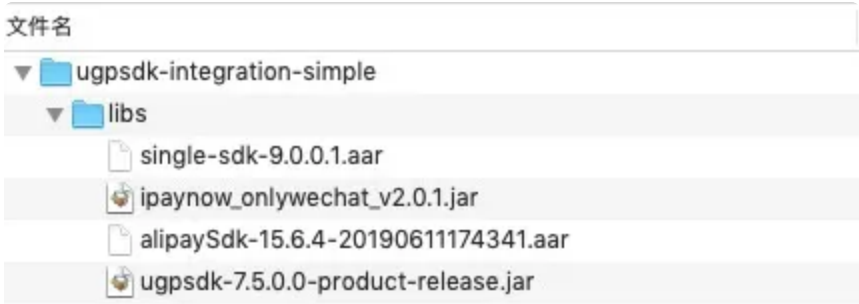


九游单机游戏_SDK_开发参考说明书_Android API_2025

1. 开发环境搭建

1.1 SDK包

将九游游戏SDK接入依赖库包中的 ugpsdk-integration-simple.zip 解压，将得到以下目录：



说明如下：

- libs下的全部.jar文件拷贝到游戏工程libs目录下；
- 在游戏工程的assets目录下创建一个文件夹命名为“UCPaySDK”，将从开放平台下载下来对应该游戏的pay.png支付文件放进去

1.2 Android Studio

将九游游戏SDK接入包中的single-sdk-x.x.aar和alipaySdk-x.x.x-xxxxx.aar文件拷贝到游戏工程的arrlibs目录并修改游戏Module的build.gradle文件，添加repositories域：

```
1 repositories {
2     flatDir {
3         dirs 'aar'
4     }
5 }
```

如果已经存在了repositories域，可添加到已存在的域中即可

为dependencies域添加打包aar的配置：

```
dependencies {
    // 其他配置...

    // 将libs目录下的aar文件打入游戏apk包
    fileTree(dir: 'libs', include: '**/*.aar')
        .each { File file ->
            dependencies.add('compile', [name: file.name.lastIndexOf('.').with {
                it != -1 ? file.name[0..
```

注：aar接入方式时，2个aar内的Androidmanifest.xml文件已经包含SDK接入所需的权限及activity声明，无需在游戏工程再次写入，否则造成冲突； 但其中“cn.uc.gamesdk.activity.PullupActivity”的data android:scheme 需要根据不同游戏配置不同的ng+gameId，故此activity需要在游戏manifest内重新写入，并通过配置tools:node="replace" 来覆盖aar的manifest配置。

1.3 修改AndroidManifest.xml文件

打开您项目的AndroidManifest.xml文件，进行如下修改：

1. 指定游戏图标需使用带有九游角标的图标；
2. 包名加上“.aligames后缀”（已在九游上线过的游戏，保持原包名不改）；
3. 增加权限声明（参见AndroidManifest.xml文件示例中所示的权限为SDK所必需的，游戏可根据需要额外增加其它权限声明）；
4. 增加2个alipay所用的activity声明com.alipay.sdk.app.PayResultActivity、com.alipay.sdk.app.AlipayResultActivity
5. 增加 cn.gundam.sdk.shell.activity.ProxyActivity 的声明(注：launchMode请保持standard或者不设置)。
6. 增加 cn.gundam.sdk.shell.activity.ThemeProxyActivity 的声明(注：launchMode请保持standard或者不设置)。
7. 增加 cn.gundam.sdk.shell.content.FileProvider 的声明
 - a. 检查 android:authorities="\${applicationId}.gamesdk.fileprovider"的最终打包输出
 - b. 正常情况下，\${applicationId} 会被自动替换成游戏的包名

8. 设置supports-screens 节点;
9. 增加 cn.uc.paysdk.SDKActivity 的声明
10. 增加 cn.uc.paysdk.service.SDKService 的声明
11. 增加 cn.gundam.sdk.shell.service.ResProxyService 的声明

修改后的AndroidManifest.xml文件如下所示：

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3  package="cn.lori.fighter.aligames" android:versionCode="1" android:vers
4  ionName="1.0"
5  xmlns:tools="http://schemas.android.com/tools" >
6      <supports-screens
7          android:anyDensity="true"
8          android:largeScreens="true"
9          android:normalScreens="true"
10         android:resizeable="true"
11         android:smallScreens="true" />
12
13         <uses-sdk android:minSdkVersion="14" android:targetSdkVersion="26" />
14
15         <uses-permission android:name="android.permission.INTERNET" />
16         <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"
17     />
18         <uses-permission android:name="android.permission.ACCESS_NETWORK_STAT
19     E" />
20         <uses-permission android:name="android.permission.WRITE_EXTERNAL_STOR
21     AGE" />
22         <uses-permission android:name="android.permission.READ_PHONE_STATE" /
23     >
24         <uses-permission android:name="android.permission.GET_TASKS" />
25         <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDO
26     W" />
27         <uses-permission android:name="android.permission.RECORD_AUDIO"/>
28         <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTIN
29     GS"/>
30
31         <uses-permission android:name="android.permission.READ_SETTINGS" />
32         <uses-permission android:name="android.permission.WRITE_CALENDAR" />
33         <uses-permission android:name="android.permission.REQUEST_INSTALL_PAC
34     KAGES" />
35         <uses-permission android:name="com.android.alarm.permission.SET_ALAR
36     M" />
37         <uses-permission android:name="android.permission.READ_EXTERNAL_STORA
38     GE" />
39
40         <queries>
41             <package android:name="cn.ninegame.gamemanager" />
42         </queries>
```

