Chess

# VISION

The program should function as a standard, playable chess game.

# MAJOR FEATURES

* Board and pieces display
* Pieces move and capture according to standard rules
  + Low priority: En passant and castling
* Game ends on checkmate
  + Low priority: draw and forfeit
* Accepts user input
  + Two human players are required
  + Input format is consistent with the format used for the command-list file
  + Low priority: read moves from a file
  + Low priority: Computer AI for second player

# TEAM Chess Masters

## MEMBERS

Dylan Roberts (Team Lead)

Joe Reed (Scrum Master)

Travis Eggett

Josh Conlon

## DECISION MAKING PLAN

The group will discuss options and questions and attempt to arrive at a unanimous consensus. If such a consensus cannot be reached, a poll will be held in the Zoom chat (or whatever mode of communication is available), and the majority vote will rule.

# REQUIRED TECHNOLOGIES

The application will be a command-line application that prints new information whenever a view update takes place. The application will be written in Java 8 using Maven. Testing will be conducted manually (i.e., not with any particular testing framework). The project will be stored in a Git repository, hosted in a private repository on GitHub, using the Pull Request system.

## UNDERSTOOD TECHNOLOGIES

Our team is largely familiar with Java 8 and Git.

## NEW TECHNOLOGIES/CONCEPTS

The Maven dependency management/build system will be new for some of us. We can use this to simplify sharing code and working across different development environments.

# DEVELOPMENT PLAN

## PROTOTYPE

We will have a prototype ready by the end of the first sprint. The board and pieces will be displayed, and some basic but realistic piece movement will exist via user input.

## BETA VERSION

We will have a viable beta version ready by the end of the second sprint. The game will appear to function smoothly for user input, with all basic rules implemented. A game may be won via checkmate. All high priority targets will be complete.

## FEEDBACK AND TESTING

* Correctness
  + Piece movement
  + Checkmate is detected
  + Input is interpreted correctly
* Ease of use
  + Whose turn is it?
  + Am I in check?
  + Is my move legal?
  + How do I submit a move?

## FINAL REFINEMENT

Bug fixes will be important in this stage. If we still have special moves to implement, try to add them. Improve UI/UX according to tester feedback. Sound effects may be added.