

## Facebook Integration Pro Plugin by Game Slyce

### Table of Contents

Overview .....	2
Basic Configuration: .....	3
Detail of Classes and Methods.....	10
Namespace.....	10
Classes.....	10
ConstantData.cs: .....	10
GameScoreItem.cs: .....	10
ListItemInvite.cs: .....	10
ListItemLeader.cs: .....	10
Payment.cs:.....	10
Utils.cs:.....	10
FBManager:.....	11
Section - Initialization and Declarations.....	11
Section - Get and Post Current User's Score- User's all games' Score.....	11
Section – Get Friends Info.....	12
Section - Custom and Native Invite .....	12
Section – FB Init Login and Logout .....	13
Section - User FB Name, Picture - Saving for Offline Access.....	14
Section – Screenshot.....	14
Section – Get Deep Link.....	14
Section – ShareOnFacebook .....	14
Section – Canvas Payment.....	15
Section – Achievement.....	15
Section – Share Graph API.....	15
Setting up Canvas Payment, Custom Graph stories and Facebook Achievements .....	15
Sharing: .....	15
Open Graph Stories (How to Create Open Graph Object).....	19
Help and Support .....	22

## Overview

This Asset is been made with Unity version 5.1.0 and it is based on Official Facebook SDK for Unity Version 7.4.0. Your game must have canvas implementation as invitable\_friends is a game only feature. Your Game should have mechanism to get publish\_action permission from Player for score API, Achievement API, Open Graph Stories Sharing over Facebook etc. This package also includes how to create canvas payment system for your game on Facebook.com.

The package is been distributed with full source code and hence it's really customizable. You can perform any sort of customization by copy pasting features from Main.scene File which is in the Following Folder

Assets>FBPro>Demo>Scenes

The Code is written in a manner so that referencing buttons and other gameobjects is easy and simple. You'll get instant prompt upon awake if any reference gets missed. Also the listeners are been attached through script and so you don't need to worry about the Game Object and can change it easily without taking care of on listeners of many buttons.

The Package included guide on how to create open graph Objects for achievements and Open Graph Stories to simplify things for naïve users. The Official Facebook guide is cluttered and is ambiguous, so we have tried to list steps in a manner which is both easy and simple for every user.

The server scripts are included in a zip file in the directory FBPro>server

The scripts are included so that you can just change links and required field and upload it on your own server to see things working. There are Facebook links that let you know how to create open graph objects without your server.

Help and support is always available. We are open for suggestions, if you want something more or thinks that something is missing or can be improved you can directly contact at [info.gameslyce@gmail.com](mailto:info.gameslyce@gmail.com) with your invoice number. Our Team will assist you as soon as possible.

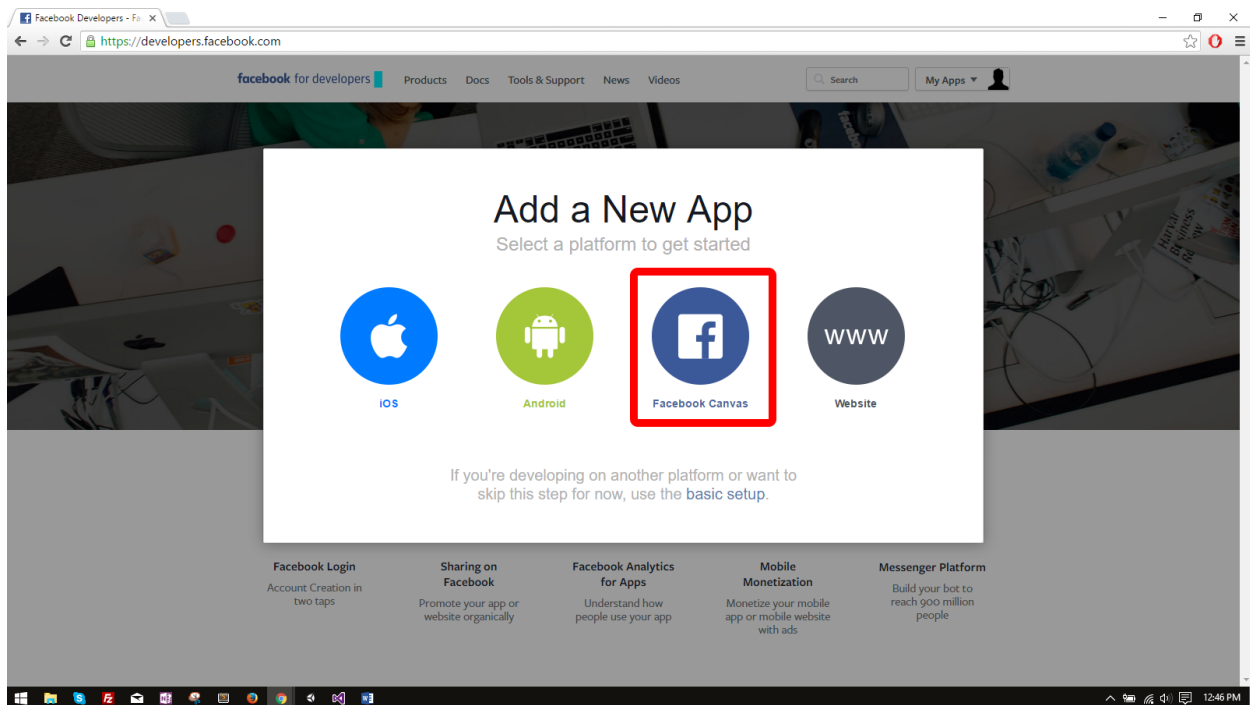
## Basic Configuration:

