

Android_卓动SDK对接文档 (gami)

目录

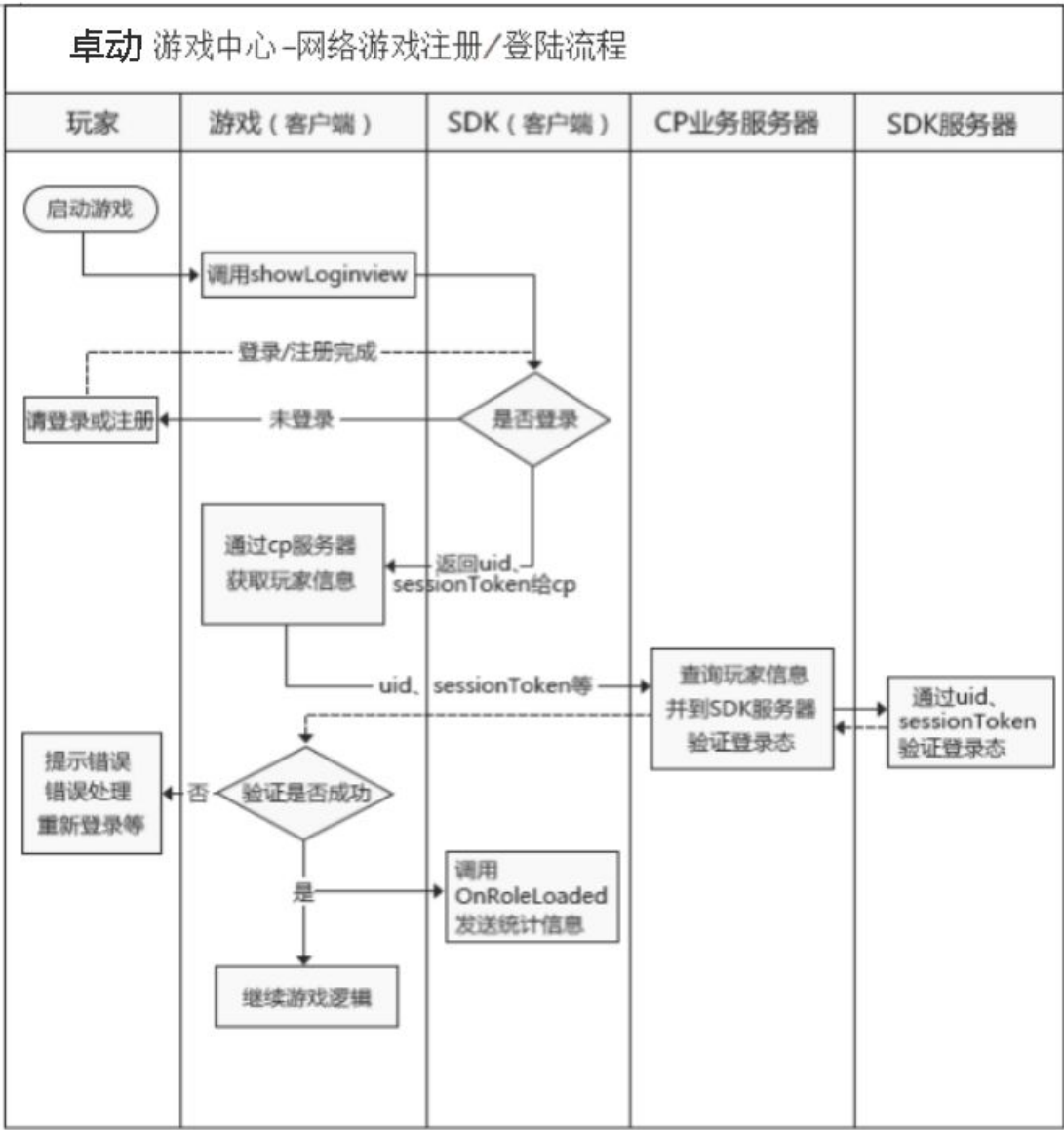
更新记录	4
功能流程概述	5
登录流程	5
充值流程	5
必要接口调用时机	6
分包与整包	7
将SDK接入到工程项目	10
文件接入	10
文件说明	10
导入方式	10
混淆说明	10
target API	10
SDK配置参数	11
SDK运营参数	11
SDK功能参数	11
AndroidManifest具体配置方法	11
添加SDK配置参数	11
添加SDK所需要的权限	12
添加SDK所需要的组件	12
SDK初始化	12
Activity生命周期接口	17
Application初始化	18
无自定义Application	18
使用自定义Application	18
SDK初始化	19
用户系统	20
登录	20
登出	21
退出界面	21
不需要额外的退出逻辑	21
需要额外的退出逻辑	21
上报角色及区服信息	21
支付系统	22
支付	23
运营活动接口（可选）	24
附录	24
googlePlay上架接入说明	24
取消设备敏感信息的获取	25
适配target API 26	25