```

35     <application android:icon="@drawable/ic_launcher_uc" android:label="@
string/app_name" >
36         .....
37         .....
38     <activity
39         android:name="cn.uc.paysdk.SDKActivity"
40         android:configChanges="keyboardHidden|orientation|screenSize"
41         android:theme="@android:style/Theme.Translucent.NoTitleBar"
42         android:windowSoftInputMode="adjustPan|stateAlwaysHidden|adju
stResize" >
43     </activity>
44
45     <service android:name="cn.uc.paysdk.service.SDKService" >
46     </service>
47
48     <activity
49         android:name="cn.gundam.sdk.shell.activity.ProxyActivity"
50         android:configChanges="keyboardHidden|orientation|screenSize"
51         android:label="@string/app_name"
52         android:theme="@android:style/Theme.Translucent.NoTitleBar"
53         android:windowSoftInputMode="adjustResize" >
54     <intent-filter>
55         <action android:name="cn.uc.gamesdk.sdkweb" />
56         <category android:name="android.intent.category.DEFAULT"
/>
57     </intent-filter>
58 </activity>
59
60     <activity
61         android:name="cn.gundam.sdk.shell.activity.ThemeProxyActivit
y"
62         android:configChanges="keyboardHidden|orientation|screenSize"
63         android:label="@string/app_name"
64         android:theme="@android:style/Theme.NoTitleBar"
65         android:windowSoftInputMode="adjustResize">
66     <intent-filter>
67         <action android:name="cn.uc.gamesdk.sdkweb"/>
68         <category android:name="android.intent.category.DEFAULT"/
>
69     </intent-filter>
70 </activity>
71
72     <!-- 请检查当前打包配置中 applicationId 是否为正确的游戏包名，如游戏包名为cn.u
c.gamesdk.demo -->
73     <!-- 打包结果应该是： android:authorities="cn.uc.gamesdk.demo.gamesdk.fi
leprovider" -->
74

```

```

85     <!-- 如有需要可以手动替换 android:authorities="${applicationId}.gamesdk.
86     fileprovider" 中的${applicationId}为游戏包名 -->
87     <provider
88         android:name="cn.gundam.sdk.shell.content.FileProvider"
89         android:authorities="${applicationId}.gamesdk.fileprovider"
90         android:exported="false"
91         android:grantUriPermissions="true"
92         tools:replace="android:authorities">
93         <meta-data
94             android:name="android.support.FILE_PROVIDER_PATHS"
95             android:resource="@xml/sdk_file_paths"/>
96     </provider>
97
98     <!-- android:taskAffinity 填上游戏的包名，如游戏包名为cn.uc.gamesdk.d
99     emo，则下面填 cn.uc.gamesdk.demo.diff -->
100    <!-- data android:scheme 里填上"ng+当前游戏的gameId"，如游戏ID是12345
101    6,则填上ng123456 -->
102    <activity
103        android:name="cn.uc.gamesdk.activity.PullupActivity"
104        android:theme="@android:style/Theme.Translucent"
105        android:taskAffinity="游戏包名.diff"
106        android:excludeFromRecents="true"
107        android:label="PullupActivity"
108        android:launchMode="singleTop"
109        android:exported="true"
110        tools:node="replace">
111        <intent-filter>
112            <action android:name="android.intent.action.VIEW"/>
113            <category android:name="android.intent.category.DEFAULT"
114            />
115            <category android:name="android.intent.category.BROWSABL
116            E"/>
117            <data android:scheme="ng游戏ID" />
118        </intent-filter>
119    </activity>
120
121    <!-- alipay sdk begin -->
122    <activity
123        android:name="com.alipay.sdk.app.H5PayActivity"
124        android:configChanges="orientation|keyboardHidden|navigation"
125        android:exported="false"
126        android:screenOrientation="behind" >
127    </activity>
128    <activity
129        android:name="com.alipay.sdk.app.H5AuthActivity"
130        android:configChanges="orientation|keyboardHidden|navigation"
131        android:exported="false"

```

