SENG300 Final Project

Game Logic Use Case Diagram

SENG 300

Dr. Steve Sutcliffe

Feb. 25th, 2025

Table of Contents 01

Play Game
Use case: Play Game
Iteration: 1
Primary Actor: Player
Goal in context: To allow the user to access a game and play a match.
Preconditions: Player has logged in and selected a game to play.
Trigger: The player presses play.
Scenario:

1. Player starts the session.
2. Player selects chess.
3. Player gets matched with another player to play chess.

Post conditions: Allows the user to have a match against another player.
Exceptions:

1. The user input systems are not functional.
2. The software malfunctions.
3. The game crashes.
4. The game has bugs.

Priority: High priority, core functionality that is required to make the system functional and usable for players.
When available: Should be available at any point once the user is logged in.
Frequency of use: Very frequent. Used by players to make use of the gaming system.
Channel to actor: Graphical user interface that player accesses from their computer.
Secondary actors: Second Player
Channel to secondary actors: Graphical user interface that player accesses from their computer.
Open issues: N/A

Move Piece
Use case: Move Piece
Iteration: 1
Primary Actor: Player
Goal in context: To allow the user to make a move in chess and checkers
Preconditions: Player has started a game of either chess or checkers
Trigger: The player clicks on the piece they intend to move, then selects the square on the grid they want to move to.
Scenario:

1. Player is playing chess
2. It is the player's turn
3. Player wants to move rook to b6
4. Player clicks on rook and selects b6

Post conditions: The piece changes location and either the game ends, or the other the player gets a turn.

Exceptions:

1. The user input systems are not functional.
2. The software malfunctions.
3. The game crashes.
4. The game has bugs.
5.

Priority: High priority, core functionality that is required to make the system functional and usable for players.
When available: Should be available at any point while the game is still ongoing, and it is the player's turn.
Frequency of use: Very frequent. Used by players to make use of the gaming system.
Channel to actor: Graphical user interface that player accesses from their computer.
Secondary actors: Second Player.
Channel to secondary actors: Graphical user interface that player accesses from their computer.
Open issues: N/A

Castling
Use Case: Castling
Iteration: 1
Primary Actor: Chess Player 1
Goal in context: Perform a castling move to safeguard the king.
Preconditions:

1. A game of chess is ongoing and it is Chess Player 1's turn to move.
2. There are no pieces between the king and the chosen rook.
3. The king and the chosen rook have not made a move.
4. The king is not in check, will not pass through a check, and not land on a check after castling.

Trigger: Chess Player 1 decides to perform the castling move and all the preconditions are met.

Scenario:

1. A game of chess is being played between two players.
2. As the game goes by, Chess Player 1 wants to perform castling in order to place their king in a safer position and their rook in a move active location.
3. Chess Player 1 is able to perform the move if all preconditions have been met.
4. If preconditions are not met, notify Chess Player 1 as to why (visually).
5. Castling is performed and the game continues.

Postconditions: Castling has been successfully performed and the king and rook have performed castling, their positions updating where the king moves two squares in the direction of the rook and the rook to the opposite side of the king.

Exceptions:

1. The king has already been moved.
2. The chosen rook for castling has already been moved.
3. Coding error that allows illegal moves without fulfilling the preconditions.

Priority: Medium, it is an element of chess used often but not game-ending like checkmate.
When Available: As soon as the chess player decides to make the castling move assuming they fulfill all the preconditions.
Frequency of Use: Often but at most **once** each game for each player, and on occasions not at all if decided so by the player.
Channel to Actor: Game GUI
Secondary Actors: N/A
Channel to secondary Actors: N/A
Open Issues: N/A

En Passant
Use Case: En Passant
Iteration: 1
Primary Actor: Chess Player 1
Goal in context: Capturing the opponent's pawn piece by performing the En Passant move.
Preconditions:

1. A game of chess is ongoing and it is Chess Player 1's turn to move.
2. Chess Player 2 has moved their pawn piece two squares forward from its starting position prior.
3. Chess Player 1 has an adjacent pawn to the pawn Chess Player 2 moved.