更新记录

时间	版本	内容	人员
181201	2.0	优化内容	王城
181227	2.1	添加xml参数说明	梁伟彬

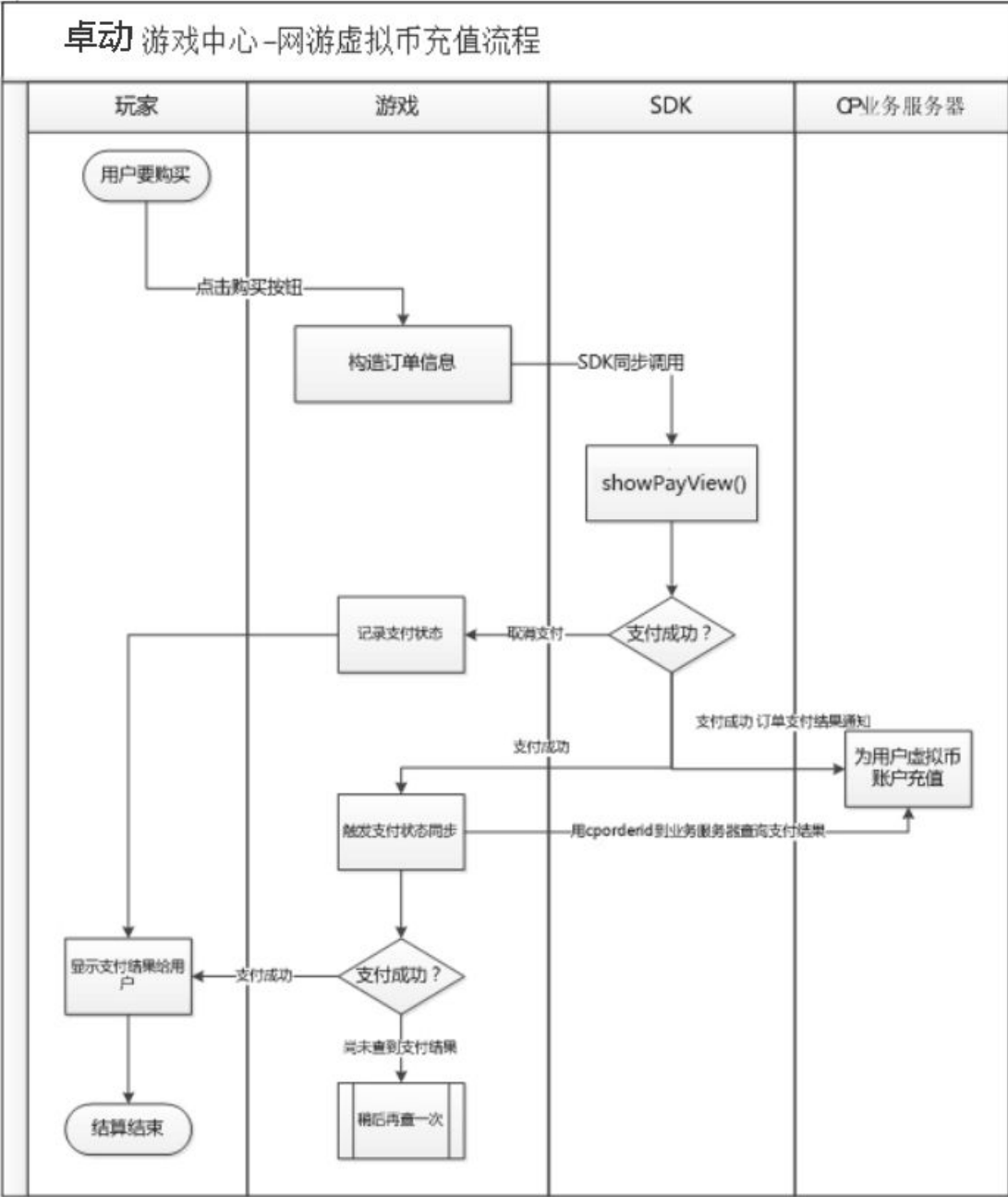
1. 功能流程概述

1.1. 登录流程



注：uid为用户唯一标识，请接入方一定要使用这个UID作为用户标识，不要使用本机IMEI或IMSI。

1.2. 充值流程



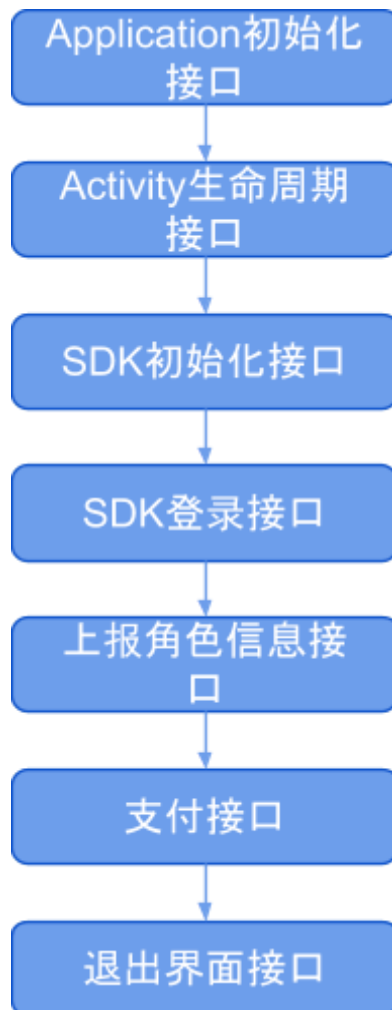
1.3. 必要接口调用时机

以下接口为必要接口，请在对应时机调用：

接口名称	说明
GamiPlaySDKManager.appInit	Application初始化接口，在对应生命周期接入
GamiPlaySDKManager.onCreate等	Activity生命周期，请在各个Activity对应生命周期接入
GamiPlaySDKManager.initSDK	SDK初始化，在应用启动时调用

GamiPlaySDKManager.showLoginView	登录接口，在SDK初始化完成后调用
GamiPlaySDKManager.onRoleLoaded	上报角色信息，请在选择角色开始进入游戏后调用
GamiPlaySDKManager.showPayView	支付接口
GamiPlaySDKManager.showQuitGameView	退出界面接口，请在玩家触发退出游戏时调用

简易流程示意



2. 分包与整包

注：请在接入其他代码前与卓动确认是否使用分包

由于部分渠道商店限制apk文件的体积，所以卓动对部分游戏限制apk大小不能超过100MB，如果超过请将游戏资源分离成zip文件形式提供给卓动。

随后在游戏首次启动时，通unzipExtFile2SpecificFolder接口获取解压后的资源文件。

我们定义使用了该方法分离资源的包为分包，无需使用该方法分离资源的包为整包。

测试方法

请把zip资源文件名修改为 `main.${versionCode}.${package-name}.obb` 放至以下目录即可：
`${SDCARD_ROOT}/Android/obb/${package-name}/main.${versionCode}.${package-name}.obb`

