SDK& Game 消息流协议

Android and IOS

Android:

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 描述 | 备注 |
| 2018年9月7日 星期五 | 1.0.5 | 初始版本 | 刘涵 |
| 2018年9月12日 星期三 | 1.0.6 | 修改内部广告缓存，游戏不需要显示调用adRequest，在需要展示广告场景中尽量早的调用adAvailable，在要展示广告的地方先调用adAvailable，返回true时再调用displayAd | 刘涵 |
| 2018年9月13日 星期四 | 1.1.0 | 增加退出游戏和闪屏结束的回调函数 | 刘涵 |
| 2018年9月19日 星期三 | 1.1.4 | 优化接入 | 刘涵 |
| 2018年9月28日 星期五 | 1.1.5 | 增加对广告位可见性的设置 | 刘涵 |
| 2019年1月29日 星期二 | 1.1.9 | 增加对gaid，应用及系统版本信息获取 | 刘涵 |
| 2019年5月13日 星期一 | 1.2.0 | 增加AppsFlyer打点 | 杨晓晶 |
| 2019年12月2日 星期一 | 1.3.0 | 将banner广告展示方法单独提出 | 董鹏飞、刘鹏 |
| 2020年9月14日星期一 | 1.4.0 | 增加抖音log上报接口 | 罗衍运、姜若泽 |
| 2020年11月3日星期二 | 1.5.0 | 增加年龄上报，地区查询，查询是否需要填写年龄 | 罗衍运、姜若泽 |
| 2021年6月3日星期四 | 1.5.1 | 增加Facebook登陆接口 | 罗衍运 |
| 2021年12月7日 |  | 广告位整理 | 关林 |

（请严格按照文档流程接入！！！）

1:Android & Game 消息相关类

com.cocoa.ad.game.GGame.java

1:在游戏主Activity的onCreate方法里设置Context

GGame.*getInstance*().setContext(**this**);

2:消息从Game流向Android

1：判断给定广告位是否有广告

**public boolean** adAvailable(String adSite);

接口说明：此接口为请求广告接口，如果当前adSite位置有广告，返回true；没有广告则返回false，并会调用加载方法去请求广告，作为缓存

| 参数 | 描述 |
| --- | --- |
| adSite | 游戏中触发广告的标识 |

2： 展示给定广告位的广告

**public void** displayAd(String adSite);

接口说明： 此接口为展示广告接口，如果AdAvailable()返回true，可调用此接口展示广告.

| 参数 | 描述 |
| --- | --- |
| adSite | 游戏中触发广告的标识 |

3：设置Banner广告位可见性

**public void** changeSiteVisibility(String adSite, int visibility);

接口说明： 此接口为设置banner广告位是否显示或隐藏的接口；如果AdAvailable()返回true，可调用此接口来设置广告的显示隐藏

注：调用此接口必须优先调用AdAvailable()是否为true，如果没有调用AdAvailable()接口或者AdAvailable()接口返回值为false，调用此接口将无效。

| 参数 | 描述 |
| --- | --- |
| adSite | 游戏中触发广告的标识 |
| visibility | 0:可见 1:不可见 （广告不会展示，如果是banner 则banner的位置不可见） |

4:关闭原生广告

**public** void closeNativeAd();

5:关闭banner

**public void** closeBannerAd();

6:进入广告场景

**public** void entryAdScenario(String Scenario);

系统相关

1:获取系统版本

**public** String getSystemVersion();

2:获取应用版本

**public** String appVersion();

3:获取GAID

**public** String getDeviceGAID();

4:事件打点

**public** void logEvent(String eventName, String eventType);

5:调用系统震动

**public** static void runShake(Context context, long time, int power);

6:年龄上报

**public void** submitAge(**int** age)

| 参数 | 描述 |
| --- | --- |
| age | 年龄 |

11:登录接口

**public void** login(String type)

| 参数 | 描述 |
| --- | --- |
| type | qq：qq登录  wechat：微信登录  onekey：一键登录 |

4：消息从Android/iOS流向Game

1：一次广告请求，加载到广告时调用

UnityPlayer.*UnitySendMessage*(**“GSdk"**, **"onAdAvailable"**, params);

