

CPSC 3367 – Mobile Apps Development

Homework 03 · Max grade: 100%

Download or clone [Resources project](#) and make the following changes:

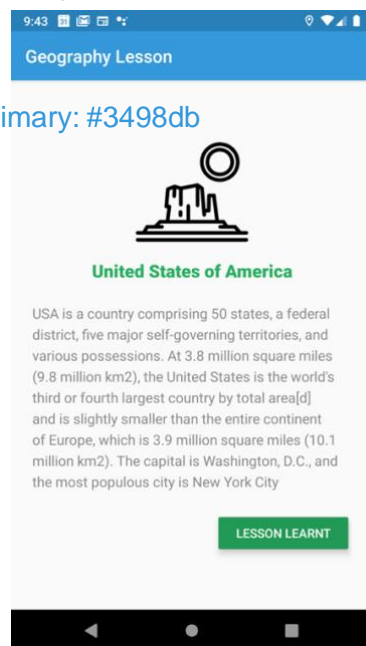
- Replace hardcoded padding, margin, text size, width, and height values used in styles.xml and activity_main.xml with their corresponding dimen resource. You'll have to create a new dimens resource file to define the values to be used.
- Use appropriately alternate resources to have a different look and feel depending on the system language. In particular, we'll use for each of the two considered languages the following specific properties:
 - Representative vector icon
 - Title
 - Description
 - Color palette

Some notes:

- The default language will be English and the other one considered will be Spanish (es)
- When we set the language of the device/emulator as English, the application will look like the one in the image on the left. For its part, when we set the language of the device/emulator as Spanish, the application will look like the one on the right:

colorPrimaryDark: #2980b9

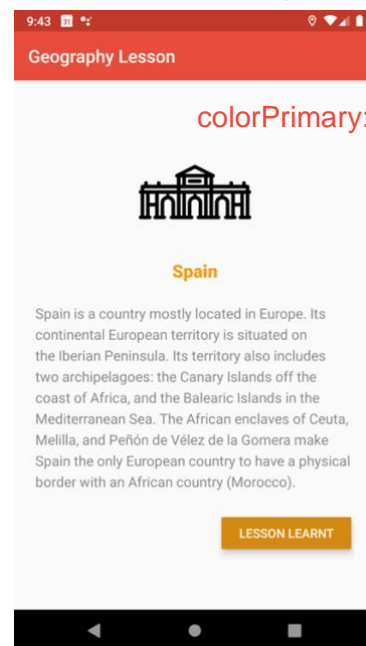
colorPrimary: #3498db



colorAccent: #27ae60

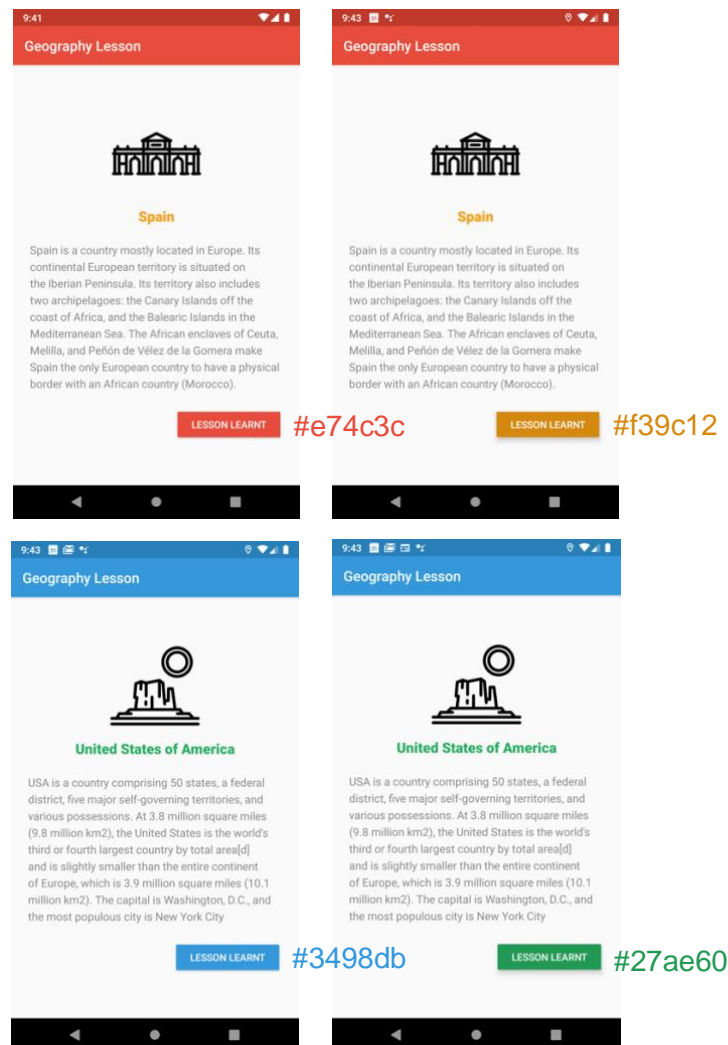
colorPrimaryDark: #c0392b

colorPrimary: #e74c3c



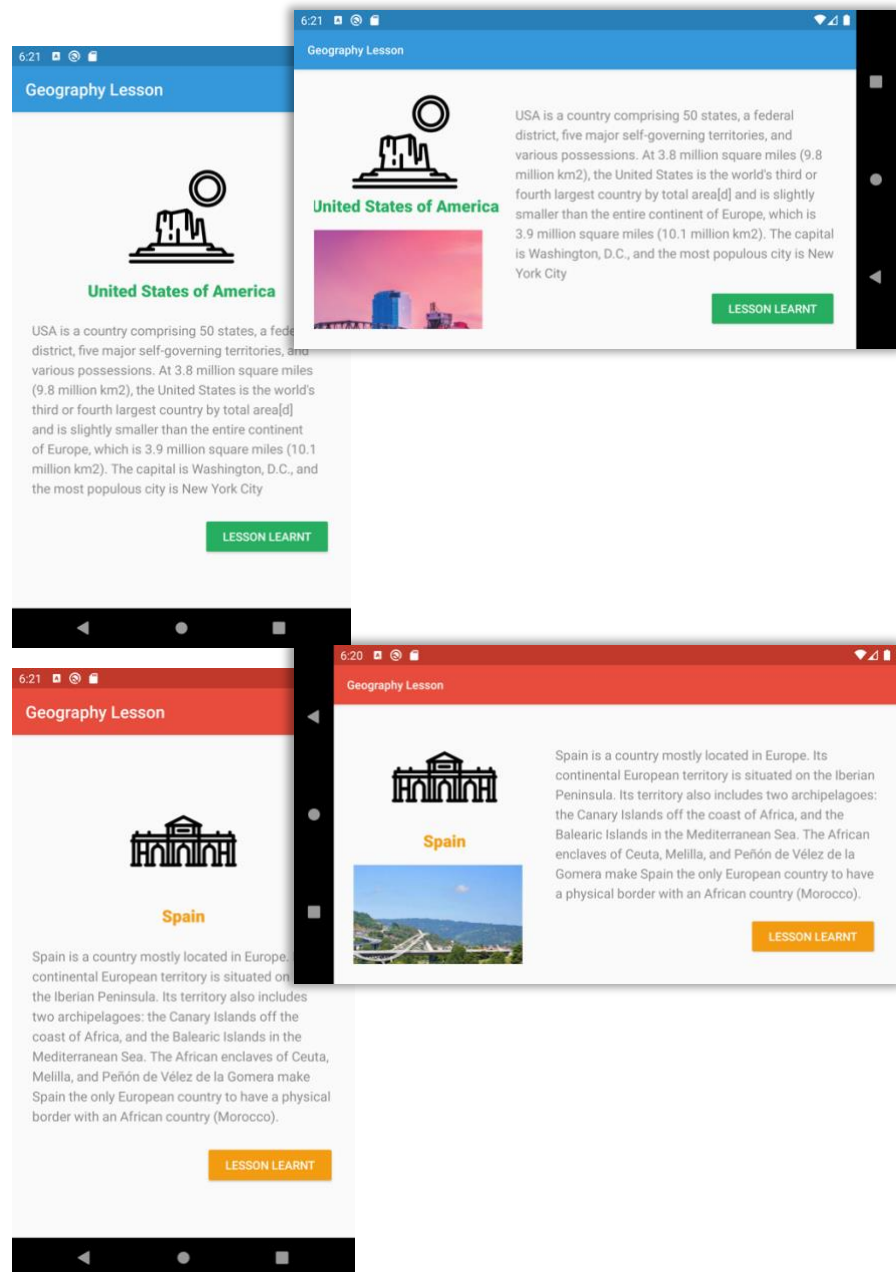
colorAccent: #f39c12

- As we can see in the images, we'll have to use a different color, string, and drawable resources depending on the selected language. You'll find the strings and pictures that you need in the [Google Drive folder](#)
- Note that:
 - In each language, the button has two different colors, one specific for the pressed state and another one for any other state. Just define colors to support this.



- We have language-specific strings and colors, but we also have strings and colors that are common to both languages.
- The description string won't be set in the layout XML file. The corresponding TextView has to be initialized with the provided strings in the MainActivity java class.
- Define appropriately alternate resources to have a different layout depending on the screen orientation. The layout for the portrait orientation is the one already included in the project.

The landscape layout can be found in the `landscape_layout.xml` file available in the Google Drive folder. Next, you can see how both orientations look like for each considered language. Note that in landscape orientation, the additional image displayed is language-specific. You can find the required bitmap images in the Google Drive folder.



- Once you have made the changes, please test everything works as expected. Check the [geography_app.webm](#) video for better detail of the expected result.

Summary:

- **[10%]** Replace hardcoded numeric values with dimen resources (Obviously, you'll have to create a dimens file).
- **[10%]** Replace hardcoded strings with string resources for every GUI text component, with the exception of the TextView with the "country_description_text" id.
- **[20%]** Define the alternative resources needed to have different content and look n feel depending on the device language.
- **[10%]** Get the description string value from resources.
- **[10%]** Initialize the text property of the TextView element with the "country_description_text" id by using the value retrieved in the previous TODO point.
- **[10%]** Define button_color resource. Remember, the button will have a specific color for pressed state and another one for the rest.
- **[20%]** Add a new layout for the MainActivity to be displayed on landscape orientation.
- **[10%]** Make sure the appropriate bitmap image is displayed for the two considered languages.