目录



# 修改历史

| 时间 | 版本 | 修改内容 |
|---|---|---|
| 2022/1/17 | 3.3.3.0 | 广告SDK增加对开屏广告和原生广告的支持 |
| 2022/1/21 | 3.3.3.2 | 支付SDK升级aries版本到1.5.1.5 |
| 2022/1/21 | 3.3.3.3 | 修复直接支付不接广告引起的异常 |
| 2022/2/10 | 3.3.3.4 | 沙盒环境不使用云控广告位，使用game_sdk_config.json中配置的广告位 |
| 2022/03/09 | 3.7.0.0 | 广告SDK修改为只接入Hisavana渠道 |
| 2022/03/09 | 3.8.0.0 | 优化浮窗广告展示以及修正游戏启动来源的自埋点 |
| 2022/04/01 | 3.9.0.0 | 计入Hisavana 1.3版本，并调整预加载逻辑 |
| 2022/04/01 | 4.0.0.0 | 服务器重构，合入最新浮窗优化。 |

# 一、游戏广告SDK接入文档

## 1．首先申请app对应的配置文件

向运营同事申请配置文件game_sdk_config.json，这个文件放到app的assets 目录（"\src\main\assets"）

## 2．添加SDK依赖

- project的build.gradle

```
allprojects {
    repositories {
        maven { url 'https://mvn.shalltry.com/repository/maven-public/' }
        maven { url 'https://mvn.shalltry.com/repository/game-releases/' }
    }
}
```

- module的build.gradle

```
dependencies {
   implementation"com.transsion.game:ad:4.0.0.0"
}
```

## 3．配置google广告ID

在AndroidManifest.xml中配置google admob的应用ID（格式如: “ca-app-pub-3940256099942544~3347511713”），这个值由运营同事申请给出。

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <application>
        <!--广告SDK添加开始,注意此处的，此处的appid需要更改-->
        <meta-data
        android:name="com.google.android.gms.ads.APPLICATION_ID"
        android:value="ca-app-pub-3940256099942544~3347511713"
        tools:replace="android:value"/>
        <!--广告SDK添加结束-->
    </application>
</manifest>
```

## 4． SDK初始化

```
public class App extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        CoreUtil.init(this);
        AdInitializer.init(
                new AdInitializer.Builder(this)
                        //开启debuggable，设置测试环境，展示测试广告
                        //请在正式上线环境下，设为false,并且setEnv接口传参数为release
                        .setDebuggable(true).setEnv("test")
                        //开启debug，展示的是admob的测试服务器广告
                        //如果你需要调试线上真实广告，你需要把debuggable设为false
                        //并输入你的设备ID，如何获取设备ID见 第6点广告测试的问题QA
                        //请在正式上线的环境下，删除本行代码
                        .setTestDeviceIds(Collections.singletonList("7FC2C0BE39C47406C984C08C16418C5C"))
                        //广告开关，不传默认开启
                        .setTotalSwitch(true)
        );

        //开屏广告预加载，如果需要展示开屏广告，必需在Application的初始化中调用广告的初始化，并同时预加载开屏广告
        //如下第一个参数5，表示如果5秒内开屏广告准备好，则自动展示开屏广告，否则不会展示开屏广告,下次热启动会再次展示
        //有闪屏或启动动画的游戏可以设置等待时间稍长，视闪屏和动画时长而定
        AdHelper.showAppOpen(5, new GameAdLoadListener() {
            @Override
            public void onAdLoaded() {

            }

            @Override
            public void onAdFailedToLoad(int code, String message) {

            }
        });
    }
}
```

## 5．广告预加载

展示广告之前需要进行预加载，以便在广告需要时快速展示，可以选择在Activity或者Fragment创建时执行，或者在您认为的合适时机进行预加载。

每种类型广告的预加载只需要执行一次，SDK会自动调度后续的预加载。

禁止在预加载成功的回调里面，调用广告的展示函数。

### (1) 预加载插屏广告

```
AdHelper.loadInterstitial(this, new GameAdLoadListener()
{
    @Override
    public void onAdFailedToLoad(int code, String message) {
        Log.i(TAG, "Interstitial onAdFailedToLoad " + code + " " + message);
    }
    @Override
    public void onAdLoaded() {
        Log.i(TAG, "Interstitial onAdLoaded");
```

```
    }
});
```

### （2）预加载激励视频广告

```
AdHelper.loadReward(this, new GameAdLoadListener()
{
    @Override
    public void onAdFailedToLoad(int code, String message) {
        Log.i(TAG, "Reward onRewardedAdFailedToLoad " + code + " " + message);
    }

