**Product Backlog**

User launches program and sees a title/introduction screen/message

Allow user to select a timed game or non-timed game

Create 9x5 board

\*Change board size

Make sure initial game board is valid

Draw pieces

Create button/function to start and restart game

User input box

User enters a move by inputting board positions

User enters a move by defined click-and-move process

Maintain board state

User sees a display of the current board state

Track whose turn it is

Display whose turn it is

Check to see if both players still have pieces.

Keep track of pieces that are gone

Display removed pieces on the side of the board

Check to see if player has any moves to perform. If they don’t, they lose

Check if maximum number of moves has been reached.

User gets response of invalid if proposing an invalid move

Computer identifies all valid moves

Computer can randomly select a valid move

Board evaluation function determines winner/loser

Highlight the pieces that have valid moves available

\*Show all valid moves for a selected piece

Only allow users to effect pieces they control

Allow diagonal and forward movement

Disallow pieces to move to a spot with a piece in it

Track where pieces have been, they are not allowed to be in the same space

Track multiple jumps

Captures 2 different ways

If a capture move can be made, it must be made

Allow to move only if cannot capture

Allow user to pass turn

\*Allow user to undo a move

Min/max tree used for AI

Computer looks ahead one move in tree

Alpha-beta pruning implemented at one level

Alpha-beta pruning implemented across all levels

Iterative deepening implemented

Make a board that can be accessed remotely

Make an AI that can be played against remotely

Timer for AI

Timer for User

Visual for AI’s last move

Visual for user’s last move

Display count down timer

Program AI with staggles

Difficulty settings on the AI

Make funny names for the AI difficulty settings

\*Make a chat function

\*Make AI taunt in chat