CS 443 Mobile Applications Xu Huang Lin

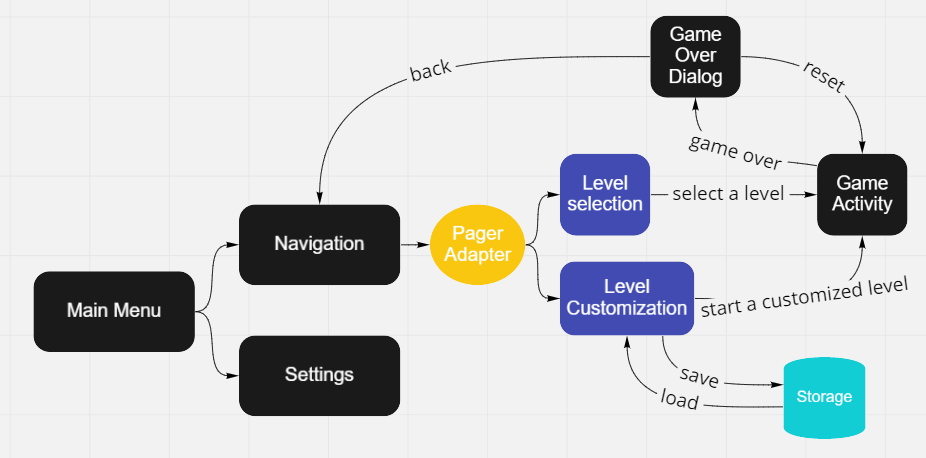
Prof. Bo Sheng 5/21/21

**Project Description Document**

1. **Project Statement**

I want to develop a customizable chess game application where the chess rules apply but the board size and the setting of pieces vary. You can use it to play chess with the computer or another person. I end up developing this because I wanted to make an app that makes me want to continuing developing after this course is over. Users who like chess but want something different might benefit from this application. There are plenty of regular chess apps on the Google Play store, but I do not think there is one that supports DIY. No special requirement for using this app, as long as the system is Android 5.0 or above.

1. **Application Design**



* No modules nor external services are used.
* This application is targeting smartphones and tablets.
* In the main activity, a user can press “Start” button to go to the navigation activity, or press “Settings” button to go to the settings activity.
* Settings:
  + Can turn on or off to rotate pieces when it is black’s move
* Navigation activity containing a tab to navigate between levels and customize page.
  + In levels page, a user can select which level and whether to not play with the computer or with the computer as black or white.
  + In customize page, a user can create a level by setting the dimension of the board and the placement of pieces. A customized level can be saved to or loaded from the local storage before starting.
* In the game activity, a user can play the chess. The chessboard will show hints for move option when a piece is selected and highlight for the last move. There will a text indicator showing who is the player to move now.
  + Game over dialog pops up when the game is over. A user can choose to reset the game or go back to the level selection page.

1. **Application Implementation and Evaluation**

* The navigation activity is using PagerAdapter to navigated between the levels page and the customization page.
* The game view is using View object’s canvas to draw the board and pieces. The user interaction with the chess board is handled using onTouchEvent() function of the View object. Canvas is updated whenever a user touches or after a move is made.
* The chess board mechanism is handled using the Chessboard class, including making moves and checking the game state.
* The state of the game is represented using the GameState class. It includes various information for the chess mechanism and for the calculation of the computer player.
* The way how game over is determined is by keeping a count of pieces. This way can support multiple King pieces.
* Each piece is represented using Enum Piece. The data structure that is representing the chess board is a 2d array of Enum Piece. Level options are represented as an array of 2d array of Enum Piece in Levels.java file. A new level can be added simply by adding a 2d array of Enum Piece in this file.
* The computer player is handled by the PlayerAI class. The algorithm is using MinMax and iterative deepening search.
* I test whether the application has bugs, and all the functionalities are working properly by running the app and play with myself or with the computer player.
* I test the performance of drawing by selecting the pieces and make moves, then see if there is any noticeable delay.
* I test the performance of the computer player by looking at the Logcat and see the maximum depth of the iterative deepening and the score of the favorite move.
* Had few bugs caused the application to crash. They were mostly caused by null pointer exceptions. Often can be solved quickly, but few times had to use the debug mode to find which line of code caused the issue.
* Had a bug that missed up the chess board. The cause was that I did not copy the 2d array that represents the board properly. Had to run a for loop to clone each array within the 2d array.
* Had a bug that made the wrong move options. Due to how the board is represented using a 2d array, the way how to access a block with x and y values should be board[y][x], but what I did was board[x][y].
* Had few minor bugs related to the algorithm in PlayerAI class. Had to use the debug mode to trace the issue.

1. **References**

Got the inspiration of this project from the assignment 4 of my Artificial Intelligence course, which is taught by Prof. Marc Pomplun.

1. **Experiences and Thoughts**

It will be nice to have the online functionality included, such as sharing customized levels and play with others online. I was thinking about supporting up to 4 players. I could write a better getScore() function to further improve the computer player. Also, there are some issues created in the GitHub repository which are features I want to add, but not now since the time is not enough. I will consider adding above functionalities maybe in the future.