Product Specification

Chess

Product purpose

The team has been tasked to program a chess application. The team will do it's best to create a fully functional chess application with all rules. The goal is to provide a chess aplication which will function as a traning ground, but also bring hours of joy and entertainment for chess enthusiast, newcomers and experts alike.

Functional requirements

- A player should be able to choose wether to play as black or white
- The system should have a visible chess board with chess pieces which the player can move
- The system should calculate a player score based on performance during a chess game and display it
- System should validate player moves according to chess rules
- A player should be able to move chess pieces according to chess rules
- A player can select a chess piece by clicking it
- A player may choose another chess piece by clicking on it
- A selected game piece must belong to the active player
- The player moves a piece by clicking and dragging it to desired tile
- A player should be able to forfeit a game at any point
- A game is ended immediately a player forfeits
- A confirmation window should pop up when a player forfeits

- The system should let the player play against another player locally
- A player should be able to play against a computer
- A player should be able to choose difficulty against computer
- A player should be able to undo moves on easy mode against computer
- The system should highlight possible moves for a chess piece when selected on easy mode against computer
- A player should be able to undo up to 3 moves against computer
- The gameboard will be reverted to the previous state when a move is undone
- A player should be able to knock out an enemy chess piece by doing a legal move onto a tile with an enemy chess piece
- The captured pieces should be displayed in a box next to the chess board

