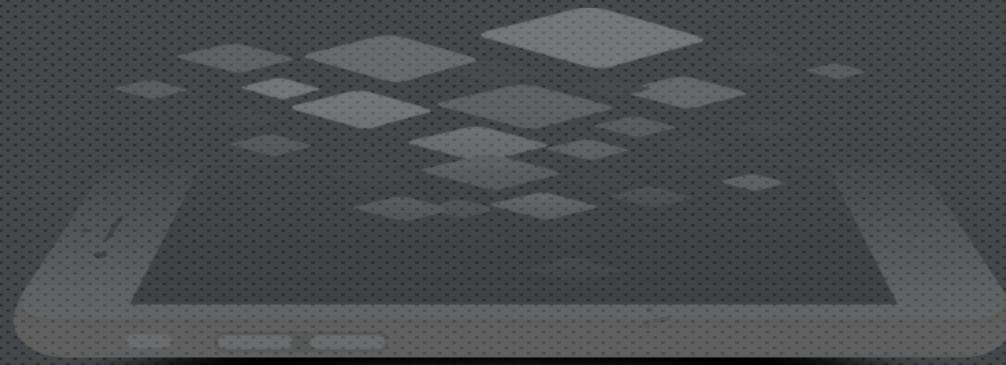


Unity Plugin for Android User Guide



애드립 Unity 플러그인 적용을 위한 문서입니다.

애드립 Unity 플러그인을 사용하기 위해서 다음 항목이 필요합니다.

- 애드립 API 키
- 애드립 최신 SDK(4.2.1.1 이상), SubView
- Google Play services 라이브러리
- 각 광고 플랫폼 사이트에서 발급받은 APP - ID 및 최신 SDK

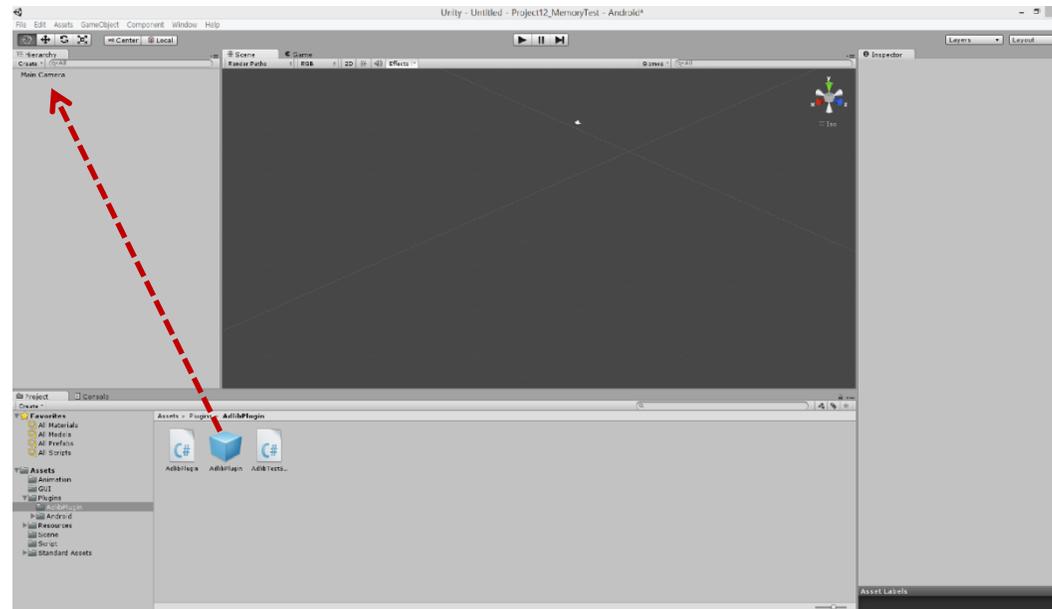
이 애드립 Unity 플러그인은 애드립 SDK (4.2.1.1)과 Unity 4.5 에서 작성 되었으며, 이전 버전에 대해서는 테스트가 이루어 지지 않았습니다.

※ 중요 ※

Unity 적용 시 애드립 SDK 는 ver.4.2.1.1 이상의 라이브러리를 사용하시기 바랍니다.

Jar 파일에 리소스를 포함하는 몇몇 플랫폼들은, 해당 리소스를 유니티에서 불러오지 못해 노출이 되지 않을 수 있습니다.

