

Lab 7

Lab 7a

Write an app that includes a pair of TextViews and a Button. When the user taps the button, the background color of the TextViews will change for a few seconds.

Use the existing class Color. A color is represented by an int, where each byte represents the alpha, red, green, and blue channels.

A single class will extend AsyncTask. When the user taps the button, create an instance of the inner class and start it. In doInBackground change the color value, which is passed to doProgressUpdate periodically. In doProgressUpdate change the background color of the Textview.

Lab 7b

Code the same app as in 7a but use an instance of Thread to change the TextView. Call activity.runOnUiThread (runnable) from run().

Example of a class extending *Thread* and calling *runOnUiThread*

```
private class Colorize extends Thread {
    :
    public void run() {
        // Create a suitable color - color
        activity.runOnUiThread( new SetColor(int color) );
        // Pause thread e.g. 50ms
    }
}

private class SetColor implements Runnable {
    private int color;

    public SetColor(int color) {
        this.color = color;
    }

    public void run() {
        call the method to change the background color in the Text View 1
        call the method to change the background color in the Text View 2
    }
}
```

Lab 7c

Extend TextView and implement AsyncTask as a single class. Implement AsyncTask so that the Textview's background color changes gradually over a short period (about 20 times/second for a few seconds). Color changes can start when calling the method colorize ().

One solution is to have a single class that extends AsyncTask. Change the color value from doInBackground. The color value is passed to doProgressUpdate periodically. In doProgressUpdate change the background color of the Textview.

Lab 7d

Update 7c with an inner class that extends Thread. Each time the color in the TextView is changed, call `invalidate()` from the run method. After calling `invalidate`, the system will call `onDraw()`, so that the background can be drawn again.

The call to `invalidate` must be performed from the UI thread. One way to get the code to be performed by the UI thread from a class that is a View is using the POST method (see below). Below is the “post” call placed within the body of the TextView’s update method. The easiest way is to call the update method from the run method in the class that inherits the Thread.

```
private void update() {
    post(new Runnable() {
        public void run() {
            invalidate();
        }
    });
}
```

TextView class should then override `onDraw`:

```
protected void onDraw(Canvas canvas) {

    // Use canvas + a Paint object with the right color
    // When you color the background

    super.onDraw(canvas); // Called last as text visible
}
```

Lab 7e

Improve 7d with a `java.util.Timer` that calls `invalidate` periodically. Google it to see how the Timer class works. Reuse as much as possible from 7d.