

玉米移动广告 Android Unity3D SDK

接入说明文档

V3.2.2

目录

1.概述	3
1.1.面向读者	3
1.2.开发环境要求	3
2.SDK 下载	3
2.1.SDK 下载	3
2.2.三方 SDK 下载	4
3.接入步骤	4
3.1.添加资源文件	4
3.2.快速接入	6
3.2.1 环境配置	6
3.2.2 横幅广告形式接入	7
3.2.3 插屏广告形式接入	10
3.2.4 激励视频广告形式接入	11
3.3 是否 GOOGLEPLAY 发布版本	13
3.4. LOG	13
3.5.调试模式	14
4.混淆	17
5.编译	18

1.概述

1.1.面向读者

本产品面向需要在 Unity 产品中接入玉米移动广告 Unity3D SDK 的开发人员

1.2.开发环境要求

OS : Windows、Mac、Linux

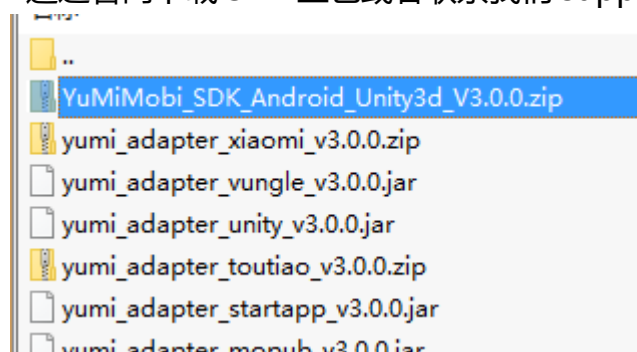
Android SDK : 2.1 及以上

IDE : Unity64-5.2.3f1

2.SDK 下载

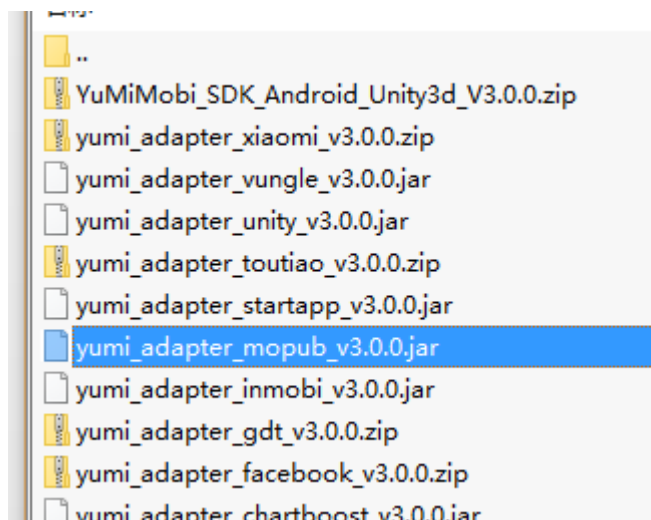
2.1.SDK 下载

通过官网下载 SDK 主包或者联系我们 support@yumimobi.com



2.2.三方 SDK 下载

在玉米 SDK jar 包中选择需要接入的三方平台（三方平台的 jar 包可以通过名称进行区分），接入工程中



3.接入步骤

3.1.添加资源文件

1、添加 SDK 需要的资源文件：

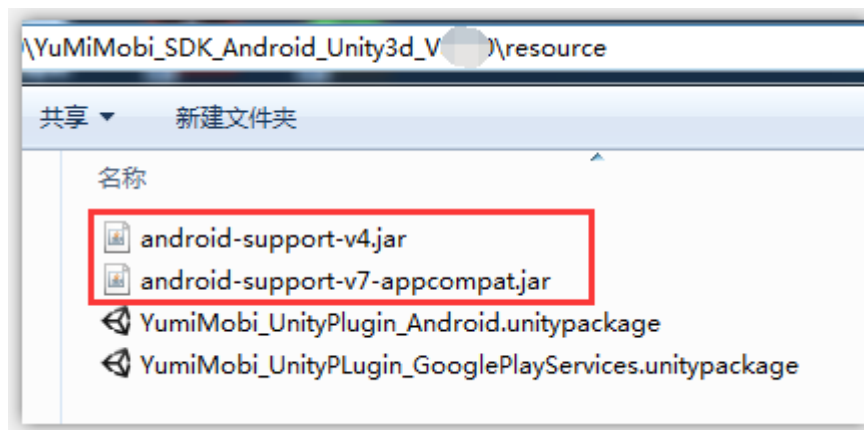
（1）将 **resource\YumiMobi_UnityPlugin_Android.unitypackage** 导入到工程中。如果资源文件跟项目中现有资源文件有冲突，请尽量使用我们提供的资源文件，若存在其他问题可联系我们 support@yumimobi.com，我们会尽快反馈。资源文件不添加可能导致广告无法正常展示。