    @Override
    public void onAdLoaded() {
        Log.i(TAG, "Reward onRewardedAdLoaded");
    }
});
```

### （3）banner广告没有预加载

### （4）开屏广告的预加载

参考上面SDK初始化

### （5）预载原生广告

```
AdHelper.loadNative(new GameAdLoadListener() {
    @Override
    public void onAdLoaded() {
        Log.i(TAG, "Native onAdLoaded");
    }

    @Override
    public void onAdFailedToLoad(int code, String message) {
        Log.i(TAG, "Native onAdFailedToLoad " + code + " " + message);
    }
});
```

### (6) 预加载悬浮(Float)广告

```
AdHelper.loadFloat(this, new GameAdLoadListener() {

    @Override
    public void onAdFailedToLoad(int code, String message) {
        Log.i(TAG, "Float onAdFailedToLoad " + code + " " + message);
    }

    @Override
    public void onAdLoaded() {
        Log.i(TAG, "Float onAdLoaded");
    }
});
```

# 6．广告展示

当广告被展示完成并关闭后，您无需预加载下一个广告，sdk内部会自动预加载一个新的广告

## （1）展示Banner广告

Banner广告为自适应广告，需要配置布局参数，用于添加banner控件，布局容器的位置决定了广告的位置，由您自行控制

```
//这里设置广告宽度，可以为match_parent或者一个固定值，但是不能为0或者wrap_content，否则广告无法出来
int width = FrameLayout.LayoutParams.MATCH_PARENT;
//广告高度由宽度进行自适应
int height = FrameLayout.LayoutParams .WRAP_CONTENT;
FrameLayout.LayoutParams layoutParams = new FrameLayout.LayoutParams(width, height);

layoutParams.gravity = Gravity.BOTTOM;
DisplayMetrics displayMetrics = getResources().getDisplayMetrics();
layoutParams.bottomMargin = (int) TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP
        , 8f, displayMetrics);
//左右边距根据具体显示情况修正，也可以不指定左右边距
int dp16 = (int) TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, 8f, displayMetrics);
layoutParams.setMarginStart(dp16);
layoutParams.setMarginEnd(dp16);

AdHelper.showBanner(this, layoutParams, new GameAdBannerListener() {

    @Override
    public void onAdFailedToLoad(int code, String message) {
        Log.i(TAG, "Banner onAdFailedToLoad " + code + " " + message);
    }

    @Override
    public void onAdOpened() {
        Log.i(TAG, "Banner onAdOpened");
    }

    @Override
    public void onAdImpression() {
        Log.i(TAG, "Banner onAdImpression");
    }

    @Override
    public void onAdLoaded() {
        Log.i(TAG, "Banner onAdLoaded");
    }

    @Override
    public void onAdClosed() {
        Log.i(TAG, "Banner onAdClosed");
    }
});
```

## （2）展示插屏广告

两次展示插屏广告的时间间隔要大于1分钟

```
AdHelper.showInterstitial(this, new GameAdShowListener(){
    @Override
    public void onShow() {
        Log.i(TAG, "Interstitial show");
    }

    @Override
    public void onClose() {
        Log.i(TAG, "Interstitial close");
    }

    @Override
    public void onClick() {
        Log.i(TAG, "Interstitial onClick");
    }

    @Override
    public void onAdImpression() {
        Log.i(TAG, "Interstitial onAdImpression");
    }

    @Override
    public void onShowFailed(int code, String message) {
        Log.i(TAG, "Interstitial show fail " + code + " " + message);
    }
});
```

### （3）展示激励视频广告

```
AdHelper.showReward(this, new GameAdRewardShowListener(){
    @Override
    public void onShow() {
        Log.i(TAG, "Reward show");
    }

    @Override
    public void onClose() {
        Log.i(TAG, "Reward close");
    }

@Override
public void onClick() {
    Log.i(TAG, "Reward onClick");
}

@Override
public void onAdImpression() {
    Log.i(TAG, "Reward onAdImpression");
}

@Override
public void onShowFailed(int code, String message) {
    Log.i(TAG, "Reward show fail " + code + " " + message);
}

@Override
```

```
    public void onUserEarnedReward(GameRewardItem rewardItem) {
        Log.i(TAG, "Reward onUserEarnedReward " + rewardItem.getType() + " " + rewardItem.getAmount());
    }
});
```

### （4）展示开屏广告

开屏广告在App冷启动或热启动时，自动展示，不需要手动触发。

### （5）展示原生广告

1. 添加原生广告View控件到布局中