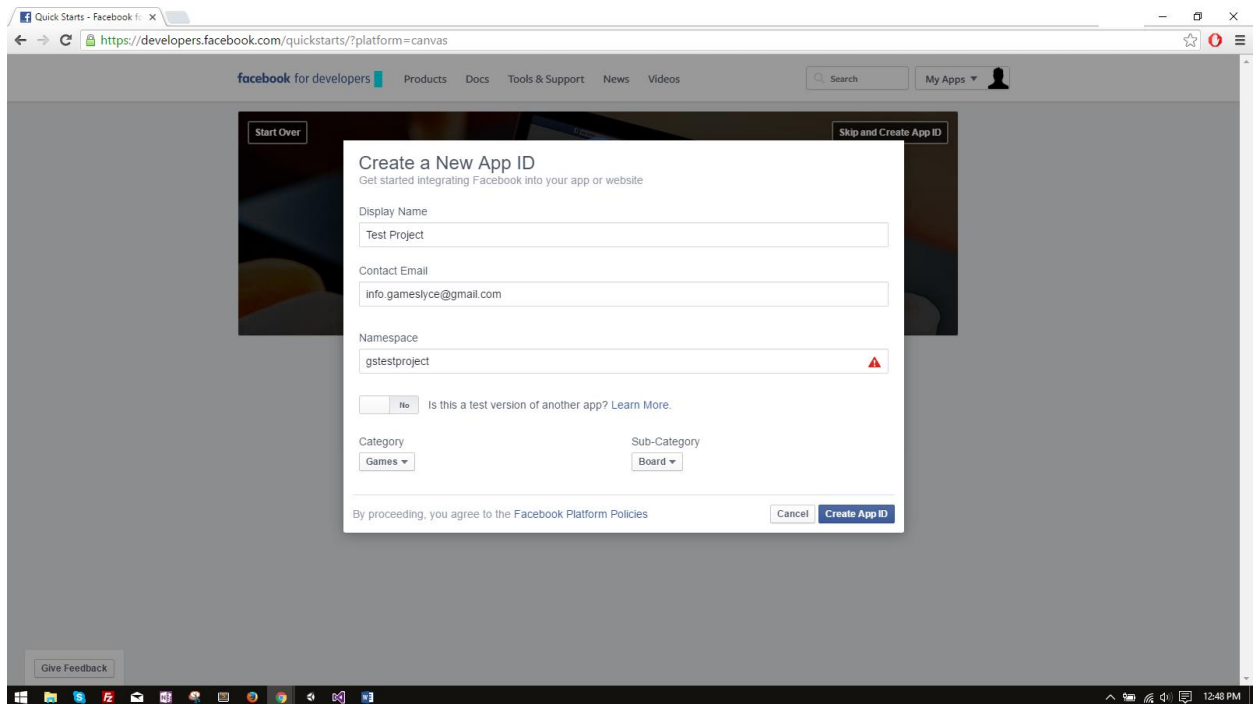
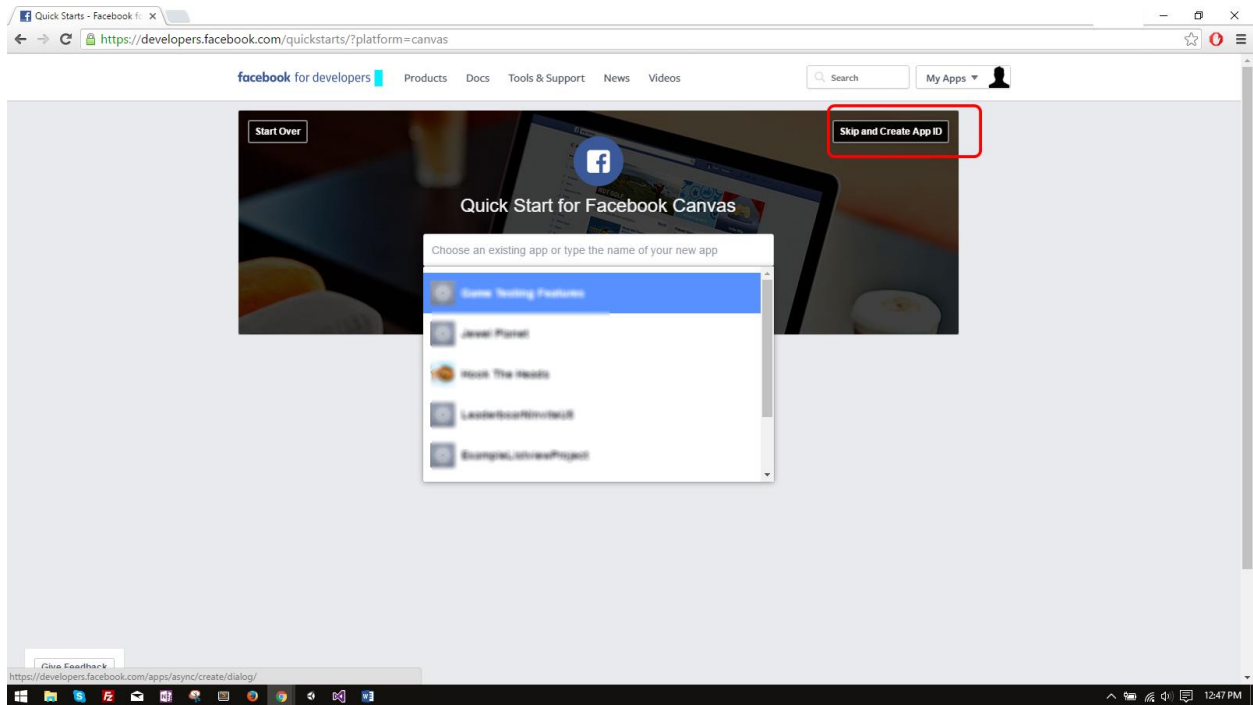
First of all Import the Unity Package which you have downloaded from Unity Asset Store.

Now Download the Latest Official Facebook SDK for Unity from the link below

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

Go to the Link <https://developers.facebook.com> and create a New App at Facebook Developer Dashboard. Don't forget to select it as A Game as some of the Features included in this Package are game only.



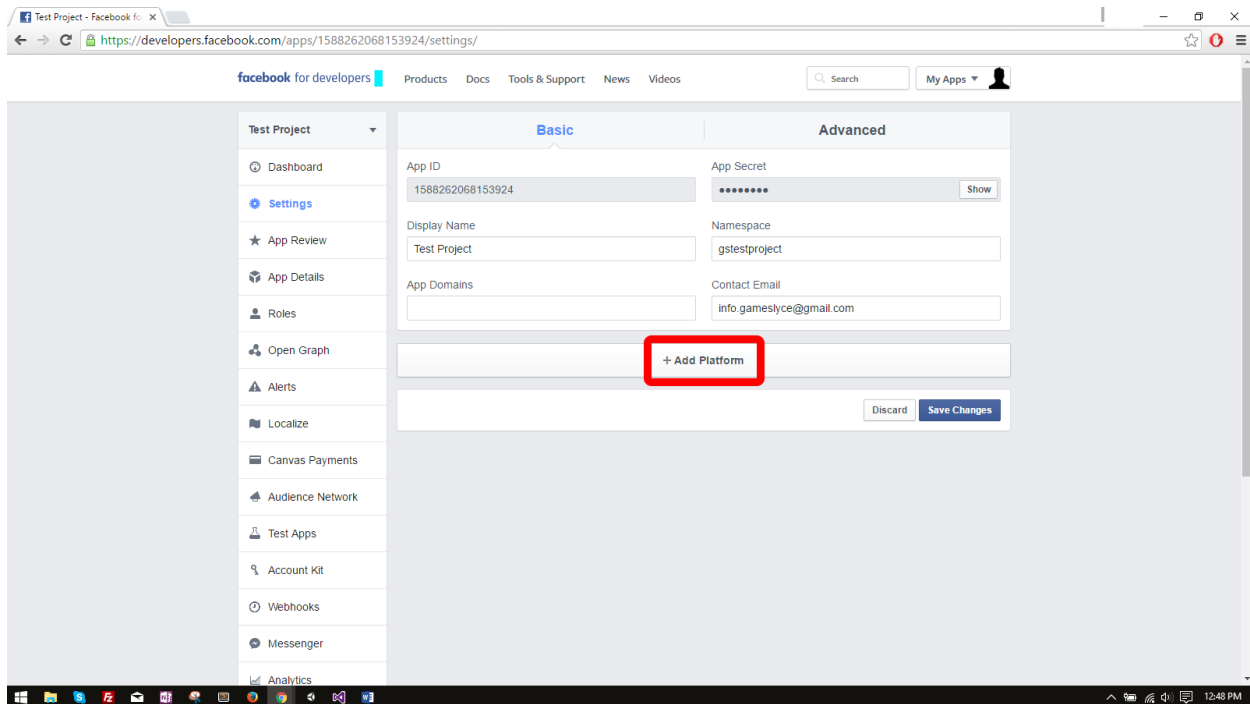
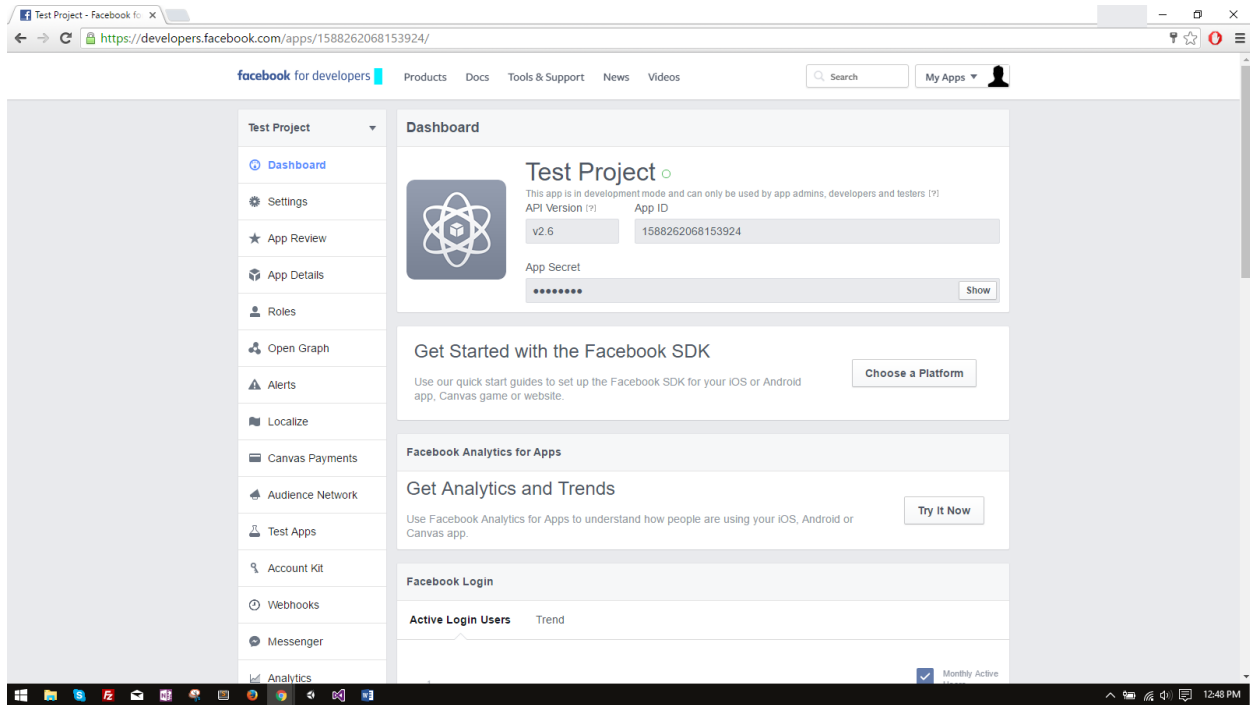