(2) 当应用需发布到 googlePlay 平台时, 若 Assest/plugins/Android 中没有关于 googlePlayServices 相关 Jar 或者 aar 文件时, 请导入 YumiMobi_UnityPLugin_GooglePlayServices.unitypackage



(3) 当 Assest/plugins/Android 没有 support-v7 或者 support-v4 相关 jar 或者 aar 文件时, 请把以下文件拷贝到../Assest/plugins/Android 文件夹中



2、增加三方平台 SDK :三方平台 adapter(yumi_adapter_*****_v*.*.jar) 在 adapter 文件夹下, 如需接入三方平台, 可添加到\Assets\plugins\Android 下。

3、AndroidManifest.xml 中权限增加 :

(1) 可选权限 : 可视需求添加到您的 Assets\plugins\Android\AndroidManifest.xml 文件中。若添加以下权限, 有助提高广告填充

```
<!-- yumi sdk start -->
<uses-permission android : name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android : name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android : name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION" />
```

```
<uses-permission android:name="android.permission.CALL_PHONE"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.WRITE_CALENDAR"/>
<!-- yumi sdk end -->
```

(2) 动态获取权限

当您的应用 targetSdkVersion 为 23 及以上时,调用以下方法进行权限检查并且弹窗提示用户授权。

该方法在实例化广告之前调用,并且需要添加 android-support-v4.jar。

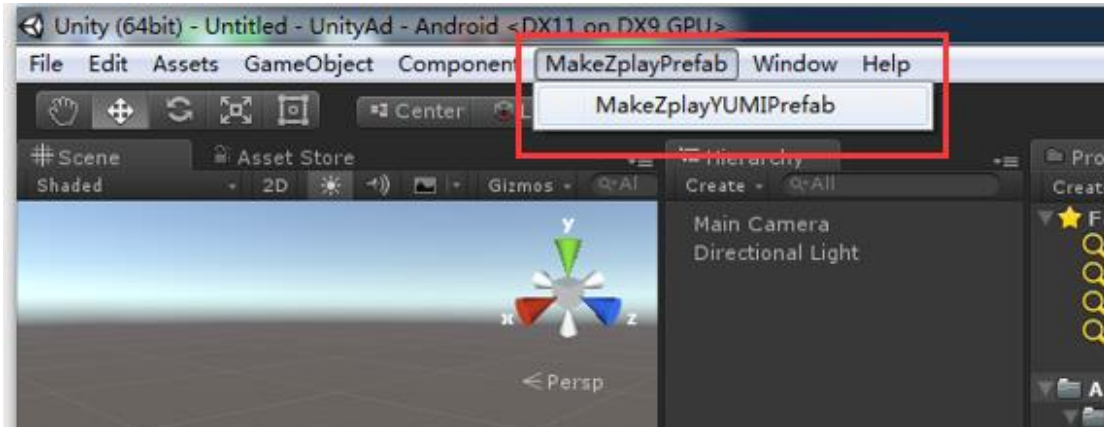
```
YumiUnityAdUtils.CheckPermission();
```

注:三方平台注册组件,请查看文档《YumiMobi 聚合广告 SDK-三方支持列表》

3.2.快速接入

3.2.1 环境配置

1. 点击 MakeZplayPrefab/MakeZplayYUMIPrefab 生成 ZplayYUMIHelper prefab 到 Hierarchy, 此 prefab 会全程跟随所有场景



2. 上述步骤完成之后,在 ZplayYUMIHelper 类内设置您的广告位 ID、渠道号、版本号(渠道号和版本号为非必选项)。

注: ChannelID是指应用发布的渠道标示,填写后YUMI平台可根据渠道ID进行数据统计和效果分析;以Popstar!消灭星星官网正版为例,当游戏发布到三星渠道时,需要将setChannelID(channelStr)设置为

setChannelID('SamSung')。

渠道标示为YUMI平台生成信息,不可随意修改;