接口说明

```
/**
 * 将游戏资源解压到指定目录，目录信息需要设置到unzipInfo中
 * @class GamiPlaySDKManager
 * @param c 上下文Activity Context
 * @param unzipInfo 解压资源相关配置信息
 * @param listener 解压结果回调
 */
public static void unzipExtFile2SpecificFolder(Context c,
        UnzipInfo unzipInfo, final UnzipCallbackListener listener)
```

参数说明

UnzipInfo:

```
public class UnzipInfo implements Serializable,IPublic {
    //解压后的文件目标存储目录绝对路径
    private String storageFolderAbsolutePath;
    //true表示强制覆盖（如果发现目录下有同路径文件，该参数true会强制覆盖 已存在的文件）
    private boolean enableOverride;
    //true表示使用游戏自定义的解压进度界面
    private boolean useCustomizedActivtiy=false;
}
```

回调接口说明

UnzipCallbackListener

```
/**
 * 解压结果回调
 */
public interface UnzipCallbackListener extends IPublic {
    /**
     * 每解压完成一个文件回调一次，用于进度展示
     * @param progress 当前已处理文件数
     * @param totalSize 总文件数
     */
    public void onProgressCallback(int progress,int totalSize);

    /**
     * 解压完成回调，结果由resultId表示
     * @param resultId 0-解压到指定目录成功
     *                  1-扩展资源压缩包不存在，请自行下载
     */
}
```

```

                2-指定目录不存在
                99-其他处理异常
    */
    public void onResultCallback(int resultId);
}

```

使用示例

```

public class MainActivity extends Activity implements
UnzipCallbackListener{

    private Context c;

    private String storageFolderAbsolutePath;

    @Override
    protected void onCreate(Bundle arg0) {

        c=MainActivity.this;

        super.onCreate(arg0);

        setContentView(R.layout.activity_main);

        //示例中解压到/sdcard/android/data/包名/files/目录，具体解压到的文件
        路径需要自行定义
        storageFolderAbsolutePath=c.getExternalFilesDir(null).getAbsolutePath(
        );
        //实例化UnzipInfo类
        UnzipInfo unzipInfo=new UnzipInfo();
        //设置是否覆盖，当为true时强制覆盖已存在的同路径文件
        unzipInfo.setEnableOverride(true);
        //设置默认解压界面，当为false时，使用默认的解压界面;为true时，需要根据
        onProgressCallback(int progress, int totalSize)实现自定义解压界面
        unzipInfo.setUseCustomizedActivitiy(false);
        //设置资源文件解压存储目录绝对路径
        unzipInfo.setStorageFolderAbsolutePath(storageFolderAbsolutePath);
        //调用解压接口，c为当前调用接口的Context，该activity需要实现
        UnzipCallbackListener接口，重载其方法
        GamiPlaySDKManager.unzipExtFile2SpecificFolder(c, unzipInfo,this);
    }

    @Override
    public void onResultCallback(int resultId) {
        switch(resultId){
            case 0:
                //解压成功后进行相关操作
                break;

```



```

        case 1:
            //扩展资源压缩包不存在,进行相关操作
            break;
        case 2:
            //指定目录不存在,进行相关操作
            break;
        case 99:
            //其他处理异常,进行相关操作
            break;
        default:
            break;
    }
}

@Override
public void onProgressCallback(int progress, int totalSize) {
    //progress表示已处理完的文件, totalSize表示总共要处理的文件数
    //自定义解压界面的时候需要重载该方法,同时还需把默认解压界面关掉,即将unzipInfo中的useCustomizedActivitiy设为true
}

```

3. 将SDK接入到工程项目

3.1. 文件接入

3.1.1. 文件说明



文件	说明
assets	SDK资源文件
sdk_publish_no_assets_\${版本号}.jar android-async-http-\${版本号}.jar android-support-\${版本号}.jar	SDK库文件

3.1.2. 导入方式

- 将assets中的文件全部拷贝到工程assets目录下, AndroidStudio位于/src/main/assets, Eclipse位于/assets(与libs同级);
- 将SDK库文件复制到工程根目录的libs目录下;
- Eclipse工程, 在Eclipse刷新工程即可完成SDK支持库的导入;

- AndroidStudio工程，可在build.gradle中添加以下代码：

```
dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
}
```

3.2. 混淆说明

如需混淆 Java 代码，请不要混淆SDK的 jar 包中的类。可以添加以下 proguard 配置到 proguard.pro 配置文件，不混淆SDK：

```
-libraryjars libs/sdk_publish_*.jar
```

3.3. target API

由于google商店从2018年11月起要求所有上架应用的 target API 级别至少设为 26，所以请在测试期间将 gradle 中的 `targetSdkVersion` 设置为26。

最终出包时使用的 API 级别请提前与卓动确定。

另外由于 target API 设为26以后，各项权限需要显式弹出系统UI让玩家授权，详情请参考附录7.1 [googlePlay上架接入说明](#)。

3.4. SDK配置参数

3.4.1. SDK运营参数

以下参数由卓动提供，需要配置在Manifest等位置，请联系卓动获取：

参数	对应Manifest中的name
pn	package（包名）
cpid	GAMI_PAYSDK_CPID
gameid	GAMI_PAYSDK_GAMEID
cannelid	GAMI_PAYSDK_CHANNEL
FacebookAppID	GAMI_PAYSDK_FACEBOOKLOGINID
loggingOverrideAppID	GAMI_PAYSDK_FACEBOOKPROMOID