```

117         android:screenOrientation="behind" >
118     </activity>
119     <activity
120         android:name="com.alipay.sdk.app.PayResultActivity"
121         android:configChanges="orientation|keyboardHidden|navigation"
122         android:exported="true"
123         android:launchMode="singleInstance"
124         android:theme="@android:style/Theme.Translucent.NoTitleBar" >
125         <intent-filter>
126             <category android:name="android.intent.category.DEFAULT"
127             />
128         </intent-filter>
129     </activity>
130     <activity
131         android:name="com.alipay.sdk.app.AlipayResultActivity"
132         android:exported="true"
133         android:launchMode="singleTask"
134         android:theme="@android:style/Theme.Translucent.NoTitleBar" >
135     </activity>
136     <!-- alipay sdk end -->
137     <service
138         android:name="cn.gundam.sdk.shell.service.ResProxyService"
139         android:enabled="true"
140         android:exported="true" />
141

```

2. API总体机制说明

2.1 SDK合规说明

SDK合规规则说明

- 1、初始化时，游戏传入是否已申请权限的标志，SDK程序按需申请所需的权限
 - 避免游戏申请后，再申请，导致重复申请而违规
 - 依赖的权限已经在《九游隐私政策》申请
 - 技术对接，参考SDK接入文档的init接口说明
- 2、如果权限用户没有授权，SDK程序不会调用对应的API，避免违规
- 3、针对用户已经授权的权限，SDK程序通过缓存策略降低对应API的调用，有效控制读取的频率

详情请看：
<https://usdpdown.game.uodoo.com/cpssgame/resources/publicfiles/template/九游游戏SDK合规指南.pdf>

2.2 消息通知机制

SDK的大部分API调用使用了事件通知的方式回调给游戏,游戏通过创建一个SDKEventReceiver对象实现SDK的每个事件类型并通过UCGameSDK.defaultSdk().registerSDKEventReceiver(eventReceiver)接口注册后,就可以获取接口调用的结果。

创建SDKEventReceiver的方法大致如下:

Java

```
1 private SDKEventReceiver eventReceiver = new SDKEventReceiver() {
2     @Subscribe(event = SDKEventKey.ON_INIT_SUCC)
3     private void onInitSucc() {
4         appendText("初始化成功");
5     }
6 };
```

2.1.1 注册与反注册

1. 在游戏Activity onCreate中调用
UCGameSDK.defaultSdk().registerSDKEventReceiver(eventReceiver)注册
2. 在游戏Activity onDestroy 中调用
UCGameSDK.defaultSdk().unRegisterSDKEventReceiver(eventReceiver)反注册

注意：如果回调方法涉及UI操作，请务必保证在UI线程执行；如果回调方法涉及耗时操作，请务必保证在异步线程执行。

2.1.2 关键字段说明如下

字段	说明
----	----

SDKEventReceiver	所有接收SDK事件的对象必须从SDKEventReceiver派生，或者直接如上例创建SDKEventReceiver的对象，对象创建出来后调用registerSDKEventReceiver方法注册后，即完成了事件接收的准备。
Subscribe	注册SDK事件的注解，event表示事件的类型，可选值请参考SDKEventKey类。
onInitSucc	此例中用于处理“初始化成功”事件的方法定义，方法名可以随意取，方法参数规格需参照各SDK功能接口的定义说明。

2.1.3 SDKEventKey事件如下

方法	说明
ON_INIT_SUCC	初始化成功
ON_INIT_FAILED	初始化失败
ON_LOGIN_SUCC	登录成功
ON_LOGIN_FAILED	登录失败
ON_CREATE_ORDER_SUCCESS	订单创建成功
ON_PAY_USER_EXIT	支付界面退出
ON_LOGOUT_SUCC	注销成功
ON_LOGOUT_FAILED	注销失败
ON_EXIT_SUCC	退出成功
ON_EXIT_CANCELED	取消退出，选择继续游戏

3. API使用说明

所有的api都需要通过获取UCGameSdk的实例对象来调用，UCGameSdk对象可以通过以下方法获取：

```
1 UGameSdk sdk = UGameSdk.defaultSdk();
```

此为一个单例对象，可以在任意地方获取调用。

3.1 初始化SDK(必须接入)

首先，您需要在程序开始的地方初始化UGameSdk对象，通过调用initSdk方法初始化九游游戏SDK。在初始化失败的状态下，游戏 **不应** 继续调用SDK的其余API，此方法必须在UI线程中调用。游戏启动后，非初始化失败或异常情况下，**不允许多次调用初始化接口**。

3.1.1 方法定义

```
1 void initSdk(final Activity activity, final SDKParams params)
2     throws AliLackActivityException
```

3.1.2 参数说明

- 输入参数说明:
-

参数	说明
activity	游戏的Activity对象，不可为null。
params	不可传null。

- 异常说明:

异常名称	说明
AliLackActivityException	未传入Activity时抛出此异常。

- SDKParams参数说明（定义在SDKParamKey）：

•

参数定义	参数名称	类型	是否必填	说明
GAME_PARAMS	gameParams	ParamInfo	是	1. setGameId将游戏gameId传入该参数从开放平台-游戏管理-SDK接入-获取参数那里拿 2、 setOrientation设置屏幕方向，PORTRAIT竖屏、LANDSCAPE横屏，不设置默认为竖屏，横屏游戏必须设置为LANDSCAPE
GAME_HAD_REQUEST_PERMISSION	params	boolean	否	true，表示游戏已经申请权限，sdk内部将不再申请权限 默认false

3.1.3 代码示例

Java

