**Gaminfinity SDK**

**Programming Guide**

V1.1.0.0

*for*

Android SDK v1.0.0.0

iOS SDK v1.0.0.0

文件修訂紀錄

|  |  |  |
| --- | --- | --- |
| 版號 | 日期 | 說明 |
| V1.0.0.0 | 2014-08-08 | 第一版 |
| V1.1.0.0 | 2014-08-14 | 增加iOS SDK以及相關說明 |

**目錄**

[1. 介紹 4](#_Toc395710610)

[2. 說明 4](#_Toc395710611)

[3. Gaminfinity SDK方法 6](#_Toc395710612)

[3.1 getAccountId 6](#_Toc395710613)

[4. Gaminfinity SDK時序圖 8](#_Toc395710614)

[4.1直接試玩的時序圖 8](#_Toc395710615)

[4.2 Facebook帳號登入的時序圖 9](#_Toc395710616)

[4.3 測試 SDK的時序圖 9](#_Toc395710617)

[5. Android使用範例 10](#_Toc395710618)

[6. iOS使用範例 13](#_Toc395710619)

[7. 相關檔案下載網址 16](#_Toc395710620)

[7.1 Gaminfinity SDK for Android 16](#_Toc395710621)

[7.2 Gaminfinity SDK Demo for Android 16](#_Toc395710622)

[7.3 Gaminfinity SDK for iOS 16](#_Toc395710623)

[7.4 Gaminfinity SDK Demo for iOS 16](#_Toc395710624)

[7.5 Facebook Login範例 for Android 16](#_Toc395710625)

[7.6 Facebook SDK for Android下載網址 16](#_Toc395710626)

[7.7 Facebook SDK for iOS下載網址 17](#_Toc395710627)

[8. 各環境的設定值 18](#_Toc395710628)

[8.1 單機環境 18](#_Toc395710629)

[8.2 測試環境 18](#_Toc395710630)

[8.3 Live環境 18](#_Toc395710631)

[9. 附錄 19](#_Toc395710632)

[9.1 錯誤代碼定義 19](#_Toc395710633)

[9.2 Facebook Login範例 20](#_Toc395710634)

[9.3 問題排除 28](#_Toc395710635)

[9.4 登入頁面實作參考 29](#_Toc395710636)

# 介紹

本文件說明Gaminfinity SDK，如何提供遊戲取得裝置的唯一識別碼的呼叫方法，以作為遊戲立即試玩的帳號，Gaminfinity SDK並提供Facebook帳號綁定裝置的呼叫方法，以實現玩家使用Facebook帳號登入後，與裝置綁定的需求。

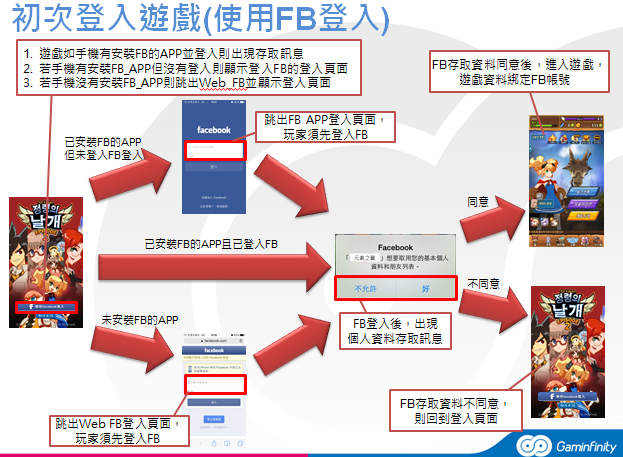
# 說明

Gaminfinity SDK提供取得登入遊戲的帳號識別碼的服務

Gaminfinity SDK所提供的API數量：1個。

Game開發商需要準備的工作

1. 使用Gaminfinity SDK所提供的API，使用時機與使用方法請參考後續章節。
2. 整合Facebook SDK。
3. 按照FB帳號登入規格實作相關程式碼。只要依照Facebook SDK規範，預設功能即能達到需求：(a)當玩家沒有安裝Facebook App時，登入Facebook，則使用Facebook網頁登入介面；(b)當玩家有安裝Facebook App時，登入Facebook，則使用Facebook App登入介面。



Facebook Login 功能，請參考Facebook官方網站說明https://developers.facebook.com/docs/facebook-login/v2.0