渠道名称	ChannelID
三星	SamSung

```
ZplayYUMIHelper.cs
ZplayYUMIHelper
1 using System.Collections;
2 using UnityEngine;
3
4 public class ZplayYUMIHelper : MonoBehaviour
5 {
6     public static ZplayYUMIHelper Instance;
7
8     //----- yu mi slot id -----//
9     private string banner_slotID = "";
10    private string interstitial_slotID = "";
11    private string media_slotID = "";
12    //-----channel id -----//
13    private const string channelId = "";
14    //-----version id -----//
15    private const string versionId = "";
16
```

3. ZplayYUMIHelper 类 Start () 方法体内有以下代码

```
void Start () {
    //Remind users to get permission
    yuMiUnityAD.CheckPermission();
    //Init video
    InitMedia();
    //Init Interstitial
    InitInterstitialAd();
    //Whether to open the log information (false:close, true: open)
    Logger.SetDebug(true);
    //After the initial video screen is completed, wait for 0.2 seconds to start requesting the video and the interstitial
    StartCoroutine(Reques(0.2f));
}
```

initMedia() //为初始化视频，如不需要视频则注释掉

InitInterstitialAD() //为初始化插屏。如不需要插屏则注释掉

ZplayLogger.setDebug(false) //为 log 信息类。传 true 显示日志，false 不显示日志

3.2.2 横幅广告形式接入

1、必须接入方法：此方法直接影响广告是否可正常展示

1). 调用以下方法展示横幅：

```
ZplayYUMIHelper.Instance.ShowBannel();
```

2). 调用以下方法隐藏横幅：

```
ZplayYUMIHelper.Instance.DismissBanner();
```

3). 调用以下方法恢复展示横幅：

```
ZplayYUMIHelper.Instance.ResumeBanner();
```

2、可选接入方法：回调方法可帮助开发者获取 SDK 状态；自适应可提升广告展示效果

1) 横幅广告回调方法：开发者可根据回调方法获悉 SDK 状态，并根据自身应用逻辑需要做在此基础上做相关处理。若不接入此方法，不影响广告的展示上报等效果。

横幅广告回调类 BannerAdCallbackListener 实现了 YumiUnityAdUtils.BannerAdCallbackListener 接口，有以下方法：

onBannerPreparedFailed(string data)	当横幅加载失败时回调，可以通过 data 获得失败原因。
onBannerPrepared(string data)	当横幅加载成功时回调
onBannerExposure(string data)	当横幅展示成功时回调
onBannerClosed(string data)	当横幅关闭时回调
onBannerClicked(string data)	当横幅点击时回调

接入示例：

```
#region Banner Callback
//横幅加载失败回调方法
public void onBannerPreparedFailed(string data)
{
    ZplayLogger.Log("onBannerPreparedFailed:" + data);
}
//横幅加载成功回调方法
public void onBannerPrepared(string data)
{
    ZplayLogger.Log("onBannerPrepared:" + data);
}
```

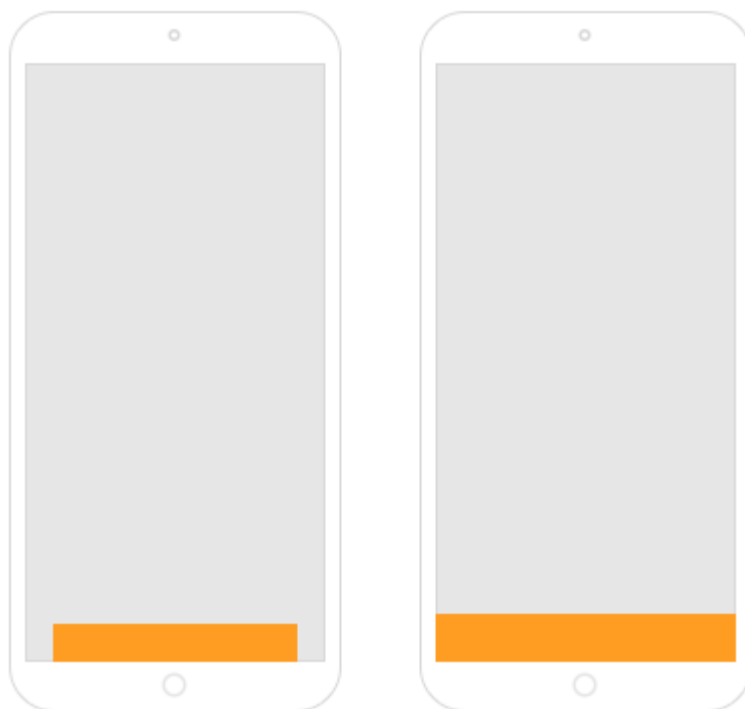


```
//横幅展示成功回调方法
public void onBannerExposure(string data)
{
    ZplayLogger.Log("onBannerExposure:" + data);
}
//横幅关闭事件回调方法
public void onBannerClosed(string data)
{
    ZplayLogger.Log("onBannerClosed:" + data);
}
//横幅点击事件回调方法
public void onBannerClicked(string data)
{
    ZplayLogger.Log("onBannerClicked:" + data);
}
#endregion
```

