Contrasting Chess

Software Design Document

Troy Neubauer

Date 2/22/2021

CS 225, Spring 2020

Embry-Riddle Aeronautical University

Daytona Beach campus

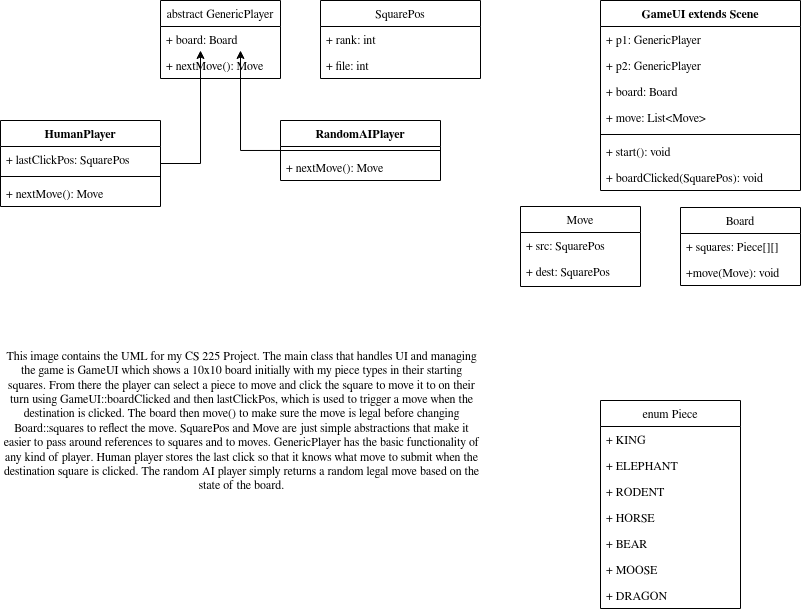
1 Aerospace Boulevard

Daytona Beach, FL 32114

**Introduction:**

Contrasting Chess is a turn based strategy game similar to chess, where users can play with other players on the same computer or against an AI.

**UML:**

****

**Problem Description:**

Contrasting Chess is loosely to chess in overall feel, but with very different moves for different piece types, creating a whole new kind of strategy! Players compete to checkmate their king first, but unlike chess stalemates don’t exist. This software will check all moves for legality so that no invalid position is ever reached.

**User Stories:**

Provide a listing of the user stories you created in the Agile Tracking Sheet for reference.

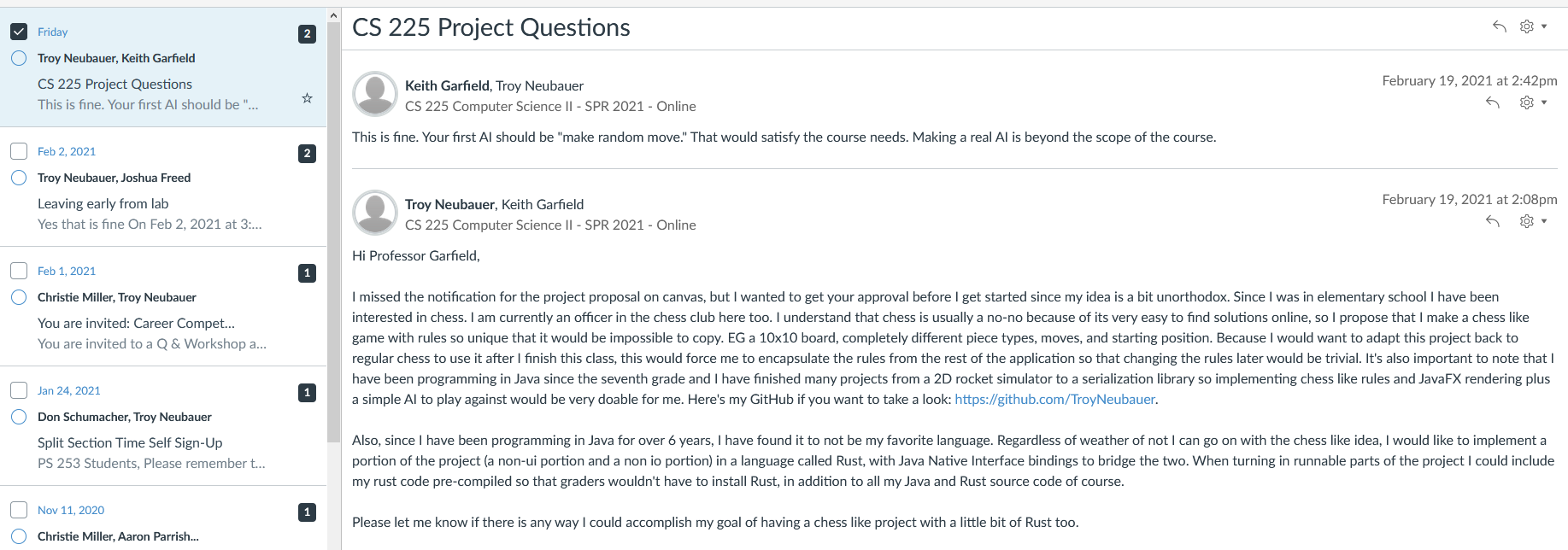
**<User story> ID**

|  |
| --- |
| As a player I want a window to open with the starting position 1 |
| As a player I want pieces to display on the board 2 |
| As a player I to move pieces 3 |
| As a developer I want to be able to load a Rust library from a .dll file 4 |
| As a user I want the software to jump to a chess position a FEN and PGN file 5 |
| As a user I want non check moves to be checked for legality 6 |
| As a user I want to play against an AI that plays random legal moves (illegal checks remain) 7 |
| As a user I want to only be able to play only legal moves (no illegal moves while in check) 8 |
| As a user I want to play against an AI that plays better than random moves 9 |
| As a user I want the AI to be stronger at the game 10 |
| As a user I want to be able to run a contrasting chess server and play games with anyone in the world 11 |
| As a user I want to be able to run the application in my web browser without having to download anything 12 |

**Problem Solution:**

P2:

To abide by restrictions established when reaching out to the professor to make a chess-like game:

 Every piece will be different to chess with the exception of the king along with other rules listed below. The five other piece type’s names and moves are TDB along with the starting position. The rules I have decided on are below:

* Like chess:
  + The king cannot execute any regular move that places him in danger (check)
  + The goal is to checkmate your opponent
    - This is a position where the opponent’s king is attacked and has no legal moves
  + While in check the only allowed moves are ones that bring the king out of check
* Unlike chess:
  + The king can move one square horizontally (like chess), or one square diagonally after not moving for one move (unlike chess).
  + Castling doesn’t exist because rooks don't exist
  + En-passant pawn capturing doesn’t exist because pawns don't exist
  + Any move that puts your opponent in stalemate (a position where a player has no legal moves) is illegal
  + Different:
    - piece moves
    - starting position
    - board size (10x10 vs 8x8)
    - standard time control rules for classical games (TBD)

**References:**

P2:

**Appendices:**

P2:

[https://youtu.be/U4ogK0MIzqk - Coding Adventure: Chess AI | Sebastian Lague](https://youtu.be/U4ogK0MIzqk)