```
1  ParamInfo gpi= genGameParams();//请自行生成对象
2  gpi.setGameId(119474); // 从九游开放平台“SDK接入”获取自己游戏的参数信息
3  //设置SDK屏幕方向
4  //LANDSCAPE：横屏，横屏游戏必须设置为横屏,PORTRAIT：竖屏
5  gpi.setOrientation(UCOrientation.PORTRAIT);
6  SDKParams sdkParams = new SDKParams();
7  sdkParams.put(SDKParamKey.GAME_PARAMS, gpi);
8  //如果游戏已经申请了权限，不想sdk主动请求权限，要通过参数告知sdk
9  sdkParams.put(SDKParamKey.GAME_HAD_REQUEST_PERMISSION, true);
10 UCGameSdk.defaultSdk().initSdk(this.getActivity(), sdkParams);
```

init方法没有返回值，初始化结果通过事件通知回调给游戏：

```

1 private SDKEventReceiver eventReceiver = new SDKEventReceiver() {
2     @Subscribe(event = SDKEventKey.ON_INIT_SUCC)
3     private void onInitSucc() {
4         appendText("初始化成功");
5     }
6     @Subscribe(event = SDKEventKey.ON_INIT_FAILED)
7     private void onInitFailed(String desc) {
8         appendText("初始化失败:" + desc);
9     }
10 };

```

3.2 登录（必接）

本文档无包括登录接口，如确认游戏需要接入九游账号体系，请先确认游戏支持一个账号对应一个角色进度，同一个设备可更换多个不同的账号且每个账号与角色一一对应，角色信息保存在服务端而非客户端本地，切换设备不会导致角色丢失。

如确认需要接入登录且符合上述要求，请联系对接技术获取登录相关文档，并提供gameId给对接技术配置登录方可使用，否则调用接口直接走登录失败回调，desc报错“当前游戏无法使用登录”。

技术联系方式：<https://game.open.uc.cn/document/doc/detail/29>。

3.2.1 方法定义

```

1 void login(Activity activity, SDKParams params)
2     throws AliLackAcitivityException, AliNotInitException

```

3.2.2 参数说明

输入参数说明：

参数	说明
activity	游戏的Activity对象，不可为null。
params	可传null。

异常说明：

异常名称	说明
AliNotInitException	未初始化或正在初始化时调用此接口将抛出此异常。
AliLackAcitivityException	Activity对象为null时将抛出此异常。

3.2.3 代码示例

账号登录

```

1 try {
2     UGameSdk.defaultSdk().login(this.getActivity(), sdkParams);
3 } catch (AliNotInitException e) {
4     //未初始化或正在初始化时，异常处理
5 } catch (AliLackAcitivityException e) {
6     //activity为空，异常处理
7 }

```

像初始化接口那样在事件接收对象处理SDK的回调结果：

```

1 private SDKEventReceiver eventReceiver = new SDKEventReceiver() {
2
3     @Subscribe(event = SDKEventKey.ON_LOGIN_SUCC)
4     private void onLoginSucc(String sid) {
5         //sid即token，需发送给游戏服务器做登录校验获取accountId用户唯一标识，客户端无法获取用户唯一标识
6         appendText("登录成功,sid:" + sid);
7     }
8
9     @Subscribe(event = SDKEventKey.ON_LOGIN_FAILED)
10    private void onLoginFailed(String desc) {
11        appendText("登录失败:" + desc);
12    }
13
14 };

```

当登录成功时SDKEventKey.ON_LOGIN_SUCC事件被触发，同时通过带回登录sid给游戏，sid为动态token，每次登录都会改变，需发送给游戏服务器做登录校验获取accountId用户唯一标识，客户端无法获取用户唯一标识，参考服务端接口协议1.3.1章节——用户会话验证接口

3.3 充值(必接)

3.3.1 充值操作

3.3.2 方法定义

Java

```
1 void pay(Activity activity, SDKParams params)
2     throws AliLackAcitvityException, AliNotInitException, IllegalArgumentE
    xception
```

3.3.3 参数说明

- 输入参数说明：
-

参数	说明
activity	游戏的Activity对象，不可为null。
params	不可传null。

- 异常说明：
-

异常名称	说明
AliNotInitException	未初始化或正在初始化时调用此接口将抛出此异常。
IllegalArgumentException	SDKParams为空时调用此接口将抛出异常。
AliLackAcitvityException	传入的activity为空时抛出此异常。

- SDKParams参数说明（定义在SDKParamKey）：

参数定义	类型	是否必填	说明
APP_NAME	String	是	应用名称,将显示在支付界面
AMOUNT	String	是	充值金额,最多保留小数点后2位, 单位为元。 例：10.00，整数金额可不传小数点部分，建议金额由服务端格式化好，客户端直接使用 游戏服务端（若有回调）必须根据九游回调的金额值进行校验并下发相应价值的虚拟货币。
PRODUCT_NAME	String	是	商品名称,将显示在支付界面
NOTIFY_URL	String	否	服务器通知地址，如果为空则以开放平台配置地址作为通知地址，优先取客户端地址(建议通过游戏服务端配置参数后读取,无服务器接收的不传即可)，长度不超过100
ATTACH_INFO	String	否	服务器通知地址透传参数
CP_ORDER_ID	String	否	cp充值订单号，需要保证用户每次充值订单号的唯一性，长度不超过30

3.3.2 代码示例