Trigger: Chess Player 1 decides to perform an En Passant capture and all the preconditions are met.

Scenario:

1. A game of chess is being played between two players.
2. As the game goes by, Chess Player 2 moves their pawn two squares forward.
3. Chess Player 1 has a pawn that happens to be adjacent to Chess Player 2's pawn which had just performed a two square move forward.
4. Chess Player 1 has the option to perform the move En Passant and does so.
5. If Chess Player 1 refuses to do so then the move is no longer possible unless all preconditions are met once again.

Postconditions: If Chess Player 1 has chosen to perform En Passant, then Chess Player 2's pawn is captured and Chess Player 1's pawn is moved behind this captured pawn.

Exceptions:

1. Chess Player 1 has chosen not to perform En passant so move is then no longer possible in the next turn.
2. The opponent's pawn was not moved two squares forward from its starting position (If it instead made two moves one square each.)
3. The pawn making the capture is not directly adjacent to the opponent's pawn.
4. Some coding error to allow an illegal move of En Passant without fulfilling preconditions.

Priority: Low, it is an element of chess but not game-ending like checkmate and not Higher in priority as castling.
When Available: As soon as the both player's pawn meets the precondition but only during the turn the preconditions are met by the player performing the move.
Frequency of Use: Rarely, can only occur when the preconditions have been met.
Channel to Actor: Game GUI
Secondary Actors: Chess Player 2 (Opponent/Player being checkmated).
Channel to secondary Actors: Game GUI
Open Issues: N/A

Place Piece
Use case: placePiece
Iteration: 1
Primary Actor: Player
Goal in context: Allow the player to place a piece (specific to connect 4 and tic tac toe)
Preconditions: There is a game ongoing between the player and another player, and it's this player's turn.
Trigger: Player is ready to make a move.
Scenario:

1. Player turn to place piece
2. Player chooses cell on board for piece to be placed to

Post conditions: If cell is open and move is legal based on piece movements, piece is placed in that cell
Exceptions:

1. Move is illegal (cell is taken)

Priority: High
When available: When it is the player's turn.
Frequency of use: every turn so long as the game is ongoing
Channel to actor: Controller class, player chooses piece and placement
Secondary actors: N/A
Channel to secondary actors: N/A
Open issues: N/A

Surrender
Use case: Surrender
Iteration: 1
Primary Actor: Player
Goal in context: allow player to surrender the game
Preconditions: there is a game ongoing between the player and another player
Trigger: player wants to surrender
Scenario:

1. Player thinks they cannot win
2. Player presses surrender button

Post conditions: Game ends, player gets a loss, enemy player gets a win
Exceptions:

1. Game is already over

Priority: Medium
When available: All game
Frequency of use: not often, likely rarely
Channel to actor: Controller class, GUI button to surrender
Secondary actors: N/A
Channel to secondary actors: N/A
Open issues: N/A

In Check
Use case: inCheck
Iteration: 1
Primary Actor: Player
Goal in context: player is in check in chess
Preconditions: Enemy player moves a piece in a way where it puts the King piece at risk.
Trigger: King is then in check
Scenario:

1. Enemy player's move puts player's king in check
2. 2. Player is blocked from doing any moves that would not move king to safety or lock the check with another piece

Post conditions: Player must stop the check either by moving the king out of the way or blocking the check with another piece
Exceptions:

1. There is no way for the player to save the king, making this a checkmate not a check

Priority: High, required for proper chess functionality
When available: All game
Frequency of use: Often
Channel to actor: Controller class
Secondary actors: N/A
Channel to secondary actors: N/A
Open issues: N/A

No Pieces
Use Case: noPieces
Primary Actor: Player/System?
Goal in Context: When the opponent player has no pieces
Preconditions:

1. One Player has no pieces on the game board

Trigger: When a player checkmates an opponent or has no pieces on board or has no other valid moves left.
Scenario:

1. Two Players are playing the game until
2. Player 1 checkmates Player 2
3. And Player 1 captures Player 2 King
4. Player 2 has noPieces left
5. Player 1 Wins

