CMSC 436 Lab 7

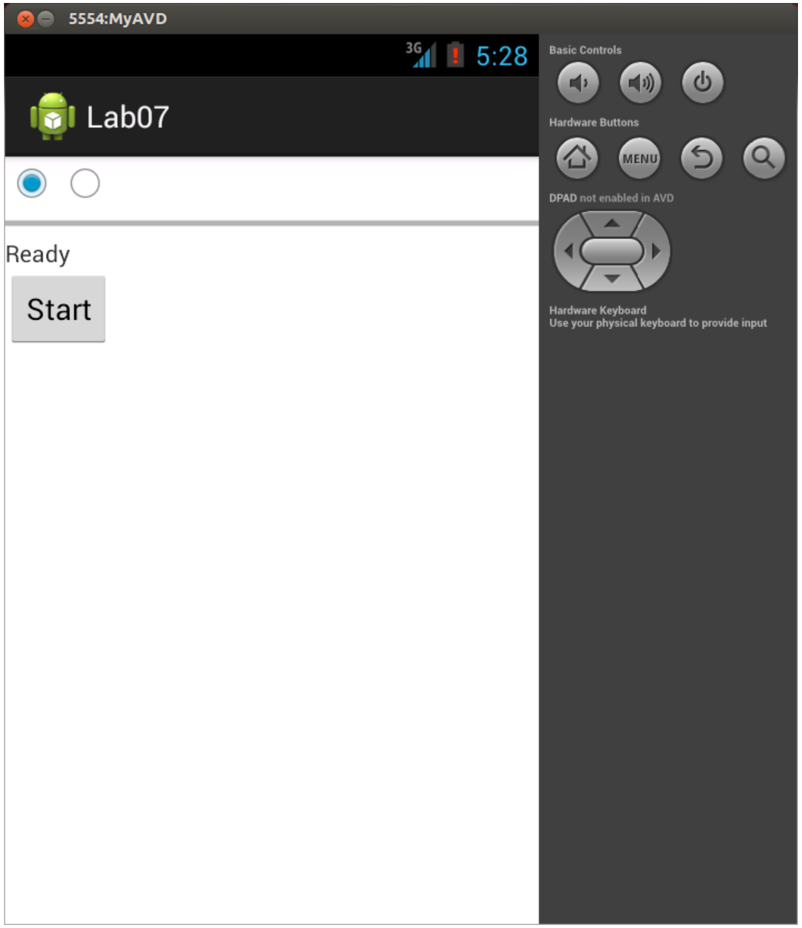
Process and Threads

Overview

- For this lab you will practice threading in Android by directly creating Threads and by using AsyncTask
- The information needed to do this lab can be found on the Android developer site at
  http://developer.android.com/guide/components/processes-and-threads.html
  http://developer.android.com/reference/android/os/AsyncTask.html

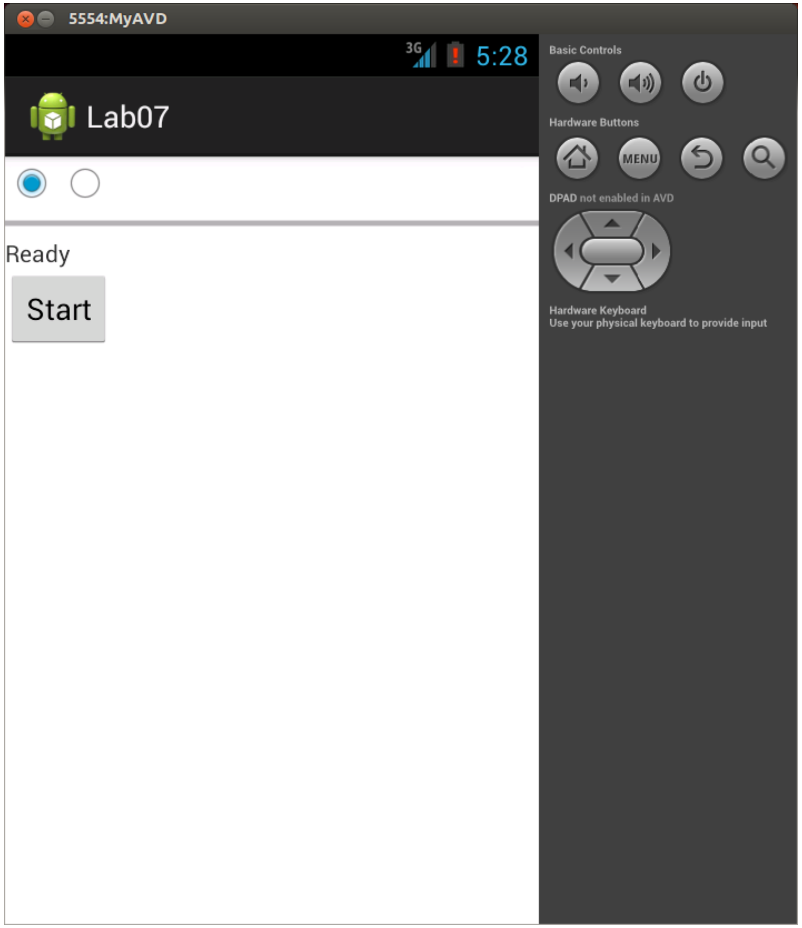
Overview

- Your activity should contain two RadioButtons (oriented horizontally), a ProgressBar, a TextView, and a start Button



