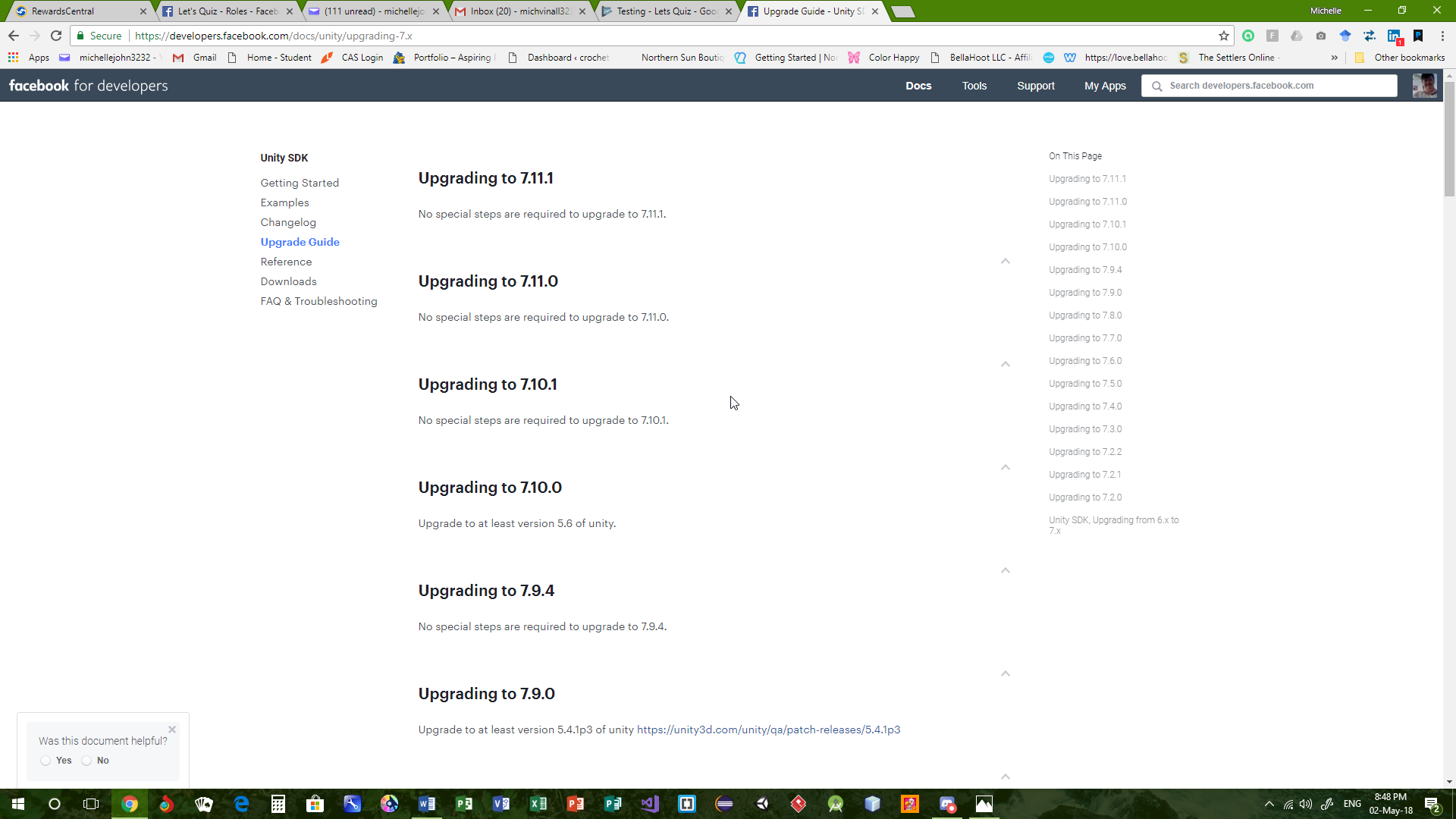
Review Updates to Facebook SDK



Example Scripts For Facebook SDK

using UnityEngine;

using UnityEngine.UI;

using System.Collections;

using System.Collections.Generic;

using Facebook.Unity;

using System;

public class FacebookManager : MonoBehaviour

{

private static FacebookManager instance;

public static FacebookManager Instance

{

get {

if (instance == null) {

GameObject fBM = new GameObject ("FBManager");

fBM.AddComponent<FacebookManager> ();

}

return instance;

}

}

public bool isLoggedIn{ get; set; }

public string profileName{ get; set; }

public Sprite profilePic{ get; set; }

public string appLinkURL{ get; set; }

public string uriPic{ get; set; }

void Awake()

{

DontDestroyOnLoad (this.gameObject);

instance = this;

isLoggedIn = true;

}

public void InitFB()

{

if (!FB.IsInitialized)

{

FB.Init (SetInit, OnHideUnity);

} else {

isLoggedIn = FB.IsLoggedIn;

}

}

void SetInit()

{

if (FB.IsLoggedIn) {

Debug.Log ("FB is logged in");

GetProfile ();

} else {

Debug.Log ("FB is not logged in");

}

isLoggedIn = FB.IsLoggedIn;

}

void OnHideUnity(bool isGameShown)

{

if (!isGameShown) {

Time.timeScale = 0;