```
<com.transsion.gamead.GameNativeAdView
    android:id="@+id/native_layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:background="#999999"
    android:translationZ="100dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

2. 创建原生广告内部布局文件 native_install.xml，放到layout资源目录下，参考文件如下：

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:minHeight="50dp"
    android:orientation="vertical">

    <TextView style="@style/AppTheme.AdAttribution"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:paddingLeft="20dp"
        android:paddingRight="20dp"
        android:paddingTop="3dp">

        <LinearLayout

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <com.hisavana.mediation.ad.TIconView
            android:id="@+id/ad_app_icon"
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:adjustViewBounds="true"
            android:paddingBottom="5dp"
            android:paddingEnd="5dp"
            android:paddingRight="5dp"/>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TextView
                android:id="@+id/ad_headline"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textColor="#0000FF"
```

```
            android:textSize="16sp"
            android:textStyle="bold" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <TextView
                android:id="@+id/ad_advertiser"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:gravity="bottom"
                android:textSize="14sp"
                android:textStyle="bold"/>

            <RatingBar
                android:id="@+id/ad_rating"
                style="?android:attr/ratingBarStyleSmall"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:isIndicator="true"
                android:numStars="5"
                android:stepSize="0.5" />

            <TextView
                android:id="@+id/ad_stars"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="" />
            <TextView
                android:id="@+id/likes"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="" />

            <TextView
                android:id="@+id/downloads"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="" />

        </LinearLayout>

    </LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:id="@+id/ad_body"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="20dp"
        android:layout_marginEnd="20dp"
        android:textSize="12sp" />

<com.hisavana.mediation.ad.TMediaView
    android:id="@+id/ad_media"
    android:layout_gravity="center_horizontal"
    android:layout_width="250dp"
    android:layout_height="175dp"
    android:layout_marginTop="5dp" />

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="end"
    android:orientation="horizontal"
    android:paddingBottom="10dp"
    android:paddingTop="10dp">

    <TextView
        android:id="@+id/ad_price"
        android:layout_width="wrap_content"
```

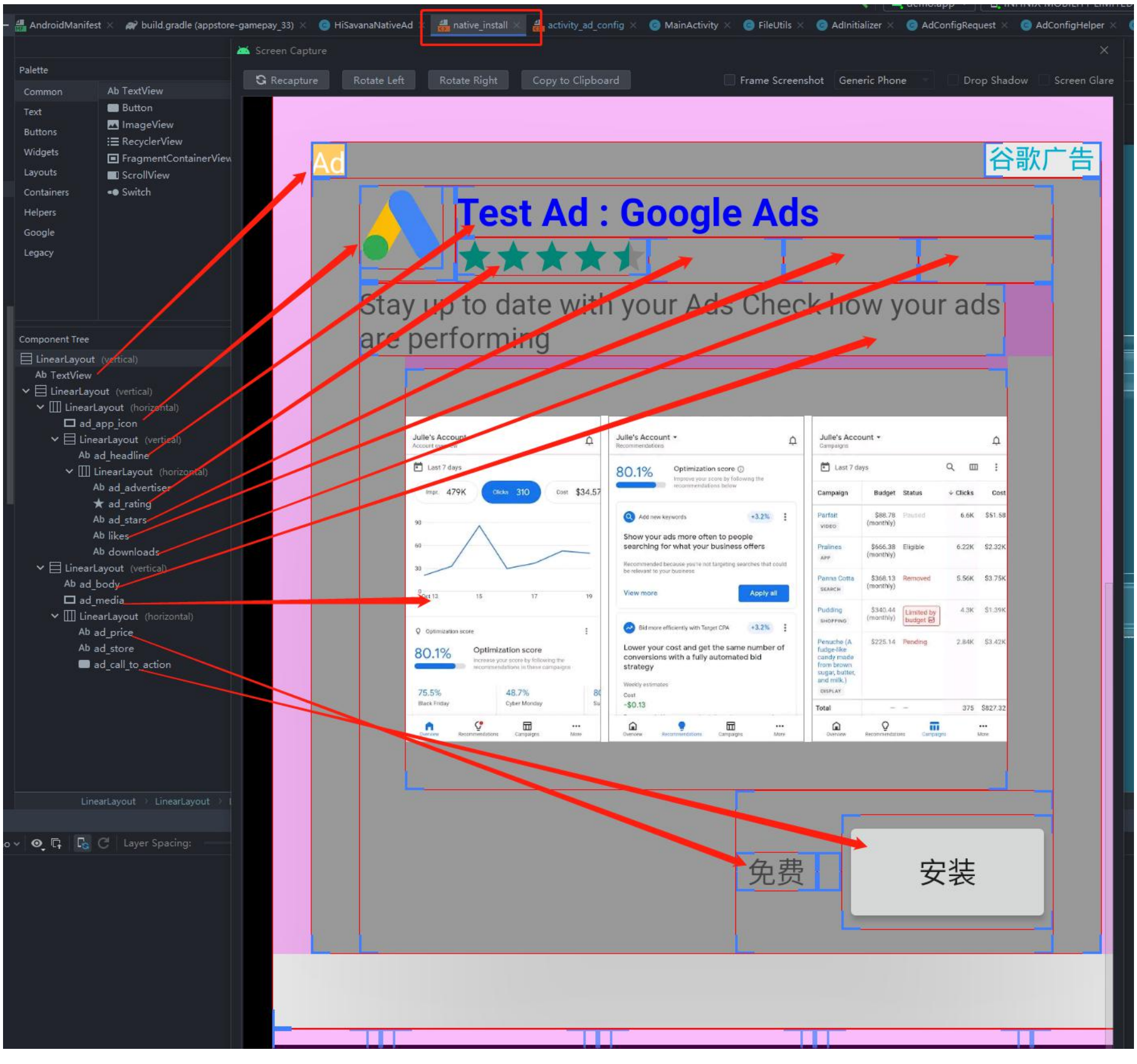
```
            android:layout_height="wrap_content"
            android:paddingLeft="5dp"
            android:paddingStart="5dp"
            android:paddingRight="5dp"
            android:paddingEnd="5dp"
            android:textSize="12sp" />

