

Exercises:

Buttons and Similar Clickable Widgets

Presumably you've tried out Buttons in previous exercises, so I will let you try some of the other Widgets here. There are probably more exercises here than you will have time to do, so order the problems based upon which types of widgets you want to practice with.

- 1.** Make a row containing two ImageButtons. Put a blank TextView at the bottom. Have the first image be red, and have it turn the TextView red when you click it. Have the second image be blue, and have it turn the TextView blue when you click it. Use a single image file for each ImageButton. For some red and blue images, try <http://findicons.com/search/red> and <http://findicons.com/search/blue>. Or, steal the icons out of my exercise solutions folder (ButtonsExercises). To change the color, use the TextView's setBackgroundColor method with arguments of Color.RED and Color.BLUE.
- 2.** Add a row with three RadioButtons with labels Red, Yellow, and Blue. Have them turn the TextView at the bottom the appropriate color.
- 3.** Add a row with three ToggleButtons. Have the labels be Red/Not Red, Yellow/Not Yellow, and Blue/Not Blue. Turn the TextView the appropriate color when the ToggleButton is clicked. For "Not *Whatever*", use black.
- 4.** Add another row of RadioButtons with the same labels as in problem 2. But this time, take no action when the RadioButtons are clicked. Instead, put a Button underneath that says "Set Color Chosen Above". Change the color when that Button is clicked. Hint: RadioGroup has a getCheckedRadioButtonId method. Note that this is not the *index* (0 for the first radio button, 1 for the second, etc.). Instead it is an int representing the *id* (i.e., the id as defined via android:id in the XML). So, you have to do a "switch" or "if" comparing the ids (R.id.red_radio_button, R.id.blue_radio_button, etc.).

In principle, you could use indexOfChild to find the index instead of the id, but doing so is a bit complicated. In some ways, this a deficiency in the design of RadioGroup, but people also argue that you should be able to rearrange the order in which radio buttons are visually laid out without changing the Java code, so they would say that you shouldn't be trying to use the index in the first place. Spinners (combo boxes), covered in the next lecture, have a very simple and straightforward way of finding the index of the selection.