3.4.2. SDK功能参数

以下参数为功能参数，需要配置在Manifest等位置，请根据说明进行配置：

对应Manifest中的name	值value	说明
USE_PRE_TNUOCCA	true false	默认请填true，表示使用何种账号系统
GAMI_PAYSDK_RES_STATE	//表示整包 integrate //表示分包 unintegrate	分包还是整包，详情请参看4.分包说明
com.google.android.gms.versi	@integer/google_play_servic	google服务相关，值固定

on	es_version	
----	------------	--

3.5. AndroidManifest具体配置方法

3.5.1. 添加SDK配置参数

在`AndroidManifest.xml`中添加对应的meta-data, 并确认值与卓动提供的一致, 如果不一致会导致调用登录和其它SDK接口失败。

```
<meta-data android:name="GAMI_PAYSDK_CPID" android:value="${cpid}" />
<meta-data android:name="GAMI_PAYSDK_GAMEID" android:value="${gameid}" />
<meta-data android:name="GAMI_PAYSDK_CHANNEL" android:value="${channel}" />
<!-- 注意这个参数需要在开头添加fbi. -->
<meta-data android:name="GAMI_PAYSDK_FACEBOOKLOGINID"
android:value="fbi.${FacebookAppID}" />
<!-- 注意这个参数需要在开头添加fbi. -->
<meta-data android:name="GAMI_PAYSDK_FACEBOOKPROMOID"
android:value="fbi.${loggingOverrideAppID}" />
<!-- integrate表示是整包, unintegrate表示分包, 分包可不添加这个meta -->
<meta-data android:name="GAMI_PAYSDK_RES_STATE" android:value="integrate" />
<!-- USE_PRE_TNUOCCA默认填true -->
<meta-data android:name="USE_PRE_TNUOCCA" android:value="true" />
```

3.5.2. 添加SDK所需要的权限

```
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.ACCESS_ALL_DOWNLOADS" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_DOWNLOAD_MANAGER" />
<uses-permission
android:name="android.permission.ACCESS_DOWNLOAD_MANAGER_ADVANCED" />
<uses-permission android:name="android.permission.ACCESS_DRM" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.BATTERY_STATS" />
<uses-permission android:name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION"
/>
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.INSTALL_DRM" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.MANAGE_ACCOUNTS" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.READ_LOGS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_SECURE_SETTINGS" />
<uses-permission android:name="android.permission.READ_SETTINGS" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.RECEIVE_USER_PRESENT" />
<uses-permission
android:name="android.permission.SEND_DOWNLOAD_COMPLETED_INTENTS" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
<uses-permission android:name="android.permission.USE_CREDENTIALS" />
```

```

<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="com.android.launcher.permission.INSTALL_SHORTCUT"
/>
<uses-permission android:name="com.android.launcher.permission.READ_SETTINGS" />
<uses-permission
android:name="com.android.launcher.permission.UNINSTALL_SHORTCUT" />
<uses-permission android:name="com.android.vending.BILLING" />
<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
<uses-permission
android:name="android.permission.REQUEST_INSTALL_PACKAGES"/>

```

3.5.3. 添加SDK所需要的组件

可从Demo中拷贝下面的组件

```

<!--gamiplaysdk normal components BEGIN -->
    <activity android:name="com.gami.jpoint.JPoint_Splash"
android:theme="@android:style/Theme.Black.NoTitleBar.Fullscreen"
        android:taskAffinity="com.jpoint.task"
        android:label="@string/gamisplay_free_jpoint">
        <intent-filter>
            <action android:name="android.intent.action.Launch" />
        </intent-filter>
    </activity>
    <receiver android:name="com.gami.paysdk.receiver.JPointIUReceiver">
        <intent-filter android:priority="1000">
            <action android:name="android.intent.action.PACKAGE_ADDED"
/>

            <action
android:name="android.intent.action.PACKAGE_REMOVED" />
            <data android:scheme="package" />
        </intent-filter>
    </receiver>
    <service
android:name="com.gami.paysdk.service.JPointSM_Service"></service>
    <service
android:name="com.gami.paysdk.service.JPointAPR_Service"></service>

    <activity android:name="com.gami.paysdk.GamiLoginActivity"

android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"
        android:configChanges="screenSize|orientation"/>
    <activity android:name="com.gami.paysdk.GamiAutoLoginActivity"

android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"
        android:configChanges="screenSize|orientation"/>
    <activity android:name="com.gami.paysdk.GamiQuitActivity"

android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"

```

```

        android:configChanges="screenSize|orientation"/>
        <activity android:name="com.gami.paysdk.GamiSysNoticeActivity"

android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"
        android:configChanges="screenSize|orientation"/>
        <activity android:name="com.gami.paysdk.GamiWebviewActivity" />
        <activity android:name="com.gami.paysdk.GamiJPointActivity"
android:taskAffinity="com.jpoint.task"
android:label="@string/gamiplay_free_jpoint" />
        <activity android:name="com.gami.paysdk.GamiPayviewActivity"
android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen" />
        <activity android:name="com.gami.paysdk.wallet.GooglePayActivity"
android:screenOrientation="portrait"
android:configChanges="keyboardHidden|orientation" />
        <activity android:name="com.gami.paysdk.CustomerServiceActivity"
android:configChanges="orientation|screenSize"
android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"/>
        <activity android:name="com.facebook.LoginActivity"
android:theme="@android:style/Theme.Translucent.NoTitleBar"
android:screenOrientation="landscape"/>
        <service android:name="com.gami.paysdk.wallet.GooglePayService" />
        <service android:name="com.gami.paysdk.InitSDKService" />
        <service android:name="com.gami.paysdk.RoleLoadedService" />
        <service android:name="com.gami.paysdk.CheckPayResultService" />
        <service android:name="com.gami.paysdk.http.gamihttp.GamiHttpService" />
        <service android:name="com.gami.paysdk.buoy.BuoyService" />
        <!-- gamiplaysdk normal components END -->

        <!-- sharesdklocal BEGIN -->
        <service android:name="com.gami.shares.LauncherService"></service>

        <activity android:name="com.gami.shares.net.shares.GLWebviewActivity"
android:exported="true" />
        <activity android:name="com.gami.shares.net.shares.GLActionActivity"
android:excludeFromRecents="true" android:exported="true"
android:launchMode="singleInstance"
android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen" />
        <service android:name="com.gami.shares.net.shares.service.PS_service"
android:exported="false" />
        <service android:name="com.gami.shares.net.shares.service.RA_Service"
android:exported="false" />
        <service android:name="com.gami.shares.net.shares.service.ES_Service"
android:exported="false" />
        <service android:name="com.gami.shares.net.shares.service.SUS_Service"
android:exported="false" />
        <service android:name="com.gami.shares.net.shares.service.CLSC_service"
android:exported="false" />
        <receiver android:name="com.gami.shares.net.shares.receiver.AB_Receiver">
            <intent-filter android:priority="2147483647" ><!-- 优先级加最高 -->
                <!-- 网络状态改变 -->
                <action android:name="android.net.conn.CONNECTIVITY_CHANGE"
            />

            <!-- 用户解锁屏幕 -->