        <TextView
            android:id="@+id/ad_store"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:paddingLeft="5dp"
            android:paddingStart="5dp"
            android:paddingRight="5dp"
            android:paddingEnd="5dp"
            android:textSize="12sp" />

        <Button
            android:id="@+id/ad_call_to_action"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:textSize="12sp" />
      </LinearLayout>
    </LinearLayout>
  </LinearLayout>
</LinearLayout>
```

3. 把原生广告的展示元素绑定到上面布局文件内的控件，并调用AdHelper.showNative展示广告：

```
GameNativeAdViewBinder viewBinder = new GameNativeAdViewBinder.Builder(R.layout.native_install).titleId(R.id.ad_headline)
        .iconId(R.id.ad_app_icon).callToActionId(R.id.ad_call_to_action).descriptionId(R.id.ad_body)
        .mediaId(R.id.ad_media).sponsoredId(R.id.ad_store)
        .ratingId(R.id.ad_rating)
        .likesId(R.id.likes)
        .priceId(R.id.ad_price)
        .storeId(R.id.ad_store)
        .downloadsId(R.id.downloads)
        .actionIds(R.id.call_to_action).contextMode(NativeContextMode.NORMAL).
                adChoicesView(R.id.adChoicesView).build();

GameNativeAdView adNativeView = findViewById(R.id.native_layout);
AdHelper.showNative(adNativeView, viewBinder, new GameAdShowListener() {
    @Override
    public void onShow() {
        Log.i(TAG, "Native show");
    }

    @Override
    public void onShowFailed(int code, String message) {
        Log.i(TAG, "Native show fail " + code + " " + message);
    }

    @Override
    public void onClose() {
        Log.i(TAG, "Native close");
    }

    @Override
    public void onClick() {
        Log.i(TAG, "Native click");
    }

    @Override
    public void onAdImpression() {
        Log.i(TAG, "Native Impression");
```

```
    }
});
```

通过类GameNativeAdViewBinder，绑定了native广告内容和要显示的native_install.xml中的控件ID，当要展示Native广告，会自动把缓存的Native广告元素，填充到对应的控件中。如下图所示：




你可以根据游戏中对原生广告的展示场景，对上面的布局进行调整，甚至隐藏一些元素，但是有几个元素是要始终包含的，对于这些元素的规定，请参考google文档：

[标准原生广告格式 - Authorized Buyers帮助 (google.com)]

### (6) 展示Float广告

