Tech Note

Cardboard VR Projects for Android, Using Google VR SDK

Jonathan Linowes, June 2016

This article provides technical notes for the book *Cardboard VR Projects for Android*, for readers interested in using the updated Google VR Cardboard Java SDK 0.8, released in May, 2016. For more information on the book, and to purchase it, use the following links:

- `https://www.packtpub.com/application-development/cardboard-vr-projects-android`
- `https://www.amazon.com/Cardboard-Projects-Android-Jonathan-Linowes/dp/1785887874`

In May 2016, Google introduced **Google VR** (`https://developers.google.com/vr/`), including two mobile **virtual reality** (**VR**) platforms consisting of an update on Cardboard VR and introducing the new higher quality Daydream platform. According to Google, Cardboard is intended for *bite-size VR experiences and works with almost any smartphone on Android or iOS*. Daydream, coming in Fall 2016, ups the Android game in mobile VR providing *rich, responsive, and immersive experiences with hardware and software built for VR*.

That same month, Matt Schoen and I released the new book, *Cardboard VR Projects for Android*, a 360-page how-to guide for building Cardboard apps with the Java SDK and Android Studio.

The book was in the works for 6 months. One of our technical reviewers, Scott Dolim, a software engineer at Google, is a member of the Google VR development team. As industry secrecy often dictates, Scott couldn't share the details about the new Daydream platform directly with us, except for saying that "expect things to change but you should be OK". Indeed, things have changed, and yes, we're OK.

Google VR SDK 0.8 versus Cardboard SDK 0.7

The projects in our book use Cardboard SDK version 0.7 (released in March, 2016). We protected our readers from future SDK releases by including all the libraries and source code on our GitHub repository (`https://github.com/cardbookvr`) and also as file downloads from the Packt Publishing website. We also offered the readers the specific Git commit within the Google SDK repository, accessible via the following commands:

```
git clone https://github.com/googlevr/gvr-android-sdk.git
```

```
git checkout 67051a25dcabbd7661422a59224ce6c414affdbc -b sdk07
```

That being said, if you would like to use the current GVR SDK, follow the instructions in this document.

The current version 0.8 SDK (May, 2016) has been rebranded from "Cardboard" to "Google VR", or GVR, and replaces the Cardboard SDK. The top-level package is now `com.google.vr.sdk`. The `CardboardView` and `CardboardActivity` classes are now called `GvrView` and `GvrActivity`, respectively. The SDK library has been renamed from `core.aar` to `base.aar`. Other changes can be found in the release notes (`https://developers.google.com/vr/android/release-notes#google_vr_sdk_for_android_v080`).

This document does not entertain the differences between Cardboard and Daydream VR platforms, although it is very promising. Daydream appears to be a superset of the Cardboard viewer SDK, in particular, adding support for a handheld input controller. Any apps built with Google VR SDK, including the projects presented in our book, should run on both platforms.

Clone the GVR Android SDK sample app

To begin, obtain a copy of the GVR Android SDK by cloning the sample app from the `gvr-android-sdk` GitHub repository (`https://github.com/googlevr/gvr-android-sdk`). You can use a Git GUI such as GitHub Desktop (`https://desktop.github.com/`) or SourceTree (`https://www.sourcetreeapp.com/`). If you're comfortable with command line, run the following from the directory where you want to clone the code:

```
    git clone https://github.com/googlevr/gvr-android-sdk.git
```

Then make a note of the location of the library files that we will need:

- `libraries/base/base.aar`
- `libraries/common/common.aar`

The SDK includes additional libraries that we do not use in the projects in this book, but they could be useful for your other projects, including support for spatialized 3D audio, the new Daydream Bluetooth input controller, as well as widgets to facilitate inserting VR media content directly into regular 2D Android applications.