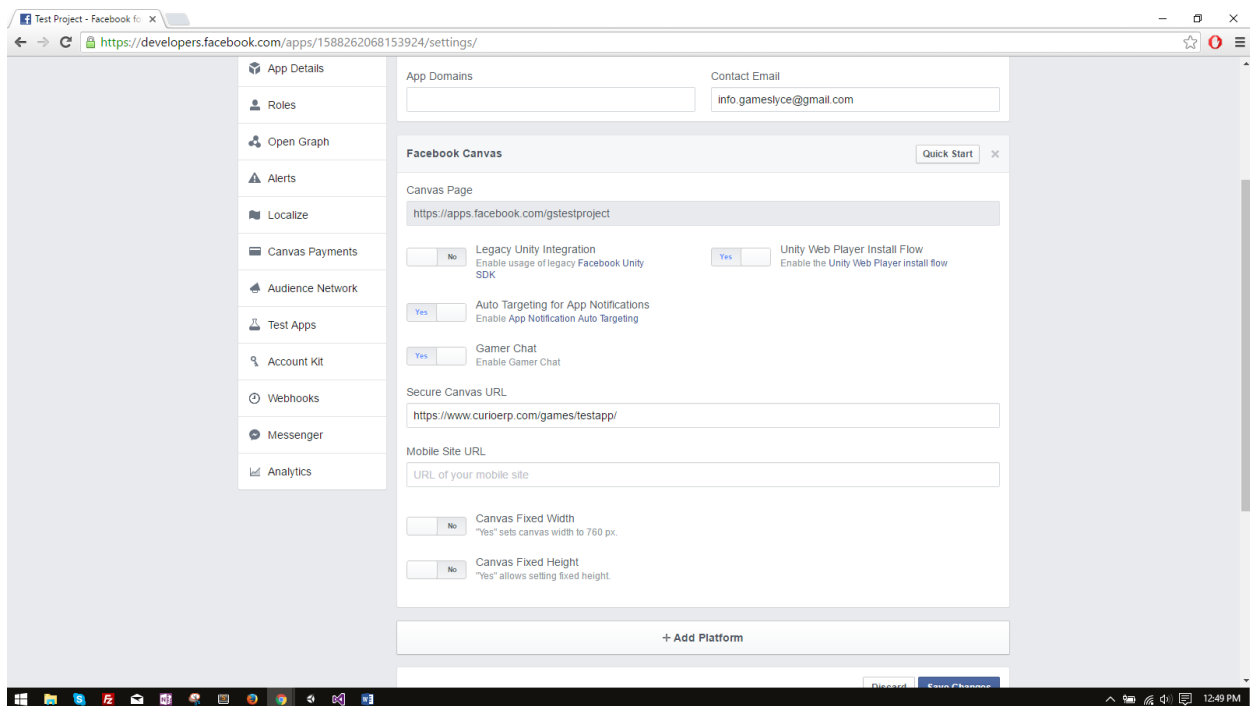
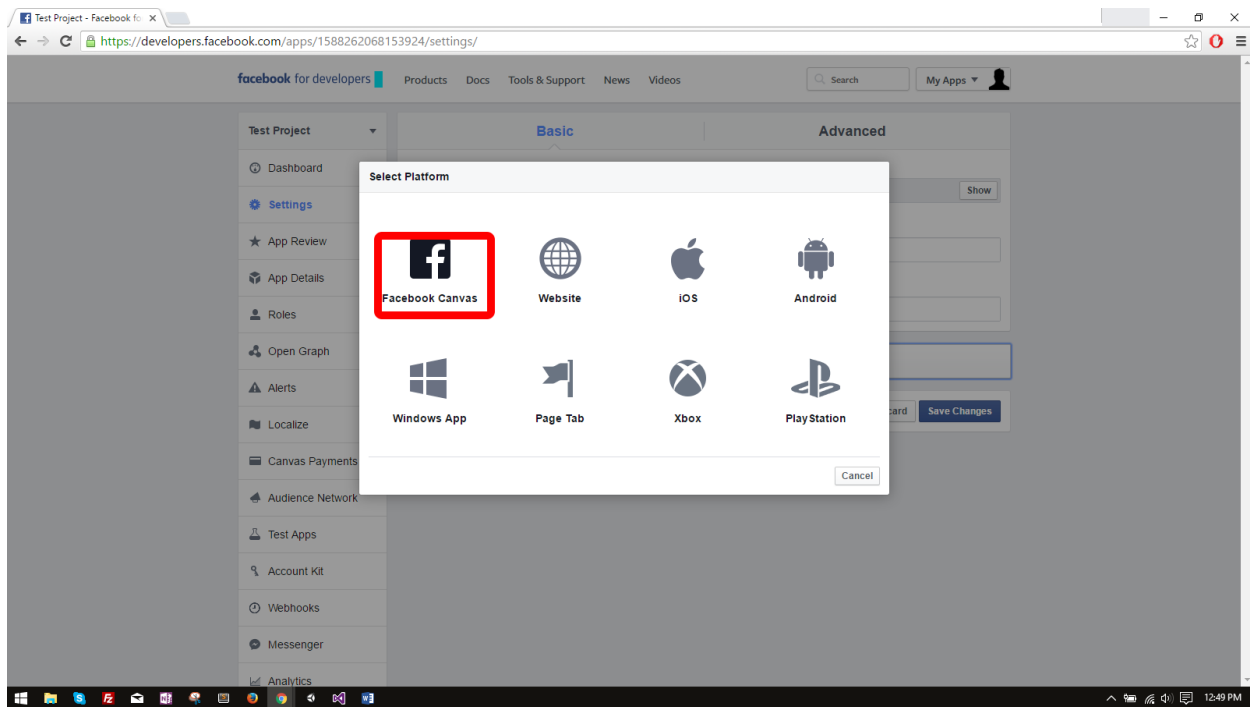
After you have created the Facebook App Go to the Link  
<https://developers.facebook.com/apps/<You App ID>/settings/>

