# APPLOVIN ADS IMPLEMENTATION

# - Import Applovin SDK for unity

# 1. Android and iOS

**-**Add Script: ApplovinListener.cs to the game object with name: ApplovinListener

# 2. iOS

**Link the Following Frameworks in Your Project**

- AdSupport

- AVFoundation

- CoreGraphics

- CoreMedia

- CoreTelephony

- SafariServices (NEW)

- StoreKit

- SystemConfiguration

- UIKit

- WebKit (NEW)

- libz.tbd (NEW)

## Enable Xcode Flags

The **-ObjC** flag **must** be added for static libraries to run correctly.

To enable the -ObjC flag, click on your project settings, go to *Build Settings*, search for *Other Linker Flags* and add **-ObjC**

# Add this code to gameobject

# AppLovinManager.cs

public class ApplovinListener: MonoBehaviour  
 {  
 private const string SDK\_KEY = "your\_key";  
 private bool \_isPreloadingRewardedVideo = false;  
 public static AppLovinManager Instance;  
 private void Awake()  
 {  
 if (Instance == null)  
 {  
 DontDestroyOnLoad(gameObject);  
 Instance = this;  
 }  
 else if (Instance != this)  
 Destroy(gameObject);  
 }  
 void Start()  
 {  
 AppLovin.SetSdkKey(SDK\_KEY);  
 AppLovin.InitializeSdk();  
// AppLovin.SetTestAdsEnabled("true");  
 AppLovin.SetUnityAdListener(gameObject.name); // gameobject with name ApplovinListener  
 Invoke("LoadVideoDelay", 12f);  
 }  
  
 private void LoadVideoDelay()  
 {  
 AppLovin.LoadRewardedInterstitial();  
 if (GameSettings.settings.boughtRemoveAds)  
 AppLovin.HideAd();  
 }  
 public bool ShowInterstitial()  
 {  
 if (!AppLovin.HasPreloadedInterstitial())  
 {  
 }  
 else  
 {  
 AppLovin.ShowInterstitial();  
 return true;  
 }  
 AppLovin.PreloadInterstitial();  
 return false;  
 }  
 public bool InterstitialIsReady() // if readey inter  
 {  
 var result = AppLovin.HasPreloadedInterstitial();  
 if (result)  
 {  
 }  
 else  
 AppLovin.PreloadInterstitial();  
 return result;  
 }  
 public bool ShowRewardedInterstitial() // show video  
 {  
 if (!AppLovin.IsIncentInterstitialReady())  
 {  
 \_isPreloadingRewardedVideo = true;  
 AppLovin.LoadRewardedInterstitial();  
unity }  
 else  
 {  
 \_isPreloadingRewardedVideo = false;  
 AppLovin.ShowRewardedInterstitial();  
 AppsFlyerManager.Instance.TrackVideoAppLovinView();  
 return true;  
 }  
 return false;  
 }  
 public void ShowBanner()  
 {  
 if (GameSettings.settings.boughtRemoveAds)  
 {  
 return;  
 }  
 AppLovin.ShowAd(AppLovin.AD\_POSITION\_CENTER, AppLovin.AD\_POSITION\_BOTTOM);  
 AppsFlyerManager.Instance.TrackBannerAppLovinView();  
 }  
 void onAppLovinEventReceived(string ev)  
 {  
 if (ev.Equals("LOADBANNERFAILED")) // fail to load banner  
 YuanAds.Instance.ShowBanner();  
 if (ev.Equals("REWARDAPPROVEDINFO"))// if user complete video and call reward function here

{   
 // Process an event like REWARDAPPROVEDINFO:100:Credits  
 char[] delimiter = {'|'};  
 string[] split = ev.Split(delimiter);  
 // Pull out the amount of virtual currency.  
 double amount = double.Parse(split[1]);  
 // Pull out the name of the virtual currency  
 string currencyName = split[2];  
 // Do something with this info - for example, grant coins to the user  
 // myFunctionToUpdateBalance(currencyName, amount);  
 Log("Rewarded " + amount + " " + currencyName);  
 AppLovin.LoadRewardedInterstitial();  
 }  
// }  
 // Check if this is a Rewarded Video preloading event  
 else if (\_isPreloadingRewardedVideo && (ev.Equals("LOADED") || ev.Equals("LOADFAILED")))  
 {  
 \_isPreloadingRewardedVideo = false;  
 if (ev.Equals("LOADED"))  
 {  
// RewardedVideoButtonTitle.text = REWARDED\_VIDEO\_BUTTON\_TITLE\_SHOW;  
 }  
 else  
 {  
// RewardedVideoButtonTitle.text = REWARDED\_VIDEO\_BUTTON\_TITLE\_PRELOAD;  
 }  
 }  
 }  
 }