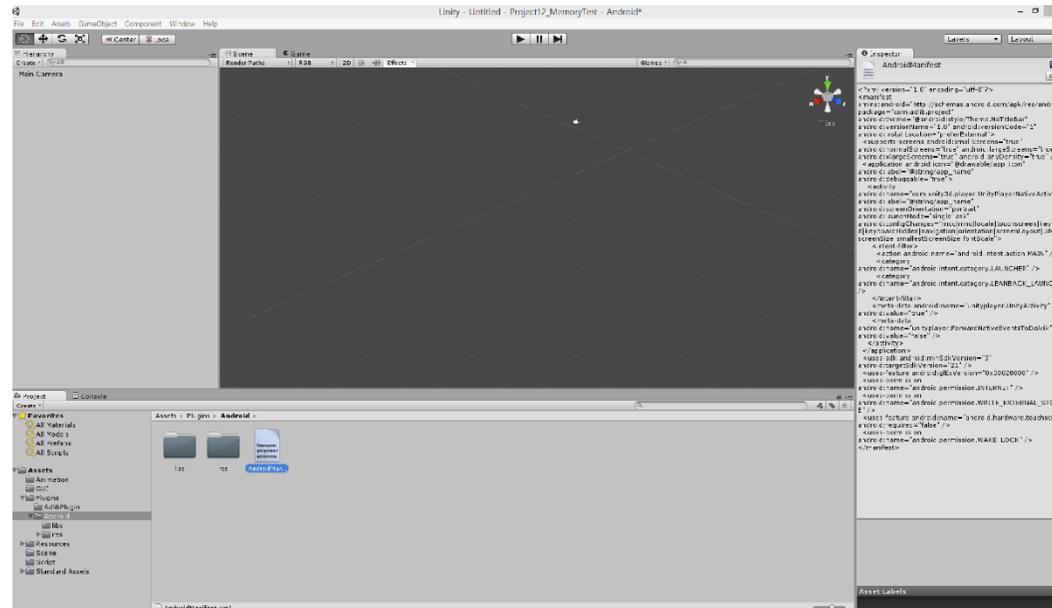
1. Assets > Import Package > Custom Package 메뉴를 클릭합니다.
2. AdlibUnityPlugin.unitypackage 파일을 선택하고, 모든 항목을 Import 합니다.
3. Assets/Plugins/AdlibPlugin/ 폴더에 있는 AdlibPlugin prefab 을 Unity scene으로 드래그 합니다.



1. 안드로이드 빌드 후,

ProjectName/Temp/StagingArea 폴더에 있는 AndroidManifest.xml 파일을 복사하여 Assets/Plugins/Android 폴더에 넣습니다.

2. Assets/Plugins/Android/AndroidManifest.xml 에 애드립 및 타 광고 플랫폼들 사용에 필요한 Activity와 퍼미션을 추가합니다.



- 아래의 권한을 추가하지 않을 경우 Exception이 발생합니다.
- 애드립 SDK 최소 버전은 9 입니다.(Android2.3이상)
- 타플랫폼의 최소버전이 애드립 보다 높다면 변경하시기 바랍니다.

```
<uses-sdk android:minSdkVersion="9" />
```

```
<!-- 애드립 실행에 필요한 권한 각 플랫폼 별로 요청하는 권한이 모두 다르기 때문에 아래 권한들을 모두 추가하는 것을 추천합니다. -->
```

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

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

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

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

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

```
<!-- 여기까지 애드립 사용을 위한 필수 권한 -->
```

```
<!-- 플랫폼에 따라 아래의 권한을 추가하시기 바랍니다. (애드립의 필수는 아닙니다.) -->
```