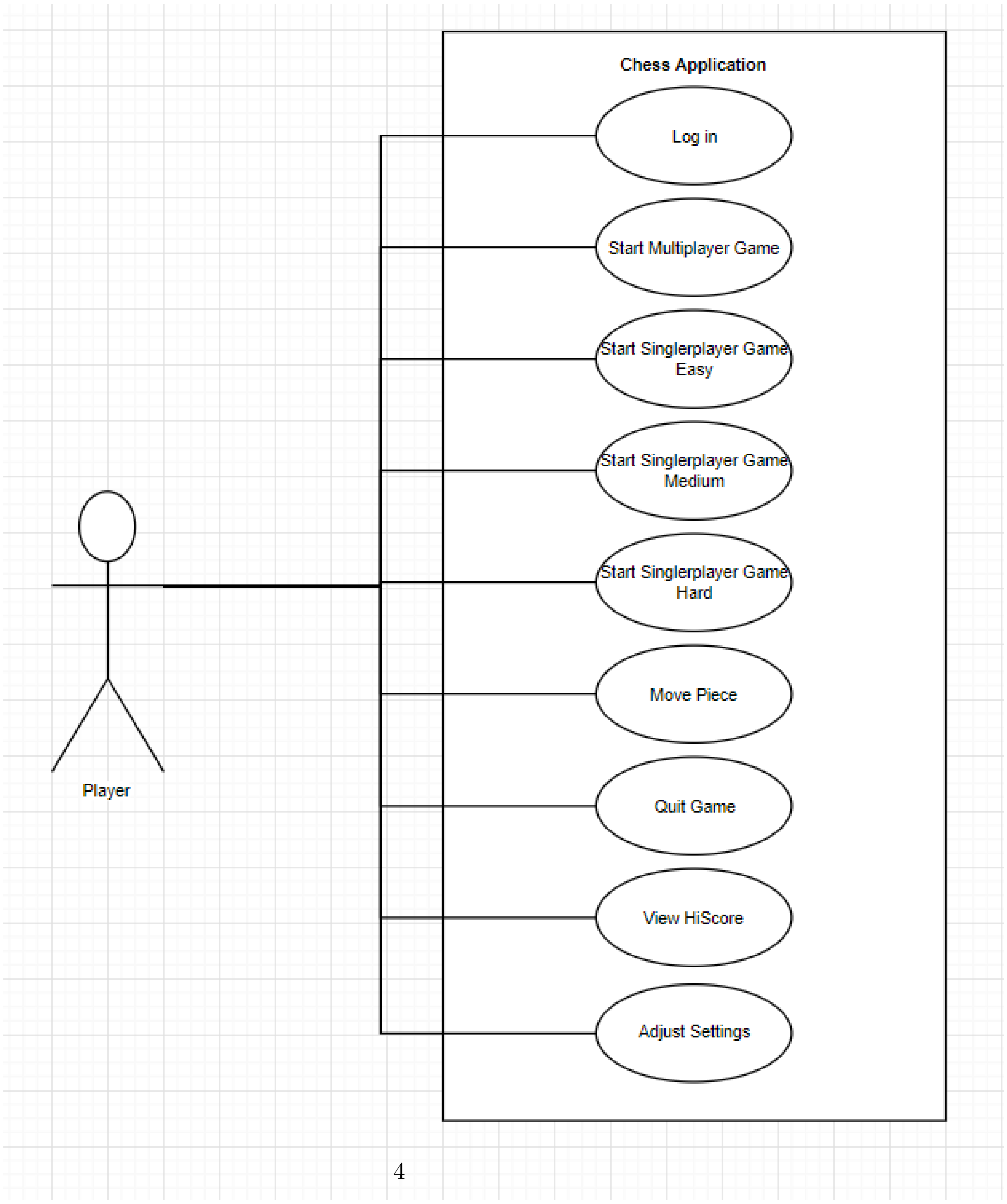
Non-functional requirements

- The application should be executable on most mainstream OS

User stories

- As a user, I can choose wether to play as black or white.
- As a user, I can choose diffuculty against the computer.
- As a user, I can undo moves when playing on easy difficulty against the computer.
- As a user, I can view the highscores.
- As a user, I can close the application.
- As a user, I can view tips & tricks
- As a user, I can move a pawn 2 squares the first time I move it, 1 square otherwise. I can't move it backwards or horizontally, and I can only knock out enemies diagonally. Special moves I can perform is ”en passant” and ”pawn promotion”.
- As a user, I can move a bishop horizontally and vertically as long it's not obstructed. I can't change direction the same turn. Special moves I can perform is ”castling”.
- As a user, I can move a knight either 2 squares horizontally and 1 square vertically, or 2 squares vertically and 1 square horizontally.
- As a user, I can move a bishop diagonally as long it's not obstructed. I can't change direction the same turn.
- As a user, I can move a king 1 square in every direction. I can't move the king into a position that checks him.
- As a user, I can move a queen in every direction as long she's not obstructed.
- As a user, I can forfeit a match.

Use case diagram



Fully dressed use case

Use Case 1: Move chess piece

Scope: Chess application

Level: User goal

Primary Actor: Player

Stakeholders and Interests:

> Player: The player wants to be able to move his chess pieces accordingly to what the rules allows
>
> Programmer: The programmer wants the application to always follow the rules of a standard chess match

Preconditions: Player has started a game and it's the player's turn

Postconditions: Player has successfully moved chosen chess piece and it's opponents turn

Main Success Scenario:

1. It's player's turn. Player chooses piece to move.
2. Player drags chosen piece to desired square.
3. System validates player's move.
4. Player does a legal move and the chess piece stays on the chosen square
5. It's opponents turn

Extensions:

4a: Illegal move

1. Player's chess piece is taken back if move is illegal
2. Player has to make a different move

4b: Knocking out enemy piece

1. Player does a legal move and the chess piece lands on an enemy chess piece
2. Knocked out chess piece is removed from the board and displayed in the captured pieces window

   3. Player's chess piece stays on the chosen square

 4c: Putting opponent into check

   1. Player does a legal move and puts enemy king in check
   2. Player's chess piece stays on the chosen square
   3. Opponent MUST move their king out of check

 4d: Putting opponent into checkmate

   1. Player does a legal move and puts enemy king in checkmate
   2. Player wins the match

Use case 2: Player turn

Scope: Chess application

Level: User goal

Primary Actor: Player

Stakeholders and Interests:

 Player: The player wants to know what they can do when it's their turn

 Programmer: The programmer wants to ensure that the player is granted all possible options the player should have

Preconditions: Player has started a game and it's the player's turn

Postconditions: Player's user input is executed

Main Success Scenario:

1. It's the players turn
2. Player chooses piece and moves it
3. Opponents turn

Extensions:

 2a: Pausing the game (against copmuter)

   1. Player hits the pause button
   2. Pause pop up appears
   3. Application waits until player presses resume

2b: Player forfeits (against computer)

1. Player hits the forfeit button
2. Confirmation pop up appears
    - 2a. Player confirms forfeit
        1. Match ends, player lose
        2. Application goes back to main menu
    - 2b. Player cacncels forfeit
        1. Match continues

2c: Player forfeits (against another player)

1. Player hits the forfeit button
2. Confirmation pop up appears
    - 2a. Other player accepts
        1. Match ends, forfeiting player lose, other win
        2. Application goes back to main menu
    - 2b. Other player cancels
        1. Match continues
        2. Forfeiting player can't forfeit until next turn

2d: Draw by agreement

1. Player hits the draw button
2. Confirmation pop up appears
    - 2a. Other player accepts
        1. Match ends in draw, no winner
    - 2b. Other player declines
        1. Match continues
        2. Requesting player can't request another draw until next turn