Set Namespace for your game, Scroll down and click Add platform and select Facebook Canvas from the dialog box. It is a requirement that you app must have a canvas Implementation for the invitable\_friends API to work for more details see the link below

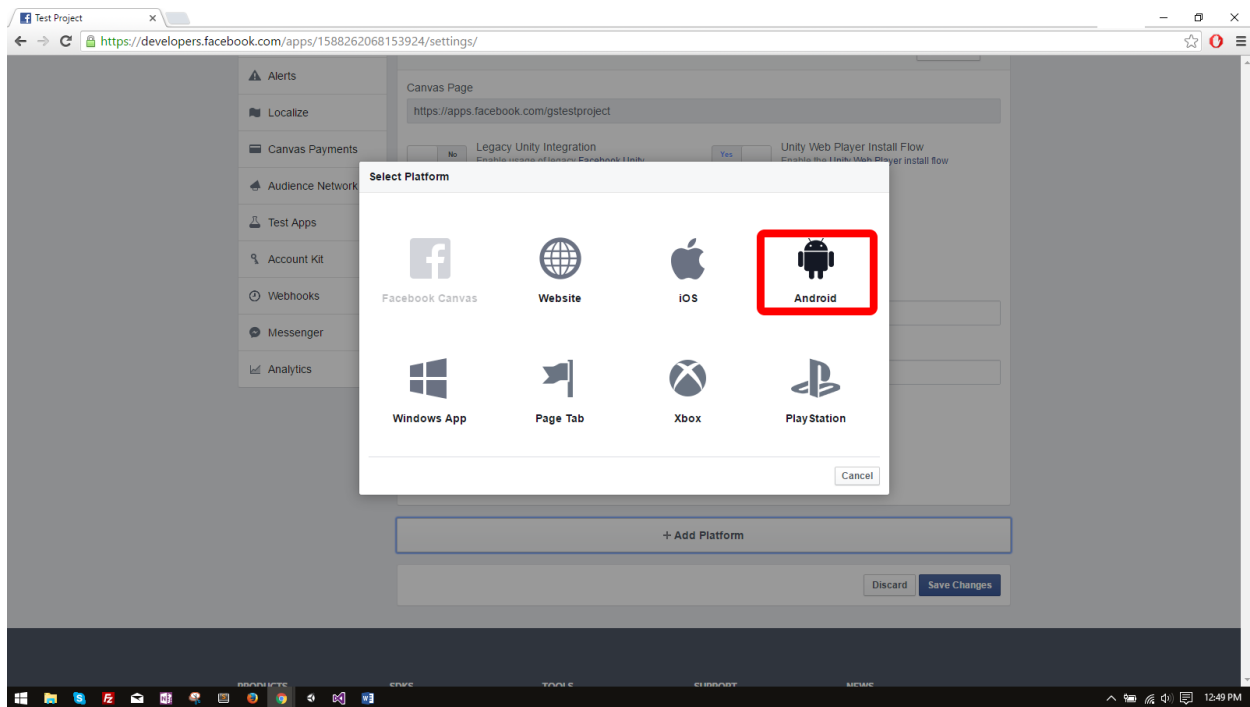
<https://developers.facebook.com/docs/games/services/gamerequests>

Fill in the Facebook Canvas Details as Shown in the Link below.



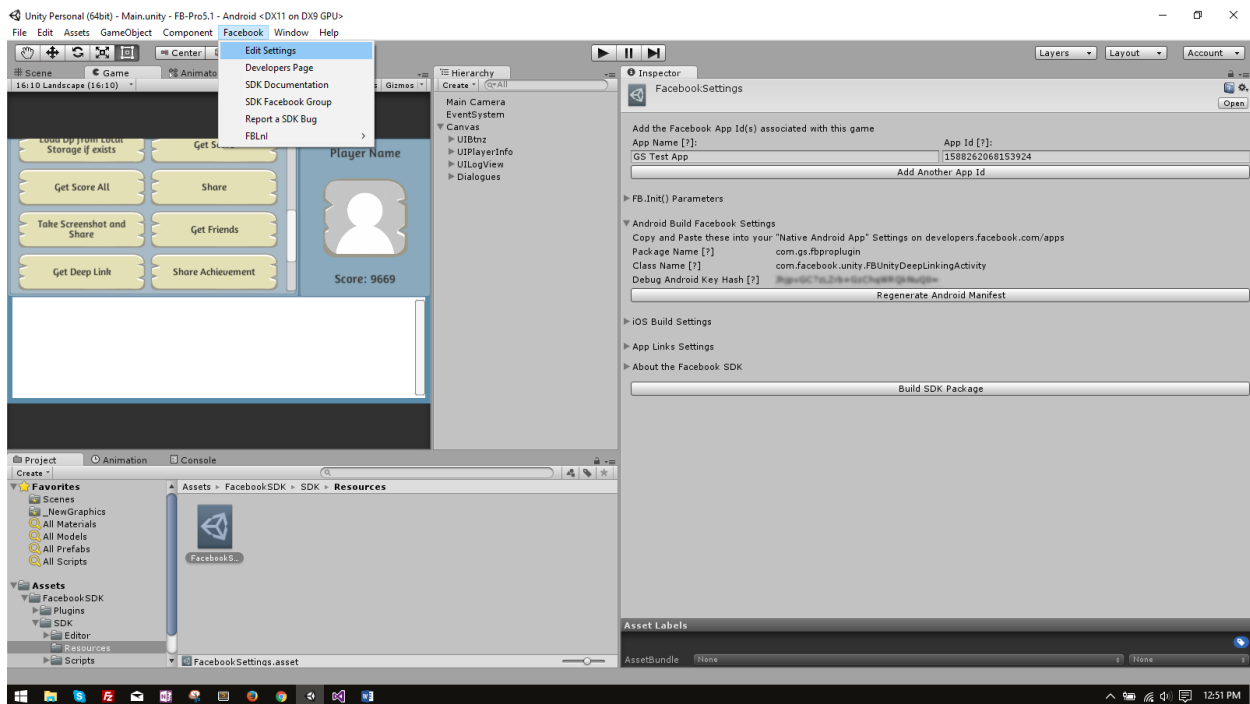


After Adding Facebook Canvas Now click Add Platform again and select Android



From Within Unity Click Facebook in the Top Menu Bar and Select "Edit Settings" from the drop down menu