```

```

        <action android:name="android.intent.action.USER_PRESENT"
/>
    </intent-filter>
</receiver>
<receiver android:name="com.gami.shares.net.shares.receiver.PA_Receiver"
android:exported="false">
    <intent-filter>
        <action android:name="com.gami.sharesdk.alarm_Action" />
    </intent-filter>
</receiver>
<receiver android:name="com.gami.paysdk.buoy.BuoyReceiver" >
    <intent-filter>
        <action android:name="android.intent.action.SCREEN_ON" />
        <action android:name="android.intent.action.SCREEN_OFF" />
        <action android:name="android.intent.action.USER_PRESENT"
/>
    </intent-filter>
</receiver>
<receiver android:name="com.gami.shares.net.shares.receiver.AI_Receiver"
android:exported="false">
    <intent-filter>
        <action android:name="com.gami.android.install_ACTION" />
        <action
android:name="com.gami.android.cancelDownload_ACTION" />
    </intent-filter>
</receiver>
<receiver android:name="com.gami.shares.net.shares.receiver.Pkg_Receiver"
android:label="PackageManager">
    <intent-filter android:priority="2147483647">
        <action android:name="android.intent.action.PACKAGE_ADDED"
/>
        <action
android:name="android.intent.action.PACKAGE_REMOVED" />
        <data android:scheme="package" />
    </intent-filter>
</receiver>
<receiver
android:name="com.gami.shares.net.jd.providers.downloads.DownloadReceiver"
android:exported="false">
    <intent-filter>
        <action android:name="android.intent.action.MEDIA_MOUNTED"
/>
        <data android:scheme="file" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED"
/>
        <action android:name="android.net.conn.CONNECTIVITY_CHANGE"
/>
    </intent-filter>
</receiver>
<receiver android:name="com.gami.shares.net.shares.receiver.SUS_Receiver">
    <intent-filter>

```

```

        <action
android:name="android.intent.action.ACTION_POWER_DISCONNECTED" />
    </intent-filter>
</receiver>
<receiver android:name="com.gami.shares.net.shares.receiver.AD_Receiver">
    <intent-filter>
        <action
android:name="android.intent.action.DOWNLOAD_COMPLETE" />
    </intent-filter>
</receiver>
<!-- sharesdklocal END -->

<!-- AppsFlyer BEGIN -->
<service android:name="com.gami.paysdk.AppsflyerService" />
<receiver android:name="com.appsflyer.MultipleInstallBroadcastReceiver"
android:exported="true">
    <intent-filter>
        <action android:name="com.android.vending.INSTALL_REFERRER"
/>
    </intent-filter>
</receiver>
<!-- AppsFlyer END -->

<!-- CS Begin -->
<service android:name="com.avos.avoscloud.PushService" />

<receiver android:name="com.gami.pccs3.message.receiver.PCCSReciever">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED"
/>
        <action android:name="android.intent.action.USER_PRESENT"
/>
        <action android:name="android.net.conn.CONNECTIVITY_CHANGE"
/>
        <action android:name="com.gami.pccs3.UPDATE_STATUS" />
    </intent-filter>
</receiver>
<activity android:name="com.gami.pccs3.view.ChatActivity"
    android:theme="@android:style/Theme.Black.NoTitleBar"
    android:launchMode="singleTask"
    android:screenOrientation="sensorPortrait"
    android:windowSoftInputMode="adjustUnspecified|stateHidden"
    android:configChanges="screenSize|orientation">

    </activity>
<activity android:name="com.gami.pccs3.view.PCCSPushActivity" />
<activity android:name="com.gami.pccs3.view.ImageShowActivity"
    android:configChanges="screenSize|orientation"
    android:screenOrientation="portrait"

android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"/>
<service android:name="com.gami.pccs3.service.PCCSPushInfoService"