2) 横幅自适应屏幕宽度： 可根据需要修改 isMatchWindowWidth 参数状态，建议使用默认 false 状态，提高广告展示效果

非自适应

自适应



横幅设置广告容器的同时，玉米 SDK 提供了 boolean 类型的参数：isMatchWindowWidth。可修改 boolean 类该参数为横幅宽度是否充满全屏，当该参数为 false 时横幅的宽度为容器宽度；当该参数为 true 时横幅的宽度为屏幕宽度。默认状态为 false

```
ZplayYUMIHelper.Instance.ShowBannel(true);
```

3.2.3 插屏广告形式接入

1、必须接入方法：此方法直接影响广告是否可正常展示

1) 调用以下方法展示插屏：

```
ZplayYUMIHelper.Instance.ShowInterstitialAD();
```

2、可选接入方法：接入以下方法可帮助开发者获取 SDK 状态

2) 插屏广告回调方法：开发者可根据回调方法获悉 SDK 状态，并根据自身应用逻辑需要做在此基础上做相关处理。若不接入此方法，不影响广告的展示上报等效果。

插屏广告回调类 `InterstitialAdCallbackListener` 实现了 `YumiUnityAdUtils.InterstitialAdCallbackListener` 接口，有以下方法：

<code>onInterstitialPreparedFailed(string data)</code>	当插屏加载失败时回调，可以通过 data 获得失败原因。
<code>onInterstitialPrepared(string data)</code>	当插屏加载成功时回调。
<code>onInterstitialExposure(string data)</code>	当插屏展示成功时回调
<code>onInterstitialClosed(string data)</code>	当插屏关闭时回调
<code>onInterstitialClicked(string data)</code>	当插屏点击时回调

接入示例：

```
#region InterstitialAd Callback
//插屏加载失败回调方法
public void onInterstitialPreparedFailed(string data)
{
    ZplayLogger.Log("onInterstitialPreparedFailed:" + data);
}
//插屏加载成功回调方法
public void onInterstitialPrepared(string data)
{

```

```

        ZplayLogger.Log("onInterstitialPrepared:" + data);
    }
    //插屏展示成功回调方法
    public void onInterstitialExposure(string data)
    {
        ZplayLogger.Log("onInterstitialExposure:" + data);
    }
    //插屏关闭事件回调方法
    public void onInterstitialClosed(string data)
    {
        ZplayLogger.Log("onInterstitialClosed:" + data);
    }
    //插屏点击事件回调方法
    public void onInterstitialClicked(string data)
    {
        ZplayLogger.Log("onInterstitialClicked:" + data);
    }
}
#endregion

```

3.2.4 激励视频广告形式接入

1、必须接入方法：以下方法直接影响广告是否可正常展示

1) 调用以下方法询问激励视频是否加载完成：

```
ZplayYUMIHelper.Instance.IsMediaPrepared();
```

//此加载完成函数调用逻辑已经提供在 ZplayYUMIHelperupdate 类 update 函数中，详情如下：

```

void Update () {
    //Ask about video every 5 seconds is prepared
    if (!rotaIsMediaPrepared)
    {
        timeNum += Time.fixedDeltaTime;
        if (timeNum > 5f)
        {
            timeNum = 0;
            IsMediaPrepared();
        }
    }
}

