|  |  |  |
| --- | --- | --- |
| **NAME** | **Start Game** | |
| **USE CASE NO.** | **1** | |
| **DESCRIPTION** | Initializes the SDL chess game when the player starts a new session. | |
| **ACTORS** | Player | |
| **PRECONDITIONS** | Game must be launched successfully. | |
| **POST CONDITIONS** | Board and pieces are displayed, and turn system is initialized. | |
| **TRIGGER** | Player clicks “Play 1v1” | |
| **DESCRIPTION** | **STEP** | **ACTION** |
| 1 | Player opens the application |
| 2 | Player clicks "Play 1v1" |
| 3 | SDL window initializes |
| 4 | Board is rendered, pieces are placed |
|  |  |
|  |  |
| **STEP EXTENSTIONS** | **STEP** | **BRANCHES** |
| 3 | If SDL fails, display error and exit |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **ALTERNATE COURSES** | **STEP** | **ACTION** |
| 2 | If no input, game stays on intro screen |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **NAME** | **Move a Piece** | |
| **USE CASE NO.** | **2** | |
| **DESCRIPTION** | Handles movement of a piece from its source to a valid destination. | |
| **ACTORS** | Player | |
| **PRECONDITIONS** | The game is active and it is the player's turn. | |
| **POST CONDITIONS** | The piece is moved, the board updates, and the turn switches. | |
| **TRIGGER** | Player clicks a valid piece and using the mouse moves it to a valid square | |
| **DESCRIPTION** | **STEP** | **ACTION** |
| 1 | Player selects a piece. |
| 2 | System calculates valid moves for the selected piece. |
| 3 | Player moves mouse piece to a valid square |
| 4 | Player clicks on the square and the piece is placed |
|  |  |
|  |  |
| **STEP EXTENSTIONS** | **STEP** | **BRANCHES** |
| 4 | If destination is invalid, input is ignored |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **ALTERNATE COURSES** | **STEP** | **ACTION** |
| 2 | If it is not the player's turn, the move is blocked. |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **NAME** | **Pawn Promotion** | |
| **USE CASE NO.** | **3** | |
| **DESCRIPTION** | Promotes pawn to queen when it reaches the final rank, if no queen is already on board. | |
| **ACTORS** | Player | |
| **PRECONDITIONS** | Pawn reaches 8th rank (white) or 1st rank (black) | |
| **POST CONDITIONS** | Pawn becomes a queen | |
| **TRIGGER** | Player moves pawn to last rank | |
| **DESCRIPTION** | **STEP** | **ACTION** |
| 1 | Pawn reaches the last row |
| 2 | System checks for existing queen |
| 3 | If none exists, pawn is promoted to queen |
| 4 | Board is updated |
|  |  |
|  |  |
| **STEP EXTENSTIONS** | **STEP** | **BRANCHES** |
| 2 | If queen already exists, promotion denied |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **ALTERNATE COURSES** | **STEP** | **ACTION** |
| 3 | No other promotions supported yet |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **NAME** | **Illegal Move Handling** | |
| **USE CASE NO.** | **4** | |
| **DESCRIPTION** | Prevents the player from performing moves that violate chess rules. | |
| **ACTORS** | Player | |
| **PRECONDITIONS** | Game is running, move attempted | |
| **POST CONDITIONS** | Move is blocked | |
| **TRIGGER** | Player clicks invalid destination | |
| **DESCRIPTION** | **STEP** | **ACTION** |
| 1 | Player selects piece |
| 2 | Player moves piece to and clicks invalid square |
| 3 | System checks legality |
| 4 | Move is rejected |
|  |  |
|  |  |
| **STEP EXTENSTIONS** | **STEP** | **BRANCHES** |
| 3 | Rule violation results in rejection |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **ALTERNATE COURSES** | **STEP** | **ACTION** |
| 2 | Player retires a legal move |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **NAME** | **Turn-Based System** | |
| **USE CASE NO.** | **5** | |
| **DESCRIPTION** | Ensures turn alternation between two players. | |
| **ACTORS** | Player | |
| **PRECONDITIONS** | One valid move is completed | |
| **POST CONDITIONS** | Turn changes to other player | |
| **TRIGGER** | Completion of a move | |
| **DESCRIPTION** | **STEP** | **ACTION** |
| 1 | Valid move is completed |
| 2 | Turn switches to other player |
| 3 | Current player disabled |
|  |  |
|  |  |
|  |  |
| **STEP EXTENSTIONS** | **STEP** | **BRANCHES** |
| 3 | Wrong-player input is ignored |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **ALTERNATE COURSES** | **STEP** | **ACTION** |