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

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

```
<meta-data android:name="unityplayer.ForwardNativeEventsToDalvik" android:value="true" />

<application
    android:icon="@drawable/icon"
    android:label="@string/app_name" >

<!-- 구글 광고 ID 사용을 위해 꼭 추가해주세요. -->
<meta-data android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />

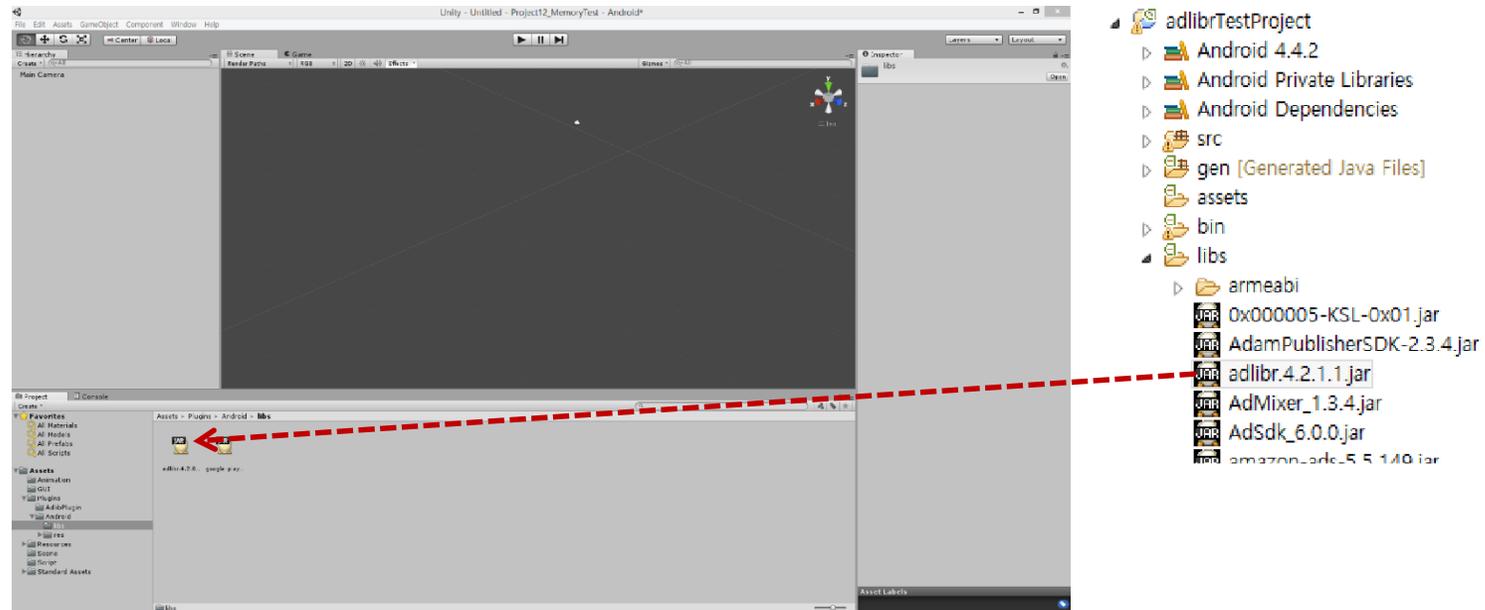
<!-- 애드립 사용을 위해 꼭 추가해주세요. -->
<activity
    android:name="com.mocoplex.adlib.AdlibDialogActivity"
    android:theme="@android:style/Theme.Translucent"
    android:configChanges="orientation|keyboard|keyboardHidden" />

<activity android:name="com.mocoplex.adlib.AdlibWebBrowserActivity"
    android:configChanges="orientation|keyboard|keyboardHidden" />

<activity android:name="com.mocoplex.adlib.AdlibVideoPlayer"
    android:theme="@android:style/Theme.NoTitleBar"
    android:configChanges="orientation|keyboard|keyboardHidden" />
</application>
```

- 광고 플랫폼 별 필수 Activity 들은 샘플 프로젝트의 /adlibUnity/AndroidManifest-sample/AndroidManifest.xml 및 각 플랫폼의 SDK문서를 참고해주시기 바랍니다.

