Laboratory – User Interfaces

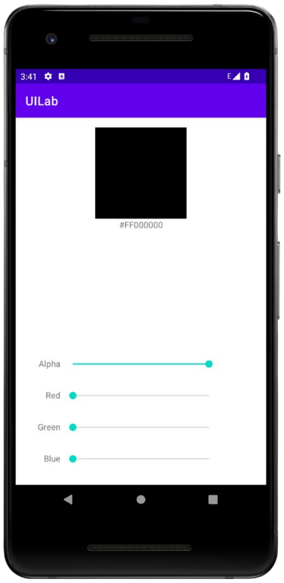
Learn how to define a user interface for your app

Objectives

Familiarize yourself with the UI classes, such as Views, ViewGroups, and Layouts. Create a simple application that presents one view with a background color equal to the current value of an ARGB color specification (Alpha, Red, Green, and Blue). The app will also present four controls to the user, one for each component of the ARGB color specification. As the user manipulates these controls the background of the colored view should update in real time. Once you've completed this Lab you should understand how to define a UI with multiple UI elements, and how to link manipulation of the controls with other actions visible on the UI.

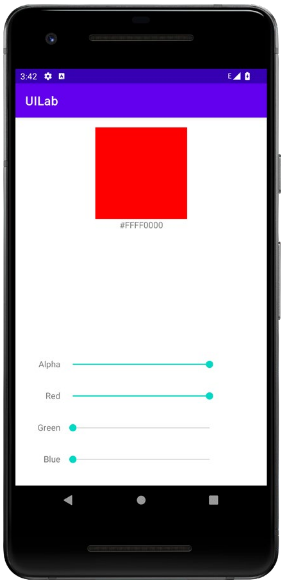
The UI Lab

This lab involves an app called UILab. When it runs, the app displays a user interface like that shown below.



This application comprises a single Activity called "MainActivity." From top to bottom, MainActivity displays a colored View, a TextView displaying the current color specification. Underneath that it displays four SeekBars. Each SeekBar has an associated TextView, displaying which color specification component that SeekBar controls.

When the manipulates a control the colored view changes to track the updated color specification. For example, if the user moves the Red SeekBar all the way to the right, the colored View updates as shown below.



See the screencast, LabUI.mp4, that's included in the Lab directory.

Testing

There is one test case with several evaluation points. Each evaluation occurs at a step labelled “evaluation point”

This test case operates as follows:

1. Start the UI app in portrait mode
2. Check the current color is #FF000000 and that the colored view is black (evaluation point 1)
3. Move the Red SeekBar all the way to the right
4. Check the current color is #FFFF0000 and that the colored view is red (evaluation point 2)
5. Move the Green SeekBar all the way to the right
6. Check the current color is #FFFFFF00 and that the colored view is yellow (evaluation point 3)
7. Move the Blue SeekBar all the way to the right
8. Check the current color is # FFFFFFFF and that the colored view is white (evaluation point 4)
9. Move all SeekBars all the way to the left
10. Check the current color is #00FFFFFF and that the colored view is white (evaluation point 5)

After completing your solution, you will record a screencast while performing a manual test. Afterward, you will submit your code and the screencast via git. You can record a screencast using services available in the Logcat console. See: https://developer.android.com/studio/debug/am-video

Submission

When you are ready just commit your solution to your repo on GitLab by running the following commands:

```
% git add path/to/changed/files
% git commit -m “completed Lab6_UI”
% git push origin main
```

Note: if you have not already pushed this branch to your repo on GitLab you will need to make a slight modification for this first time and run this instead:

```
git push –u origin main
```

This sets up tracking between your local branch and a branch with the same name on your repo in GitLab.

Some Implementation Notes

We are providing template code and layout resources for this application. Only modify the areas marked with the word TODO.

We have done our testing on an emulator using a Pixel 5 AVD with API level 31. To limit configuration problems, you should test your app against a similar AVD.

Your SeekBars’ values should range from 0 to 255.