# Gaminfinity SDK方法

適用於Android和iOS。

提醒：本章節在語法上使用pseudo方式說明，針對 Android與iOS會有獨立章節的使用說明

## 3.1 getAccountId

**目的**

向Gaminfinity SDK提出請求，取得玩家的登入識別碼。

**輸入參數**

String accessToken //該玩家登入Facebook後， 有效的access token

String url //聯繫Gaminfinity Server的URL

其中

accessToken 可以為NULL或有效的Facebook access token字串

url 可以為NULL或各環境的Gaminfinity Server URL

**使用時機**

當玩家選擇【立即試玩】或【Facebook帳號登入】時呼叫本方法。

下表為帶入不同的參數，會得到不同的結果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | 使用時機 | accessToken | url | 說明 |
| 1 | 立即試玩 | NULL | NULL | 本機端測試模式，取得的AccountID = A123456789 |
| 2 | 有效的URL | 取得玩家登入識別碼 |
| 3 | 使用Facebook  帳號登入 | access token | NULL | 本機端，取得的AccountID = A123456789 |
| 4 | access token | 有效的URL | 取得玩家登入識別碼 |

註：當使用本機端測試模式時，在debug log會看到 "stand-alone mode is ON"，否則會看到"stand-alone mode is OFF"

使用本API時，需要實作接收結果的method

規格為 void onGetAccountId(int result, String accountId)

**輸出參數**

int result

當result = 1  時，代表綁定成功。

當result = -n 時，代表有錯誤發生。錯誤代碼定義請見【附錄錯誤代碼章節】

String accountId //登入遊戲所使用的玩家識別碼

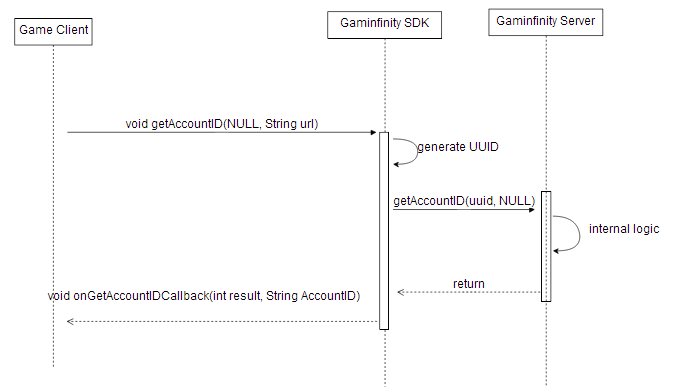
# Gaminfinity SDK時序圖

以下為時序圖各階段的說明

* Game Client：使用Gaminfinity SDK的遊戲客戶端程式
* Facebook SDK：由 Facebook提供的SDK
* Gaminfinity SDK：由樂無極提供的SDK
* Gaminfinity Server：由樂無極提供的Web Service
* Facebook：泛指Facebook所屬的網站