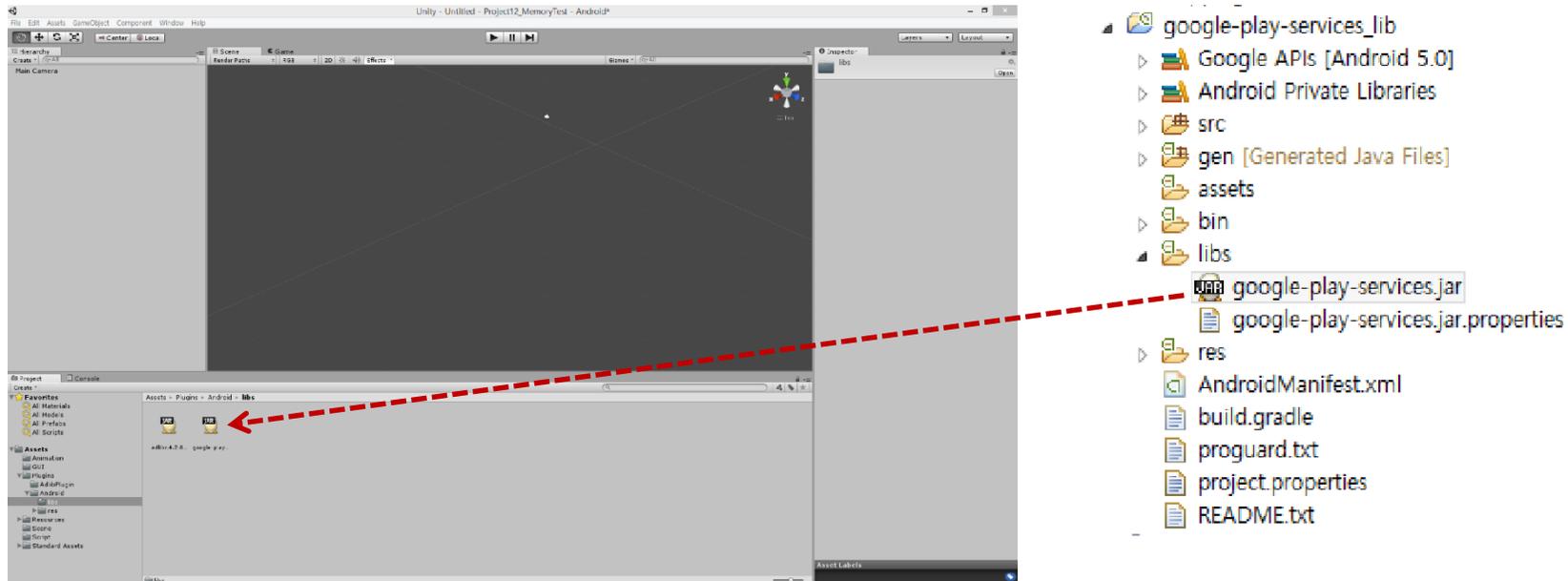
샘플 프로젝트에 포함되어 있는 SDK를 libs/adlibr.x.x.x.jar 를 찾아서 Assets/Plugins/Android/libs 폴더로 드래그하여 추가합니다.



Google Play Services 관련하여 라이브러리를 추가를 해야 합니다.

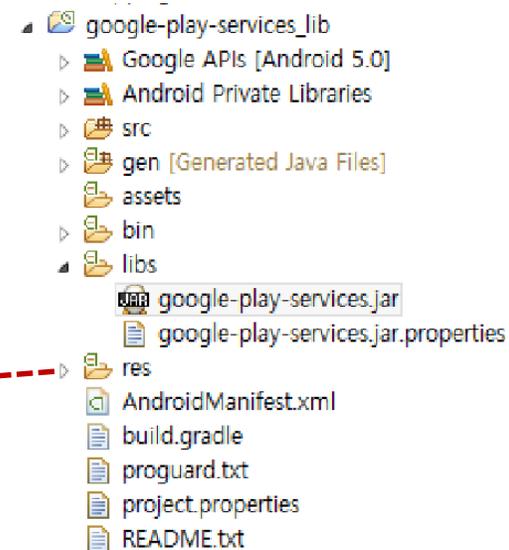
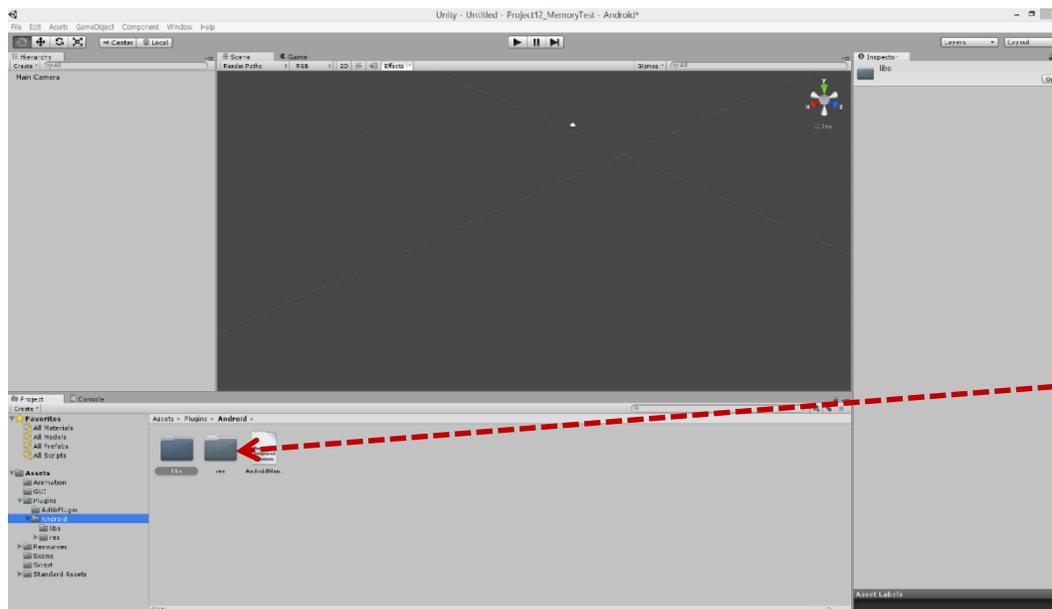
android-sdk/extras/google/google_play_services/libproject/google-play-services_lib/libs