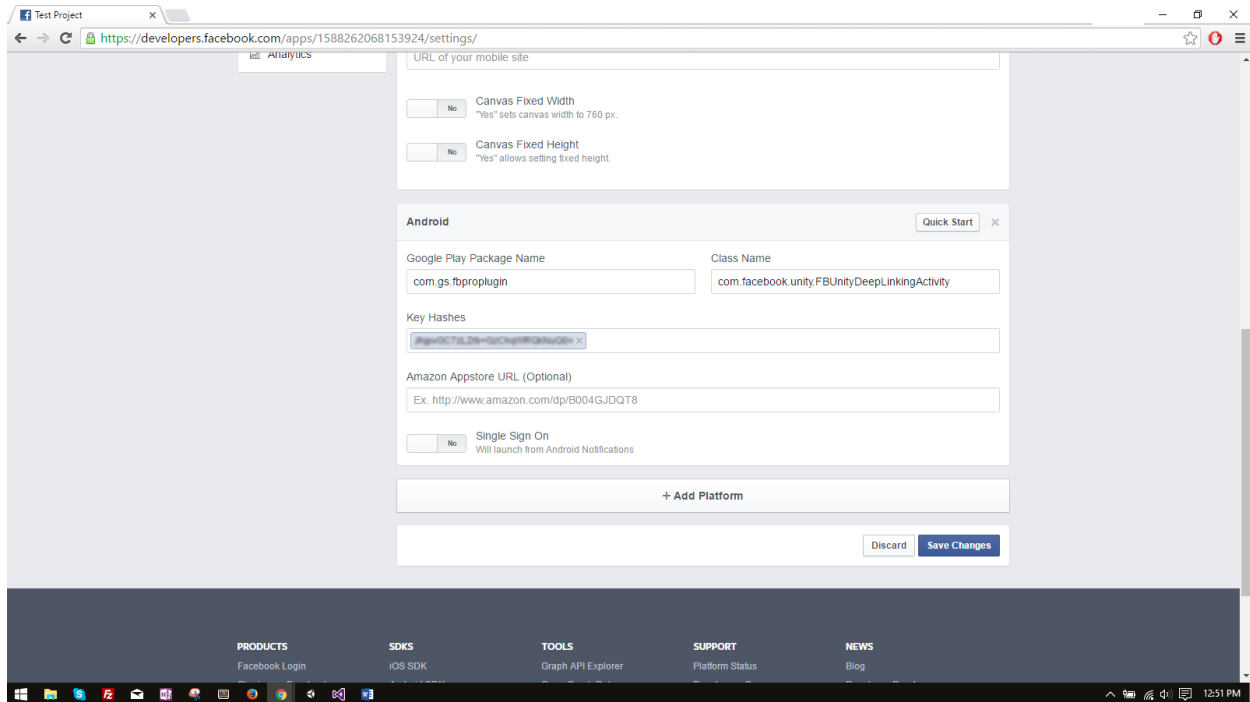
Now Add App ID from the Facebook Developer Page to Unity Editor



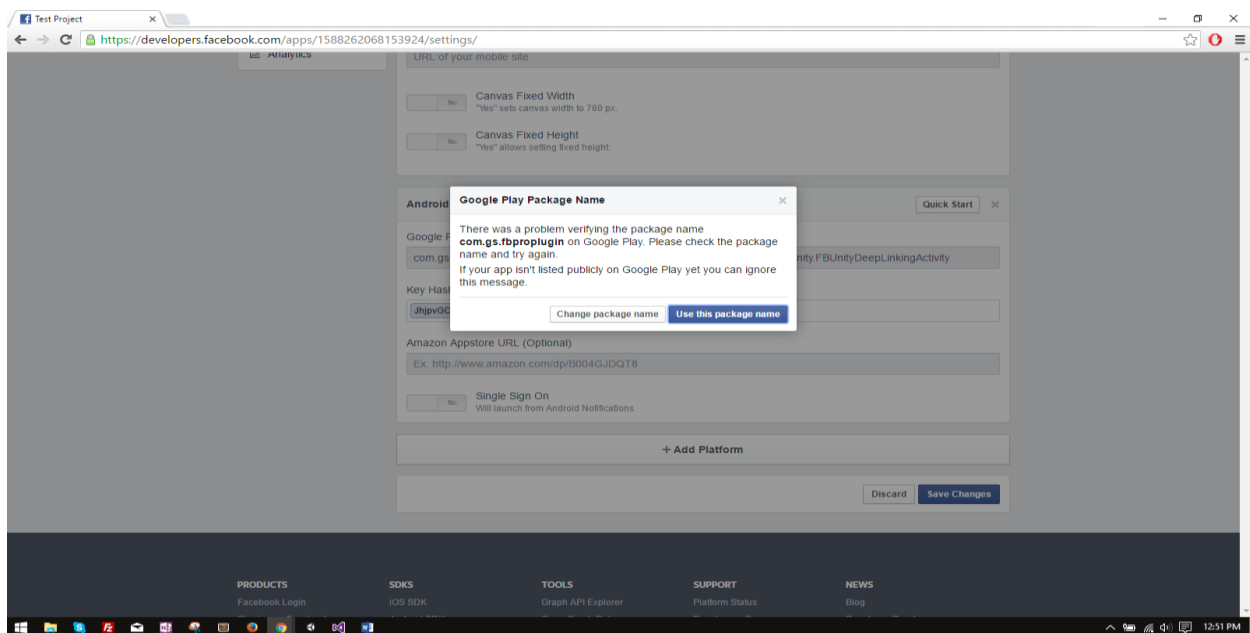
Fill in Android Platform Data from the Unity Editor to the Facebook Developer Page.

If you are on Windows and unable to see the keyhash to put on developer console than you need to install and configure openssl see the link below for steps in details.

<https://docs.google.com/document/d/1iSOelDm8cD-6iGEJehgczBuad308CepYpW4Dfksui6M/edit>



After that save changes.

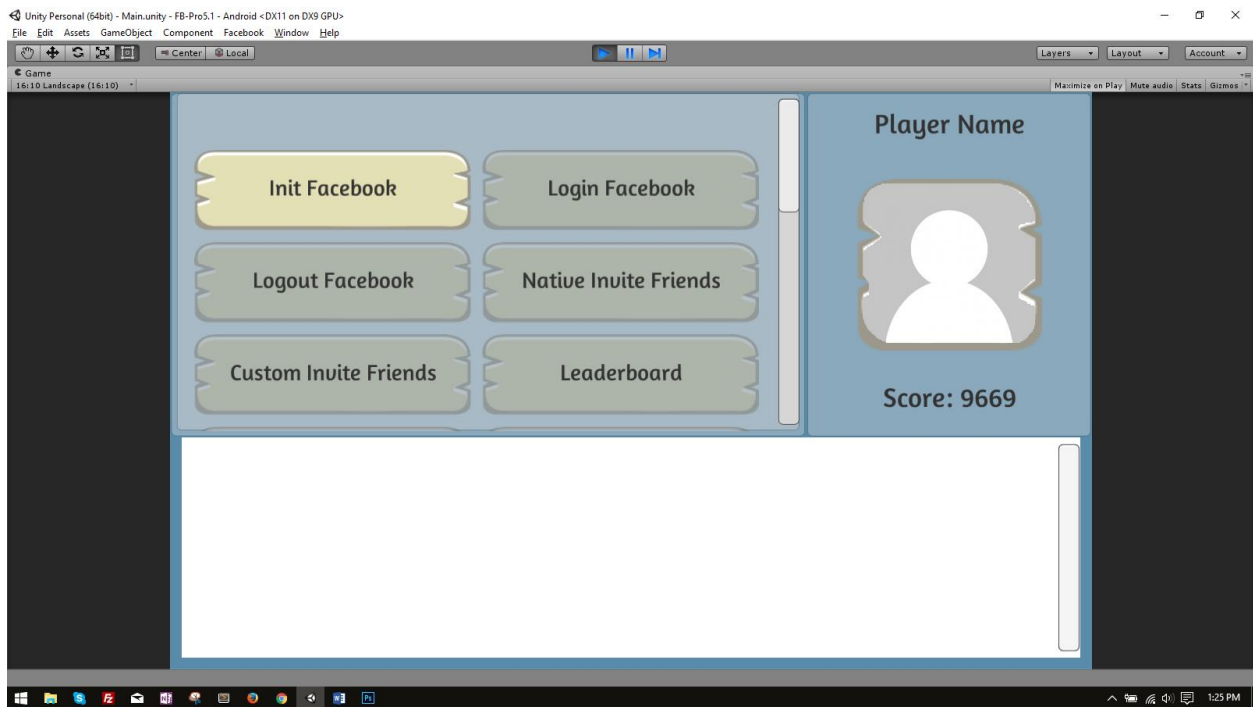




Now you are good to start testing demo scene.