## 4.1直接試玩的時序圖

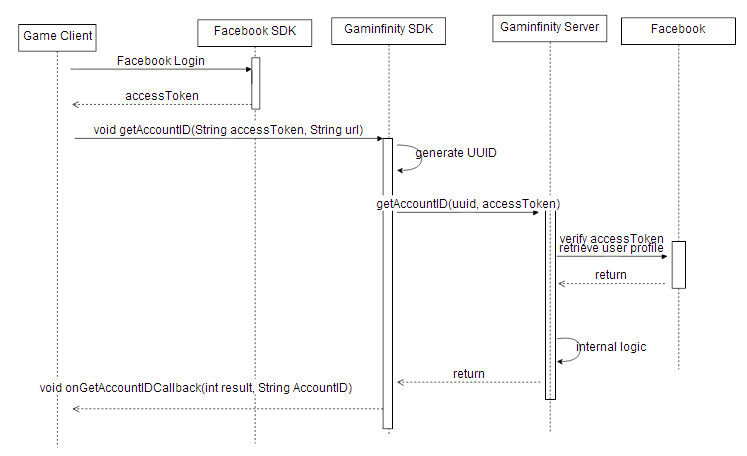
當玩家按下【直接試玩】的按鈕



時序圖1

## 4.2 Facebook帳號登入的時序圖

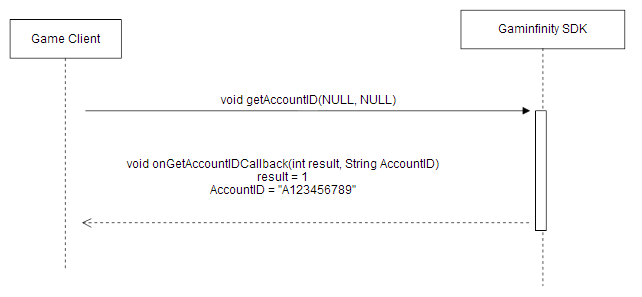
當玩家按下【Facebook帳號登入】的按鈕



時序圖2

## 4.3 測試 SDK的時序圖

單機測試用

  
時序圖3

# Android使用範例

本範例使用的開發工具，SDK資訊如下

* 開發工具：Android SDK ADT: adt-bundle-windows-x86-20140624
* SDK
  + Facebook SDK for Android: v3.15.0
  + Gaminfinity SDK for Android

匯入Gaminfinity SDK到遊戲專案

Gaminfinity SDK名稱：GaminfinitySDK.jar

匯入Facebook SDK到遊戲專案

請參考附錄Android Facebook Login範例。

在呼叫Gaminfinity SDK提供的API之前，需實作下列事項

1. 使用Gaminfinity SDK的package
2. 繼承Gaminfinity SDK所定義的介面
3. 宣告
4. 建立instance。這裡必須傳入Activity的實體。除了將帳號回傳的用途之外，也需使用該 Activity來取得Android裝置的唯一識別碼。
5. 實作Gaminfinity SDK所定義介面

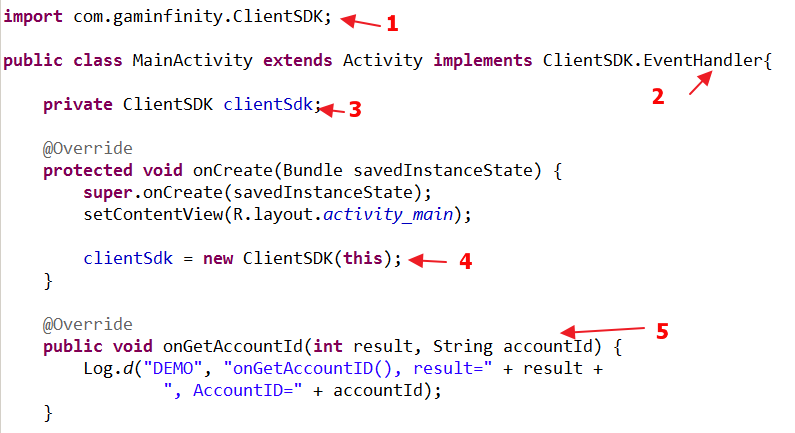


圖5-1

完成後，就可以開始呼叫Gaminfinity SDK提供的API

(1) 當玩家選擇【立即試玩】

clientSdk.getAccountId(null, "http://GaminfinityServerURL");

注意：

accessToken參數需傳入null

url參數請參考【各環境的設定值】章節

(2) 當玩家選擇【Facebook帳號登入】

clientSdk.getAccountId("Facebook accessToken", "http:// GaminfinityServerURL");

注意：

accessToken參數需傳入有效的Facebook Access Token

url參數請參考【各環境的設定值】章節

# iOS使用範例

本範例使用的開發工具，SDK資訊如下

* 開發工具：XCode 5.1.1
* SDK
  + Facebook SDK for iOS: v3.17.0
  + Gaminfinity SDK for iOS

匯入Gaminfinity SDK到遊戲專案

Gaminfinity SDK名稱：libGaminfinitySDK.a以及GaminfinitySDK.h

在呼叫Gaminfinity SDK提供的API之前，需實作下列事項

1. 使用Gaminfinity SDK include file

#import "GaminfinitySDK/GaminfinitySDK.h"

1. 繼承Gaminfinity SDK所定義的介面

@interface ViewController : UIViewController **<EventHandler>**

1. 宣告

GaminfinitySDK\* gaminfinitySdk;

1. 建立instance

gaminfinitySdk = [[GaminfinitySDK alloc]init];

gaminfinitySdk.delegate = self;

1. 實作Gaminfinity SDK所定義介面

- (void) onGetAccountId:(int)result AccountId:(NSString \*)accountId{

…

}

完成後，就可以開始呼叫Gaminfinity SDK提供的API

(1) 當玩家選擇【立即試玩】

[gaminfinitySdk getAccountId:nil ServerUrl: @"http://GaminfinityServerURL"];

