Template

Use Case:
Iteration: 1
Primary Actor:
Goal in context:
Preconditions:
Trigger:
Scenario:

1.

Postconditions:
Exceptions:

1.

Priority:
When Available:
Frequency of Use:
Channel to Actor:
Secondary Actors:
Channel to secondary Actors:
Open Issues:

1.

Checkmate

Use Case: Checkmate in Chess
Iteration: 1
Primary Actor: Chess Player 1
Goal in context: To deliver a checkmate and win the game by placing the opposing Chess Player 2 in a position where it cannot escape the attack.
Preconditions: A game of chess is ongoing and it is Chess Player 1's turn to move.
Trigger: The player's move places the opponent's king in a position where it is in check and no legal move exists for it to escape check,
Scenario:

1. A game of chess is being played between two players
2. As the game goes by, Chess Player 1 recognizes that their opponent's king is in a position for checkmate.
3. A player moves their pieces to attack the opponent's king.
4. The king is unable to move or block the attack and no longer has any possible moves outside of accepting a capture in the next move.
5. A "checkmate" is declared by Chess Player 1.
6. The game ends with Chess Player 1 winning.

Postconditions: The opponent's king is checkmated and the player wins the game. The victory is recorded along with the captures of pieces made in the session.
Exceptions:

1. The opponent's king can escape.
2. The opponent can use another piece to block or capture the attack piece.
3. The opponent made an illegal move due to a coding error.

Priority: High, it is an end goal in the game and one way to ensure that the match concludes.
When Available: As soon as the opponent's king is in a position where no legal moves are possible to avoid check.
Frequency of Use: At most **Once** each game, but can not occur at all if draw is called.
Channel to Actor: Game GUI
Secondary Actors: Chess Player 2 (Opponent/Player being checkmated).
Channel to secondary Actors: Game GUI
Open Issues: N/A

Castling

Use Case: Castling in Chess
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. The king and the chosen rook have not made a move.
3. The king is not in check, will pass through a check, or lands on a check during castling.

Trigger: The player decides to perform the 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. The player is able to perform the move if all preconditions have been met.
4. Castling is performed and the game continues.

Postconditions: Castling has been successfully performed and the king and rook have castling, their positions updating.
Exceptions:

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

Priority: Medium, it is an element of chess but not game-ending like checkmate.
When Available: As soon as the opponent's king is in a position where no legal moves are possible to avoid check.
Frequency of Use: At most **Once** each game, but can not occur at all if draw is called.
Channel to Actor: Game GUI
Secondary Actors: Chess Player 2 (Opponent/Player being checkmated).
Channel to secondary Actors: Game GUI
Open Issues: N/A

En Passant

Use Case:

Iteration: 1

Primary Actor:

Goal in context:

Preconditions:

Trigger:

Scenario:

1.

Postconditions:

Exceptions:

1.

Priority:

When Available:

Frequency of Use:

Channel to Actor:

Secondary Actors:

Channel to secondary Actors:

Open Issues: