Chess Game

Use Cases Development

Use case number: 1

Use case name: Play chess

Use case description: Allows two players to play a chess game

Actors: Subscribers (interested on using the system for playing)

Preconditions:

- User subscribed and correctly logged in; game created
- System properly initialized
- The system has selected who is the player it will give turn in first place

Postconditions:

- Game ended and stored

Basic flow:

1. The system gives turn to one player
2. The system requests movement to the player
3. The system runs player's time
4. Player that has the turn proposes an ordinary correct movement of one of his pieces from one square of the board to another square of the board.
5. The system executes the movement
6. The system stops player's time
7. The system notifies movement to players
8. The system records movement in the trace
9. Repeat steps 2 to 7 while there is no winner
10. Notify winner to players
11. The system saves trace of game

Extensions:

2.a. The player cannot do any legal movement

1. The player is not in check

2. The system notifies the players there is a Stalemate

3. The game ends in a draw

4. Return to step 11 of the basic flow

2.b. In the chessboard there is one of the automatic draw situations (king vs king | king vs king + bishop | king vs king + knight | king + bishop vs king + bishop where bishops are in squares of the same colour.

1. The system notifies the players the game ends in a draw

2. Return to step 11 of the basic flow

2.c. There has been no capture or pawn move in the last fifty moves by each player (Fifty-move rule)

1. The last move has not been a checkmate

2. The system notifies the players the game ends in a draw

3. Return to step 11 of the basic flow

4.a. The player proposes an illegal movement (moving the piece makes the king be in check or the piece is not able to move to the proposed square)

1. The system notifies to user the error

2. Return to step 2 of the basic flow

4.b. The proposed movement is a castling (either kingside or queenside)

1. The king and the chosen rook are on the player's first rank.

2. Neither the king nor the chosen rook has previously moved.

3. There are no pieces between the king and the chosen rook.

4. The king is not currently in check.

5. The king does not pass through a square that is attacked by an enemy piece.

6. The king does not end up in check.

7. Return to step 5 of the basic flow

4.c. The proposed movement is a promotion

1. System asks the player for which piece the moved pawn wants to be replaced (Queen, knight, rook or bishop)

2. Player selects the desired piece

3. The system executes the movement

3. The pawn is replaced by the chosen piece

4. Return to step 5 of the basic flow

4.d. The proposed movement is an “En Passant”

1. The players pawn is standing on the 5th rank

2. The previous movement of the enemy has been moving a pawn from 2nd rank to 4th rank and the moved ended in a square either in the right or left of the selected pawn for the proposed movement

3. Return to step 5 of the basic flow

4.e. The player proposes draw

1. The system notifies the opponent the proposal of draw

2. The opponent accepts

3. The system notifies to the player the opponent has accept

4. The system notifies to both players the game ends in a draw

5. Return to step 11 of the basic flow

2.a. The opponent declines the proposal

1. Return to step 2 of the basic flow

7.a. The movement checks opponent’s king

1. System notifies movement to players

2. System notifies check to king

3. Return to step 8 of the basic flow

7.b The movement checkmate’s the opponent’s king

1. System notifies movement to players

2. System notifies checkmate

3. The player is declared as winner

4. Return to step 8 of the basic flow

*a. A player runs out of time

1. The opponent is declared as winner of the chess game

2. The system records in the trace that the player has ran out of time

2. Return to step 9 of the basic flow

Subfunction use case number: 1

Subfunction use case name: Proposed movement is correct

Subfunction use case description: Identifies if the proposed movement is correct

Preconditions:

- Player has introduced a movement into the system

Postconditions:

- Movement correctly analysed

Glossary:

y_c ≡ current vertical position of the piece

x_c ≡ current horizontal position of the piece

x_m ≡ number of squares to move the piece sideward

y_m ≡ number of squares to move the piece vertically

Basic flow:

1. The user proposes a move
2. System checks that the origin square contains a player's piece
3. The system splits the movement in vertical and horizontal displacement (ex. 1 square forward & 2 to the right)
4. Movement satisfies $x_c - 1 \leq |x_m| \leq 8 - x_c$, $y_c - 1 \leq |y_m| \leq 8 - y_c$, $|x_m| + |y_m| \neq 0$, and that the piece is not blocking check (Basis of correct movement)
5. The movement is appropriate for the piece
6. There is no piece in the trajectory path
7. The player's king is not in check
8. Return to step 5 of the basic flow

2.a. There is no piece in the source square

1. Go to Exception 4.a. of the basic flow

4.a. Movement does not satisfy $x_c - 1 \leq |x_m| \leq 8 - x_c$, $y_c - 1 \leq |y_m| \leq 8 - y_c$, $|x_m| + |y_m| \neq 0$, and that the piece is not blocking check