注意：

accessToken參數需傳入nil

ServerUrl參數請參考【各環境的設定值】章節

(2) 當玩家選擇【Facebook帳號登入】

[gaminfinitySdk getAccountId:fbAccessToken ServerUrl: @"http:// GaminfinityServerURL"];

注意：

accessToken參數需傳入有效的Facebook Access Token

serverUrl參數請參考【各環境的設定值】章節

# 相關檔案下載網址

## 7.1 Gaminfinity SDK for Android

<https://github.com/stevechen0923/GaminfinitySDK/tree/master/Android/>

## 7.2 Gaminfinity SDK Demo for Android

https://github.com/stevechen0923/GaminfinitySdkDemo/tree/master/Android

## 7.3 Gaminfinity SDK for iOS

<https://github.com/stevechen0923/GaminfinitySDK/tree/master/iOS>

## 7.4 Gaminfinity SDK Demo for iOS

https://github.com/stevechen0923/GaminfinitySdkDemo/tree/master/iOS

## 7.5 Facebook Login範例 for Android

<https://github.com/stevechen0923/FacebookLoginTest/>

## 7.6 Facebook SDK for Android下載網址

<https://developers.facebook.com/docs/android>

## 7.7 Facebook SDK for iOS下載網址

<https://developers.facebook.com/docs/ios>

# 各環境的設定值

## 單機環境

無需與Gaminfinity Server聯繫，主要目的是方便遊戲開發，避免因為Gaminfinity Server因素，影響到遊戲登入。

注意：請勿使用此環境設定，與遊戲App正式上架。

Gaminfinity Server URL = NULL

## 8.2 測試環境

與Gaminfinity Server測試環境聯繫，需要有Internet。主要目的是讓開發更接近於實際環境。

注意：請勿使用此環境設定，與遊戲App正式上架。

Gaminfinity Server URL = http://219.87.94.74/sns/new\_bind\_uuid.php

## 8.3 Live環境

與Gaminfinity Server Live環境聯繫。

注意：遊戲App正式上架，需使用本環境設定。

Gaminfinity Server URL = <http://OOO>(尚未建置完成)

# 附錄

## 9.1 錯誤代碼定義

1 成功

以下從Gaminfinity Server所捕捉到的錯誤

-1 參數錯誤

-2 Facebook Access Token無效

-3 Facebook帳號綁定錯誤

以下從Gaminfinity SDK所捕捉到的錯誤

-100 一般內部錯誤(包含網路連結錯誤)

-101 網路逾時(timeout為5秒)

-102 網路傳輸強制被中斷

-103 網路傳遞的資料格式錯誤

## 9.2 Facebook Login範例

實作項目：

1. Custom Facebook Login Button
2. 取得Facebook access token
3. Custom Facebook Logout Button

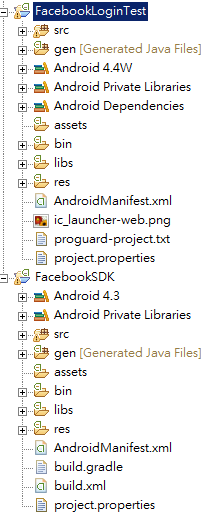
實作說明：

實作平台為Android，使用Facebook SDK，版本為3.15.0。詳細內容以及原理，請參考Facebook官方網站文件。

請先申請Facebook App ID，並產生適當的Key hashes。

專案檔案架構

FacebookLoginTest為本範例專案。FacebookSDK為Facebook SDK官方的專案檔案。



本範例只需要專注於FacebookLoginTest專案裡面的檔案，FacebookSDK則無需做任何修改。

在FacebookLoginTest專案裡面需加入FacebookSDK到專案的Library裡面。

**(1)在\res\values\strings.xml檔案裡面，加入您申請的facebook app id**

<string name="app\_id">716592228435883</string>

**(2)在AndroidManifest.xml檔案裡面，加入Internet存取權限，以及給facebbok使用的meta-data，以及Activity**

<uses-permission android:name="android.permission.INTERNET"/>

<meta-data android:name="com.facebook.sdk.ApplicationId" android:value="@string/app\_id" />