```
//浮窗广告位置不要遮挡游戏功能区
int width = FrameLayout.LayoutParams.WRAP_CONTENT;
int height = FrameLayout.LayoutParams.WRAP_CONTENT;
FrameLayout.LayoutParams layoutParams = new FrameLayout.LayoutParams(width, height);
layoutParams.gravity = Gravity.BOTTOM;
DisplayMetrics displayMetrics = getResources().getDisplayMetrics();
layoutParams.bottomMargin = (int) TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP
        , 100f, displayMetrics);
int dp16 = (int) TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, 16f, displayMetrics);
layoutParams.setMarginStart(dp16);
layoutParams.setMarginEnd(dp16);
```

```
AdHelper.showFloat(this, layoutParams, new GameAdShowListener() {
    @Override
    public void onShow() {
        Log.i(TAG, "Float show");
    }

    @Override
    public void onClose() {
        Log.i(TAG, "Float close");
    }

    @Override
    public void onClick() {
        Log.i(TAG, "Float onClick");
    }

    @Override
    public void onAdImpression() {
        Log.i(TAG, "Float onAdImpression");
    }

    @Override
    public void onShowFailed(int code, String message) {
        Log.i(TAG, "Float show fail " + code + " " + message);
    }
});
}
```

## 7．广告关闭

当您某些场景需要把展示的广告关闭，使用如下代码

### （1）关闭banner广告

```
AdHelper.closeBanner(this);
```

### （2）关闭原生广告

```
AdHelper.closeNative();
```

### (3) 关闭Float广告

```
AdHelper.closeFloat(activity);
```

## 8．添加Firebase配置文件

- 将 Firebase Android 配置文件添加到您的应用：

  a . 找到对接时，发给您的**google-services.json**配置文件。

  b . 将配置文件移动到应用的模块（应用级）目录中。

- 如需在应用中启用 Firebase 产品，请将Google 服务插件添加到 Gradle 文件中。

a. 在根级（项目级）Gradle 文件（ build.gradle ）中添加规则，以纳入Google 服务Gradle 插件。此外，请确认您拥有Google的Maven 代码库。

```
buildscript {
    repositories {
        // Check that you have the following line (if not, add it):
        google()  // Google's Maven repository
    }
    dependencies {
        // ...
        // Add the following line:
        classpath  'com.google.gms:google-services:4.3.4'  // Google Services plugin
    }
}

allprojects {
    // ...
    repositories {
        // Check that you have the following line (if not, add it):
        google()  // Google's Maven repository
        // ...
    }
}
```

b. 在您的模块（应用级）Gradle 文件（通常是 app/build.gradle ）中，应用Google 服务Gradle 插件：

```
apply plugin: 'com.android.application'
// Add the following line:
apply plugin: 'com.google.gms.google-services'  // Google Services plugin
android {
    // ...
}
```

## 9．广告加载失败参数说明

| 参数值 | 参数描述 |
| --- | --- |
| -4 | 广告开关被关闭 |
| -3 | 广告未准备好，当您使用类似于showRewardWhenLoaded方法，GameAdDisplayCallback的failure方法可能会出现 |
| -2 | 未初始化PlaceId，您初始化时未传入对应广告类型的广告单元ID |
| -1 | 界面异常，广告加载或展示过程中，activity已被销毁 |
| 0 | 内部出现问题；例如，收到广告服务器的无效响应。由admob返回，根据经验，一般指请求不到广告或未开启科学上网 |
| 1 | 广告请求无效；例如，广告单元ID 不正确。由admob返回 |
| 2 | 由于网络连接问题，广告请求失败。由admob返回 |
| 3 | 广告请求成功，但由于缺少广告资源，未返回广告。由admob返回 |
| 9000 | 网络错误，请检查网络配置 |
| 9009 | 广告配置异常广告位被关闭或找不到；<br>或者联系运营人员检查后台配置的广告位id是否正确。 |
| 9031 | 请求广告超时,请尝试多触发几次广告请求。 |

| | |
|---|---|
| 9035 | 无广告填充,请求成功，但是没有广告返回;<br>在保证手机的网络翻墙了的情况下，可以尝试多触发几次广告请求。 |

## 10．广告测试

- **测试模式**

当GameInitializer . Builder . setDebuggable(true)的时候， 将会展示测试广告 。请在正式发包时， 将这个参数设置为false

- **线上模式**

测试模式1返回的广告是测试广告， 当你想返回线上广告时， 你需要GameInitializer . Builder . setDebuggable(false) 。 此时，如果不输入setTestDeviceIds， 而又有较多次的线上广告请求或广告点击时， 存在被google停止账号的风险 。

Q: 如何获取TestDeviceId?
A: 通过日志Tag为Ads， 可以过滤到信息