```
1  SDKParams params = new SDKParams();
2  sdkParams.put(SDKProtocolKeys.APP_NAME, "坦克大战");
3  sdkParams.put(SDKProtocolKeys.PRODUCT_NAME, "金币");
4  sdkParams.put(SDKProtocolKeys.AMOUNT, "2.33");
5  sdkParams.put(SDKProtocolKeys.NOTIFY_URL, "http://192.168.1.1/notifypage.do");
6  sdkParams.put(SDKProtocolKeys.ATTACH_INFO, "你的透传参数");
7  sdkParams.put(SDKProtocolKeys.CP_ORDER_ID, "20160000101001");
8  // 各字段说明, 见SDKProtocolKeys参数表
9
10 try {
11     UCGameSdk.defaultSdk().pay(this.getActivity(), sdkParams);
12 } catch (IllegalArgumentException e) {
13     //传入参数错误异常处理
14 } catch (AliNotInitException e) {
15     //未初始化或正在初始化时, 异常处理
16 } catch (AliLackActivityException e) {
17     //activity为空, 异常处理
18 }
```

支付回调:


```

1 private SDKEventReceiver eventReceiver = new SDKEventReceiver() {
2
3     @Subscribe(event = SDKEventKey.ON_CREATE_ORDER_SUCC)
4     private void onPaySucc(Bundle data) {
5         Log.i(TAG, "此处为支付成功回调:( callback data = " + data.getString("response"));
6
7         // 在此执行发货，执行成功后需要往data添加如下参数，否则会重复通知订单
8         data.putString("result", Response.OPERATE_SUCCESS_MSG);
9
10        // 如果执行发货失败需要往data添加如下参数
11        // data.putString("result", Response.OPERATE_FAIL_MSG);
12    }
13
14    @Subscribe(event = SDKEventKey.ON_PAY_USER_EXIT)
15    private void onPayFailed(String msg) {
16        Log.i(CLASS_NAME, "支付失败: msg = " + msg);
17    }
18 };

```

支付成功响应参数json说明（定义在PayResponse）：

CP_ORDER_ID	CP_ORDER_ID	String	调用支付时传入的订单ID
TRADE_ID	TRADE_ID	String	交易号
PAY_MONEY	PAY_MONEY	String	支付金额
PAY_TYPE	PAY_TYPE	String	支付类型
ORDER_STATUS	ORDER_STATUS	String	订单状态：00成功，01失败，99退款
ORDER_FINISH_TIME	ORDER_FINISH_TIME	String	订单完成时间
PRO_NAME	PRO_NAME	String	道具名称
EXT_INFO	EXT_INFO	String	商品扩展信息
ATTACH_INFO	ATTACH_INFO	String	附加透传信息

3.4 退出SDK(必须接入)

当游戏退出前 必须 调用该方法，进行清理工作。如果游戏直接退出，而不调用该方法，可能会出现未知错误，导致程序崩溃。

3.4.1 调用说明（重要）

该接口需要游戏在退出前调用，游戏等待该接口的回调通知后，再执行游戏的退出。

此接口需要在UI线程中调用。

3.4.2 接口定义

Java

```
1 public void exit(Activity activity, SDKParams params)
2     throws AliLackAcitivityException, AliNotInitException
```

3.4.3 参数说明

- 输入参数说明：
-

参数	说明
activity	游戏Activity对象
params	传null。

- 异常说明：
-

异常名称	说明
AliNotInitException	未初始化或正在初始化时调用此接口将抛出此异常。

3.4.4 代码示例

```

1 try {
2     UCGameSdk.defaultSdk().exit(this.getActivity(), null);
3 } catch (AliNotInitException e) {
4     //未初始化或正在初始化时，异常处理
5 } catch (AliLackAcitivityException e) {
6     //activity为空，异常处理
7 }

```

添加SDK退出回调：