As described in the Google VR's Getting Started guide (`https://developers.google.com/vr/android/get-started`), you can build and run the sample Treasure Hunt app by opening Android Studio, choosing **Open an Existing Project**, and selecting the `android-sdk` directory. Then, with your phone connected to the PC, select the `samples-treasurehunt` target and click on **Run** to compile and run the application on your phone.

Create a new Google VR project

In the *Cardboard VR Projects for Android* book's *Chapter 2*, *The Skeleton Cardboard Project*, we saw how to build a skeleton Cardboard project, which was useful as a starting point for other projects in the book. The following instructions show how to adapt it to work with SDK 0.8.

Open Android Studio and start a new project. As explained in the book, make sure to specify a **Minimum SDK** no less than API 19 (Android 4.4, KitKat). Choose **Empty Activity** and name it `MainActivity`. Click on **Finish**.

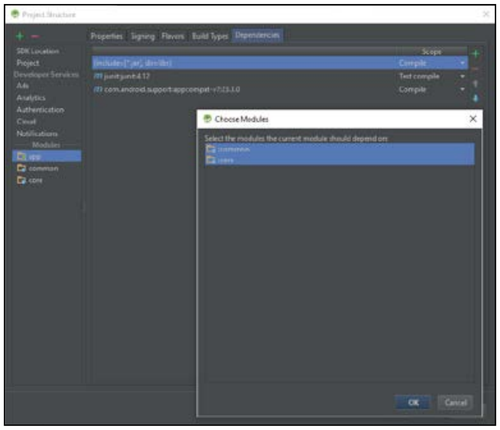
Add the SDK libraries

Add the `.aar` GVR SDK library files to your project. For each, the `base.aar` and `common.aar` libraries perform the following steps:

1. Navigate to **File** | **New** | **New Module...**.
2. Select **Import .JAR/.AAR Package**, click on **Next**.
3. Use the file browser (**...** or press the *Shift* and *Enter* keys) to find and select the `base.aar` file. Then click on **OK**.
4. Click on **Finish**.
5. Repeat these steps for the `common.arr` module.

Next, we'll add the new modules as dependencies to our main app:

1. Navigate to **File** | **Projects Structure** | **Modules** (on the left-hand side) | **app** (your app name) | **Dependencies** (tab) | **+** (on the right-hand side) | **Module Dependency**.
2. Choose both modules from the list, click on **OK**. Then click on **OK** again in the **Project Structure** box:



The AndroidManifest.xml file

There are no changes required to the `AndroidManifest.xml` file, apart from the one specified in *Chapter 2*, *The Skeleton Cardboard Project* of the book. A copy of the file is shown in the following. Make sure to replace the `package=` string with the one for your specific app:

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

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.NFC" />
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission
    android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <!-- Make accelerometer and gyroscope hard requirements for
    good head tracking. -->
    <uses-feature
```

```
            android:name="android.hardware.sensor.accelerometer"
            android:required="true" />
        <uses-feature
            android:name="android.hardware.sensor.gyroscope"
            android:required="true" />

        <uses-sdk
            android:minSdkVersion="19"
            android:targetSdkVersion="22" />
        <uses-feature
            android:glEsVersion="0x00020000"
            android:required="true" />

        <application
            android:allowBackup="true"
            android:label="@string/app_name"
            android:supportsRtl="true">
            <activity
                android:name=".MainActivity"
                android:configChanges=
                "orientation|keyboardHidden|screenSize"
                android:label="@string/app_name"
                android:screenOrientation="landscape">
                <intent-filter>
                    <action android:name="android.intent.action.MAIN"
                    />

                    <category android:name=
                    "android.intent.category.LAUNCHER" />
                    <category android:name=
                    "com.google.intent.category.CARDBOARD" />
                </intent-filter>
            </activity>
        </application>

    </manifest>
```

We would recommend you to still specify `targetSdkVersion="22"` to avoid the Android's new permissions system (which would complicate the example projects in our book). Note that Google's VR sample apps also target version 22 (at the time of writing), presumably, for similar reasons. (At Google I/O 2016, Google announced that an upcoming version Android Studio will add a tool to help developers migrate code to the Android M Permission model, allowing you to inject boilerplate code to handle the requests and different outcomes).

In addition to the `AndroidMainfest.xml` file, you also need to modify the `build.gradle` file for the app module to specify `targetSdkVersion 22`, as follows:

```
...
defaultConfig {
    minSdkVersion 19
    targetSdkVersion 22
    ...
}
...
```

You may be asked to resync the project as the Gradle file has changed. Go ahead and select **Sync Now**.

Refer to the pages 43 and 44 of the book for further discussion on this issue.

The activity_main.xml file

The `activity_main.xml` file is very similar to the one specified in *Chapter 2*, *The Skeleton Cardboard Project* of the book, only it now needs to reference `com.google.vr.sdk.base.GvrView` (instead of `com.google.vrtoolkit.cardboard.CardboardView`) and set the ID to match our new naming conventions, `android:id="@+id/gvr_view"`. A copy of the file is shown in the following:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    tools:context=".MainActivity">

    <com.google.vr.sdk.base.GvrView
        android:id="@+id/gvr_view"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true" />
</RelativeLayout>
```

The MainActivity class

We will define `MainActivity` to extend `GvrActivity` and implement `GvrView.StereoRender`, instead of `CardboardActivity` and `CardboardView.StereoRenderer`, respectively:

```
MainActivity extends GvrActivity implements
GvrView.StereoRenderer {
```

Then add the *implements methods* stub using the intellisense context menu (the light bulb icon), or navigate to **Code** | **Implements Methods...** on the main menu.

Next, edit `onCreate()` by adding the `GvrView` instance, as follows:

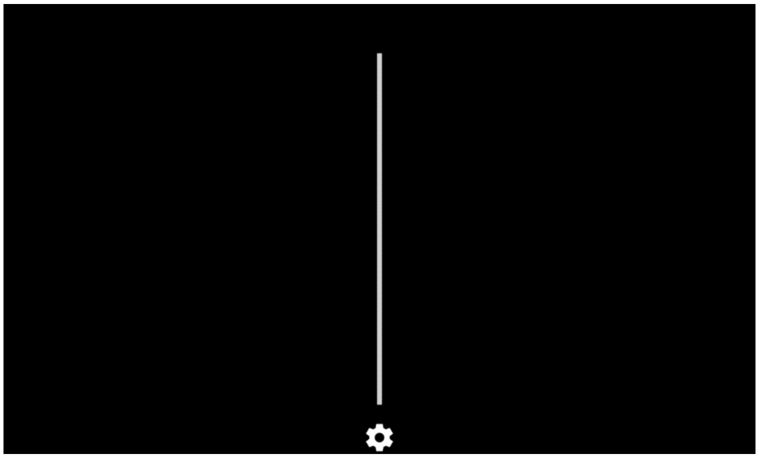
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    GvrView gvrView = (GvrView) findViewById(R.id.gvr_view);
    gvrView.setRenderer(this);
    setGvrView(gvrView);
}
```

The is very similar to the instructions given in the book, only `CardboardView` is replaced by `GvrView`.

Build and run

At this point, you should be able to build and run the skeleton app with no build errors. The app will only draw a blank Cardboard scene, including the signature vertical line, separating the stereoscopic views and a gear icon, as shown in the following image:



Convert existing Cardboard projects

To convert an existing Cardboard Java app from 0.7 SDK to 0.8, open it in Android Studio and perform the following steps.

This also applies to the `RenderBox` library built in *Chapter 5*, *RenderBox Engine*. At the end of the chapter, we made a new project, `RenderBoxLib`, and built the `Renderbox` module library, resulting in `renderbox/build/outputs/aar/renderbox-debug.aar` (and `renderbox-release.aar`):

1. To upgrade the `common.aar` library module, simply copy the new `common.aar` file into the project's existing `common/` module directory.
2. To replace the `core` module with base, it's easier to delete the old one. Delete the entire `core/` directory from your project (for example, using Windows Explorer).
3. Then **navigate to File** | **New Module**, select **Import .JAR/.AAR Package**, and then locate the `base.aar` file.
4. From the menu bar, **navigate to File** | **Projects Structure** | **Modules** (on the left-hand side) | **app** (your app name) | **Dependencies** (tab). Remove ":`core`" by selecting it and pressing the minus (`"-"`) sign on the right-hand side. Then press the plus (**"+"**) sign on the right-hand side, choose **Module Dependency**, and select "**:base**" to add it.
5. As mentioned earlier, we recommend you still specify the target SDK Version **22** to avoid the Android's new permissions system (which would complicate the example projects in our book). Check the `build.gradle` file for the app module and make sure that it specifies `targetSdkVersion 22`.
6. You may be asked to resync the project as the Gradle file has changed. Go ahead and select **Sync Now**.
7. If the project is using our `RenderBox` library (Chapters 6 to 9), replace the old `renderbox-debug.aar` file with the updated one for SDK 0.8.
8. Modify the `activity_main.xml` file, replacing the reference with `com.google.vrtookit.cardboard.CardboardView` with `com.google.vr.sdk.base.GvrView`, and set the ID, `android:id="@+id/gvr_view`.
9. In `MainActivity.java`, set the class to `extends GvrActivity implements GvrView.StereoRenderer`. In case the projects use the `RenderBox` lib, it `extends GvrActivity implements IRenderBox`.
10. Delete the now-invalid import lines at the top of the file, and let Android Studio repopulate these for the updated SDK.

11. Lastly, replace the references of the old SDK with the new names, for example:

Old SDK	**GVR SDK 0.8**
`CardboardView`	`GvrView`
`setCardboardView()`	`setGvrView()`
`R.id.cardboard_view`	`R.id.gvr_view`
`Cardboard#getCardboardDeviceParams()`	`Gvr#getGvrViewerParams()`

CardbookVR GitHub projects

The GitHub repository for the *Cardboard VR Projects for Android* book is located at `https://github.com/cardbookvr` and includes the following repositories. Each repository will now have new branches with completed projects compatible with GVR SDK 0.8.0 named `sdk080`:

- `skeleton` (*Chapter 2*, *The Skeleton Cardboard Project*)
- `cardboardbox` (*Chapter 3*, *Cardboard Box*)
- `launcherlobby` (*Chapter 4*, *Launcher Lobby*)
- `renderboxdemo` (*Chapter 5*, *RenderBox Engine*)
- `solarsystem` (*Chapter 6*, *Solar System*)
- `gallery360` (*Chapter 7*, *360-Degree Gallery*)
- `modelviewer` (*Chapter 8*, *3D Model Viewer*)
- `visualizevr` (*Chapter 9*, *Music Visualizer*)

Additionally, the `RenderBox` library repository, `renderboxlib`, has separate branches for the incremental states at the end of each chapter, which is, for example, presently tagged, `after-ch5`, updated for SDK 0.8.0, and named `after-ch5-sdk080`.

Summary

Google can be expected to continually improve and update Java SDK for Google Cardboard, which was recently rebranded as Google VR and includes the higher fidelity Daydream VR platform. Newer versions of Android OS and Daydream devices will run older Cardboard apps. However, if you download the latest SDK, you may need to update your code to accommodate the recent changes and improvements. Important changes were introduced in the SDK version 0.8, released in May 2016. The top-level package is now `com.google.vr.sdk`. The `CardboardView` and `CardboardActivity` classes are now called `GvrView` and `GvrActivity`, respectively. The SDK library has been renamed from `core.aar` to `base.aar`. Upgrading existing projects is straightforward. Adapting existing tutorials and books, like the ones in the *Cardboard VR Projects for Android* book, is also straightforward, but necessary if you use current version of the SDK.