```

调用视频是否加载完成函数 IsMediaPrepared()，建议每五秒请求一次。加载成功返回 true，未成功返回 false。加载成功后 GetRotaIsMediaPrepared 设置

为 true，暂停循环调用 IsMediaPrepared()函数。

```
/// <summary>
/// Video Is Prepared
/// </summary>
1 个引用
public void IsMediaPrepared()
{
    bool isPrepared = yuMiUnityAD.IsMediaPrepared();
    if (isPrepared)
    {
        //The video load completes the shutdown of the rotation request video logic
        GetRotaIsMediaPrepared = true;
        Logger.LogError("yumiMobi SDK Media IsMediaPrepared : true");
    }
    else
    {
        Logger.LogError("yumiMobi SDK Media IsMediaPrepared : false");
        //The video screen does not have 5 seconds to request a screen for loading
    }
}
```

在视频广告关闭回调中再次开启轮询加载视频

```
public void onMediaClosed(string data)
{
    Logger.LogError("yumiMobi SDK Media Closed : " + data);
    //Once the video screen is successful, it will be retrained to see if the video screen is loaded
    ZplayYUMIHelper.Instance.GetRotaIsMediaPrepared = false;
}
```

2)调用以下方法展示激励视频：

```
ZplayYUMIHelper.Instance.ShowMedia();
```

3)激励视频广告回调方法：

激励视频广告回调类 MediaAdCallbackListener 实现了 Assets.YumiUnityAdUtils.MediaAdCallbackListener 接口，有以下方法：

onMediaExposure(string data)	当激励视频展示成功时回调
onMediaClosed(string data)	当激励视频关闭时回调
onMediaClicked(string data)	当激励视频点击时回调。注意：根据平台不同，此方法不保证 100%回调
onMediaIncentived(string data)	当激励视频播放完成，可以获取奖励时回调。 注意：如果未完成播放，不会回调此方法。另外，此方法永远在 onMediaClosed ()之后触发。

接入示例：

```
#region MediaAd Callback
// 激励视频奖励回调方法
public void onMediaIncentived(string data)
{
    ZplayLogger.Log("onMediaIncentived:" + data);
}
// 激励视频展示完成回调方法
public void onMediaExposure(string data)
{
    ZplayLogger.Log("onMediaExposure:" + data);
}
// 激励视频关闭事件回调方法
public void onMediaClosed(string data)
{
    ZplayLogger.Log("onMediaClosed:" + data);
}
// 激励视频点击事件回调方法
public void onMediaClicked (string data)
{
    ZplayLogger.Log("onMediaClicked:" + data);
}
#endregion
```

3.3 是否 GooglePlay 发布版本

注：如果您的 APP 是在 GooglePlay 发布，请设置以下方法。

```
ZplayYUMIHelper.Instance.SetApplsGooglePlayVersions();
```

