Android Developer Codelab

You can help develop the most widely installed operating system in the history of Earth. Yes, you're here to embark upon the journey of becoming an Android platform engineer.

Although the path is challenging, the Android team strives to simplify your journey, every release. And the team makes improvements every day through direct work in the Android Open Source Project (AOSP).

So sit back, fire up a terminal, and let's make history.

[Goals]

The mission of this codelab is:

1. To give you a idea of what the developer workflow is like for Android engineers working on the platform (the operating system).

[Environment]

Typically, users build and develop on the workstation directly. Because you may be working in various terminals, and many of the commands used are terminal-specific, you will need to rerun them in each terminal session. Specifically, these include the `source build/envsetup.sh` and `lunch` commands.

[Build the code]

To build Android, you must select a target device type to build with the `lunch` command. A target is a device permutation, such as a specific model or form factor.

The device target included below, `aosp_cf_x86_64_phone-userdebug`, enables you to build the Cuttlefish virtual Android device for testing without a physical device.

1. Set up your environment for building Android devices by running the following command from the root of your source code checkout:

```
cd ~/aosp

source build/envsetup.sh
```

2. Pass the build target to the lunch command, like this:

```
lunch aosp_cf_x86_64_phone-userdebug
```

3. Build the code from anywhere in your checkout with:

```
m
```

Expect the first build to take hours. Subsequent builds take significantly less time.

Make a change

Update the source code following this example changelist.

1. From the root of your checkout (`aosp/` directory), navigate to the `frameworks/native` Git project:

```
cd frameworks/native
```

2. Start a temporary project with this command:

```
repo start codelab .
```

3. Edit `SurfaceFlinger.cpp` to include the updates from the changelist at the following location:

```
aosp/frameworks/native/services/surfaceflinger/SurfaceFlinger.cpp
```

4. Find these two lines:

```
postFrame();
postComposition();
```

5. Replace those two lines with the following:

```
postFrame();
postComposition();
mClientColorMatrix = mat4(vec4{1.0f, 0.0f, 0.0f, 0.0f}, vec4{0.0f, -1.0f, 0.0f,
0.0f},
                          vec4{0.0f, 0.0f, -1.0f, 0.0f}, vec4{0.0f, 1.0f, 1.0f,
1.0f});
updateColorMatrixLocked();
```

6. Build the code:

```
m
```

[Revert your change]

Normally, post-testing and upon review and approval, you submit your change in Gerrit and merge it into the repository.

Instead, for the purposes of this codelab, revert your changelist by clicking **Abandon** in Gerrit.

Then abandon the associated temporary branch in the `frameworks/native` project directory (or its subdirectories):

```
repo abandon codelab .
```

At this point, you're done! Nice work!