

接入说明

1. 导出并复制项目

从 Unity 导出 Android 项目（A），并复制一份（B），作为 Android Studio 编辑的项目。之后每次导出都用 A，然后将 A 项目里 src\main\assets\bin 中的文件拷贝到项目 B 中相应的位置（因为每次导出，都只有这一部分数据会发生变化）。

2. 合并资源文件

将 res 中的文件全部复制到项目里 src\main\res 中，如果有冲突，则直接覆盖。

资源文件已做处理，不会影响游戏运行。

3. 引入 SDK

将文件 tct_v1.1.0.jar 拷贝到项目 libs 文件夹下。

在项目中的 AndroidManifest.xml 文件中加入相关权限：

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
```

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

在 app 中的 AndroidManifest.xml 文件中加入 SDK 的登录界面声明：

```
<activity android:name="com.tadpolechain.LoginActivity"
    android:theme="@style/LoginDialog" />
```

4. 合并 activity

将 UnityPlayerActivity 文件和项目中的 UnityPlayerActivity 文件合并，替换除了包名（即第一行的 package）以外的所有内容，并将 onCreate 方法里的 SDK 初始化语句中 app key 和 secret key 替换为自己的。

【注意】该初始化方法的最后一个参数，意思为是否测试中。测试的话则为 true，调用测试服务器数据。

5. 修改版本号 and 游戏名称

如果需要修改版本号，则打开 AndroidManifest.xml 文件，修改以下内容：

```
manifest
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.game.first"
    android:versionName="1.0"
    android:versionCode="1"
    android:installLocation="preferExternal">
```

如果修改游戏名称，则打开文件 src\main\res\values\strings.xml，修改 app_name:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">好游戏</string>
</resources>
```

6. 引入 Unity 封装文件

将 TCTForUnity.cs 拷贝到项目相应目录。TCTForUnity 中的方法已经对接到 SDK，可以直接调用。调用范例如 TestClick.cs 文件所示。

7. 回调方法

部分 SDK 方法的调用需要回调，比如登录功能，TestClick.cs 中的调用方法如下所示：

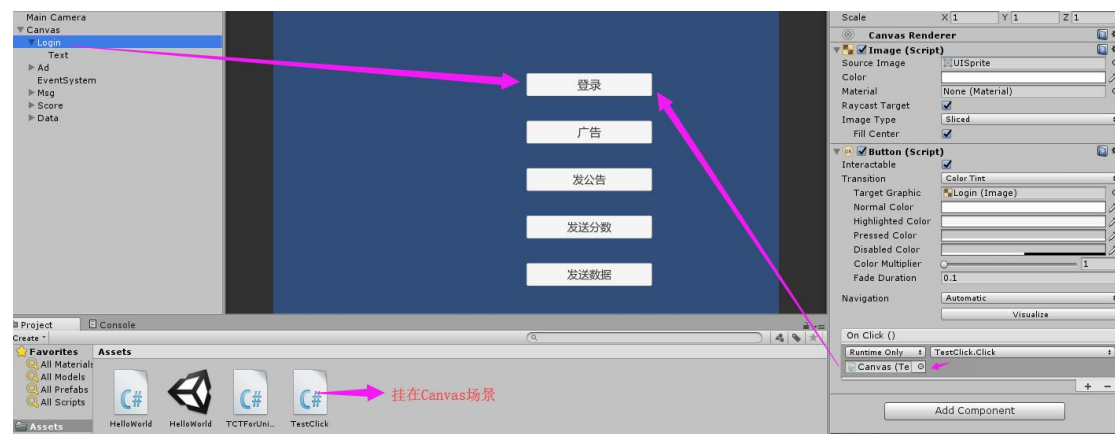
```

public void Click () {
    Debug.Log("点击了登录按钮");
    int status = TCTForUnity.getStatus();
    str = "status : " + status;
    if (status == TCTForUnity.Status_Initial)
    {
        TCTForUnity.login("Canvas", "BeCallFunc");
    }
    else {
        str = "UserId : " + TCTForUnity.getUserInfo("Id") +
            "\nNickname : " + TCTForUnity.getUserInfo("Nickname") +
            "\nAvatar : " + TCTForUnity.getUserInfo("Avatar");
    }
}

private void BeCallFunc(string content)
{
    if (content.Equals("success"))
    {
        str = "UserId : " + TCTForUnity.getUserInfo("Id") +
            "\nNickname : " + TCTForUnity.getUserInfo("Nickname") +
            "\nAvatar : " + TCTForUnity.getUserInfo("Avatar");
    }
    else
    {
        str = content;
    }
}

```

项目目录如下：



第一个参数 Canvas 代表的是脚本所挂载的场景名称，第二个参数代表的是回调方法名称。

8. 封装方法简介

登录方法(login)

调用即可登录，可以添加回调方法，回调方法的参数是一个字符串，如果登录成功则返回 `success`，然后可以获得用户信息，如果登录失败，则返回失败的原因。

上线方法(online)

如果之前登录过游戏，再次打开游戏会自动上线，此方法仅用于传递回调方法。

获得 SDK 状态(getStatus)

登录状态有以下值：

<code>TCTForUnity.Status_Unavailable</code>	0	
<code>TCTForUnity.Status_Initial</code>	1	
<code>TCTForUnity.Status_Login</code>	201	已经登录
<code>TCTForUnity.Status_Online</code>	200	已经上线
<code>TCTForUnity.Status_Offline</code>	500	已经下线

获得用户信息(getUserInfo)

可以获得字符串类型的用户信息，暂时仅支持以下参数：`Id`、`Nickname`、`Avatar`。

发送公告(showMsg)

调用此方法，传入一个字符串，则可以显示在上面的跑马灯公告栏。

发送得分(sendScore)

调用此方法，可以向服务器发送一个 float 型的分数，用户排行榜排序。此方法可以带回调。

发送其他数据(sendData)

调用此方法，可以向服务器发送一个键值对。其中 key 是一个小于 100 字节的字符串，代表了数据种类，score 是一个 float 型的分数，可以用于数据统计和分析。此方法可以带回调。

支付(pay)

调用实现支付功能，成功后可以获得 PayType 数据如下：

```
{
  "id":"79f3662c-a5f4-11e8-9a64-00163e006954", // 订单号
  "profit":0.857143, // 支付的金额，type 为 0 时，代表 TCT 金额
  "amount":1.8, // 支付的人民币金额
  "type":0 // 支付方式，0 代表 TCT 支付
}
```