3.4. Log

以下是插件提供的 Android Log 输出方法，开发者可自行判断是否使用。
增加此方法可方便查找问题：

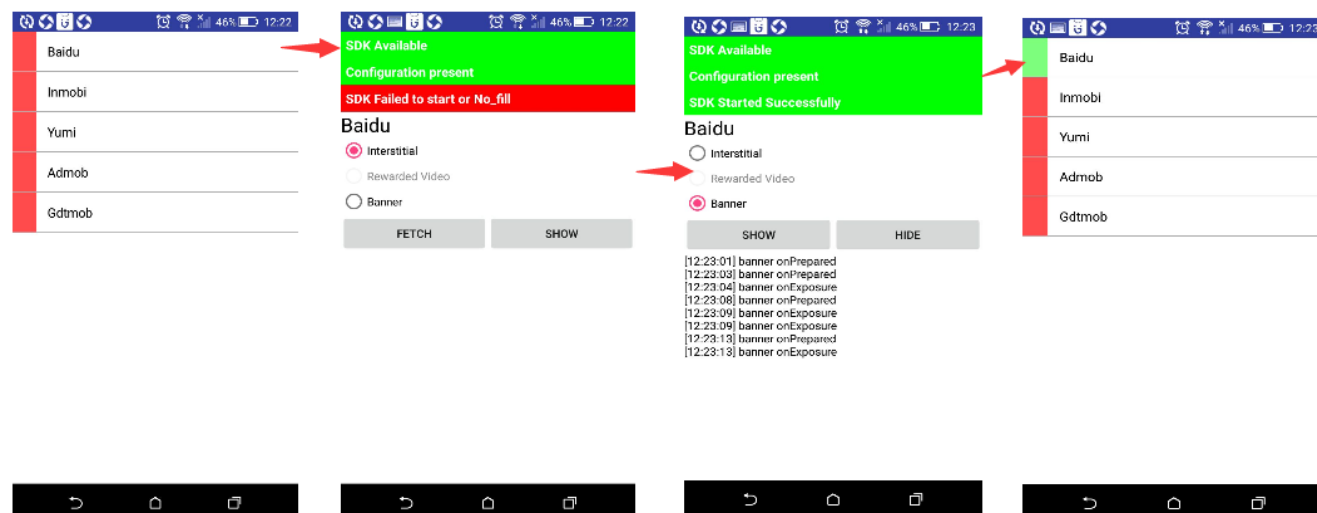
```
ZplayLogger.Log("*****");
```

3.5.调试模式

玉米广告 SDK 为开发者提供了一个检测三方平台集成状态的调试模式，如

图

:



可以通过调用以下方法启动调试模式页面。

```
ZplayYUMIHelper.Instance.StartDebugging();
```

使用步骤：

1、调用方法打开调试页面：

```
ZplayYUMIHelper.Instance.StartDebugging();
```

2、玉米 SDK 将获取配置并展示三方平台列表，进入 debug 页面：

1) 页面展示为 Searching for third party ADnetwork adapters：表示未进行配置，请检查应用的不同广告形式配置情况，如果问题仍未解决，请通过邮件联系我们：support@yumimobi.com