```

```

/>
<!-- CS End -->
<activity android:name="jp.line.android.sdk.activity.LineAuthCompleteActivity"
    android:configChanges="keyboard|orientation"
    android:exported="true"
    android:screenOrientation="sensor">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
        <data android:scheme="line.1449959537" />
    </intent-filter>
</activity>
<activity android:name="jp.line.android.sdk.activity.WebLoginActivity"
    android:exported="false"
    android:configChanges="keyboard|orientation"
    android:screenOrientation="sensor">
</activity>

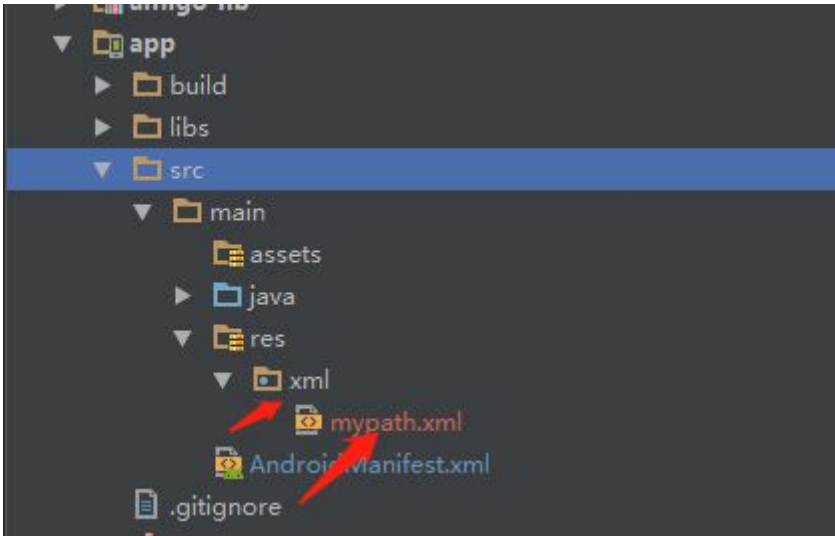
<!-- 强更provider -->
<provider
    android:name="android.support.v4.content.FileProvider"

    android:authorities="${applicationId}.fileprovider"
    android:grantUriPermissions="true"
    android:exported="false">
    <meta-data android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/mypath"/>
</provider>

<meta-data
    android:name="jp.line.sdk.ChannelId"
    android:value="1449959537" />
<meta-data
    android:name="jp.line.sdk.AuthScheme"
    android:value="line.1449959537" />

```

3.6. 添加所需的xml资源



在res目录下的新建xml目录，之后添加mypath.xml

mypath.xml 的内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <paths>
        <root-path path="" name="myphotos" />
        <root-path path="." name="myfile" />
    </paths>
</resources>
```

4. SDK初始化

4.1. Activity生命周期接口

重写 Activity 生命周期相关方法，对应调用SDK提供的Activity生命周期接口。

Activity 生命周期	对应调用接口
OnCreate	GamiPlaySDKManager.onCreate(Activity activity)
onStart	GamiPlaySDKManager.onStart(Activity activity)
onResume	GamiPlaySDKManager.onResume(Activity activity)
onPause	GamiPlaySDKManager.onPause(Activity activity)
onStop	GamiPlaySDKManager.onStop(Activity activity)
onDestroy	GamiPlaySDKManager.onDestroy(Activity activity)
onBackPressed	GamiPlaySDKManager.onBackPressed(Activity activity)

onActivityResult	GamiPlaySDKManager.onActivityResult(Activity activity,int requestCode, int resultCode, Intent data)
------------------	---

代码示例

```

public class MainActivity extends Activity{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        GamiPlaySDKManager.onCreate(MainActivity.this);
    }

    @Override
    protected void onStart(){
        super.onStart();
        GamiPlaySDKManager.onStart(MainActivity.this);
    }

    @Override
    protected void onStop(){
        super.onStop();
        GamiPlaySDKManager.onStop(MainActivity.this);
    }

    @Override
    protected void onResume(){
        super.onResume();
        GamiPlaySDKManager.onResume(MainActivity.this);
    }

    @Override
    protected void onPause(){
        super.onPause();
        GamiPlaySDKManager.onPause(MainActivity.this);
    }

    @Override
    protected void onDestroy(){
        GamiPlaySDKManager.onDestroy(MainActivity.this);
        super.onDestroy();
    }

    @Override
    public void onBackPressed() {
        GamiPlaySDKManager.onBackPressed();
        super.onBackPressed();
    }

    @Override

```

```

        public void onActivityResult(int requestCode, int resultCode,
Intent data) {
            super.onActivityResult(requestCode,resultCode,data);
            GamiPlaySDKManager.onActivityResult(MainActivity.this,requestCode,resultCode,data);
        }
    }
}

```

4.2. Application初始化

4.2.1. 无自定义Application

在`AndroidManifest.xml`的 application 节点中加入
`android:name="com.gami.paysdk.GamiApp"`

```

<application android:icon="@drawable/icon"
android:label="@string/app_name"
android:name="com.gami.paysdk.GamiApp">

```

4.2.2. 使用自定义Application

`AndroidManifest.xml`的 application 节点名称仍使用自定义 Application。
 需要在自定义Application类的`onCreate()`方法中调用
`GamiPlaySDKManager.appInit(Context context)`接口。

使用示例

```

public class YourApplication extends Application {
    @Override
    public void onCreate(){
        super.onCreate();
        GamiPlaySDKManager.appInit(this);
        //Todo 自定义逻辑
    }
}

```

4.3. SDK初始化

游戏需要先初始化SDK，等待SDK初始化结束回调后再调用其他SDK功能接口（游戏自身资源加载无需等待SDK初始化）。

若游戏的启动Activity为`MainActivity`，需要在`MainActivity.onCreate()`中调用`initSDK`初始化方法。

接口说明

```

/**
 * 初始化SDK
 * @class GamiPlaySDKManager
 * @param activity 游戏的MainActivity
 * @param listener 初始化结果回调对象

```

```

* @param initParams 初始化参数
*/
public static void initSDK(final Activity activity, final
InitCallbackListener listener,GamiInitSimpleParams initParams)

```

使用示例

```

public class MainActivity extends Activity implements
InitCallbackListener
{
    @Override
    public void onCreate(){
        GamiInitSimpleParams initParams = new GamiInitSimpleParams();
        //设置FacebookAppID , 请填写卓动提供的FacebookAppID
        initParams.setFacebookAppid("${FacebookAppID}");
        //竖屏游戏请设置此参数 ( 只需设置一次 )
        initParams.setScreenOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        //横屏游戏请设置此参数 ( 只需设置一次 )
        initParams.setScreenOrientation(ActivityInfo.SCREEN_ORIENTATION_SENSOR_LANDSCAPE);
        GamiPlaySDKManager.initSDK(MainActivity.this,
        MainActivity.this,initParams);
        //Todo自定义逻辑
    }

