CS 213 Fall 2020: Android Project

Chess OR Photos

Posted Fri, Nov 13
Due Thu, Dec 10, 11 PM in BitBucket

Worth 200 points (20% of course grade)

You will continue working in pairs.

You MUST use Git/Bitbucket to manage your repository. Your app MUST run correcty on the Nexus 4 (4.7 inch, 768x1280, xhdpi) device emulator that comes with Android Studio. Make sure your app runs on Android API Level 29 (Android 10.0 - Q) We will test only for this level.

You may serialize the data used in your app, but are not required to do so - you can use alternative formats for storage on the device.

Javadoc and UML are not required. Lateness penalties same as for Chess and Photos.

Photos

You are not required to carry over all the functionality from the Java FX implementation. Instead, you will implement a smaller set, as described below. You may reuse any of the code from your Java FX project.

Since the app will run on a personal smart phone, there is only a single user, who is the owner of the phone. So there is no logging in, and no admin functionality. Also, explicit captions are not required for photos (the filename will stand for the caption). Dates are not required either.

Your app should implement the following features:

- 30 pts Home screen. When the app comes up, it should load album and photo data from the previous session, if any, and list all albums with names in plain text. Off this "home" screen, you should be able to do the following, in one or more navigational steps.
- 40 pts Open, create, delete, and rename albums as listed in the Java FX project description. When opening an album, display all its photos, with their thumbnail images.
- 40 pts Once an album is open, you should be able to add, remove, or display a photo. The photo display screen should include an option for a slideshow, allowing you to go backward or forward in the album one photo at a time with manual controls.
- 20 pts When a photo is displayed, you should be able to add a tag to a photo. Only person and location are valid tag types; there are no typeless tags. You should also be able to delete a tag from a photo. Note: When displaying a photo, tags (if any) should be visible.
- 20 pts You should be able to move a photo from one album to another
- 50 pts You should be able to search for photos by tag (person and/or location), and matches should allow completion. For instance, if a location "New" is typed, matches should include photos taken in New York, New Mexico, New Zealand, etc. Note: You are not required to implement functionality to add new tag types, or save the matching photos in an album.

Chess

Port the terminal-based Chess program to Android: a chess app that lets two people play chess with each other on the phone. You need to carry over all the functionality from the Chess project, and implement some other new functionality.

Playing chess (120 pts)

- 30 pts Two humans can use your app to play a game of Chess - this is all the functionality from the Chess assignment.
- 20 pts Your app must draw the board with icons for the pieces, and correctly shaded squares.
- 20 pts Players must move their pieces using touch input - either dragging a piece or touching first the piece's original square and then its destination.
- 10 pts Provide an 'undo' function that will undo the last move (but no farther).
- 10 pts Provide an 'AI' button that will choose a move for the current player. Choosing randomly from the set of legal moves is sufficient.
- 20 pts Provide functional 'draw' and 'resign' buttons.
- 10 pts When the game is over, report the outcome.

Recording games (50 pts)

- 20 pts Record all games as they're being played.
- 10 pts At the conclusion of a game, offer to store it and prompt the user for a game title.
- 20 pts List all recorded games, sorted by both date and by title (user can select which view to choose).

Game playback (30 pts)

- A function that allows the user to play a selected recorded game. The selected game should be playable one move at a time, per player. The user of the app will click each time they want to see the next move (i.e. the moves are not auto-animated.)