In Unity Editor go to Asset>FBPro>Demo>Scenes and Add "Main" to your build Settings. For Through and Detailed Experience it's recommended to test the package on a Physical device as behavior of many Facebook Features is Mocked up in Unity Editor and is different than on actual devices.

Now hit Play to Test the Scene.



## Detail of Classes and Methods

### Namespace

The Package is packaged in a namespace called GS, so that you don't need to worry for the conflicting names of your own Project Classes.

### Classes

#### ConstantData.cs:

This class is used to store all the user specific data in one place. Instead of diving in all the source files to change some basic information for your game you can consult this class and replace your specific information.

#### GameScoreItem.cs:

This class is helper class that lets you store and show information for an item of user's games and game's score list.

#### ListItemInvite.cs:

This class is also a helper class that lets you store and show information in an easy manner for inviting friends to play your game in a Customized List View.

#### ListItemLeader.cs:

As the Name suggest this class is used for storing and showing information in the custom leaderboard of your game.

#### Payment.cs:

This class is used for Handling Facebook Canvas Payment. For secure and well trusted In-App Purchase experience, Facebook provide a very simple and easy to use mechanism for handling In-App items' purchasing.

#### Utils.cs:

This class is a small utility class that let you retrieve and de-serialize image URL for Facebook Friends.

## FBManager:

This Class is the Main Class and is responsible for every task of Facebook integration in this Asset.

**There are following Regions/sections in this Class. Each region/section is responsible for its specific task.**

## Section - Initialization and Declarations

In this section, all the declarations are taken placed. Here in the Awake Method of our FBManager Class, the necessary Initialization takes place.

Sprites – for check box states for the custom invite dialog

Dialog Buttons- All the dialog Buttons are reference here and there action handlers are defined and configured in this script.

Text – These UI.Text Items are referenced here including player name, score and few text items of the dialogues

List of Invites, Leaderboard, Game Score, User Friends who played this game etc

GameObjects Referencing to the Prefabs for Leaderboard, Invite, Friends-Details and all game scores of Player.

GameObjects that are parent (List parent) and will have all the items

Strings that are responsible for making Facebook API Calls.

Buttons: Reference of all the buttons that are performing different actions. All the Listeners are added through script for easy referencing and changing GameObject position in hierarchy.

Text Object that is using for showing Logs.

GameObjects for showing and hiding Dialogues, and Loader Screens

InputField Objects for searching invitable\_friends list, leaderboard list, submitting score and posting graph Message

## Section - Get and Post Current User's Score- User's all games' Score

Posting, Getting User's score and score of all games of user is managed here.

PostScore():

This method checks for publish permission and ask for it if not granted or else Call method to post score.

PostOnlyIfPermitted():

This method post's user Score to facebook by facebook score API.

GetPublishPermission():

This method is responsible for getting Facebook Publish actions permission. You can make a custom UI to make it indulge more in the game UI.

GetAppIDNScore():

This method is responsible for Getting AppID for Getting Score of current app of the current user.

GetFacebookScoreInternal():

This method gets the Facebook score from Facebook Server.

GetAllScores():

This method gets score of all the apps that user has played and apps/games which are using score api of Facebook. This particular method requires user\_games\_activity permission on Facebook.

## Section – Get Friends Info.

GetFriendsInfo():

This method is used for getting Facebook Friends Count and details of friends who have installed this game.

## Section – Leaderboard

LoadLeaderboard():

This method loads the Leaderboard JSON from Facebook Server.

CallBackLoadLeaderboard():

This method performs necessary steps for parsing JSON and displaying it in the UI.

OnValueChangedLeaderSearcher():

This method performs the Searching mechanism when user enters name of his/her friends in the Search box of Leaderboard UI.

LoadFriendsAvatar(), CreateListItemLeaderboard(), PicCallBackLeaderboard() and LoadFPicRoutine() are responsible for the miscellaneous helpful task.

## Section - Custom and Native Invite

TglSelectAllClickHandler()

This method is click handler that manages and sets the check buttons.

ChangeToggleState()

Change toggle state of Single Item.

SendInvites():

This method is used for sending Invite to the selected item in the list of invitable\_friends

CallInvite():

This method is actually a helper method to send invites to friends. The Facebook invite API requires that you send invites to no more than 50 friends in a single call. So it'll be recursive if you have and selected more than 50 friends to whom you want to send invite to.

LoadInvitableFriends():

This method is responsible for sending call to the Facebook Invite API to get Invitable Facebook Friends.

CallBackLoadInvitableFriends()

Parsing JSON and Creating List of Invitable\_Friends and Displaying Details of Items is handled here.

CreateListItemInvite, PicCallBackInvitable are helper methods for creating and showing user's friend's data

OnValueChangeInvSearcher()

This method is used to search Facebook friends. It takes the value in the input Field of Invite friends and performs search.

NativeInviteFriendsFB()

This method shows a default native facebook UI for inviting friends to the game.

## Section – FB Init Login and Logout

InitFB()

This method is used for initializing Facebook SDK. OnHideUnity and InitCallbacks are related to this method.

LoginFB():

This method logged in the user with their FB Account. LogInCallBack is the CallBack as the name implies.

LogoutFB():

This Method is responsible for logging User out of Facebook Account in the current game/app and resetting everything to the start state.

ClearOldData():

This method is used for clearing all the current data from all the UI and Setting it to the very initial State. ClearLeaderboard(), ClearGameScore(), ClearUserFriendsData() are the related items.

SetFBItems():

This method handles setting Facebook Items upon login and logout.

## Section - User FB Name, Picture - Saving for Offline Access

GetPath(): This method returns the correct path for storing/ loading user's picture for every different Operating system.

LoadPlayerPic(): Method responsible for loading player picture from Facebook by inquiry the Graph API of facebook. GetProfilePicRoutine() and LoadOrSavePicture() are related Methods.

LoadDPIIfExists(): This method is responsible for loading picture if stored in the internal storage of system.

LoadPlayerName(): This methods Load's name of Player from Facebook Server after logging into Facebook from the app.

## Section – Screenshot

TakeScreenshotNShare():

This method call the coroutine for taking screenshot and sharing it over Facebook. The Coroutine TakeScreenshot is related method.

## Section – Get Deep Link

GetDeepLink():

This method returns you deep linking JSON, which you can use to parse for giving Lives and In-App gifting, Inviting etc. and knowing that the App is launched from the Facebook Notification on web, mobile or an app on mobile.

## Section – ShareOnFacebook

ShareOnFB():

This method is used to share a customized message with Custom picture url and description on User's Facebook Wall with an Default Facebook Sharing Dialog.

## Section – Canvas Payment

BuyCoins():

This method is responsible to assist developer in implementing a purchase system in your Game on Facebook Canvas.

UpdateCoins():

Updates coins after successful purchase on Buy Purchase Dialog

## Section – Achievement

PostAchievement():

This method is used for giving user an achievement and sharing it over Facebook.

