Project 1 Proposal

Name: [Young Ha Kim]

Chess Game

(a) Description: A chess program where you can play the following modes:

- Play against the computer AI. (Algorithm listed in (d))
- A chess *cheater/advisory* mode, where you can input moves for a physical chess game with 2 players manually and get advice on the potential moves **you** intend to make.

(b) Classes:

- Game class: Manages the whole game, owns the Board class, Player classes and will determine if the game is finished and prompt each player according to the their turn. An instance of this class is an entire game in itself.
  - Methods: runGame(): runs game loop
  - Attributes: Board, Player classes, Game Mode(against computer vs cheater mode)
- Board class: Keeps the location of each piece that is currently on the board
  - Methods: drawBoard(): draws the board to the screen
  - Attributes: list of list structure to hold each piece in location a_{ij}
- Player class: Base class for player, attributes will represent the state of each player, turns taken, time spent on each turn, pieces captured etc
  - Methods: playTurn(board): returns move, input is board
- Human Player class, AI Player class: Will inherit the Player class and will be setup for input/output according to the type of player
- Piece class: Will be the base class for all chess pieces in the game
  - Methods: possibleMoves(board): returns list of possible moves given board as input
  - Attributes: currentPosition, player(of ownership), isCaptured
- King, Queen, Bishop, Knight, Rook, Pawn classes: Will inherit the Piece class for specific implementation of possible moves for each piece
- ChessProblem class: Will be the AI model class, which describes the chess game. Class implementation depends on model selected
- TrainingAlgo class: The class which will train the AI model
- ScreenUtil class: Manages output to the screen, colors, prompts etc

(c) Interaction: The board will be displayed and updated throughout the entire game. The program will prompt each player on their turn for either a move decision, or for advice on a potential move. Once a player makes a move decision, turn passes on to the next player. Each player will take turns moving pieces until the enemy King is captured or will be captured for sure in the next move (Checkmate) or if the other player concedes.

(d) Complexity: The feature I plan to implement for the chess game is the computer AI, and the advisory AI. The algorithm will be, at a minimum, DFS-ID (Depth-First Search Iterative Deepening) and possibly A-star, RL(reinforcement learning using Markov Processes) or MiniMax if performance is unsatisfactory. The choice of model will be made after some experimentation and also within the code length limitation of 750 lines.