1. Go to Exception 4.a. of the basic flow

6.a. There is a piece in the trajectory path

1. The piece is not a knight

2. Go to Exception 4.a. of the basic flow

1.a. The piece is a knight

1. Go to Step 6 of this flow

7.a. The player's king is in check

1. The proposed movement gets the king out of check

1.a. The proposed movement does not get the king out of check

1. Go to Exception 4.a. of the basic flow

Subfunction use case number: 2

Subfunction use case name: The movement is appropriate for a pawn

Subfunction use case description: Identifies if a pawn can do the proposed movement

Preconditions:

- Movement satisfies the basis of a correct movement

Postconditions:

- Movement satisfies piece's requirements

Glossary:

y_c ≡ current vertical position of the piece

x_c ≡ current horizontal position of the piece

x_m ≡ number of squares to move the piece sideward

y_m ≡ number of squares to move the piece vertically

Basic flow:

1. Its position is the initial one

2. Movement satisfies $y_m = \{+1, +2\}$ and $x_m = 0$

3. There is not another piece in the destination square

    1.a. Position is not the initial one

        1. Movement satisfies $y_m = +1$ and $x_m = 0$

        2. There is not another piece in the destination square

            1.a. Movement satisfies $y_m = +1$ and $x_m = \{-1, +1\}$.

                1. There is another piece in the destination square

                2. The piece is not a king or a player's piece

                    1.a. There is not another piece in the destination square

                        1. The pawn is standing on the self-5th rank

                        2. The previous movement of the enemy has been moving a pawn from 2nd rank to 4th rank and the moved ended in a square either in the right or left of the selected pawn for the proposed movement.

1.a. The pawn is not standing on the self-5th rank

1. Go to Exception 4.a. of the basic flow

2.a. The previous movement of the enemy has not been moving a pawn from 2nd rank to 4th rank and the moved ended in a square either in the right or left of the selected pawn for the proposed movement.

1. Go to Exception 4.a. of the basic flow

2.a. The piece is a king or a player's piece

1. Go to Exception 4.a. of the basic flow

2.a. Movement satisfies $y_m = +1$ and $x_m = \{-1, +1\}$

1. There is another piece in the destination square

2. The piece is not a king or a player's piece

1.a. There is not another piece in the destination square

1. Go to Exception 4.a. of the basic flow

2.a. The piece is a king or a player's piece

1. Go to Exception 4.a. of the basic flow

3a. There is a piece in the destination square

1. Go to Exception 4.a. of the basic flow

Subfunction use case number: 3

Subfunction use case name: The movement is appropriate for a knight

Subfunction use case description: Identifies if a knight can do the proposed movement

Preconditions:

- Movement satisfies the basis of a correct movement

Postconditions:

- Movement satisfies piece's requirements

Glossary:

y_c ≡ current vertical position of the piece

x_c ≡ current horizontal position of the piece

x_m ≡ number of squares to move the piece sideward

y_m ≡ number of squares to move the piece vertically

Basic flow:

1. The movement satisfies y_m = {-2, -1, +1, +2} and x_m = {-1, +1} or y_m = {-1, +1} and x_m = {-2, -1, +1, +2}.
2. There is not a player's piece in the destination squares

    1.a. The movement does not satisfy y_m = {-2, -1, +1, +2} and x_m = {-1, +1} or y_m = {-1, +1} and x_m = {-2, -1, +1, +2}.

        1. Go to Exception 4.a. of the basic flow

    2.a. There is a king or a player's piece in the destination square

        1. Go to Exception 4.a. of the basic flow

Subfunction use case number: 4

Subfunction use case name: The movement is appropriate for a bishop

Subfunction use case description: Identifies if a bishop can do the proposed movement

Preconditions:

- Movement satisfies the basis of a correct movement

Postconditions:

- Movement satisfies piece's requirements

Glossary:

y_c ≡ current vertical position of the piece

x_c ≡ current horizontal position of the piece

x_m ≡ number of squares to move the piece sideward

y_m ≡ number of squares to move the piece vertically

Basic flow:

4.b. The piece is a bishop

1. The movement satisfies $|x_m| = |y_m|$
2. There is not a king or a player's piece in the destination square

1.a. The movement does not satisfy $|x_m| = |y_m|$

1. Go to Exception 4.a. of the basic flow

2.a. There is a king or a player's piece in the destination square

1. Go to Exception 4.a. of the basic flow

Subfunction use case number: 5

Subfunction use case name: The movement is appropriate for a rook

Subfunction use case description: Identifies if a rook can do the proposed movement

Preconditions:

- Movement satisfies the basis of a correct movement

Postconditions:

