CST3104 – Introduction to Mobile Application Development

Final Project – Fall 2022

1. Goal

The Project will give you experience in developing software in a group environment

2. Teamwork

You will work in a team of 2 students.

3. GitHub

- It is recommended to work on GitHub for collaboration, but this is not a requirement for this project.
- If you choose to use GitHub as a team, follow these instructions:

1. One person should create a new project for the team and then upload it to GitHub using the menu option "VCS" -> "Share project on GitHub".
2. That group member must then invite the other to contribute.
    a. This is done by clicking on the "Settings" tab in GitHub, then click "Collaborators" on the left side menu and search the name to add to the project.
    b. Other team members should then clone that project to their computer and start making branches for their work.
    c. From Android Studio, select "File" -> "New" -> "Project from Version Control" -> "Git" and then paste the git URL from the main GitHub repository from the previous step.
    d. <u>You will not be able to integrate your work if you do not start by first cloning the project!</u>
    e. Write your own code on your own branch and then merge that branch on GitHub
    f. Commit, push and merge frequently
    g. Don't wait until the due day start merging the code.

4. Project Description

1. For this assignment, you will receive a starter project
2. It contains a basic code to load a list of assets in an ArrayList of objects.
3. The assets are all provided in the project:
    1.1. A list of images in the Assets folder
    1.2. A JSON file that contains the data

1.3. An Adapter class that converts the JSON file into an ArrayList of objects.

4. The application is a game where the user is provided with a picture of a US state and its capital.
5. The state to guess is randomly chosen
6. A list of names of the US states is displayed on the screen; and the user tries to guess the corresponding state.
7. If he clicks on the wrong state, a counter is incremented on a red background.
8. If he clicks on the right list item, the counter is incremented and shown on a green background, and the name of the state is displayed at the top of the screen in green.
9. Once the correct answer is found, any other click on the list will not incremented any more.
10. The user can replay the game if he clicks on a menu item on the toolbar.
11. The date and the score are stored for each play
12. The application needs to store the last, lowest and the highest scores along with the dates.
13. These scores should be displayed to the user every time the application is started.

5. Required Work

Create an application that offers the following:

1. The first screen presents a dashboard that shows the stored values for:
    - 1.1.1. Date and score for latest game result
    - 1.1.2. Date and score for lowest game result
    - 1.1.3. Date and score for highest game result
    - 1.1.4. A Play button
    - 1.1.5. An Info menu item on the toolbar
    - 1.1.6. When the user clicks on '**Play**' button, a new screen appears, showing a Game Activity
    - 1.1.7. When the user clicks on the '**Info'** icon, details about the application and its authors are displayed in a dialog.

2. The Game page consists of 2 parts:

- 1.1. The top of the screen displays the flag of the State along with the Capital.
- 1.2. The rest of the screen shows the list of all the US states.
- 1.3. The toolbar has 2 menu items:
    - 1.3.1. When the first is selected, the website of the state displayed at the top is shown
    - 1.3.2. If the second is selected, a new flag and capital is displayed at the top of the screen.

3. How the Game works:
    - 1.1. A random state is displayed at the top of the screen. Only the flag and the capital are shown.
    - 1.2. The user clicks on the items in the list, trying to guess the name of the state.

1.3. Once the user clicks on a list item, a counter is incremented.
1.4. The counter is displayed on the top right side of the screen.
1.5. If he selects an incorrect state, the area where the counter is displayed becomes red.
1.6. If he selects the correct state, the area where the counter is displayed becomes green, and the textview beside the counter displays the name of the state.
1.7. Once a correct answer is found, the counter stops incrementing, and the application does not check the subsequent list item selections. The final count stays on the green background.
1.8. To play again, the user selects the Overflow Menu item on the toolbar: a new state is displayed, the counter is reset, and the background color is reset to white.

6. Suggestions

- When the user selects a wrong or a good answer, you may want to add animation or sound effects to make the experience more engaging.
- You are free to style the application as much as you like
- To select a state to guess, you can use code such as:

```
// Returns an integer between 0 and less than maxBound
// Assume maxBound is already defined
private int chooseRandom() {
   Random rand = new Random();
   return rand.nextInt(maxBound);
}
```

-

7. Project Requirements

1. The software must have at least one activity written by each person in your team.
2. Each activity must have a help menu item that displays an AlertDialog with instructions for how to use the interface and information about the authors.
3. There must be at least one other language supported by your Activity. If you are not bilingual, then you must support both British and American English (words like colour, color, neighbour, neighbor, etc.). If you know a language other than English, then you can support that language in your application and don't need to support American English.
4. The application must use SharedPreferences or SQLite to save information about the application for use the next time it is launched.
5. The application must have properly aligned GUI elements.
6. The application should display properly when the device is rotated.
7. The functions and variables you write must be properly documented using Javadoc comments.
8. Every Java file must have a header containing at least this:

```
/**
 * Full Name:
 *
 * Student ID:
 *
```

```
 * Course: CST3104
 *
 * Term:  Fall 2022
 *
 * Assignment: Team Project
 *
 * Date :
 */
```

8. Additional Resources

1. https://www.geeksforgeeks.org/java-util-random-nextint-java/