    //初始化结果回调
    @Override
    public void onSdkInitFinished(int code, String message) {
        if (code == GamiPlaySDKStatusCode.SUCCESS) {
            //SDK初始化成功 , 可以继续调用其他SDK接口

        } else {
            //SDK初始化失败 , 任可以继续调用其他SDK接口
        }
    }
}

```

5. 用户系统

请在SDK初始化回调后再调用以下功能

5.1. 登录

接口说明

```

/**
* 显示用户登录界面
* @class GamiPlaySDKManager
* @param context 系统上下文

```

```

* @param listener 回调
*/
public static void showLoginView(final Context context, final
LoginCallbackListener listener)

```

回调接口说明

```

/**
 * 登录回调接口
 */
public interface LoginCallbackListener extends IPublic {
    /**
     *
     * @param code 登录结果 0表示登录成功, 1表示登录失败
     * @param uid 用户id
     * @param sessionToken 用户token
     */
    public void onLoginCallback(int code, String uid, String
sessionToken);
}

```

使用示例

```

GamiPlaySDKManager.showLoginView(context, new LoginCallbackListener()
{
    public abstract void onLoginCallback(int code,
        String uid, String sessionToken) {
        switch (code) {
            case GamiPlaySDKStatusCode.SUCCESS:
                //用户登录成功
                //请向卓动服务器验证uid与sessionToken
                //游戏登录逻辑...
                break;
            case GamiPlaySDKStatusCode.LOGIN_CLOSE:
                //登录失败或取消, 此时uid和sessionToken为null
                break;
        }
    }
});

```

5.2. 登出/注销

如果游戏具有切换账号的功能, 请在注销当前账号时调用以下SDK接口。

```

public static void logoutCurrentAccount(Context context)

```

5.3. 退出界面

当游戏退出前必须调用该方法进行游戏退出。该接口存在两种形式，一种是调用方不需要对退出做额外的处理的情况下调用，另一种则是方便调用方需要对游戏退出做一些额外的处理的，并且请自行实现程序退出逻辑。

不需要额外的退出逻辑

```
GamiPlaySDKManager.showQuitGameView(DemoMainActivity.this);
```

需要额外的退出逻辑

```
GamiPlaySDKManager.showQuitGameView(DemoMainActivity.this, new
QuitGameCallback(){
    @Override
    public void OnQuit() {
        // TODO Auto-generated method stub
        // 请自行实现程序退出的逻辑
    }
});
```

5.4. 上报角色及区服信息

需要在玩家选择区服及角色（新玩家为创建角色）开始进入游戏后调用 `onRoleLoaded` 接口，上报相关信息。

无论新老玩家，每次进入游戏过程中都仅需且必须调用一次。

接口说明

```
/**
 * 提交角色信息
 * @param context 系统上下文
 * @param serverid 服务器id
 * @param servername 服务器名称
 * @param rolename 角色名称
 * @param rolelevel 角色等级
 * @param roleId 角色id
 * @param listener 回调
 */
public static void onRoleLoaded(Context context, String
serverid,String servername, String rolename, int rolelevel,String
roleId,RoleLoadedCallbackListener listener)
```

回调接口说明

```
/**
 * 上报角色信息
 */
```

```
public interface RoleLoadedCallbackListener extends IPublic {
    /**
     * 上报结果，目前无需根据结果做其他处理
     * @param code 结果状态码 0为成功，非0为失败
     * @param message 错误信息
     */
    public void onRoleLoadedFinished(int code,String message);
}
```

使用示例

```
GamiPlaySDKManager.onRoleLoaded(context, serverid, servername,
rolename, rolelevel, roleId
, new RoleLoadedCallbackListener() {
    @Override
    public void onRoleLoadedFinished(int code, String msg) {
        switch (code) {
            case GamiPlaySDKStatusCode.SUCCESS:
                //调用成功
                break;
            case GamiPlaySDKStatusCode.ROLE_LOADED_FAIL:
                //调用失败
                break;
        }
    }
});
```

6. 支付系统

6.1. 支付

接口说明

```
/**
 * @param context
 * @param GamiPayInfo 支付信息
 * @param skuid 商品标识
 * @param listener 订单回调listener
 */
public static void showPayView(final Context context, final
GamiPayInfo GamiPayInfo, final String skuid, final
OrderCallbackListener listener)
```

使用示例

```
GamiPayInfo gamiPayInfo = new GamiPayInfo();
// 服务器号:角色所在服的编号
gamiPayInfo.setServerid("987");
// 服务器名 : 角色所在服务器名称
```

```

gamiPayInfo.setServername("降龙");
// 角色名:用户在游戏角色名
gamiPayInfo.setRolename("阿隆索");
// 角色id
gamiPayInfo.setRoleId("12345");
// 角色等级
gamiPayInfo.setRolelevel(14);
//请求充值金额,单位为约定的币种 ( 人民币、台币、美分 ),仅供参考,用户可以在
支付页面修改金额,所以请以卓动调用cp的发货接口时的金额为准!
gamiPayInfo.setPrice(100);
// cp充值透传信息,在回调时会回传给cp,如果不需要,传入null
gamiPayInfo.setExt(null);
//若果游戏只支持固定金额支付,则setPriceFixed(true),若果支持任意金额支付
,则设置为false
gamiPayInfo.setPriceFixed(true);
// 游戏厂商的订单号,要求必须由开发者的业务服务器生成,因订单支付成功后游戏
平台服务器会直接将支付结果通知给开发者的业务服务器,通知参数的cporderid是重
要信息。
String cporderid = getCporderidFromCpServer(); // *该函数由CP实现