위 경로의 google-play-services.jar 파일을 Assets/Plugins/Android/libs 폴더에 드래그하여 추가합니다.



Google Play Services 관련하여 리소스를 추가해야 합니다.

android-sdk/extras/google/google_play_services/libproject/google-play-services_lib
위 경로의 res 폴더를 Assets/Plugins/Android 폴더에 드래그하여 추가합니다.



애드립 외에 다른 플랫폼을 사용하실 경우

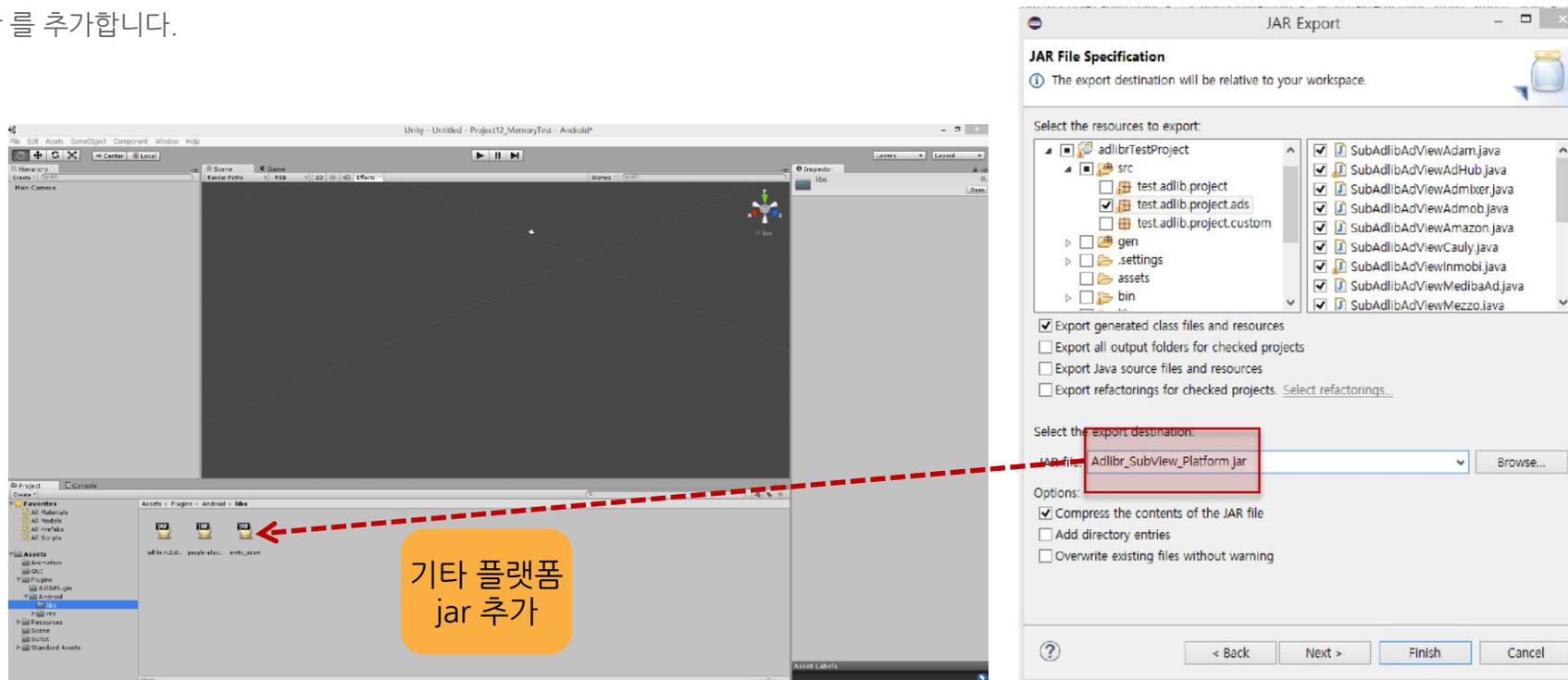
Assets/Plugins/Android/libs 폴더에 subview 관련 jar 파일을 넣습니다. (Adlibr_SubView_Platform.jar)

subview 는 샘플로 제공한 adlibrTestProject 에서 Export 하여 생성하시면 됩니다.