2：广告点击时调用

UnityPlayer.*UnitySendMessage*(**"GSdk"**, **"onAdClicked"**,params);

3：广告关闭时调用

UnityPlayer.*UnitySendMessage*(**"GSdk"**, **"onAdClosed"**, params);

4：广告展示时调用

UnityPlayer.*UnitySendMessage*(**"GSdk"**, **"onAdImpressed"**, params);

5：广告展示失败调用

UnityPlayer.*UnitySendMessage*(**"GSdk"**,**"onAdShowFailed"**, params);

6：视频播放完成时调用

UnityPlayer.*UnitySendMessage*(**"GSdk"**,**"onAdPlayCompleted"**, params);

7：广告加载错误调用

UnityPlayer.*UnitySendMessage*(**"GSdk"**,**"onAdError"**, params);

注：params 格式json，测试用的sdk,params为：{“adSite”:”广告请求时传递的adSite”，“adReqTime”:1234567890123/\*onAdAvailable 消息包含\*/}

15:微信登录成功

UnityPlayer.*UnitySendMessage*(**"GSdk"**,**"**onWeChatLoginSuccess**"**, params);

注：params 格式json,params为：

{“errorCode”:”0”,”data”:{"code":””}}

16:微信登录失败

UnityPlayer.*UnitySendMessage*(**"GSdk"**,**"**onWeChatLoginFail**"**, params);

注：params 格式json,params为：

{“errorCode”:””,”errorMsg”:“”}

errorCode错误码为wx登录返回错误码

errorMsg为wx登录返回错误msg

17:网络消息数据填充

UnityPlayer.*UnitySendMessage*(**"GSdk"**,**"**onCommonCallback**"**, params);

18:城市ID 省份ID 数据返回

UnityPlayer.*UnitySendMessage*(**"GSdk"**,**"**onIPCallback**"**, params);

IOS:

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 描述 | 备注 |
| 2018年年10⽉月24⽇日 星期三 | 1.0.0 | 初始版本 | 刘涵 |
| 2018年年12⽉月26⽇日 星期三 | 1.0.1 | 优化接⼊入接口声明；  增加震动接口和打点接口 | 刘涵 |
| 2019年年4⽉月9⽇日 星期二 | 1.0.12 | 增加获取设备标识的接  口；  增加打开浏览应⽤用商店接  口 | 刘涵 |
| 2019年11⽉月29⽇日 星期五 | 1.0.13 | 增加 display\_banner 接口 | Tony |
| 2020年9月14日星期一 | 1.0.14 | 增加抖音log上报接口 | 罗衍运 |
| 2020年11月3日星期二 | 1.0.15 | 增加年龄上报、地区信息查询、是否显示年龄弹窗接口 | 罗衍运 |

# 1：消息从Game流向iOS

## 1:设置UIViewController

extern void set\_current\_viewcontroller(const UIViewController \*controller);

游戏viewcontroller

controller

**描述**

**参数**

说明：

此接口主要是设置接入的第三方sdk 在那个view里显示。

## 2: 获取指定广告位的offer 预加载状态。

extern bool availbale\_at\_position(const char \*position);

游戏中触发广告的标识

position

**描述**

**参数**

说明：

调用此接口后，如果对应的广告位的已经预加载并已经准备好了广告内容，则返回true，如果没有准备好，则返回false，同时广告sdk会异步去请求广告服务器，然后拉取offer，预加载到对应的广告，等待下次调用此接口的时候使用。

## 3:展示给定广告位的广告(不包括banner)

extern void display\_at\_position(const char \*position);

游戏中触发广告的标识

position

**描述**

**参数**

说明：

如果对应广告位的offer已经预加载并准备好，则可以通过调用此接口来在view中显示预加载的广告。

## 4:展示Banner广告位的广告

extern void display\_banner();

说明：

调用此函数后bannner 就会在后台开始加载，所有的广告物料加载完成后会自动展示。

## 4:设置广告位可见性

extern void change\_visibility\_of\_position(const char \*position,int visibility);

|  |  |
| --- | --- |
| **参数 描述** | |
| adSite | 游戏中触发广告的标识 |
| visibility | 0:可见 1:不可见 （广告位不会展示，如果是banner 则banner广告位不可见） |

