

CS 213 Fall 2021: Android Project

Chess OR Photos (Implement one or the other, not both!)

Posted Fri, Nov 12

Due Tue, Dec 14, 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 correctly on the Nexus 4 (4.7 inch, 768x1280, xhdpi) device emulator that comes with Android Studio.

The latest version of the Android SDK is Android 12.0 (S), so may set your min-sdk to this. Alternatively, if you have an older version of Android Studio, with SDK version 11.0 (R) or 10.0 (Q), you may use those.

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-value pairs, just like in the Photos assignment (of course now limited to only person and location type tags). All matches are case insensitive, so "new york" is the same as "nEw YOrk".

Moreover, matches should allow auto completion, given a *starting* substring. For instance, when searching by location, if "New" is typed, matches should include photos taken in New York, New Mexico,

New Zealand, etc. In other words, all locations that start with the typed text. The user can then pick from this auto-completed list.

Searches apply to photos across all albums, not just to the album that may be open.

Note: You are NOT required to implement functionality to add new tag types, or save the matching photos in an album.

You may save photos to the device's gallery. Here's some info on how to access files (including photos) from storage: <https://developer.android.com/guide/components/intents-common#Storage>

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.

Note: We won't test checkmate, so if you didn't get to implement it earlier, that's ok, you won't have to do it for this assignment.

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 ONLY 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.)
-

Frequently Asked Questions

Q: Are there any particular requirements for the android projects as long as the project satisfies the functionality that is listed in the project description? For instance, should we have a certain project structure, should we be using specific UI elements, etc.?

A: No specific project structure or UI requirements.

Q: Do we have to use XML or FXML to format our android program, or can we do it natively in JavaFX?

A: You may only use Android XML (with Android tag namespace), not FXML. Also, you are required to build all your UI with XML, except for error/info dialogs where user action is limited to clicking a button for OK/Cancel/Yes/No type of thing.

Photos

Q: What happens if a photo is removed from the gallery after being added to an album, should we handle this kind of issue?

A: No, you don't need to handle this.

Q: Is the stock photo functionality required?

A: No.

Q: When we implement searching photos by tag, should we display all photos that satisfy it in general or only the photos that satisfy it in the particular album that is open?

A: Photos from **all** albums, not just the ones in the open album.

Q: If we tag image1 with tag1 in album1, and image1 is in album2 as well, should we update the tags for image 1 in album2?

A: Since there is no functionality to copy photos from one album to another, you can treat photos in different albums as independent of each other, even though they may be the same actual picture. Which means the same picture may have different tags in different albums. (A good implementation will make sure that the user doesn't import a photo into an album when the same photo is already in another album, but for the purpose of this project, we will allow it.)

Q: Are we allowed to use Picasso (or any other) library to display images in our program?

A: No.

Chess

Q: If a player does i.e. e2 e3 -> undo -> e2 d3, should the recording show the unwanted move, then the undo, and then the better move? Or is it okay if we only show the player did "e2 d3" in the recording and ignore the undo button?

A: You don't want to show the undo action, just show the final move after undo.

Q: Every time a player touches squares that would make for an illegal move, instead of popping up a dialog such as "Illegal Move, Try Again", could we just prevent them from touching illegal squares up front by making those squares non-responsive?

A: Yes, this is a much better way to handle it..

Q: About undo:

White: a2 a3

Black: a7 a6

Now it is white's move. And if white clicks undo, will it only undo the black's move? Or it will undo both?

A: Undo black's move. (Undo is done by the player that has just made the move, not the other player. So it is actually black undoing their move.)

Q: For game playback, can the user also request to see the previous move (or is it only next move)?

A: Only next move.

Q: For the GUI, do we need to implement moving a chess piece via dragging AND clicking 2 squares, or is one of the two options sufficient?

A: Either dragging, or clicking 2 squares, not both.

Q: Should the state of the game be saved and loaded if the user quits before the end? (For instance, they close the app before finishing the game and then open it again)

A: Not required.

Q: If the AI move results in a pawn promotion, should it prompt the user to pick the upgrade or should it pick randomly or should it promote to Queen by default?

A: Random piece (except pawn) or Queen are both OK.

Q: For the draw button on the chess project, does pressing draw always initiate a draw or does the draw have to be accepted/denied?

A: For the purpose of this project, pressing draw will automatically result in a draw without requiring the other player to accept/deny.