

In class Lab and Project

Overview:

This is purely an exercise to familiarize you with network communications, AsyncTask, preferences, preference activities and the action bar.

Grading:

10% - In Class: Please complete steps 1 -3 below for full credit.

90% - Out of Class: Please complete this assignment and **demonstrate** it to me by Friday March 15th

Requirements:

Create app that has/does the following;

1. On startup it should display a default background image (very important for customizing the look of an app). You can use the default image in drawable, ic_launcher.png if you wish.
2. The app will also have an action bar with a settings icon.
3. When you click this icon a preference activity will pop up.
4. This preference activity allows the user a chance to customize their background using a ListPreference. (see below for links, how tos etc.)
5. If the user does not select a custom background the default image stays
6. If the user does select a custom background the preference activity will save that background in the default preferences.
7. The main activity will have a listener set on the preference key that tracks the background, when the key changes the listener is triggered, it in turn will change the background
8. The background is changed by making a network call and downloading the appropriate image.
9. That image is then set as the background.

Helpful bits

Background URLs

<http://www.pcs.cnu.edu/~kperkins/bug.png>
<http://www.pcs.cnu.edu/~kperkins/sky.png>
<http://www.pcs.cnu.edu/~kperkins/space.png>

Setting an activity background image: Give your layout an id ((so you get at it thru java) and point it to a default drawable like so

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ClassLab2Activity"
    android:id="@+id/RelativeLayout1"
    android:background="@drawable/ic_launcher">
```

Changing the Background of the layout - note the change in BitmapDrawable. It was deprecated so I had to find a replacement.

```
@Override
protected void onPostExecute(Bitmap result) {
//    mImage = new BitmapDrawable(result); //deprecated what to do? Answer: Stackoverflow of course
    mImage = new BitmapDrawable(myContext.getResources(),result);
    if (mImage != null) {
        myContext.rl.setBackgroundDrawable(mImage);
    }
}
```

Setting up the Preferences - For a general preference outline see DemoPreferences example project. For a tutorial specific to a ListPreference see <http://android-er.blogspot.com/2010/08/listpreference.html>) for a rigorous step by step example. The following is a summary of relevant parts;

1. Define 2 arrays in res/values – first should hold user friendly names for the pictures (Bug Space Sky. The second the associated URLs for each.
2. Create a preferences.xml in the folder /res/xml/
3. Create SetPreferences.java (don't forget to add this activity to the manifest)

Setting up action Bar

See last lecture (7-2 More UI) and UI_demos project, start the preference activity whenever this icon is pressed.

Downloading image - You will do this using http in a separate thread. see AsyncDownload project, remember to set the proper permissions in the manifest

Preference listener – Make sure your preference listener is a member of your main activity (See <http://stackoverflow.com/questions/2542938/sharedpreferences->

[onsharedpreferencechangelistener-not-being-called-consistently](#) for why). This means you have something like the following;

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_class_lab2);
    myPreference=PreferenceManager.getDefaultSharedPreferences(this);

    // listen for a change to the listPref key,
    //its the key part of the key value pair that holds the URL to load
    //when this key changes then the URL has changed to so reload it
    //make this listener an instance var so it is not GCed due to it being
    //saved as a weak reference
    listener = new SharedPreferences.OnSharedPreferenceChangeListener() {
        public void onSharedPreferenceChanged(SharedPreferences prefs, String key) {
            if (key.equals("listPref")) {
                loadImage();
            }
        }
    };

    //gotta register the listener
    myPreference.registerOnSharedPreferenceChangeListener(listener);
}
```