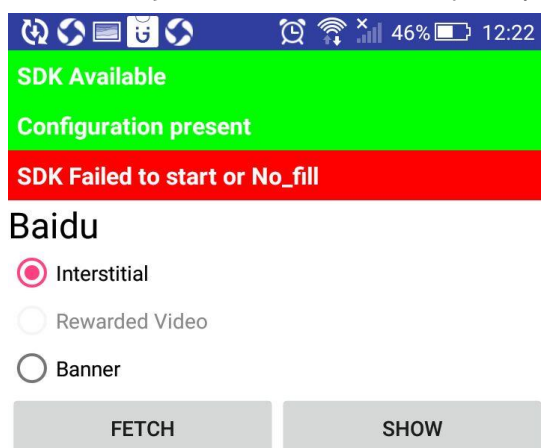


2) 广告配置后，正常展示配置平台，首次进入左侧所有平台均为红色，当某个平台正确接入并展示成功后，左侧状态为绿色。

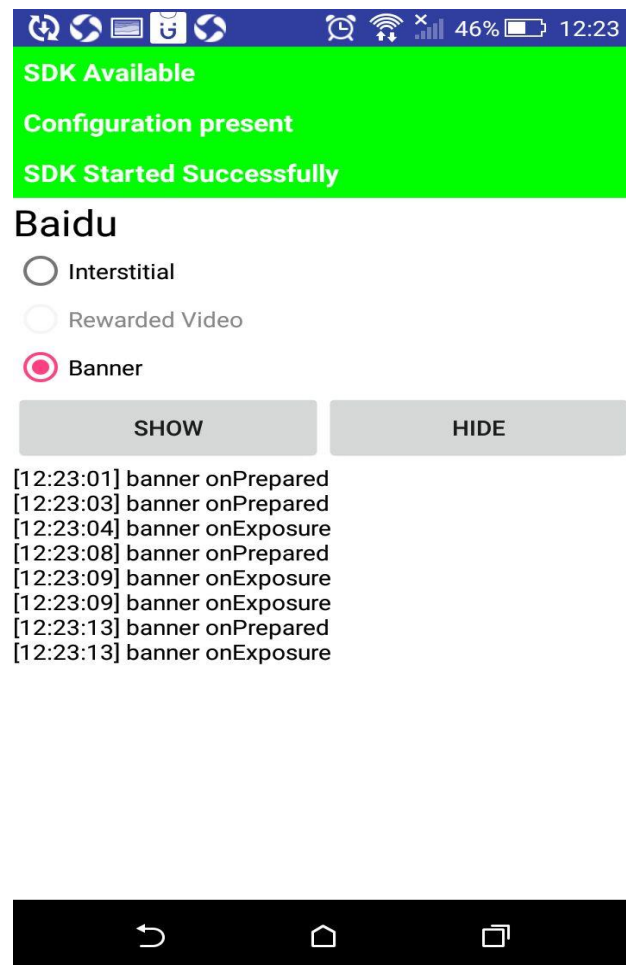


3、无论左侧状态栏颜色为何状态，均可选择一家平台点击进入：

- 1) SDK Available 为绿色时表示三方平台适配器已添加；当为红色时表示三方平台适配器未添加，回到 3.1-2 检查该平台 adapter 是否添加
- 2) Configuration present 为绿色时表示三方平台适配器组件 Manifest 已注册；当为红色时表示三方平台适配器组件 Manifest 未注册，可回到 3.1-3 检查该平台适配器组件是否添加
- 3) SDK Failed to start or No_fill 为绿色表示广告曾经展示成功过；当为红色时表示还未展示成功过广告，可继续进行下一个步骤，如果所有步骤完成后仍未红色，请邮件联系我们 support@yumimobi.com



- 4、点击 Fetch 开始请求广告，点击 Show 展示广告
- 5、广告展示成功后检查项全部变为绿色，表示该平台接入成功



6、应用发布前需要将调试模式注释掉。

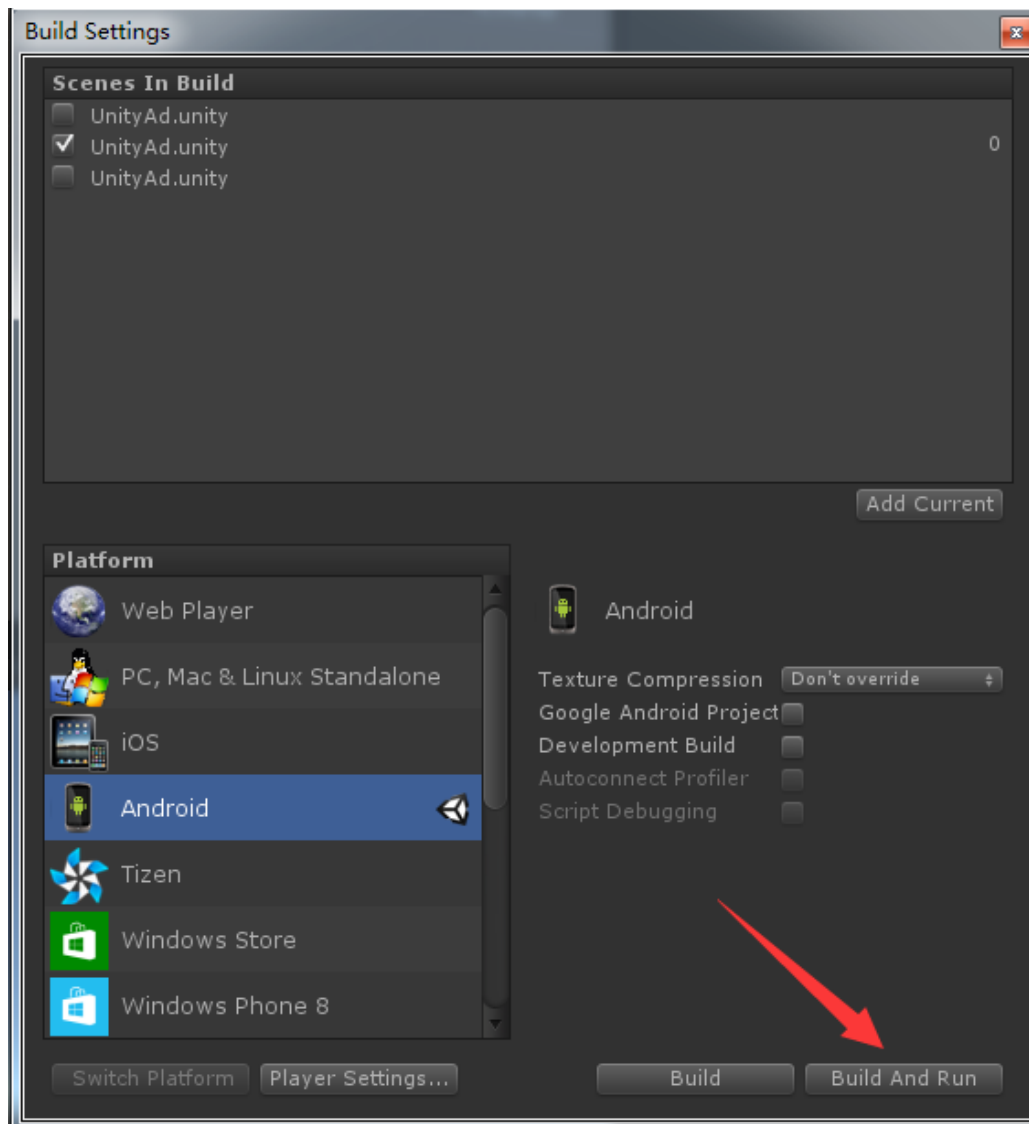
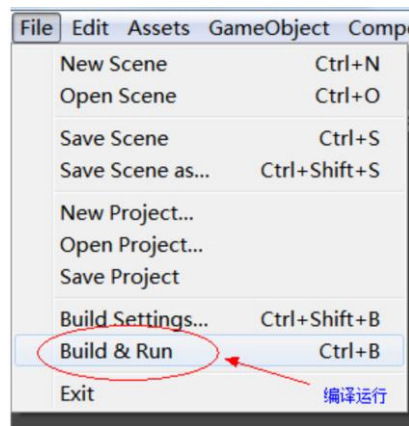
4.混淆

