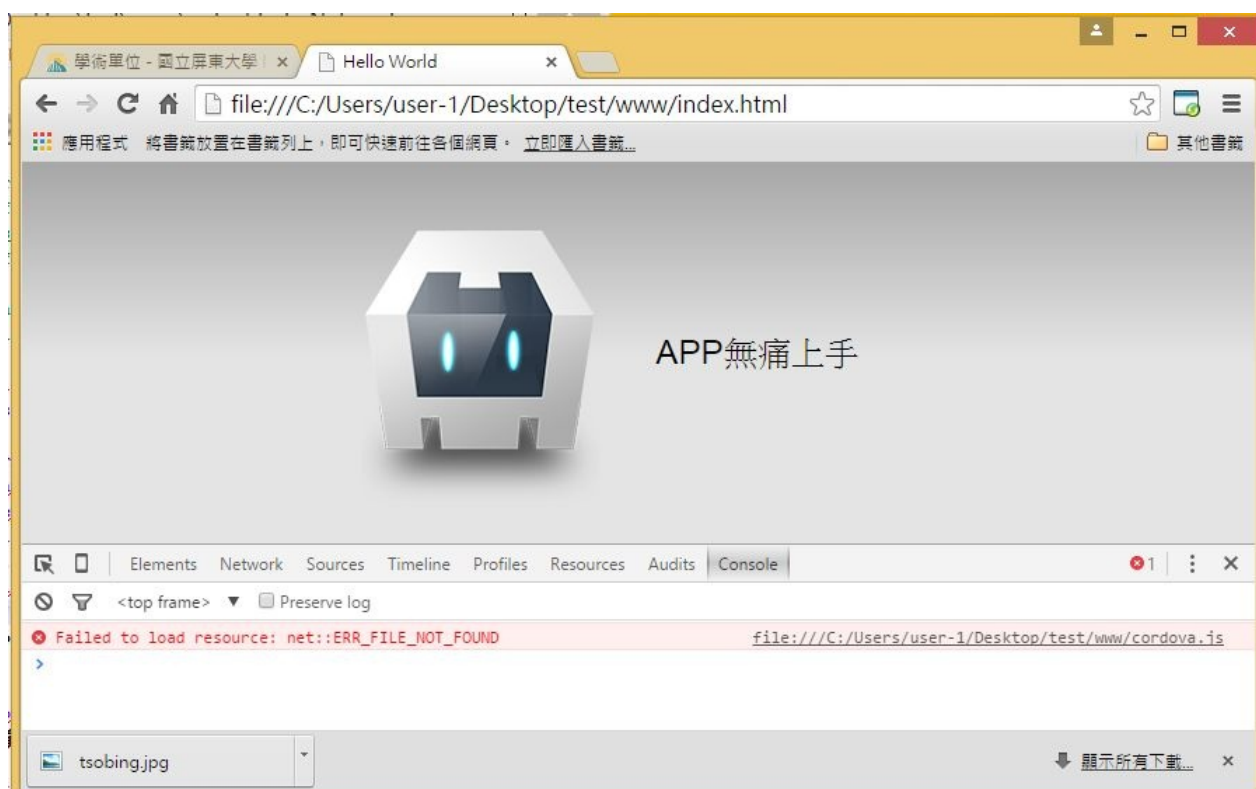


Cordova 範例

以下將提供幾個範例以供練習

修改範本範例

第一個範例以修改Cordova所提供範本，並在PC的瀏覽器下進行除錯後上傳。



第二個範例為將既有的網頁程式轉成手機程式。



第三個範例則是使用手機的GPS，Cordova抓取GPS的資訊顯示在Google Maps上，事前準備為：

- [Google Maps Android API的憑證](#)
- [Google Maps plugin for Cordova](#)

安裝方法可參考Google Maps plugin for Cordova所提供的[Tutorial for Windows](#)，這種與裝置相關的程式無法在windows的瀏覽器下進行除錯。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="format-detection" content="telephone=no" />
    <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1
    <link rel="stylesheet" type="text/css" href="css/index.css" />
    <meta name="msapplication-tap-highlight" content="no" />
    <title>Hello World</title>
    <script type="text/javascript" src="cordova.js"></script>
    <script type="text/javascript">
      var map;
```

```
document.addEventListener("deviceready", function() {
    var div = document.getElementById("map_canvas");

    // Initialize the map view
    map = plugin.google.maps.Map.getMap(div);

    // Wait until the map is ready status.
    map.addEventListener(plugin.google.maps.event.MAP_READY, onMapReady);
}, false);

function onMapReady() {
    var button = document.getElementById("button");
    document.getElementById('map_canvas2').innerHTML += 'onMapReady';

    map.getMyLocation(function(location) {
        var msg = ["Current your location:\n",
            "latitude:" + location.latLng.lat,
            "longitude:" + location.latLng.lng,
            "speed:" + location.speed,
            "time:" + location.time,
            "bearing:" + location.bearing].join("\n");

        map.addMarker({
            'position': location.latLng,
            'title': msg,
        }, function(marker) {
            marker.showInfoWindow();
        });
        map.setCenter(location.latLng);
        map.setZoom(16);
        document.getElementById('map_canvas2').innerHTML += 'getMyLocation';
    });

    button.addEventListener("click", onBtnClicked, false);
    document.getElementById('map_canvas2').innerHTML += 'onBtnClicked';
}

function onBtnClicked() {
    map.showDialog();
}
</script>
</head>
<body>
<h3>PhoneGap-GoogleMaps-Plugin</h3>
<div style="width:100%;height:400px" id="map_canvas"></div>
<button id="button">Full Screen</button>
<div id="map_canvas2"></div>
</body>
</html>
```




結論

早期的應用程式也曾是多方開發程式爭鳴的年代，後來WEB出來後，幾乎佔了80%左右的市場，相信在手機、平板、電視等多項終端裝置的開發程式市場，WEB開發終將還是會佔有一席重要的地位。