(이클립스에서 File > Export 클릭 후 Java의 JAR File 선택하여 원하는 폴더로 jar를 생성합니다.)

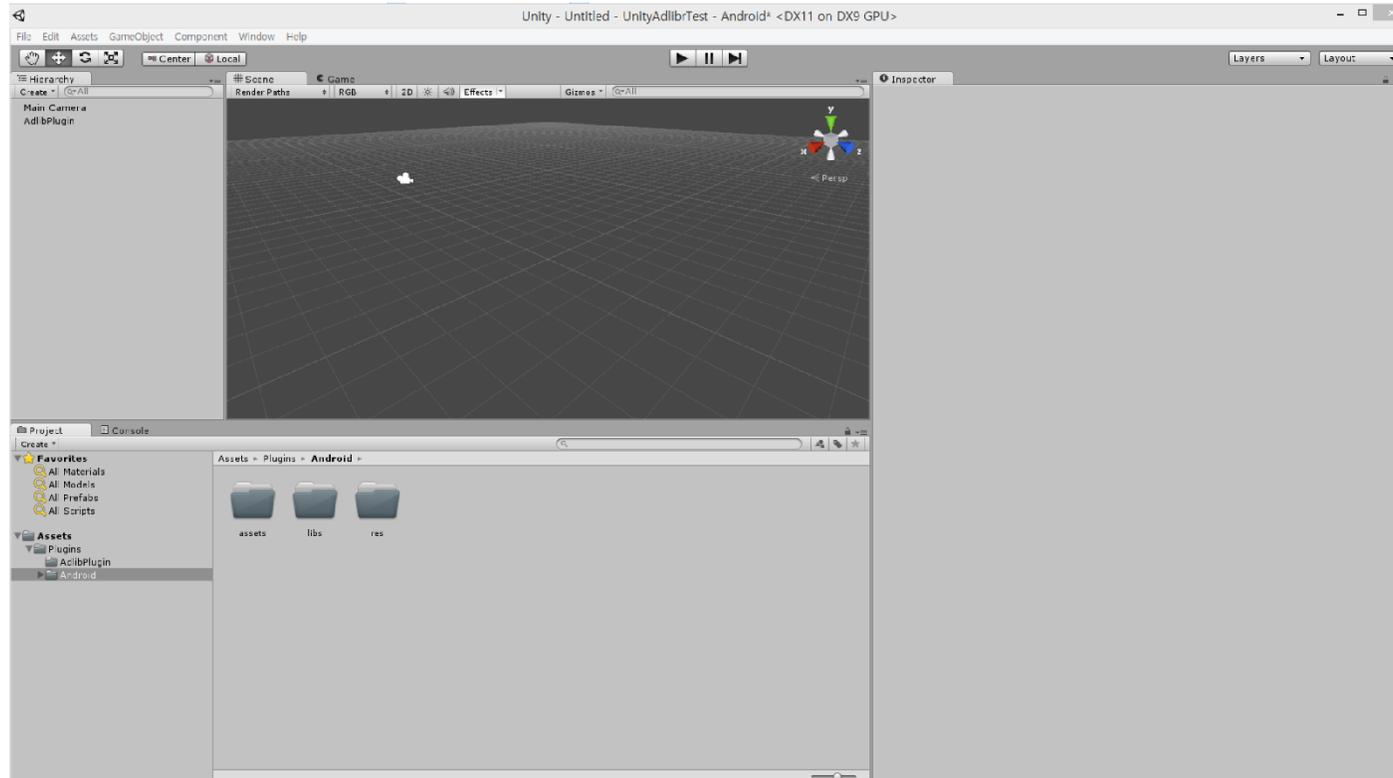
JAR 생성 이후 드래그하여 추가하시면 됩니다. (※ 주의 ※ 생성 과정에서 미리 플랫폼 API Key를 넣어두셔야 합니다.)

그리고 기타 플랫폼 jar 를 추가합니다.



광고 노출을 위한 이미지 리소스를 추가해야 합니다.

애드립 SDK (adlibr.4.2.1.1 이상) 내 **adlibr.4.2.1.x/assets** 의 assets 폴더를 Assets/Plugins/Android 폴더에 드래그하여 추가합니다.