Postconditions: Player 1 score saved and Player 2 score is removed.
Exceptions:

1. If Player # did not make a valid move or others.

Priority: MEDIUM
When available: When the opponent has noPieces left.
Frequency of Use: Only once when one player has noPieces left
Channel to Actor: System
Secondary Actors: Player #?
Channel to secondary actors: No
Open Issues: 1

Checker Capture
Use Case: Checker Capture
Primary Actor: Player/System?
Goal in Context: When player wants to take a piece/capture
Preconditions:

1. Must be a valid move

Trigger: When the player wants to capture
Scenario:

1. Two Players are playing the game
2. Player 1's turn, makes a move
3. If Player 1 piece has adjacent (diagonal) of Player 2 pieces
4. If valid move, jump over Player 2 Piece and Capture, removing Player 2 Piece off board
5. The score value is updated for Player 1

Postconditions:

1. Remove from game board
2. And add score

Exceptions:

1. If Player 1, moving piece does not have adjacent Player 2 pieces, not a valid move

Priority: MEDIUM
When available: When Player # piece can capture
Frequency of Use: Every time a Player makes a Move to capture
Channel to Actor: Return/Reflect to Board
Secondary Actors: Player #?
Channel to secondary actors: No
Open Issues: 1

Chess Capture

Use Case: Chess Capture

Primary Actor: Player/System?

Goal in Context: When player wants to take a piece/capture

Preconditions:

1. Must be a valid move

Trigger: When the player wants to capture

Scenario:

1. Two Players are playing the game
2. Player 1's turn, makes a move
3. The move is a valid move, so the piece in the spot is captured
4. The captured piece is removed and
5. The score value of the piece is added to Player 1

Postconditions:

1. Remove from game board
2. And add score

Exceptions:

1. If Player # did not make a valid move
2. If Player # valid move, but does not have a piece in the destination

Priority: MEDIUM

When available: When Player # piece can capture

Frequency of Use: Every time a Player makes a Move to capture

Channel to Actor: Return/Reflect to Board

Secondary Actors: Player #?

Channel to secondary actors: No

Open Issues: 1

Promote a Piece
Use Case: Promote a piece
Iteration: 1
Primary Actor: Chess/Checker Player 1
Goal in context: A pawn has made it to the opposing end of the board and the player wants to promote it to a higher ranking piece.
Preconditions: A player's pawn or checker piece must reach the opposite end of the board.
Trigger: A player moves their pawn/checker piece to the opposite end.
Scenario:

1. A game of chess/checkers is ongoing and it is Player 1's turn to move.
2. A pawn/checker piece is one move away from reaching the opposite end of the board from Player 1.
3. Player 1 moves their piece to the end.
4. Player 1 must then promote the piece.

Postconditions: The promoted piece becomes a king in checkers, or any other piece except a king in chess.

Exceptions:

1. Promotion outside of the player's turn.

Priority: Medium, it is an element of chess.
When Available: After preconditions are fulfilled.
Frequency of Use: Rarely.
Channel to Actor: Game GUI
Secondary Actors: N/A
Channel to secondary Actors: N/A
Open Issues: N/A

Offer a Draw
Use Case: Offer a draw
Iteration: 1
Primary Actor: Chess Player 1
Goal in context: To offer for a draw to Chess Player 2.
Preconditions: N/A
Trigger: Draw Button.
Scenario:

1. Chess Player 1 either wants to call a draw for fun or believes that the game is completely even to the point that the game will never end.
2. Chess Player 1 calls for a draw.
3. Chess Player 2 can decide whether to accept or decline the draw.
4. If Chess Player 2 accepts, the game ends.
5. If Chess Player 2 declines, the game continues.

Postconditions: The game continues until one player runs out of time or the game ends with a draw.

Exceptions:

1. The draw button is disabled due to a bug.

Priority: Medium, Not always will a draw be offered.
When Available: Always.
Frequency of Use: As much as the player wants.
Channel to Actor: Game GUI
Secondary Actors: Chess Player 2
Channel to secondary Actors: Game GUI
Open Issues:

1. What if a player wants to spam the draw button for fun.