

1. Running the program is really simple since we only have one program, the Chess game. You simply compile and run the Chess.java class and the Chess board will pop up which you can interact with
2. We used threads for the purpose of animating our chess pieces.

Inheritance was plentiful in our project, and necessary for the implementation of our pieces in the game. We had an abstract Piece class, which each specific chess piece extended.

Events in our program we used java swing components to control the timers for players, reset the game, select the pieces to move, and select the tile the pieces are to move to.

The most prevalent data structure in the program is a two dimensional array which holds all of the tiles of our chess board.

A new feature implemented is a new menu (A JOptionPane) that is displayed when moving a pawn to the opposite end of the board and allows the player to select the new piece they want to replace the pawn. A JOptionPane is also shown in the determined situations in which a king is in checkmate, and this pane will display which player is the winner.

3. We used GitHub to keep all of our up to date code available to everyone in our group. We had to be very careful while all coding at the same time to make sure we didn't have any merge errors.
4. Specific contributions:

Emma Flatland: I helped to design & build the ChessTimer class. In the King class, I worked on the constructor, the getValidMoves method, the isCheck method, and the seelfInCheck method. I also helped come up with the logic in the MovingThread class in collaboration with Nicky, and I helped to figure out how to get the images of the pieces to display on the board. For documentation, I wrote a brief description of each class, documented large segments of code in the Chess class, and documented the instance variables in the Chess class. I also helped with testing & debugging.

Nicky Morgan: One of my major contributions to this project was the logic for moves on the Bishop, Rook, and Queen pieces. I also worked on the Java Swing window display and the MovingThread which is used for animation. Much of the MovingThread was done in collaboration with Emma. The group also met many times throughout the project to work together and compare ideas, making sure we were all on the same page. I worked with Arnold for many hours to make sure everything was working correctly, working on moving pieces, and

helped with checkmate logic, but much of that was done by Arnold and I was there helping if needed.

Ethan Tubia: My contributions to the project included the assistance in the original design of piece logic in the Rook and Bishop classes as well as the addition and proofreading of javaDoc comments for each of the methods in each class. I was also in charge of the acquisition of piece images.

Arnold Bova: My main focus was piece logic and board generation. I designed and implemented all of the logic needed to build the board. I also helped work on click detection for the board so that a user can click on the screen and show where the piece clicked can move.