CIS 3515 Lab

Worksheet 5

Instructions: We are going to localize our application by adding a Spanish translation and then generating a signed APK with a new version number.
You will update the color selector application you worked on in a previous lab. You will first replace the hardcoded text values (Blue, Red, Greed, etc.) with their resource file equivalent (i.e. using a resource file to define them). You will then provide a translation for a second language.

1. Add string resources to your application to provide the labels necessary to display the word for each color from your original selection list. One approach is to use individual String resource elements, but a better approach is to use a String-array resource: https://developer.android.com/guide/topics/resources/string-resource.html

2. Modify your application code to use the string resources to display the labels in your adapter view. You can retrieve a string-array resource as follows:

```
Resources res = [context.]getResources();
String[] colorLabels = res.getStringArray(R.array.color_labels_array);
```

   - If you were previously using the **parseColor(String)** method to set the color of your adapter view views, and the color of the second activity's layout, you will find that your approach will not work with translated text. As such you will have to change your approach to setting the color. One option you could explore:

     - Maintain a second array in your code (static final String [] actualColors) that will always have the english names (or hex codes) of your colors in the same order as your resource array. Then whenever you need to set a color based on a value in the resource array, you will instead use the relevant position to access your actualColor array.

3. Create a new **res/values/strings.xml** file for your Spanish translations

   1. In Android Studio's **Project** view, right click on the **values** folder under **res** and select ***New → Android resource file***

   2. Enter ***strings*** in the File name field and select **main** *Source set* and enter **values** under Directory name

   3. Select **Locale** under Available qualifiers and click the >> button

   4. Select **es:Spanish** under Language. Leave *Any region* selected on *Specific Region Only*

   5. Once complete click on **OK**

4. Copy your string-array resource (and all other string resources) from your default strings

resource file, into your new Spanish translation file, and replace all the text values with the proper translation.

1. Ways to get translations
    1. Good → Google translate
    2. Better → Ask your Spanish speaking friend or take a Spanish course
    3. Best → Pay a translation service (Google provides one)

5. Test your application by changing the Language of your virtual device from English to Spanish. You will find the Language configuration under Device Settings. Good luck changing it back from Spanish to English.

6. Once your application has been tested, update the version number and generate a new APK
    1. Modify the file **build.gradle (Module:app)** found under **Gradle Scripts** in the project view
        1. Under default config, change **versionCode** to ***2***, and **versionName** to ***1.0.1-spanish***
    2. Click on ***Build*** and select ***Generate Signed APK***
    3. On the presented dialog, click on ***Choose existing*** and locate the certificate you generated during the previous lab. Select the alias you had created and enter the requested passwords, then click on ***Next***.

        Please note: If you are unable to find your previous key store, or are unable to recall one or both passwords, then you can create a new key store and key. Keep in mind however that had this application been previously distributed on Google Play or another distribution platform, you would have lost the ability to update the application currently deployed and would instead have to change the package name (your apps unique identifier), lose your current user base and start all over.

    4. Select the destination folder of the APK and set Build Type to ***release***
    5. Click on ***Finish***
    6. Locate and rename your generated APK to **colorpicker-release-1.0.1.apk**
    7. **Upload your renamed APK file to Blackboard. You have 1 week to complete this lab.**