说明：

此接口主要功能是设置对应位置的广告位是否可见（注意如果在对应的广告位没有预加载offer则对应的广告位也将为不可见的状态）

## 5:调用系统振动

extern void call\_system\_vibrate(void);

## 6:日志打点

extern void log\_event(const char \*name,const char \*type);

事件所属的类别

type

事件名称

name

**描述**

**参数**

## 7:获取设备标识

extern char \*open\_uuid(void);

返回当前设备的标识，卸载后重新安装会发生改变。

## 8:跳转应用商店

extern void open\_url(const char \*url);

click url

url

**描述**

**参数**

## 9:内购日志打点

 extern void log\_purchase(const char \*pid,const char \*type,float price,const char \*currency , bool isSuccess);

| 参数 | 描述 |
| --- | --- |
| pid | 产品ID（内购ID） |
| type | 商品类型 |
| price | 金额 |
| currency | 币种 |
| isSuccess | 支付是否成功 |

## 10:设置 level

 extern void set\_level(int level);

| 参数 | 描述 |
| --- | --- |
| level | 游戏等级 |

**11:抖音log上报 广告位按钮展示 广告位按钮点击 等事件（wobble man 使用）**

extern void log\_douyin\_ad\_event(const char \*site,const char \*type);

事件所属的类别

type

广告位字段

site

**描述**

**参数**

**12：发送邮件**

 extern void send\_feedback(void);

**13：Ohayoo 埋点上报（死神奥利给），**

extern void log\_douyin\_ad\_event(const char \*jsonMsg,const char \*type); // 根据事件类型上报

| 参数 | 描述 |
| --- | --- |
| jsonMsg | json字符串 |
| type | 事件 类型 |

## 14:是否展示年龄输入弹窗

extern bool isShowAntiAddictionAgePopup(void);

## 15:提交上报年龄

extern void submitAgeInfo(int age);

| 参数 | 描述 |
| --- | --- |
| age | 年龄 |

## 16:查询国家编码信息

extern char \*getAddressIpInfo(void);

## 17:三方登陆接口

extern void login(const char \*type)

| 参数 | 描述 |
| --- | --- |
| type | qq：qq登录  wx：微信登录  onekey：一键登录  appleLogin:苹果登录  FBLogin：facebook登录 |

## 18:获取阿里推送设备id

extern char \*getAliDeviceId(void);

## 19:添加设备推送别名

extern void addAlias(const char \*alias)

| 参数 | 描述 |
| --- | --- |
| alias | 1、单个设备最多添加128个别名，同一个别名最多可被添加到128个设备。  2、别名支持的最大长度为128字节 |

## 20:删除推送别名

extern void removeAlias(const char \*alias);

| 参数 | 描述 |
| --- | --- |
| alias | alias = null or alias.length = 0 时，删除设备全部别名 |

# 2：消息从Android/iOS流向Game

## 1：一次广告请求，加载到广告时调用

UnityPlayer.*UnitySendMessage*(**“GSdk"**, **"onAdAvailable"**, params);

## 2：广告点击时调用

UnityPlayer.*UnitySendMessage*(**"GSdk"**, **"onAdClicked"**,params);

## 3：广告关闭时调用

UnityPlayer.*UnitySendMessage*(**"GSdk"**, **"onAdClosed"**, params);

## 4：广告展示时调用

UnityPlayer.*UnitySendMessage*(**"GSdk"**, **"onAdImpressed"**, params);

## 5：视频播放完成时调用

UnityPlayer.*UnitySendMessage*(**"GSdk"**,**"onAdPlayCompleted"**, params);

## 6:跳转商店

UnityPlayer.*UnitySendMessage*(**"GSdk"**,**"openStoreURL"**, params);

### params中的code 取直 如下：0: 跳转成功。1:跳转失败 2:不知道是否成功

注：params 格式json，广告测试用的sdk,params为：{“adSite”:”广告请求时传递的adSite”};跳转params格式为：{“url”:”xxxxx”,”code”:”x”}.