<activity android:name="com.facebook.LoginActivity"/>

**(3)在\res\layout\activity\_main.xml檔案裏面，加入２個Button**

<Button

android:id="@+id/buttonLogin"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:onClick="onLoginButtonClick"

android:text="login" />

<Button

android:id="@+id/buttonLogout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_below="@+id/buttonLogin"

android:onClick="onLogoutButtonClick"

android:text="logout" />

**(4)在MainActivity.java裡面加入程式碼**

**package** com.example.facebooklogintest;

**import** android.app.Activity;

**import** android.content.Intent;

**import** android.os.Bundle;

**import** android.view.Menu;

**import** android.view.MenuItem;

**import** android.view.View;

**import** android.widget.Toast;

**import** com.facebook.\*;

**public** **class** MainActivity **extends** Activity {

**private** Session.StatusCallback statusCallback = **new** SessionStatusCallback();

**private** **class** SessionStatusCallback **implements** Session.StatusCallback {

@Override

**public** **void** call(Session session, SessionState state, Exception exception) {

**if** (session.isOpened()) {

Toast.*makeText*(MainActivity.**this**, "AccessToken=" + session.getAccessToken(), Toast.*LENGTH\_SHORT*).show();

}

}

}

@Override

**public** **void** onStart() {

**super**.onStart();

Session.*getActiveSession*().addCallback(statusCallback);

}

@Override

**public** **void** onStop() {

**super**.onStop();

Session.*getActiveSession*().removeCallback(statusCallback);

}

@Override

**public** **void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {

**super**.onActivityResult(requestCode, resultCode, data);

Session.*getActiveSession*().onActivityResult(**this**, requestCode, resultCode, data);

}

@Override

**protected** **void** onSaveInstanceState(Bundle outState) {

**super**.onSaveInstanceState(outState);

Session session = Session.*getActiveSession*();

Session.*saveSession*(session, outState);

}

**public** **void** onLoginButtonClick(View v){

Session session = Session.*getActiveSession*();

**if** (!session.isOpened() && !session.isClosed()) {

session.openForRead(**new** Session.OpenRequest(**this**).setCallback(statusCallback));

} **else** {

Session.*openActiveSession*(**this**, **true**, statusCallback);

}

}

**public** **void** onLogoutButtonClick(View v){

Session session = Session.*getActiveSession*();

**if** (!session.isClosed()) {

session.closeAndClearTokenInformation();

}

}

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

Settings.*addLoggingBehavior*(LoggingBehavior.*INCLUDE\_ACCESS\_TOKENS*);

Session session = Session.*getActiveSession*();

**if** (session == **null**) {

**if** (savedInstanceState != **null**) {

session = Session.*restoreSession*(**this**, **null**, statusCallback, savedInstanceState);

}

**if** (session == **null**) {

session = **new** Session(**this**);

}

Session.*setActiveSession*(session);

**if** (session.getState().equals(SessionState.*CREATED\_TOKEN\_LOADED*)) {

session.openForRead(**new** Session.OpenRequest(**this**).setCallback(statusCallback));