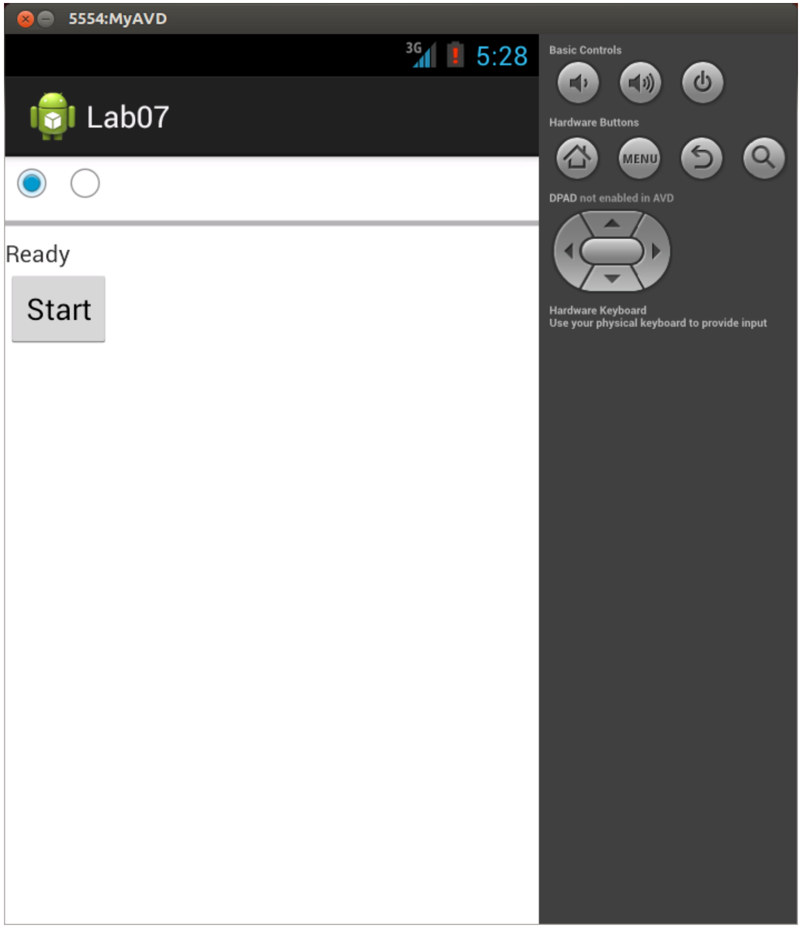
Overview

- The checked mark should continually alternate back and forth between the RadioButtons, changing once each second