```

1 private SDKEventReceiver eventReceiver = new SDKEventReceiver() {
2     @Subscribe(event = SDKEventKey.ON_EXIT_SUCC)
3     private void onExitSucc() {
4         appendText("SDK退出");
5     }
6     @Subscribe(event = SDKEventKey.ON_EXIT_CANCELED)
7     private void onExitCanceled() {
8         appendText("放弃退出，继续游戏");
9     }
10 };

```

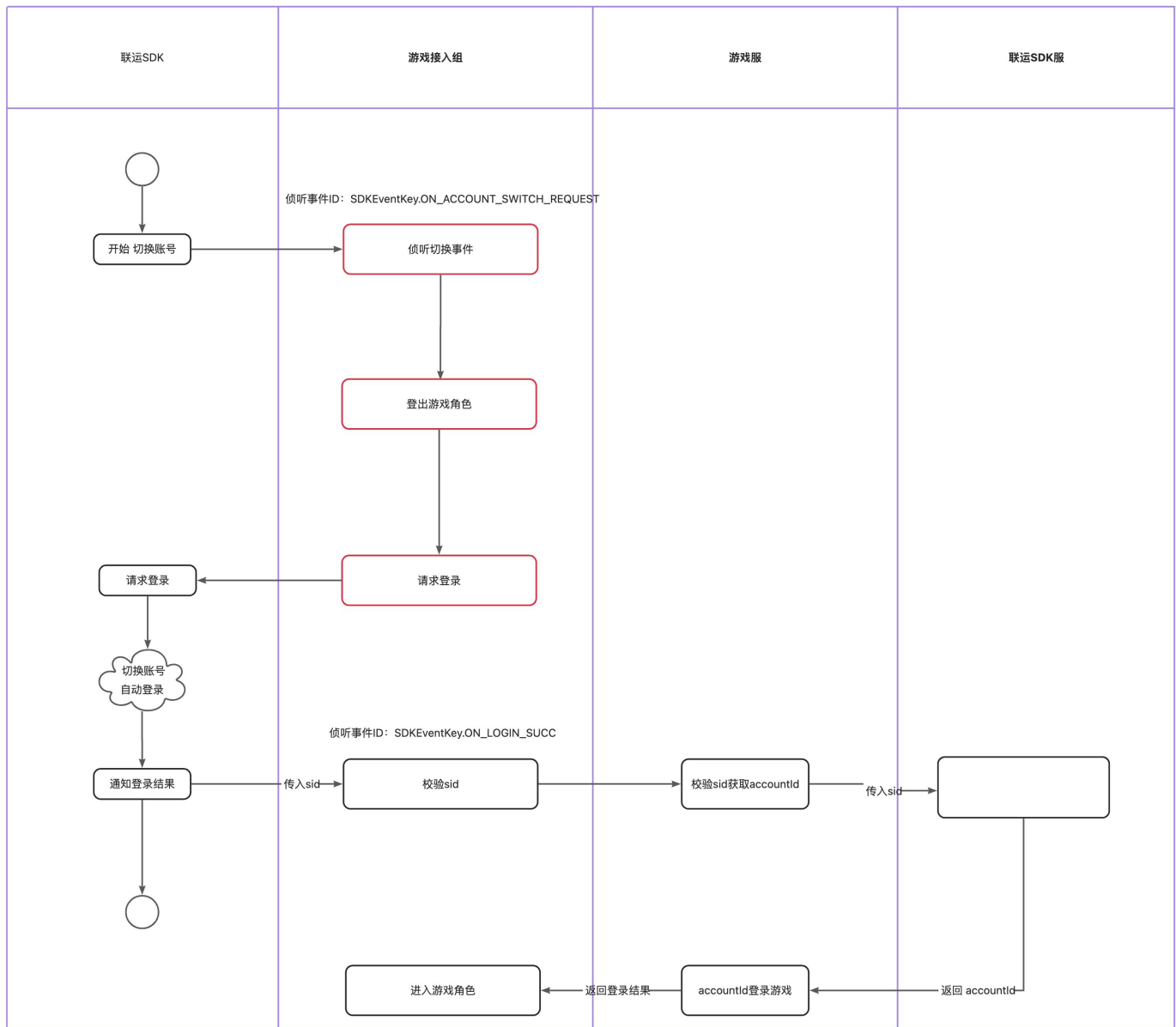
3.5 切换账号接口（必须接入）

SDK提供的被动切换账号的接口，需要游戏响应账号切换动作，最终协同完成用户账号和游戏角色的切换；更详细的信息可以查看：[《联运SDK切换账号接入指南-简化方案》](#)。

3.5.1 调用说明（重要）

该接口需要游戏响应切换指令，游戏需要退出当前游戏角色，并重新调用登录请求。

具体流程图：



3.5.2 切换账号的回调

代码示例

```
1 private SDKEventReceiver eventReceiver = new SDKEventReceiver() {  
2  
3     .....省略若干代码.....  
4  
5     @Subscribe(event = SDKEventKey.ON_ACCOUNT_SWITCH_REQUEST)  
6     private void onAccountSwitchRequest(final String sid) {  
7         // 注意: sid是兼容旧版本保留下来的, 9.5.10.2版本及以上请不要使用sid!  
8     }  
9  
10    .....省略若干代码.....  
11  
12 };  
13
```

当用户从游戏启动器启动游戏，同时要求切换账号时，
SDKEventKey.ON_ACCOUNT_SWITCH_REQUEST事件被触发；【注意：sid是兼容旧版本保留下来的，9.5.10.2版本及以上请不要使用sid！】

场景示例

在声明SDK回调的地方，进行切换账号的流程处理：

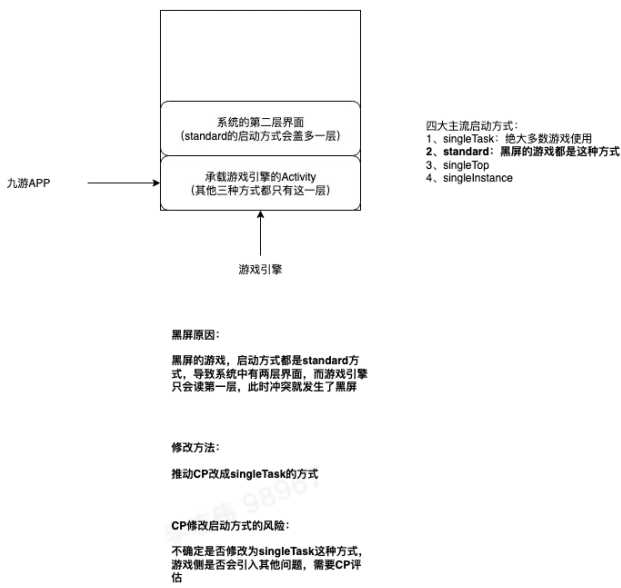
```