InitializeAdlib (string adlibKey, string adsInfo)

애드립 사용에 필요한 플랫폼 별 KEY를 setting하는 메소드로, 최초 한번 호출합니다.

- adlibKey는 string 으로 애드립 홈페이지 통해 발급받으시는 키를 넣으시면 됩니다.
- adsInfo 는 JSON 포맷의 string입니다.
- adsInfo를 구성하기 위해, 애드립 Unity 플러그인에 기본적으로 JSON for .NET library 가 추가되어 있습니다.
- 샘플 스크립트대로 형태만 일치 시킨다면 다른 JSON library를 사용하셔도 괜찮습니다

adsInfo 포맷 예)	{“ADAM” : “subview 패키지명”, “ADMOb” : “subview 패키지명”, “CAULY” : “subview 패키지명”}
메소드 호출 예)	<pre>//사용하실 플랫폼을 넣어주시기 바랍니다. (기존의 각 플랫폼 Key는 subview의 jar파일에 넣어주셔야 합니다.) JsonObjectCollection ads = new JsonObjectCollection(); ads.Add(new JsonStringValue("ADAM", "test.adlib.project.ads.SubAdlibAdViewAdam")); ads.Add(new JsonStringValue("ADMOb", "test.adlib.project.ads.SubAdlibAdViewAdmob")); ads.Add(new JsonStringValue("CAULY", "test.adlib.project.ads.SubAdlibAdViewCauly")); ads.Add(new JsonStringValue("TAD", "test.adlib.project.ads.SubAdlibAdViewTAD")); ads.Add(new JsonStringValue("NAVER", "test.adlib.project.ads.SubAdlibAdViewNaverAdPost")); ads.Add(new JsonStringValue("SHALLWEAD", "test.adlib.project.ads.SubAdlibAdViewShallWeAd")); ads.Add(new JsonStringValue("INMOBI", "test.adlib.project.ads.SubAdlibAdViewInmobi")); ads.Add(new JsonStringValue("MMEDIA", "test.adlib.project.ads.SubAdlibAdViewMMedia")); ads.Add(new JsonStringValue("MOBCLIX", "test.adlib.project.ads.SubAdlibAdViewMobclix")); ads.Add(new JsonStringValue("UPLUSAD", "test.adlib.project.ads.SubAdlibAdViewUPlusAD")); ads.Add(new JsonStringValue("MEZZO", "test.adlib.project.ads.SubAdlibAdViewMezzo")); ads.Add(new JsonStringValue("AMAZON", "test.adlib.project.ads.SubAdlibAdViewAmazon")); ads.Add(new JsonStringValue("ADHUB", "test.adlib.project.ads.SubAdlibAdViewAdHub")); ads.Add(new JsonStringValue("MEDIBAAD", "test.adlib.project.ads.SubAdlibAdViewMedibaAd")); // 애드립 홈페이지에서 발급받으신 KEY를 입력해주시기 바랍니다. AdlibPlugin.InitializeAdlib(" 53858972e4b0ef94c0636d85 " , ads.ToString()); // 테스트 광고 노출을 위한 메서드로, 상용 시 주석처리 혹은 삭제 해주시기 바랍니다. AdlibPlugin.setAdlibTestMode(true);</pre>

ShowBanner(BannerSize size, bool useHouseBanner, bool positionAtTop, int padding, BannerAlign align)

띠배너를 화면에 노출시킵니다.

SIZE	배너 사이즈 설정. BannerSize.Banner (디바이스 width X 50dp 사이즈), BannerSize.Banner_320X50 (320dp X 50dp 사이즈)
useHouseBanner	true 이면 초기 광고 로딩까지의 공백에 하우스배너 노출, false이면 하우스배너 노출하지 않음.
positionAtTop	true 이면 화면 상단, false이면 화면 하단에 위치.
padding	(optional) 상단 또는 하단으로 부터 padding(dp) 만큼 떨어진 곳에 위치. 기본값 0.
align	(optional) 배너 horizontal 정렬 설정. 기본값 Center BannerAlign.Left (좌측정렬) BannerAlign.Right (우측정렬) BannerAlign.Center (가운데정렬)
메소드 호출 예)	AdlibPlugin.ShowBanner(AdlibPlugin.BannerSize.Banner, false, false); AdlibPlugin.ShowBanner(AdlibPlugin.BannerSize.Banner_320x50, false, false, 50); AdlibPlugin.ShowBanner(AdlibPlugin.BannerSize.Banner_320x50, true, true, 0, AdlibPlugin.BannerAlign.Left);

HideBanner()

화면에 노출된 띠배너를 숨깁니다.