## Section – Share Graph API

ShareViaDialog():

This method Post a custom Graph Story with the native Facebook Dialog. User can add a custom message to it as well.

ShareViaGraph():

This method post a custom Graph story with a fully custom Dialog box. The Game's familiar UI will be more reasonable in most of the case. There will be a Dialogue where user can put the message. Sharing will be seamless without the native Facebook Dialog. It is encourage that you explicitly tell user before posting to Facebook that they are going to post it on Facebook as well as there should be an option to skip posting in your real game.

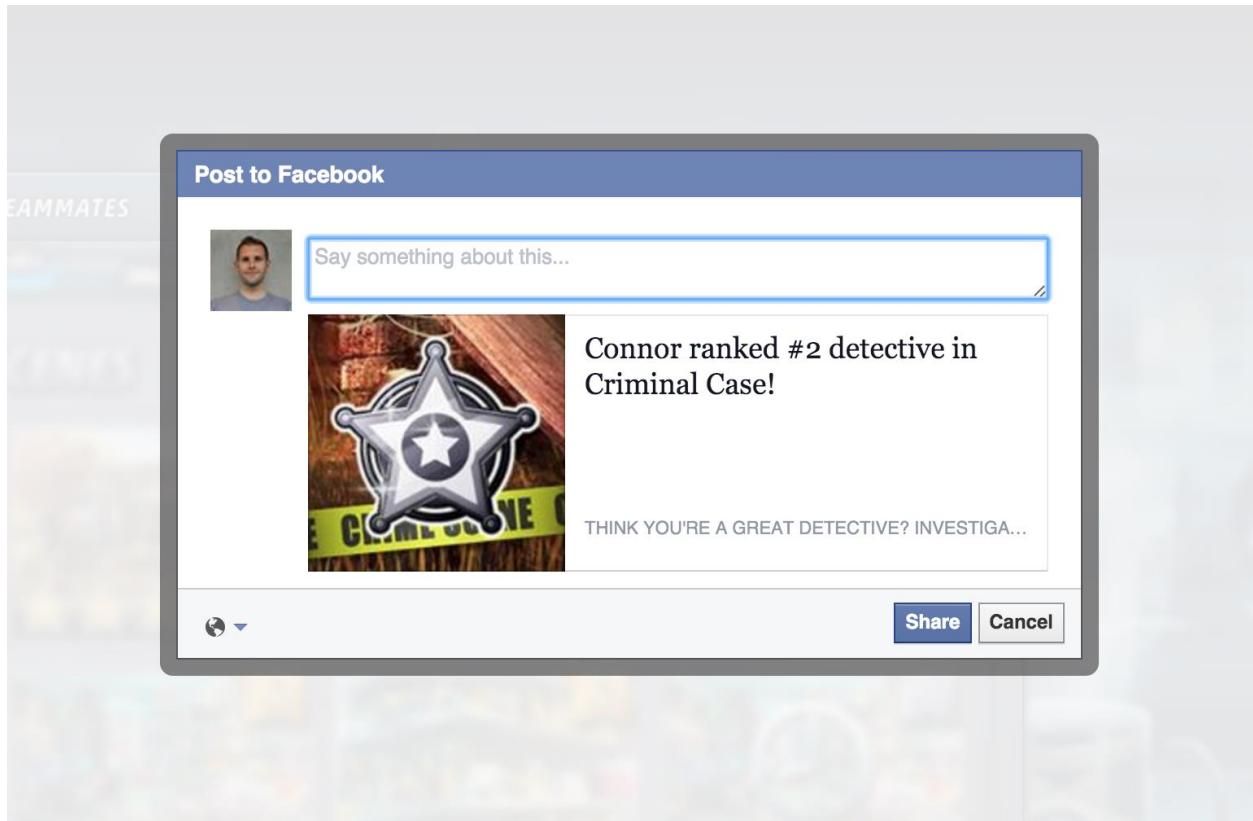
## Setting up Canvas Payment, Custom Graph stories and Facebook Achievements

Sharing:

There are 3 Types of sharing for games from Facebook. See more

<https://developers.facebook.com/docs/games/services/sharing>

Sharing Through Native Facebook Dialog:



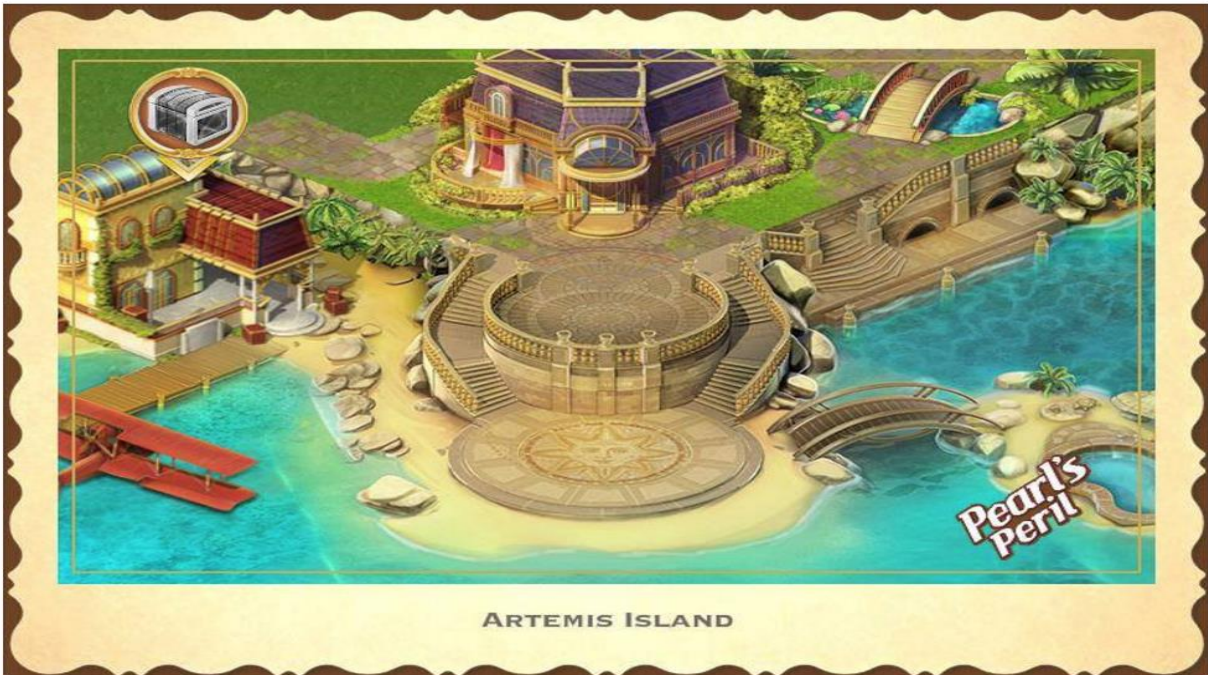
Share Picture, Video or Story through Open Graph:





**Connor Treacy** added a new photo.

Just now · Pearl's Peril · 🌐 ▼



**Connor Treacy**

11 mins · Caesars Casino · 👤 ▼



### Daily Bonus Mega Black Wedge: X5 MORE!

Enjoy the world's best free slots games with X5 your White Wedge value on your daily bonus! Play over 100 games with outstanding Black Wedge bonus!

CONNOR IS SHARING THE LUCK AFTER GETTING A BONUS WHEEL DAI...

