

Implementation

Outline of the code we need

This chess program has two main roles:

- 1) Help users learn the valid moves in the game of chess, and enforce those rules
- 2) Allow human players to play a game of chess

We must allow exactly two people to play a game of chess at the same time, and we must enforce all the rules of chess except for determining draws. The game only knows who's turn it is (white or black). To achieve this, we are planning to implement an Object-Oriented approach. We will create a parent class called Piece, which has the x and y positions, the colour and the name of the piece as attributes. We will also create a derived class for each of the types of chess pieces: King, Rook, Pawn etc. This will allow us to store the possible moves for each individual type of piece within the class, instead of storing all moves in the Piece class and selecting the valid moves based on the type.

We will also need other classes and functions to meet all the functional requirements. My idea is to have a class called Setup which asks and stores the users name and colour selection, and sets up the chess board using JavaFX. We should also have a class called Game, which holds the functions for piece selection, movement, piece removal, detecting check and checkmate. Two more classes needed to meet the functional requirements can be called Quit, which holds the functions for quitting the game and ending the game, and Replay, which stores the game state and loads the previous game when requested by the users. We will also need JavaFX code to generate the image display for the board, pieces and start-up menu, as well as updating the piece positions throughout the game, but the planning and implementation of that will be left to the members of the group who are handling the UI implementation, and are more experienced with JavaFX than I.

What problems will be difficult

Most of the group members took the C and C++ module, and the main assignment for that module was to create a garden simulation. The basic logic for the movement of the organisms in that program can be applied to the movement of the chess pieces in this program, and the detection of check and checkmate can be compared to the checking of food or mates in the organism's vicinity in the garden simulation program. There were also functions in that assignment for quitting the game and (optionally) detecting whether the game was over, so these could be applied as well.

However, this is not to say that it will be easy to implement these functions. Despite most of this group being more experienced with Java than C, it will still take time to translate the functions to Java. And while some of the basic functionalities of movement and detecting check and checkmate can be carried over, it will not be an exact match. Using nested for loops to detect positions of pieces and determining if the King is in check or checkmate after every turn could become complicated, especially when the time for each operation must take no longer than a second.

There are a few functional requirements which will also take some extra thought. None of us have much experience with JavaFX, so implementing the UI will likely be challenging, and will likely be the first main priority after we get a basic template of the code. Replaying the game and storing the positions of the pieces could also be challenging. This will likely involve writing all the piece's x and y co-ordinates, type, colour and player name to a .txt file, and reading it when the user chooses to replay their last game. We will also have to include code to detect if there has been no game in the

past yet, or the last game has been finished. This could be represented by that option in the start-up menu being greyed out, another task for the UI team.