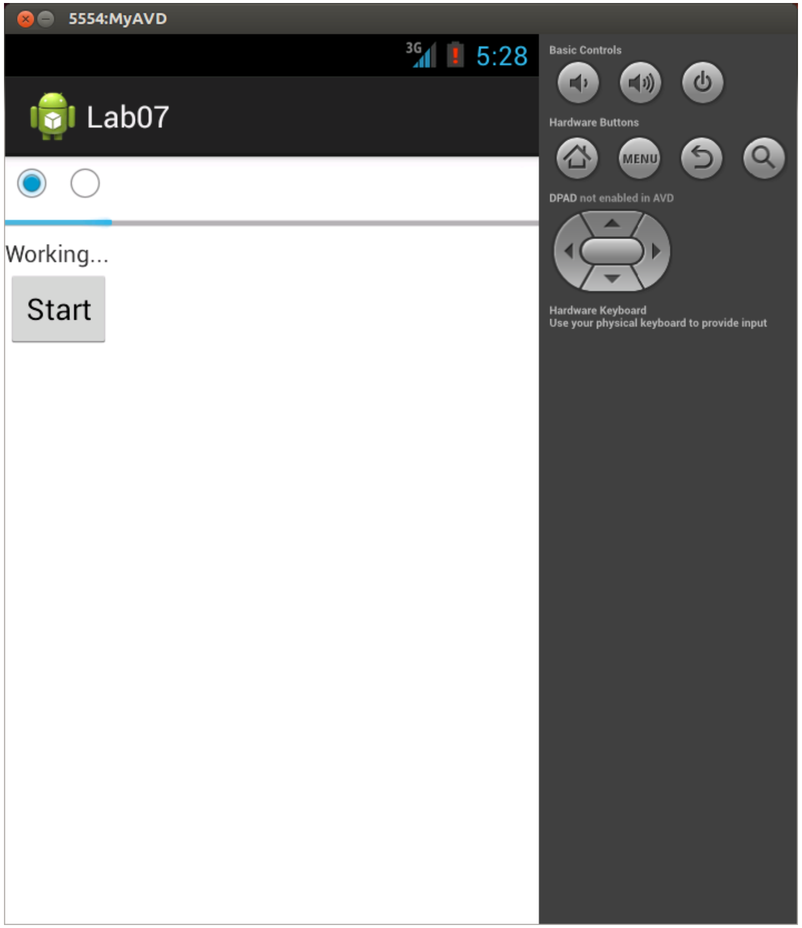
Overview

- The ProgressBar should start at 0% and the TextView should initially say “Ready”



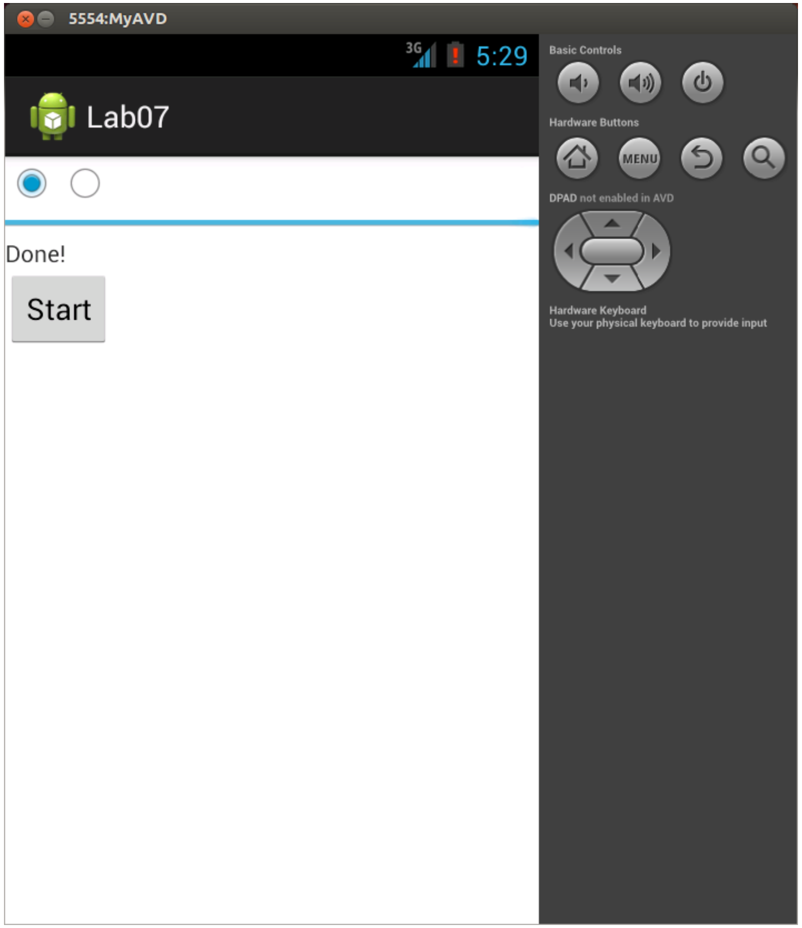
Overview

- Clicking the start Button should change the text to “Working...” and cause the ProgressBar to start incrementing by 10% each second



Overview

- When the ProgressBar reaches 100% the text should change to “Done!”



Done!

Overview

- You must implement the RadioButton blinking by directly instantiating a Thread, and implement the ProgressBar incrementing by using an AsyncTask
- Keep in mind the two rules of Android threading
  - Do not block the UI thread
  - Do not access the Android UI toolkit from outside the UI thread

Blink Thread

- An example of how NOT to do threading is the following

```
public void onClick(View v) {
    new Thread(new Runnable() {
        public void run() {
            Bitmap b = loadImageFromNetwork("http://example.com/image.png");
            mImageView.setImageBitmap(b);
        }
    }).start();
}
```

- This example accesses mImageView from outside the UI thread, which is forbidden
- See the Android developer site for how to tell UI elements to update themselves on the UI thread

Blink Thread

- You can start the blink Thread running when the Activity is Resumed
- You don't need to worry about stopping the Thread when the Activity is Paused

Progress AsyncTask

- To ease the implementation of worker threads, Android provides the AsyncTask class
- When the start button is pressed, your subclass of AsyncTask should perform the following actions
  - Set the TextView to “Working...”
  - Increment the ProgressBar by 10% every second until it reaches 100%
  - Set the TextView to “Done!”

Progress AsyncTask

- Pressing the start button again after the task is completed should reset the progress bar and perform the incrementing again, setting the text appropriately
- It doesn't matter what happens if the start button is pressed again while the incrementing is in progress