} else {

Time.timeScale = 1;

}

}

public void GetProfile()

{

FB.API ("/me?fields=first\_name", HttpMethod.GET, DisplayUsername);

FB.API ("/me/picture?type=square&height=128&width=128", HttpMethod.GET, DisplayProfilePic);

FB.GetAppLink (dealWithAppLink);

}

void DisplayUsername(IResult result)

{

if (result.Error == null) {

profileName = "" result.ResultDictionary ["first\_name"];

} else {

Debug.Log (result.Error);

}

}

void DisplayProfilePic(IGraphResult result)

{

if (result.Texture != null) {

profilePic = Sprite.Create (result.Texture, new Rect (0, 0, 128, 128), new Vector2 ());

}

}

void dealWithAppLink(IAppLinkResult result)

{

if (!string.IsNullOrEmpty (result.Url)) {

appLinkURL = result.Url;

}

}

public void share()

{

FB.FeedShare (

string.Empty,

new Uri("http://linktoga.me"),

"Hello this is the title",

"This is the caption",

"Check out this game",

new Uri("https://i.ytimg.com/vi/NtgtMQwr3Ko/maxresdefault.jpg"),

string.Empty,

shareCallBack

);

}

void shareCallBack(IResult result)

{

if (result.Cancelled) {

Debug.Log ("Share Cancelled");

} else if (!string.IsNullOrEmpty (result.Error)) {

Debug.Log ("Error on share!");

} else if (!string.IsNullOrEmpty (result.RawResult)) {

Debug.Log ("Success on share");

}

}

public void invite()

{

FB.Mobile.AppInvite (

new Uri("http://linktoga.me"),

new Uri("https://i.ytimg.com/vi/NtgtMQwr3Ko/maxresdefault.jpg"),

inviteCallBack

);

}

void inviteCallBack(IResult result)

{

if (result.Cancelled) {

Debug.Log ("Invite Cancelled");

} else if (!string.IsNullOrEmpty (result.Error)) {

Debug.Log ("Error on invite!");

} else if (!string.IsNullOrEmpty (result.RawResult)) {

Debug.Log ("Success on Invite");

}

}

public void shareWithUsers()

{

FB.AppRequest (

"Come play, beat my high score!",

null,

new List<object> (){ "app\_users" },

null,

null,

null,

null,

shareWithUsersCallBack);

}

void shareWithUsersCallBack(IAppRequestResult result)

{

Debug.Log (result.RawResult);

if (result.Cancelled) {

Debug.Log ("Challenge Cancelled");

} else if (!string.IsNullOrEmpty (result.Error)) {

Debug.Log ("Error on challenge!");

} else if (!string.IsNullOrEmpty (result.RawResult)) {

Debug.Log ("Success on challenge");

}

}

}

scripts/FacebookManager.cs.meta

scripts/FacebookScript.cs

using UnityEngine;

using UnityEngine.UI;

using System.Collections;

using System.Collections.Generic;

using Facebook.Unity;

public class FacebookScript : MonoBehaviour {

public GameObject DialogLoggedIn;

public GameObject DialogLoggedOut;

public GameObject DialogUsername;

public GameObject DialogProfilePic;

void Awake()

{

FacebookManager.Instance.InitFB();

DealWithFBMenus (FB.IsLoggedIn);

}

public void FBlogin()

{

List<string> permissions = new List<string> ();

permissions.Add ("public\_profile");

FB.LogInWithReadPermissions (permissions, AuthCallBack);

}

void AuthCallBack(IResult result)

{

if (result.Error != null) {

Debug.Log (result.Error);

} else {

if (FB.IsLoggedIn) {

FacebookManager.Instance.isLoggedIn = true;

FacebookManager.Instance.GetProfile ();

Debug.Log ("FB is logged in");

} else {

Debug.Log ("FB is not logged in");

}

DealWithFBMenus (FB.IsLoggedIn);

}

}

void DealWithFBMenus(bool isLoggedIn)

{

if (isLoggedIn) {

DialogLoggedIn.SetActive (true);

DialogLoggedOut.SetActive (false);

if (FacebookManager.Instance.profileName != null) {

Text userName = DialogUsername.GetComponent<Text> ();

userName.text = "Hello, " FacebookManager.Instance.profileName;

} else {

StartCoroutine ("waitForProfileName");

}

if (FacebookManager.Instance.profilePic != null) {

Image profilePic = DialogProfilePic.GetComponent<Image> ();

profilePic.sprite = FacebookManager.Instance.profilePic;

} else {

StartCoroutine ("waitForProfilePic");

}

}else{

DialogLoggedIn.SetActive (false);

DialogLoggedOut.SetActive (true);

}

}

IEnumerator waitForProfileName()

{

while (FacebookManager.Instance.profileName == null) {

yield return null;

}

DealWithFBMenus (FB.IsLoggedIn);

}

IEnumerator waitForProfilePic()

{

while (FacebookManager.Instance.profilePic == null) {

yield return null;

}

DealWithFBMenus (FB.IsLoggedIn);

}

public void share ()

{

FacebookManager.Instance.share ();

}

public void invite()

{

FacebookManager.Instance.invite ();

}

public void shareWithUsers()

{

FacebookManager.Instance.shareWithUsers ();

}

}