| 1 | Invalid move keeps same player’s turn |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **NAME** | **Click-Based Piece Selection** | |
| **USE CASE NO.** | **6** | |
| **DESCRIPTION** | Allows a player to click and select one of their own pieces | |
| **ACTORS** | Player | |
| **PRECONDITIONS** | It is the player’s turn | |
| **POST CONDITIONS** | The selected piece is ready to move | |
| **TRIGGER** | Player clicks a piece | |
| **DESCRIPTION** | **STEP** | **ACTION** |
| 1 | Player clicks on piece |
| 2 | System verifies turn and piece ownership |
| 3 | Piece is selected |
|  |  |
|  |  |
|  |  |
| **STEP EXTENSTIONS** | **STEP** | **BRANCHES** |
| 2 | Opponent’s piece is ignored |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **ALTERNATE COURSES** | **STEP** | **ACTION** |
| 1 | Clicking an empty square does nothing |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **NAME** | **Capture a Piece** | |
| **USE CASE NO.** | **7** | |
| **DESCRIPTION** | Replaces an opponent’s piece with the player’s own piece when a legal capturing move is made. | |
| **ACTORS** | Player | |
| **PRECONDITIONS** | Player selects their piece and clicks on an opponent’s piece in a legal capture move. | |
| **POST CONDITIONS** | Opponent’s piece is removed from the board, and the current piece moves into its square. | |
| **TRIGGER** | Player clicks on a legal destination containing an opponent’s piece. | |
| **DESCRIPTION** | **STEP** | **ACTION** |
| 1 | Player selects a piece |
| 2 | Player moves piece and clicks on a square with an opponent’s piece |
| 3 | System checks legality of capture |
| 4 | If legal, opponent’s piece is removed |
| 5 | Player’s piece moves to that square |
|  |  |
| **STEP EXTENSTIONS** | **STEP** | **BRANCHES** |
| 3 | If the move is not legal or the piece is not an opponent's, input is ignored |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **ALTERNATE COURSES** | **STEP** | **ACTION** |
| 2 | Player moves piece and clicks an empty square — process as a standard move |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **NAME** | **Invalid Turn Attempt** | |
| **USE CASE NO.** | **8** | |
| **DESCRIPTION** | Prevents the wrong player from making a move when it is not their turn. | |
| **ACTORS** | Player | |
| **PRECONDITIONS** | It is the opponent's turn | |
| **POST CONDITIONS** | No move is executed | |
| **TRIGGER** | Player clicks during opponent's turn | |
| **DESCRIPTION** | **STEP** | **ACTION** |
| 1 | Player clicks on their piece |
| 2 | System checks active player |
| 3 | If mismatch, action is ignored |
|  |  |
|  |  |
|  |  |
| **STEP EXTENSTIONS** | **STEP** | **BRANCHES** |
| 2 | Repeat input attempts are ignored |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **ALTERNATE COURSES** | **STEP** | **ACTION** |
| 1 | Game remains idle until correct player acts |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **NAME** | **Intro Screen** | |
| **USE CASE NO.** | **9** | |
| **DESCRIPTION** | Displays a static intro screen when the game is launched. | |
| **ACTORS** | System | |
| **PRECONDITIONS** | Application starts | |
| **POST CONDITIONS** | Waits for player to begin | |
| **TRIGGER** | Game is launched | |
| **DESCRIPTION** | **STEP** | **ACTION** |
| 1 | SDL loads into the background |
| 2 | Display is shown |
| 3 | System waits for click or input |
|  |  |
|  |  |
|  |  |
| **STEP EXTENSTIONS** | **STEP** | **BRANCHES** |
| 3 | On input either exits or moves to main game |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **ALTERNATE COURSES** | **STEP** | **ACTION** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **NAME** | **Visual Board Rendering** | |
| **USE CASE NO.** | **10** | |
| **DESCRIPTION** | Redraws the board and pieces after every valid move or event. | |
| **ACTORS** | System | |
| **PRECONDITIONS** | Game state has changed | |
| **POST CONDITIONS** | Visuals are updated | |
| **TRIGGER** | Valid move or state change | |
| **DESCRIPTION** | **STEP** | **ACTION** |
| 1 | Game processes the move |
| 2 | SDL clears and redraws board |
| 3 | Piece images are rendered in new positions |
|  |  |
|  |  |
|  |  |
| **STEP EXTENSTIONS** | **STEP** | **BRANCHES** |
| 3 | If image is missing, error handler runs |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **ALTERNATE COURSES** | **STEP** | **ACTION** |
| 2 | No redraw on invalid move |
|  |  |
|  |  |
|  |  |
|  |  |