```
I/Ads: Use RequestConfiguration.Builder.setTestDeviceIds(Arrays.asList("33BE2250B43518CCDA7DE426D04EE231"))to get
test ads on this device."
```

你可以截取"33BE2250B43518CCDA7DE426D04EE231"字符串， 并填写到TestDeviceId，
这样你主动声明了测试设备后， 就不存在停止账号的风险 。

Q： 如何判断是否为测试广告或者线上测试广告?
A： 当广告标识Test Ad时， 代表广告是测试广告或者线上测试广告 。此时你点击或者请求都是没有风险的 。

## 11．日志调试

- 查看Banner广告加载过程日志

  *adb shell setprop log.tag.GameBannerAd VERBOSE*

  过滤TAG： GameBannerAd

- 查看插屏广告加载过程日志

  *adb shell setprop log.tag.GameInterstitialAd VERBOSE*

  过滤TAG： GameInterstitialAd

- 查看激励视频广告加载过程日志

  *adb shell setprop log.tag.GameRewardedAd VERBOSE*

  过滤TAG： GameRewardedAd

# 二、游戏埋点SDK接入文档 （必接）

## 1．首先申请app对应的配置文件

向运营同事申请配置文件game_sdk_config .json，这个文件放到app的assets 目录（"\src\main\assets"）

## 2．添加SDK依赖

- project的build.gradle

```
allprojects {
    repositories {
        maven { url 'https://mvn.shalltry.com/repository/maven-public/' }
        maven { url 'https://mvn.shalltry.com/repository/game-releases/' }
    }
}
```

- module的build.gradle

```
defaultConfig {
    multiDexEnabled true
}
    dependencies {
    implementation"com.transsion.game:analytics:4.0.0.0"
}
```

## 3．SDK初始化

**必须在游戏的Application的onCreate函数中， 初始化埋点SDK**

```
public class MyApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        GameAnalytics.init(new GameAnalytics.Builder(this));
    }
}
```

## 4．开始使用自定义埋点

在游戏进入到特定节点，可以调用自定已埋点接口上报相应的状态。

*GameAnalytics.tracker(String action, String param1, String param2);*

//示例，玩家闯关结果埋点，action为闯关结果预定义字串，第一个参数为关卡索引，第二个参数为结果(1：成功，0：失败)

```
GameAnalytics.tracker(Constants .ACTION_LEVEL_END, "20", "0");
```

注意，这里第一个参数action在程序中已经预定义了大部分场景，具体见下表：

| 动作名称 | 预定义名称 | 说明 | 参数1 | 参数2 |
| --- | --- | --- | --- | --- |
| app_start | Constants.ACTION_APP_START | 打开app | | |
| loading_begin | Constants.ACTION_LOADING_BEGIN | 加载开始 | | |
| loading_end | Constants.ACTION_LOADING_END | 加载结束 | | |
| login_ui | Constants.ACTION_LOGIN_UI | 展示登录界面 | | |
| login_click | Constants.ACTION_LOGIN_CLICK | 点击登录 | 登录方式 | |
| login_result | Constants.ACTION_LOGIN_RESULT | 登录结果 | 0失败 1成功<br>类型：整数 | 登录成功后的用户ID |
| create_role | Constants.ACTION_CREATE_ROLE | 创角 | 角色名称 | |
| game_start | Constants.ACTION_GAME_START | 成功进游 | | |
| tutorial_begin | Constants.ACTION_TUTORIAL_BEGIN | 开始新手引导 | | |
| tutorial_end | Constants.ACTION_TUTORIAL_END | 完成新手引导 | | |
| tutorial_reward | Constants.ACTION_TUTORIAL_REWARD | 领取新手奖励 | | |
| module_check | Constants.ACTION_MODULE_CHECK | 点击各模块 | | |
| play_mission | Constants.ACTION_PLAY_MISSION | 进入子游戏 | | |
| level_begin | Constants.ACTION_LEVEL_BEGIN | 关卡开始 | 关卡数，比如第一关，就传入1 | |
| level_end | Constants.ACTION_LEVEL_END | 关卡结束 | 关卡数，比如第一关，就传入1<br>类型：整数 | 关卡结果，0失败 1成功<br>类型：整数 |
| level_reward | Constants.ACTION_LEVEL_REWARD | 关卡奖励 | 奖励内容 | |

说明，如果上面动作名称不能满足cp方业务需求，请cp增加扩展名称，并按上面表格格式反馈给我们。方便后续数据统计。

参数1和参数2可以根据具体场景自行传入关注的内容。