1 private SDKEventReceiver eventReceiver = new SDKEventReceiver() {
2
3     .....省略若干代码.....
4
5     // 步骤1, 侦听SDK切换账号指令
6     @Subscribe(event = SDKEventKey.ON_ACCOUNT_SWITCH_REQUEST)
7     private void onAccountSwitchRequest(final String sid) {
8         //注意: sid是兼容旧版本保留下来的, 9.5.10.2版本及以上请不要使用sid!
9         //步骤2, CP接入组先退出当前游戏角色
10        //这里需要注意:
11        //    1. 【重要】必须退出当前游戏角色, 否则会因为“切换场景未登出游戏角色”被打
    回
12        logoutGameRole();
13        //步骤3, 和正常登录一样, 调用sdk做登录请求
14        //这里需要注意:
15        //    1. 如果游戏的退出和重新登录在native层是异步的,
16        //        那requestLogin()应该由cp自行决定在合适的节点调用,
17        //        不要求必须放在onAccountSwitchRequest()方法中;
18        //    2. 【重要】如果游戏首页本身会自动触发登录,
19        //        那onAccountSwitchRequest()中不需要额外调用requestLogin();
20        //        否则会因为“切换账号场景重复登录”被打回;
21        //    3. requestLogin()需要保证在角色已退出的情况下调用;
22        requestLogin();
23    }
24
25    .....省略若干代码.....
26
27 };
28
29 private void logoutGameRole() {
30     CP接入组需要实现游戏角色的登出;
31 }
32
33 private void requestLogin() {
34     try {
35         UCGameSdk.defaultSdk().login(this.getActivity(), null);
36     } catch (AliNotInitException e) {
37         //未初始化或正在初始化时, 异常处理
38     } catch (AliLackActivityException e) {
39         //activity为空, 异常处理
40     }
41 }

```

3.5.3 切换账号后游戏黑屏问题（重要）

为了避免游戏在运营期，在用户发起切换账号后遇到游戏黑屏等异常，强烈建议接入时，检查游戏的Manifest文件，游戏主Activity的launchMode配置是否为singleTask，其原理如下：



3.6 可选个人信息配置（可选）

3.6.1 调用说明（重要）

该接口需要初始化调用之前调用。

此接口需要在UI线程中调用。

3.6.2 接口定义

```
Java |
1 public void setPrivacyManager(SdkPrivacyManager privacyManager) throws AliNotInitException
```

3.6.3 参数说明

输入参数说明：

参数	说明
----	----

privacyManager	个人信息配置管理器
----------------	-----------

异常说明：

异常名称	说明
AliNotInitException	初始化成功后调用此接口将抛出此异常。

3.6.4 代码示例