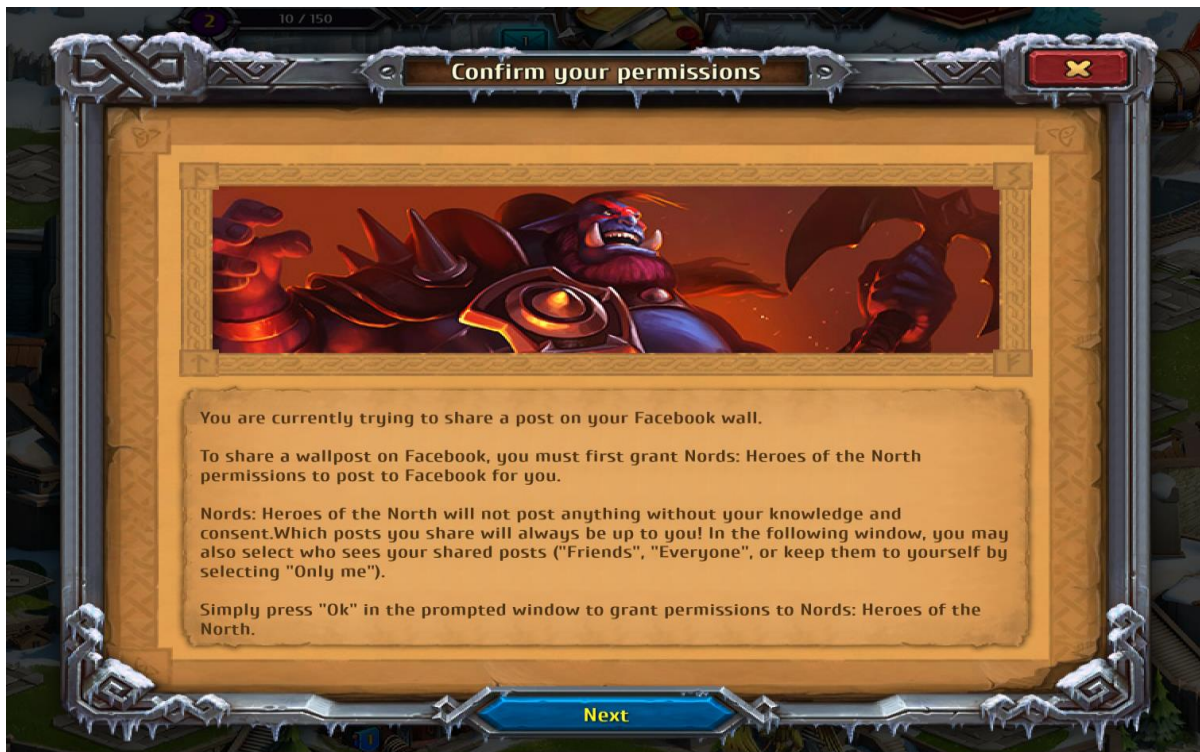
[Play](#)

Creating Open Graph Stories and sharing it over Facebook with UI that blends well in the interface and environment of your game with no Native Facebook Style Dialog.





It is recommended, needed and encouraged that you always show a proper message to let user know that this app is going to post something on his/her behalf. And there should always be a way to skip post and unsubscribe from granted Facebook permissions from within app.



## Open Graph Stories (How to Create Open Graph Object)



Let's get started on how to do this. For this example, we will send a life to our friends! As with our other post, this will be divided into two parts: One explaining the Facebook side configuration we need to do, and after we have that all set up, the appropriate C# code to make the actual call.

Head on to the [Facebook Developer Console](#), and under the menu Apps, select your game (If you don't have your game in the dropdown menu, go and create it before continuing). Now, from all the options on the left-side menu, click on Open Graph. This opens up the tab where you can manage everything Open Graph related.

Click on Object Types, and then Add Object Type. For this example, as we said, we are going to create a life.

Now we entered the editing screen of the new object. Just fill in the appropriate names for plural and singular, and scroll down to save the changes.

Now that we have the type created, we have to create an instance of that type, to use in our game. Head on back to the [Developer Console](#) of your game, and on Open Graph -> Object Types.

Stories		Action Types	Object Types
Object Types	Properties		
Game Achievement	Points, Secret		Get Code · ×
GsFbTest			Get Code · ×
+ Add Object Type			

Click on the right hand side, where it says Get Code. Now you will be prompted with a new popup that has examples of all the calls we can do with that object type (We can create a new object, update an existing one, delete, etc).

Since what we want to do is create one, just click on the link that is shown there (As it says there, keep in mind you need publish\_actions permissions to perform this. To get these permissions, in the Graph API Explorer page, we can click on Access Token and select the permissions we would like to have).

The screenshot shows the Facebook Developer Console interface. A modal window titled "Sample Code for Game Achievement" is open. It contains the following information:

- Buttons: Create, Read, Update, Delete, Markup
- Platform: HTTP
- Required Permission: publish\_actions
- Sample Code (POST request):

```
https://graph.facebook.com/me/objects/game.achievement?
access_token=ACCESS_TOKEN&
method=POST&
object=%7B%22og%3Aurl%22%3A%22http%3A%5C%2F%5C%2Fsamples.ogp.me%5C%2F161147604019932%22%2C%22og%3Atype%22%3A%22game.achievement%22%7D
```
- Close button

After following the link, we arrive at the [Graph API Explorer](#), with some data already filled in for us. The only thing we have to do, is change a few things in the parameter of the POST call. I suggest you copy and paste it into your favorite text editor, for easier editing.

```
{  
  "app_id":660672357301163,  
  "type":"testProject:life",  
  "url":"http://samples.ogp.me/1234568956",  
  "title":"Sample Life",  
  "image":"https://fbstatic-a.akamaihd.net/images/devsite/attachment_blank.png",  
  "description":""  
}
```

Now, before we fill it in with the proper values, there is a detail to keep in mind.

Now, let's fill in the corresponding values for the properties:

- ⇒ app\_id – Leave it as is. This is the ID of your game.
- ⇒ type – Leave it as is. This will be the identifier of the type of Object you are creating.
- ⇒ url – This should be the url of your game. Keep in mind to escape (\\) all special characters
- ⇒ title – The title to display.
- ⇒ image – The url to the image that you want to display.
- ⇒ description – A short description of the object.

Here is a final look at how my object looked:

```
{  
  "app_id":1234567895456,  
  "type":"testapp:life",  
  "url":"https://apps.facebook.com/hooktheheads",  
  "title":"Life",  
  "image":"https://www.curioerp.com/gameslyce/plugins/fbpro/life.png",  
  "description":"Use this life to play another time!"  
}
```

Now, go back to the Graph API Explorer page where you were, copy the contents of your text editor, and replace the value that you copied to start off with this new one, with all the

appropriate values.

If everything is correct, you should see an ID returned as a response.

That's all now you can create engaging open graph stories for your game. Just see the section "Share Graph API" to know how to create open graph stories.

For details on how to create open graph object for Stories see

<https://developers.facebook.com/docs/sharing/opengraph/custom>

For Achievement See the Link below

<https://developers.facebook.com/docs/games/services/scores-achievements#achievements>

For canvas payment see the link below

<https://developers.facebook.com/docs/unity/reference/current/FB.Canvas.Pay>

## Help and Support

More Assets from GameSlyce Design:

<https://www.assetstore.unity3d.com/en/#!/search/page=1/sortby=popularity/query=publisher:20940>

More Assets from Game Slyce:

<https://www.assetstore.unity3d.com/en/#!/search/page=1/sortby=popularity/query=publisher:10488>

Support Email: [info.gameslyce@gmail.com](mailto:info.gameslyce@gmail.com)

Facebook Page: <https://www.facebook.com/gameslyce>