메소드 호출 예)	AdlibPlugin.HideBanner();
-----------	---------------------------

ShowBannerWithPixel (BannerSize size, bool useHouseBanner, bool positionAtTop, int padding, BannerAlign align)

띠배너를 화면에 노출시킵니다.

SIZE	배너 사이즈 설정. BannerSize.Banner (디바이스 width X 50dp 사이즈), BannerSize.Banner_320X50 (320dp X 50dp 사이즈)
useHouseBanner	true 이면 초기 광고 로딩까지의 공백에 하우스배너 노출, false이면 하우스배너 노출하지 않음.
positionAtTop	true 이면 화면 상단, false이면 화면 하단에 위치.
padding	(optional) 상단 또는 하단으로 부터 padding(pixel) 만큼 떨어진 곳에 위치. 기본값 0.
align	(optional) 배너 horizontal 정렬 설정. 기본값 Center BannerAlign.Left (좌측정렬) BannerAlign.Right (우측정렬) BannerAlign.Center (가운데정렬)
메소드 호출 예)	AdlibPlugin.ShowBanner(AdlibPlugin.BannerSize.Banner, false, false); AdlibPlugin.ShowBanner(AdlibPlugin.BannerSize.Banner_320x50, false, false, 50); AdlibPlugin.ShowBanner(AdlibPlugin.BannerSize.Banner_320x50, true, true, 0, AdlibPlugin.BannerAlign.Left);

LoadInterstitialAd()

전면배너를 호출합니다.

위 메소드는 1차적으로 애드립의 전면배너를 호출합니다.

애드립의 전면배너 수신이 실패했을 경우, 대쉬보드의 전면배너 스케줄 설정된 순서대로 전면배너 수신이 성공할 때 까지, 타 플랫폼의 전면배너를 호출합니다.

메소드 호출 예)	AdlibPlugin.LoadInterstitialAd();
-----------	-----------------------------------

RequestInterstitial()

전면배너를 미리 요청하여 로드하는 경우 사용합니다.

성공이후 SDK에서 주기적으로 재요청을 하며, 실패시 요청을 중단합니다.

(애드립 플랫폼만 사용 가능합니다.)

메소드 호출 예)	<code>AdlibPlugin.RequestInterstitial();</code>
-----------	---

ShowInterstitial()

RequestInterstitial() 통해 성공한 경우 화면에 노출시킵니다. (애드립 플랫폼만 사용 가능합니다.)

메소드 호출 예)	<code>AdlibPlugin.ShowInterstitial();</code>
-----------	--

전면배너 리스너

▶ 애드립 Unity 플러그인은 다음의 리스너 이벤트를 제공합니다.

```
// loadFullInterstitialAd
public static event Action<string> ReceivedInterstitial // 전면배너 수신 성공 - string은 수신성공한 플랫폼 이름
public static event Action<string> FailedInterstitial // 전면배너 수신 실패 - string은 수신실패한 플랫폼 이름
public static event Action FailedToReceiveInterstitial // 스케줄링된 광고 플랫폼의 모든 전면배너 수신 실패
public static event Action ClosedInterstitial // 전면배너 닫힘

// showInterstitial
public static event Action ReceivedPreInterstitial // 전면배너 프리로드 수신 성공
public static event Action FailedPreInterstitial // 전면배너 프리로드 수신 실패
public static event Action ClosedPreInterstitial // 전면배너 닫힘
```

▶ += 연산자를 이용해서 아래와 같이 이벤트를 등록할 수 있습니다.

```
// 리스너 응답을 받을 메소드명을 이벤트에 등록
AdlibPlugin.ReceivedInterstitial += HandleReceivedInterstitial;

// 리스너 응답후 호출부 구현
public void HandleReceivedInterstitial(string platform) {
    print("Received Interstitial Ad : ");
    print(platform);
}
```

StopAds()

광고클릭, 홈버튼 클릭 등으로 앱이 pause 상태가 되었을 때 애드립 서비스 호출을 중지합니다.

메소드 호출 예)	AdlibPlugin.StopAds();
-----------	------------------------

RestartAds()

앱이 pause 상태였다가 다시 돌아왔을 때 애드립 서비스를 재개 합니다.

메소드 호출 예)	AdlibPlugin.RestartAds();
-----------	---------------------------