SuperG SDK for Android

Revision history

Version	Date	Changes
3.0.2	2014-03-27	-Removed ad serving: Just add your own ad serving SDK. Cliq Digital works with MADS. You can download the MADS SDK here: http://developer.madsone.com/Android_SDK -Removed the Thumbr subscription flow -Removed all assets and compiled SDK to .jar -Added Generic Event -Removed mandatory settings. All is automatically set within the SDK now. -Removed Unity Plugin for the time being.
3.0.1	2013-11-12	-Bugfix hardware acceleration (API 11) -Bugfix non-standard timezone strings.
3.0.0	2013-11-06	-Rebranded to SuperG -Event logging -Push Notifications Support -Separate Thumbr T animation Framework -AppsFlyer made optional -Performance optimizations
2.0.32	2013-07-25	-Possibility to hide the Thumbr close button (remote configuration) -Improved interstitial ads -Added demo and manual for Ad-only integration
2.0.31	2013-05-16	-Personalized Ad serving support -Technical updates
2.0.3	2013-04-29	-Ad serving support -Added SSL support -Some small bug fixes
2.0.22	2013-03-22	-Added support for alternative layout -Catch url's that contain 'openinbrowser' and open them in browser -Send along SDK version number to Thumbr server -Improved orientation behavior
2.0.21	2013-01-24	-Better Thumbr T-button resizing -Added visual SDK version to Thumbr window -Added Customizable Orientation -Added extra parameters to registration flow: default,registration,optional_registration -Added a counter to the 'opens' of the Thumbr SDK window -Let external URL's (within the SDK window) be opened in the default browser -Bug fix: better orientation handling in general -Let Thumbr server know that back end is loaded from within the SDK (via GET param (&sdk=1) + via HEADER (x-thumbr-method)) -Added version header (X-Thumbr-Version)
2.0.2		-Updated version number to match iOS version (for better release planning) -Added Animated Thumbr T-button support -Added SDK version number to Thumbr screen -Bug fixes
1.1		- Added Appsflyer support
1.0.1		- Bug fixes
1.0		- Initital version

SuperG SDK

The SuperG SDK is a lightweight set of classes that handles Event logging, AppsFlyer tracking and Push Messaging.

SuperG SDK installation

Prerequisites

- Push messaging is enabled by default
- Event logging is enabled by default. You can set custom events by following the guidelines in the part 'Optional 3'

Step 1:

- Unzip the SDK package.
- Add the TestApp project to Eclipse to understand the methods better.

Step 2:

Add the following jar files to your build path:

```
SuperG_SDK/libs/SuperG-SDK-3.0.2.jar

SuperG_SDK/libs/AF-Android-SDK-v2.3.1.1.jar

SuperG_SDK/libs/android-support-v13.jar

SuperG_SDK/libs/gcm.jar
```

Step 3:

Copy the contents of SuperG_SDK/res/raw to your own res/raw

If the raw directory does not exist, simply drag the raw folder into your res folder.

```
The raw directory contains the 6 default push message alert sounds.
```

Step 4:

Update your AndroidManifest.xml with at least the settings below.

A lot of permissions are asked, this is because (external) advertising functionality requires most of them!!!

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

    <uses-sdk
        android:minSdkVersion="8"
```

```
    android:targetSdkVersion="19" />

<permission
    android:name="com.yourcompany.yourapp.permission.C2D_MESSAGE"
    android:protectionLevel="signature" />

<uses-permission android:name="com.gkxim.tqhung.thumbr.dev_demo.permission.C2D_MESSAGE" /
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.READ_CALENDAR" />
<uses-permission android:name="android.permission.WRITE_CALENDAR" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.DISABLE_KEYGUARD" />

<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
        android:name="com.yourcompany.yourapp.MainActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

    <!-- You need the Google Play SDK library added for this meta data line.-->
    <meta-data android:name="com.google.android.gms.version" android:value="@integer/goog

    <receiver android:name="com.appsflyer.MultipleInstallBroadcastReceiver" android:expor
         <intent-filter>
             <action android:name="com.android.vending.INSTALL_REFERRER" />
         </intent-filter>
    </receiver>
    <receiver
        android:name="com.cliqdigital.supergsdk.utils.GCMReceiver"
        android:permission="com.google.android.c2dm.permission.SEND" >
        <intent-filter>
            <action android:name="com.google.android.c2dm.intent.RECEIVE" />
            <action android:name="com.google.android.c2dm.intent.REGISTRATION" />
            <category android:name="com.yourcompany.yourapp" />
        </intent-filter>
    </receiver>
    <service android:name="com.cliqdigital.supergsdk.utils.GCMIntentService" />

</application>
```

```
</manifest>
```

Step 5:

Update the *activity file*, where SuperG will be called (usually your MainActivity)

Import at least these libraries

```
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;

import com.cliqdigital.supergsdk.SuperG;
import com.cliqdigital.supergsdk.SuperG.OnPushMessageListener;
import com.cliqdigital.supergsdk.utils.EVA;
```

Define the 'superG' object and add or edit onPause and onResume methods

```
SuperG superG;

@Override
protected void onPause() {
    superG.pause(this);
    super.onPause();
}
@Override
protected void onResume(){
    superG.resume();
    super.onResume();
}
```

Modify your onCreate function

```
@Override
protected void onCreate(Bundle savedInstanceState) {

    //CALL THE SUPERG LIBRARY AND REGISTER SETTINGS
    superG=new SuperG(this);

    superG.setPushMessageListener(new OnPushMessageListener(){
        @Override
        public void onPushEvent(Intent intent) {
            try {
                String action = intent.getExtras().getString("action");
                String message = intent.getExtras().getString("message");

                Log.i("SuperG","Push message was received. Message is:"+message);

                if(action.equals("customEvent")){
                    Log.i("SuperG","Action '"+action+"' was received. Message is:"+message);
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

    });
```

```
    // DEFAULT STUFF
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

Optional: Extra events logging

Basic events are logged and stored by default. You can use the following extra methods to store other events, anywhere in your application:

First get an instance of the EVA class:

```
EVA eva = new EVA();
```

Choose any of these events

Generic event

```
//usage: eva.genericEvent(context, generic_key, generic_value)
eva.achievementEarned(this, "MyEvent","It Fired!");
```

Achievement event

```
//usage: eva.achievementEarned(context, achievement_name)
//example:
eva.achievementEarned(this, "MyAchievement");
```

Click event

```
//usage: eva.click(context, clicked_item)
//example:
eva.click(this, "SomeButton");
```

Purchase event

```
//usage: eva.purchase(context, currency, payment_method, price, purchased_item)
//example:
eva.purchase(this, "EUR", "in-app-purchase", "0.99", "ProStatus");
```

Start Level event

```
//usage: eva.startLevel(context, app_mode, level, score_type, score_value)
//example:
eva.startLevel(this, "basic", "1", "points", "0");
```

Finish Level event

```
//usage: eva.finishLevel(context, app_mode, level, score_type, score_value)
//example:
eva.finishLevel(this, "basic", "1", "points", "120");
```

Upsell event

```
//usage: eva.upSell(context, currency, payment_method)
//example:
eva.upSell(this, "EUR", "in-app-purchase");
```

Optional 2: installation_id => server

If you are using a back end that triggers events that are used by SuperG Push messaging, you need to pass the installation*id. You can obtain the installation*id like this:

```
EVA eva = new EVA();
eva.getInstallationId(context);
```

Proguard

Add these lines to your Proguard configuration:

```
-dontwarn com.unity3d.**
-dontwarn com.appsflyer.**
```

If Proguard still gives you errors, please look at the proguard.cfg in the Test App.