如果您的工程需要混淆编译，请在混淆文件内增加以下内容。

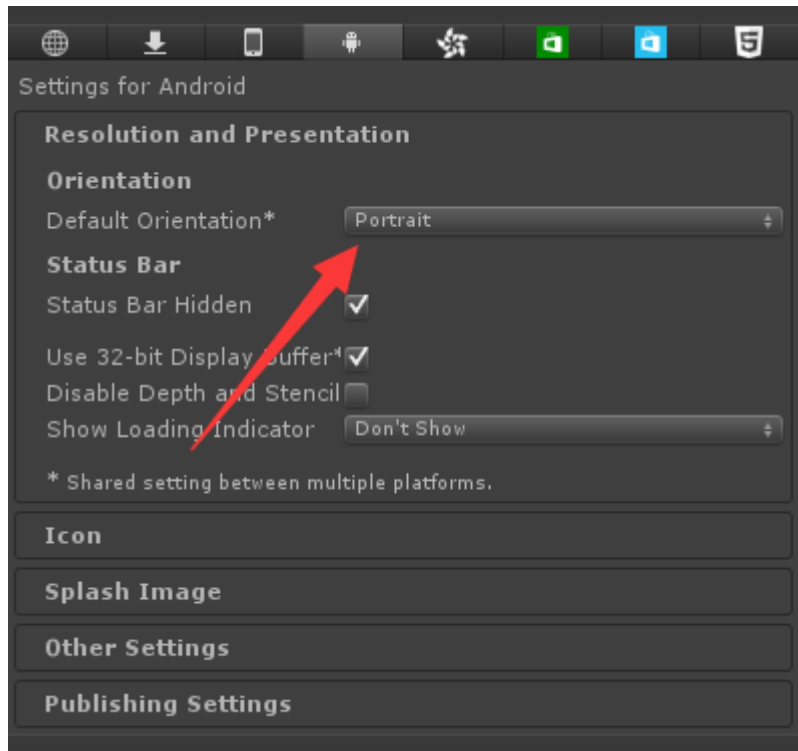
```
-keepattributes Exceptions , InnerClasses , Signature , Deprecated ,  
SourceFile ,LineNumberTable ,*Annotation* ,Synthetic ,EnclosingMethod  
-keep class com.yumi.android.sdk.ads.** { *;}  
-keep class com.yumi.android.sdk.ads.self.**{*;}  
-keep class com.yumi.android.sdk.ads.selfmedia.**{*;}
```

5.编译

1、点击 File , 选择 Build&Run 进行编译。



2、请注意设定 Unity3D 项目中的屏幕方向和 Android 项目中 AndroidManifest.xml 中所设置的保持一致，否则启动 app 后旋转屏幕可能会导致崩溃。（如下图）



点击【Build And Run】等待进度条完成