}

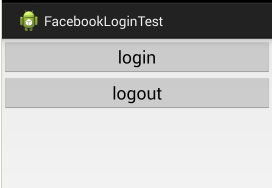
}

}

}

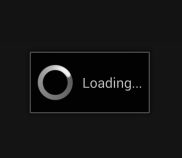
**(5)編譯完成後執行**

FacebookLoginTest程式畫面



按下login button後

會出現loading畫面，此loading畫面由Facebook SDK自行顯示



如果玩家沒有安裝Facebook App，則會顯示Facebook登入網頁



在Facebook網頁登入畫面中，點選下圖紅色框部分，會自動轉到Google商店，讓玩家下載Facebook App

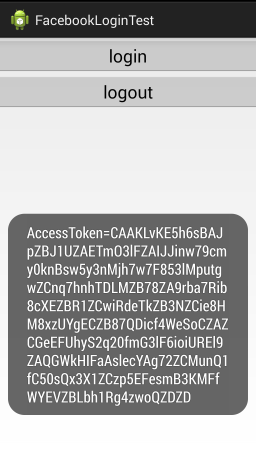




如果玩家安裝Facebook App後，執行FacebookLoginTest的login功能，就會顯示Facebook App的登入畫面。



執行登入後，可以看到access token，顯示在螢幕上約１-２秒。



## 9.3 問題排除

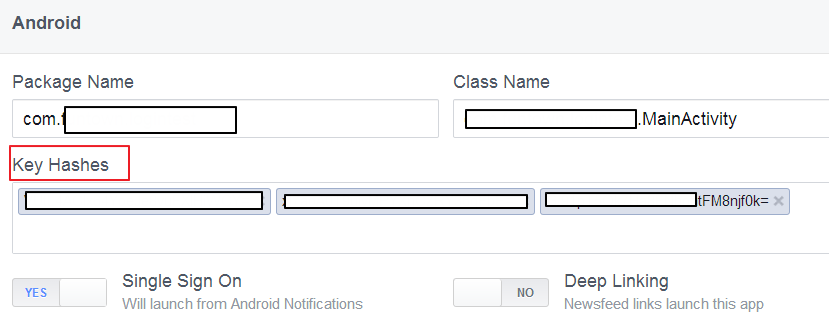
1. 當無法連結到Gaminfinity Server情況發生，但需要Gaminfinity SDK持續回覆玩家識別碼，

此時可以在呼叫getAccountId()時，傳入NULL的url，此時Gaminfinity SDK就不會嘗試連結到Gaminfinity Server，並回覆一個測試用的玩家識別碼，該數值為A123456789。

2. Facebook一直無法登入

這部分並非本SDK的範圍，但還是將問題排除方式記錄在這裡，主要是因為常遇到這樣的問題。

問題描述：按照程式碼實作完成後，嘗試登入Facebook，一直無法成功登入。其中常遇到的問題是在Facebook App設定中，沒有將您的App所產生的Key Hashes填入所導致。



## 9.4 登入頁面實作參考

(1) 無論Facebook帳號登入或尚未登入，都會顯示【立即試玩】

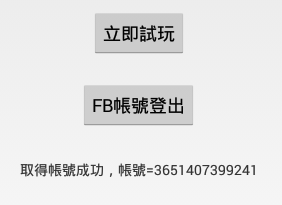
本方法不用在UI的登入介面上不用做任何處理，只要將【Facebook帳號登入】Button和【立即試玩】Button同時顯示即可。

在Android 的 GaminfinitySdkDemo專案裡面，將本路徑\GaminfinitySdkDemo\src\com\gaminfinity\example \MainActivity.java 檔案中內容

修改如下即可看到效果。(iOS GaminfinitySDKDemo專案沒有提供這項程式碼)

private static final Boolean OPTION\_HIDE\_INSTANT\_LOGIN\_BUTTON\_WHEN\_FB\_LOGIN = **false**;

下圖為當FB帳號登入成功後，玩家還是可以使用【立即試玩】，取得到另外一組帳號



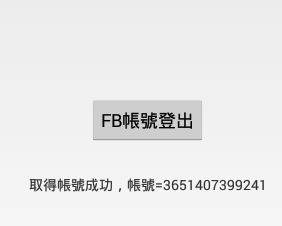
(2) Facebook帳號登入成功後，尚未登出Facebook之前，不顯示【立即試玩】Button

在GaminfinitySdkDemo專案裡面，將本路徑\GaminfinitySdkDemo\src\com\gaminfinity\example \MainActivity.java 檔案中內容

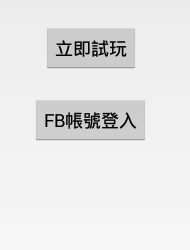
修改如下即可看到效果。

private static final Boolean OPTION\_HIDE\_INSTANT\_LOGIN\_BUTTON\_WHEN\_FB\_LOGIN = **true**;

Facebook帳號登入後，隱藏【立即試玩】Button



Facebook帳號登出後，顯示【立即試玩】Button



(3) 當GaminfinitySDK將AccoutId回傳給呼叫的主程式時，主程式將資料顯示在UI上，會出現UI threading的錯誤，可以使用下列方式排除。

**Android 可以利用runOnUiThread來解決**

@Override

public void onGetAccountId(final int result, final String accountId) {

…

this.runOnUiThread(new Runnable() {

public void run() {

TextView tv = (TextView)findViewById(R.id.textView1);

tv.setText("onGetAccountID(), result=" + result + ", AccountID=" + accountID);

}

});

}

**iOS可以利用performSelectorOnMainThread來解決**

- (void) onGetAccountId:(int)result AccountId:(NSString\*)accountId

{

…

[self performSelectorOnMainThread:@selector(updateUI:)

withObject: [NSString stringWithFormat: @”result=%d, accountId=%@", result, accountId]

waitUntilDone:YES];

}

- (void) updateUI: (NSString\*) msg{

\_label.text = msg; //顯示在\_label上面

}