- Movement satisfies piece's requirements

Glossary:

y_c ≡ current vertical position of the piece

x_c ≡ current horizontal position of the piece

x_m ≡ number of squares to move the piece sideward

y_m ≡ number of squares to move the piece vertically

Basic flow:

4.c. The piece is a rook

1. The movements satisfies $|x_m| \neq 0$ and $|y_m| = 0$ or $|x_m| = 0$ and $|y_m| \neq 0$
2. There is not a king or a player's piece in the destination square

1.a. The movement does not satisfy $|x_m| \neq 0$ and $|y_m| = 0$ or $|x_m| = 0$ and $|y_m| \neq 0$

1. Go to Exception 4.a. of the basic flow

2.a. There is a king or a player's piece in the destination square

1. Go to Exception 4.a. of the basic flow

Subfunction use case number: 6

Subfunction use case name: The movement is appropriate for a queen

Subfunction use case description: Identifies if a queen can do the proposed movement

Preconditions:

- Movement satisfies the basis of a correct movement

Postconditions:

- Movement satisfies piece's requirements

Glossary:

y_c ≡ current vertical position of the piece

x_c ≡ current horizontal position of the piece

x_m ≡ number of squares to move the piece sideward

y_m ≡ number of squares to move the piece vertically

Basic flow:

4.d. The piece is a queen

1. The movement satisfies $|x_m| = |y_m|$, $|x_m| \neq 0$ and $|y_m| = 0$ or $|x_m| = 0$ and $|y_m| \neq 0$

2. There is not a king or a player's piece in the destination square

1.a. The movement does not satisfy $|x_m| = |y_m|$, $|x_m| \neq 0$ and $|y_m| = 0$ or $|x_m| = 0$ and $|y_m| \neq 0$

1. Go to Exception 4.a. of the basic flow

2.a. There is a king or a player's piece in the destination square

1. Go to Exception 4.a. of the basic flow

Subfunction use case number: 7

Subfunction use case name: The movement is appropriate for a king

Subfunction use case description: Identifies if a king can do the proposed movement

Preconditions:

- Movement satisfies the basis of a correct movement

Postconditions:

- Movement satisfies piece's requirements

Glossary:

y_c ≡ current vertical position of the piece

x_c ≡ current horizontal position of the piece

x_m ≡ number of squares to move the piece sideward

y_m ≡ number of squares to move the piece vertically

Basic flow:

1. The piece is in the initial position
2. The movement satisfies x_m = {-1, 0, +1} and y_m = {-1, 0, +1}
3. There is not a king or a player's piece in the destination square

1.a. The piece is not in the initial position

1. The movement satisfies x_m = {-1, 0, +1} and y_m = {-1, 0, +1}
2. There is not a king or a player's piece in the destination square

1.a. The movement does not satisfy x_m = {-1, 0, +1} and y_m = {-1, 0, +1}

1. Go to Exception 4.a. of the basic flow

2.a. There is a king or a player's piece in the destination square

1. Go to the Exception 4.a. of the basic flow

2.a. The movement does not satisfy x_m = {-1, 0, +1} and y_m = {-1, 0, +1}

1. The movement is x_m = 2 and y_m = 0
2. The kingside rook is in the initial position

3. The two squares to de right of the piece are empty

4. Move the king 2 squares to the right

5. Move the kingside tower 2 squares to the left

1.a. The movement is $x_m = -3$ and $y_m = 0$

1. The queenside rook is in the initial position

2. The three squares to the left of the piece are empty

3. Move the king 3 squares to the left

4. Move the queenside tower 3 squares to the right

1.a. The queenside rook is not in the initial position

1. Go to Exception 4.a. of the basic flow

2.a. The three squares to the left of the piece are not empty

1. Go to Exception 4.a. of the basic flow

2.a. The kingside rook is not in the initial position

1. Go to Exception 4.a. of the basic flow

3.a. The two squares to de right of the piece are not empty

1. Go to Exception 4.a. of the basic flow

3.a. There is a king or a player's piece in the destination square

1. Go to Exception 4.a. of the basic flow

Use case number: 2

Use case name: Create New Game

Use case description: Allows a player to start a new chess game

Actors: Subscribers (interested on using the system for playing)

Preconditions:

- User subscribed and correctly logged in
- System properly initialized

Postconditions:

- New game correctly created

Basic flow:

1. A user selects the option of creating a new game in the main menu
2. The System generates the 8x8 chess board
3. The system generates the chess pieces of both players and assigns the correspondent initial position to each piece.
4. The system searches another player for playing
5. The system randomly selects which subscriber will take white pieces and which the black ones
6. The system welcomes both players

Extensions:

1.a. The user selects another option in the menu

1. The system ends the use case

*a. The player selects the option of returning to the main menu

1. Return to step 1 of the basic flow