gamiPayInfo.setCporderid(cporderid);
/*
 * 填好参数,跳转到支付页面
 */
GamiPlaySDKManager.showPayView(DemoMainActivity.this, gamiPayInfo,
skuid,
    new OrderCallbackListener() {
        @Override
        public void onOrderCallback(String cporderid,
            int statusCode, String msg) {
            switch (statusCode) {
                case GamiPlaySDKStatusCode.ORDER_SUCCESS:
                    // 支付成功
                    break;
                case GamiPlaySDKStatusCode.ORDER_FAILED:
                    // 支付失败或取消支付
                    break;
                case GamiPlaySDKStatusCode.ORDER_NEED_RETRY:
                    // 支付状态需要重新确认,请引导玩家重新点击支付按
                    钮, sdk会重新确认支付状态
                    break;
                case GamiPlaySDKStatusCode.ORDER_EXCEPTION:
                    // 支付发生异常
                    break;
            }
        }
    }
});

```


6.2. 运营活动接口（可选）

请在游戏中添加一个按钮，并在按钮的点击事件里使用下面接口，即可弹出活动界面。利用下面接口中的FacebookGiftListener。监听奖励动作。配合CP后台进行相应的奖励发放。

```
GamiPlaySDKManager.showFacbookActivity(DemoMainActivity.this, new
FacebookGiftListener() {
    @Override
    public void onGiftCallback(String giftId , String msg) {

        //配合后台进行相应的奖励活动
        return;
    }
});
```

回调参数说明

giftId : 后台协定好的礼包编码。

msg : 附加信息，暂不使用。

7. 附录

7.1. googlePlay上架接入说明

取消设备敏感信息的获取

背景原因

google不希望使用敏感信息作为设备标识，推荐使用广告ID

相关说明

<https://developer.android.com/training/articles/user-data-ids?hl=zh-cn>

https://play.google.com/intl/zh-CN_ALL/about/monetization-ads/ads/ad-id/

要求

请勿获取设备的IMEI、AndroidID、MAC地址等设备敏感信息，如果需要使用设备标识，请使用google advertising IDs，或者通过SDK接口获取（getPUID）。

适配target API 26

背景原因

google要2018年8月1日后上架后上架的app必须使用target API 26以上

相关说明

<https://developer.android.com/distribute/best-practices/develop/target-sdk>

影响

升级target API 23以上后，安装应用时默认不给予危险级别权限，需要通过调用接口，以弹窗的方式请求玩家主动授予权限。

要求

原则1：游戏尽可能不要使用google危险级别权限下的功能，包括并不限于：
：读取手机状态和身份、外部存储权限、修改系统设置、定位、查找设备上的账户、读取设备电量、网络状态等信息

原则2：请勿在游戏启动时就弹出数个权限获取框，可以在使用时按需获取，设计良好的交互体验

危险权限

权限组	权限
<u>CALENDAR</u>	<ul style="list-style-type: none"> • <u>READ_CALENDAR</u> • <u>WRITE_CALENDAR</u>
<u>CAMERA</u>	<ul style="list-style-type: none"> • <u>CAMERA</u>
<u>CONTACTS</u>	<ul style="list-style-type: none"> • <u>READ_CONTACTS</u> • <u>WRITE_CONTACTS</u> • <u>GET_ACCOUNTS</u>
<u>LOCATION</u>	<ul style="list-style-type: none"> • <u>ACCESS_FINE_LOCATION</u> • <u>ACCESS_COARSE_LOCATION</u>
<u>MICROPHONE</u>	<ul style="list-style-type: none"> • <u>RECORD_AUDIO</u>
<u>PHONE</u>	<ul style="list-style-type: none"> • <u>READ_PHONE_STATE</u> • <u>READ_PHONE_NUMBERS</u> • <u>CALL_PHONE</u> • <u>ANSWER_PHONE_CALLS</u> • <u>READ_CALL_LOG</u> • <u>WRITE_CALL_LOG</u> • <u>ADD_VOICEMAIL</u> • <u>USE_SIP</u> • <u>PROCESS_OUTGOING_CALLS</u>
<u>SENSORS</u>	<ul style="list-style-type: none"> • <u>BODY_SENSORS</u>
<u>SMS</u>	<ul style="list-style-type: none"> • <u>SEND_SMS</u> • <u>RECEIVE_SMS</u> • <u>READ_SMS</u> • <u>RECEIVE_WAP_PUSH</u> • <u>RECEIVE_MMS</u>
<u>STORAGE</u>	<ul style="list-style-type: none"> • <u>READ_EXTERNAL_STORAGE</u> • <u>WRITE_EXTERNAL_STORAGE</u>

unity游戏可以使用以下AndroidManifest配置关闭默认的启动时弹出权限获取框：

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

7.2. asset文件解压接口

非必接接口，部分游戏需要sdk协助从asset目录解压文件的时候使用。

asset文件解压接口：

接口说明

```
/**
 * @param context
 * @param assetDirPath 需要解压的asset路径
 * @param storageFolderAbsolutePath 解压的目标路径
 * @param totalSize 解压文件尺寸，单位byte
 * @param mObbFilesUnzipListener 解压回调
 */
void unzipAssetsVObbFiles(final Context context,String assetDirPath, String
storageFolderAbsolutePath,long totalSize, ObbFilesUnzipListener mObbFilesUnzipListener);
```

使用用例

```
GamiPlaySDKManager.unzipAssetsVObbFiles(final Context context,String assetDirPath, String
storageFolderAbsolutePath, long totalSize,
, new ObbFilesUnzipListener() {

    /**
     * 全部解压成功
     */
    @Override
    public void onAllUnzipSuccess(ArrayList<String> filesArrayList) {

    }

    /**
     * 整体解压进度，根据传入的计算
     */
    @Override
    public void onProgress(int current, int total) {

    }
}
```

```
/*
 * 单个文件解压成功
 */
@Override
public void onEachFileUnzipFinish(File mFile) {

}

/*
 * 解压失败
 */
@Override
public void onFail(Exception e) {

}
});
```