```

1 // 必须在初始化调用之前设置
2 try {
3     // 设置SdkPrivacyManager后，SDK内降不再采集可选的个人信息，
4     // 直接采用接入方注入的SdkPrivacyManager相关方法提供的返回值
5     UCGameSdk.defaultSdk().setPrivacyManager(new SdkPrivacyManager() {
6         @Override
7         public boolean hadShowPrivacyDlg() {
8             // 返回游戏是否已经展示过隐私弹窗
9             return super.hadShowPrivacyDlg();
10        }
11
12        @Override
13        public String getImei() {
14            // 返回游戏采集到的Imei，不采集请直接传空字符串
15            return "";
16        }
17
18        @Override
19        public String getImsi() {
20            // 返回游戏采集到的Imsi，不采集请直接传空字符串
21            return "";
22        }
23
24        @Override
25        public String getAndroidId() {
26            // 返回游戏采集到的AndroidId，不采集请直接传空字符串
27            return "";
28        }
29    });
30 } catch (AliNotInitException e) {
31     e.printStackTrace();
32 }
33 // 初始化
34 UCGameSdk.defaultSdk().initSdk(this.getActivity(), sdkParams);
35 // 禁用SSID和BSSID采集
36 GamePrivacyManager NUL.disableWifiCollection();

```

3.7 信息采集配置（可选）

可选个人信息字段	个人信息采集目的	功能配置方案及示例
----------	----------	-----------

设备信息：IMEI	用以统计用户（oaid）维度的游戏激活以及游戏崩溃等UV+PV数据。	<pre>//设置联运SDK初始化时获取设备的imei //仅需调用一次即可 UCGameSdk.defaultSdk().setPrivacyManager(new SdkPrivacyManager() { @Override public String getImei() { // 返回游戏采集到的Imei，不采集请直接传空字符串 return ""; } });</pre>
设备信息：IMSI	用以统计用户（oaid）维度的游戏激活以及游戏崩溃等UV+PV数据。	<pre>//设置联运SDK初始化时获取设备的imei //仅需调用一次即可 UCGameSdk.defaultSdk().setPrivacyManager(new SdkPrivacyManager() { @Override public String getImei() { // 返回游戏采集到的Imei，不允许采集请直接传空字符串 return ""; } });</pre>
设备信息：AndroidId	用以统计用户（oaid）维度的游戏激活以及游戏崩溃等UV+PV数据。	<pre>//设置联运SDK初始化时获取设备的AndroidId //仅需调用一次即可 UCGameSdk.defaultSdk().setPrivacyManager(new SdkPrivacyManager() { @Override public String getAndroidId() { // 返回游戏采集到的Imei，不采集请直接传空字符串 return ""; } });</pre>

设备信息：OAID	用以统计用户（oaid）维度的游戏激活以及游戏崩溃等UV+PV数据。	<pre>//允许联运SDK登录时获取设备的oaid //仅需调用一次即可 GamePrivacyManager.enableOaidCollection() //不允许联运SDK登录时获取设备的oaid //仅需调用一次即可 GamePrivacyManager.disableOaidCollection()</pre>
网络标识：SSID、BSSID	用于获取用户网络标识，以根据网络标识过滤风控策略，保障用户账号登录、支付过程安全	<pre>//允许联运SDK登录和支付过程获取设备的网络标识 //仅需调用一次即可 GamePrivacyManager.enableWifiCollection() //不允许联运SDK登录和支付过程获取设备的网络标识 //仅需调用一次即可 GamePrivacyManager.disableWifiCollection()</pre>

4. 接入规范

“游戏客户端”集成“游戏SDK”开发时，须遵循以下要求：

4.1. 游戏包名称命名规范

游戏客户端的AndroidManifest.xml中游戏包名定义加上“.aligames”后缀。如原游戏包名为 package="com.lori.app", 则修改为 package="com.lori.app .aligames ", 已在九游上线过的游戏，保持原包名不改

4.2. icon图片

游戏图标需增加九游角标，角标位置在接入包内“角标logo及使用说明”文件夹下，添加角标后在AndroidManifest.xml中指定。如android:icon="@drawable/iconuc"。其中iconuc为在原游戏图标基础上制作的能表明九游渠道的图标文件。

以上两点规范都表现在AndroidManifest.xml中，以下为一个AndroidManifest.xml的例子：

```
Java |
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3  package="cn.lori.fighter.aligames" android:versionCode="1" android:versi
   onName="1.0" >
4      .....
5      <application android:icon="@drawable/ic_launcher_uc"
6                  android:label="@string/app_name" >
7          .....
8          .....
9      </application>
10 </manifest>
```

4.3. 代码混淆要求

如果游戏发布时采用proguard进行代码混淆，请在proguard配置文件加入以下代码，以避免对SDK进行混淆，否则会造成SDK部分功能不正常。

```
1 -libraryjars './libs/alipaySdk-20160825.jar' #填写支付宝sdk的路径
2 -libraryjars './libs/classes.jar' #如果不是采用aar接入, 填写游戏SDK jar 的路径
3 -keepattributes InnerClasses,Signature,SourceFile,Exceptions,LineNumberTable,*Annotation*
4 -keep class * extends cn.gundam.sdk.shell.even.SDKEventReceiver {*; }
5 -keep class cn.gundam.sdk.shell.utdid.**{*;*;}
6 -keep class android.**{
7     <methods>;
8     <fields>;
9 }
10 -keep class cn.uc.**{
11     <methods>;
12     <fields>;
13 }
14 -keep class cn.gundam.**{
15     <methods>;
16     <fields>;
17 }
18 -keep class com.alipay.**{
19     <methods>;
20     <fields>;
21 }
22 -keep class com.ta.**{
23     <methods>;
24     <fields>;
25 }
26 -keep class com.ut.**{
27     <methods>;
28     <fields>;
29 }
30 -keep class org.json. **{
31     <methods>;
32     <fields>;
33 }
```

5. 游戏客户端开发要点（重要，必读）

“游戏客户端”集成“SDK客户端”的开发工作要点如下：

1. 实现“初始化SDK”功能：（必须接入）

a. 实现初始化回调处理； b. 调用初始化SDK方法；

2. 实现“充值”功能（如开充值必须接入）：

a. 实现充值回调处理； b. 设置充值订单信息； c. 调用充值方法；

3. 实现“退出SDK”功能（必须接入）

a. 调用退出SDK方法；

5.1 【温馨提示1】 必须在UI线程中调用的接口列表如下：

- 初始化接口（initSDK）；
- 充值接口（pay）；
- 退出接口（exitSDK）

5.2 【温馨提示2】：

引用SDK后，发生编译错误时，可以试试执行一下IDE的“Clean”（或类似方法）以清除不一致的编译缓存文件。

附录一：接入注意事项及常见问题解决方案

请查看开放平台单机SDK接入常见问题集：<https://game.open.uc.cn/document/doc/detail/324>