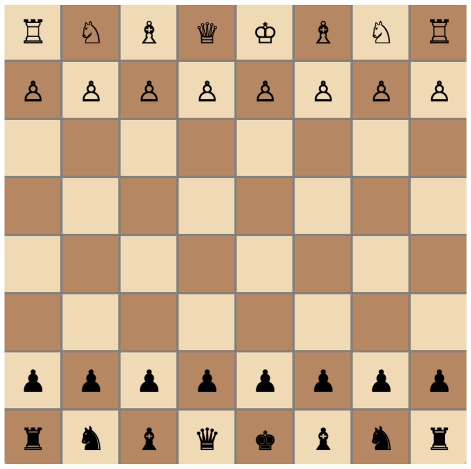
Chess Game GUI Design: Project Instruction



Objective:

In Phase 2 of your Chess Game Project, you will develop a Graphical User Interface (GUI) that displays the chessboard and allows two players to move pieces on the board. You will be focusing on the visual aspects and basic functionality of the board in this phase, without needing to worry about implementing the full game rules.

Requirements:

1. **Chessboard GUI**:
   - Create a GUI that displays an 8x8 chessboard. You should use **Swing** Java GUI framework.
   - The chessboard should be displayed with alternating light and dark squares.
2. **Chess Pieces**:
   - Display the chess pieces for both players (White and Black) on their respective initial positions on the board.
   - Each piece should be visually distinct, representing the appropriate chess pieces (Pawn, Rook, Knight, Bishop, Queen, King).
3. **Piece Movement**:
   - Allow the user to interact with the chess pieces using a **mouse**:
     - **Click and move**: A piece should be able to move to a selected destination square by clicking on it and then clicking the destination square.
     - **Drag and drop**: Alternatively, pieces can be moved by dragging a piece and dropping it onto the destination square.
   - Movement validation is **not** required at this stage—each piece can move to any square.

4. **Capturing Pieces**:
    - If a piece is moved to a square that is already occupied by the opponent's piece, the opponent's piece should be **captured** and disappear from the board.
    - The captured piece will no longer be displayed on the board.
5. **Endgame Notification**:
    - If a player captures the opponent's **King**, a **pop-up window** should declare that player as the winner.
    - After declaring the winner, the game should terminate.
6. **Extra GUI Features:**
    - Select at least two extra features listed below and implement them in your project.
    - If you implemented more than two, each extra one would receive 5 extra points.

Submission Guidelines:

1. **Code Submission**:
    - Implement the above requirements in your project.
    - Include clear Javadoc comments for your classes and methods.
    - Push your code to your GitHub repository with meaningful commit messages.
2. **Document Submission**:
    - Prepare a brief document (PDF) explaining:
        - A screenshot of the GUI at the start of the game.
        - A screenshot of moving several pieces, and some pieces got captured.
        - A screenshot of capturing the King, and a window pops up to declare the winner.
        - If you implemented other extra features, also add corresponding screenshots and explanations to demonstrate your design.
        - Any questions, comments, or necessary explanations of your submission.
3. **File Submission**:
    - Submit the following files:
        - Your source code GitHub repository link. (You can continue use the same repository as phase 1, put your phase 2 code in a separate folder or a separate branch)
        - A PDF document has required screenshots of your working GUI and explanations.

Extra GUI Features

Each team should implement at least two extra GUI features to get full credits. If you implemented more than two of them, each additional one will receive 5 extra points.

GUI Feature 1: Menu Bar with Game Controls

Design and implement a menu bar at the top of the chess game window with the following menu items:

- **New Game**: Starts a new game, resetting the board and all game data.
- **Save Game**: Allows the player to save the current state of the game (you can choose a format like serialization or a text file).
- **Load Game**: Loads a previously saved game, restoring the board and player positions from the saved file.

Requirements:

- Ensure that the "Save Game" and "Load Game" functionalities work correctly, so players can resume a saved game from the exact state they left off.
- The "New Game" option should reset the board to its initial setup.

GUI Feature 2: Settings Window for Customization

Design a **Settings Window** that allows players to customize the chessboard and piece appearance. The settings should include:

- **Board Background Color/Style**: Allow users to select different colors or styles for the board (e.g., classic wooden, modern gray).
- **Chess Piece Color/Style**: Allow users to select different designs or colors for the chess pieces.
- **Board Size**: Enable the user to adjust the size of the chessboard (e.g., small, medium, large).

Requirements:

- Implement an "Apply" button that updates the chessboard in real-time according to the user's selections.
- Ensure that the board and pieces change their appearance dynamically after the settings are applied.

- The settings window should be accessible through the menu bar or another clear interface element.

GUI Feature 3: Game History Panel with Undo Button

Add a **separate panel** that tracks the history of moves and captures made during the game. This panel should include:

- **Move History**: Display the sequence of moves, indicating the piece, starting position, and ending position.
- **Captured Pieces**: Show which pieces have been captured by each player, displaying the captured pieces.
- **Undo Button**: Include an **Undo** button that allows players to revert to the previous game state.

Requirements:

- The undo button should accurately revert the game to the previous move, including restoring any captured pieces.
- The move history should be updated in real-time as players make moves.

GUI Feature x: Propose your own

Propose your cool GUI feature design ideas and let the instructor know before the submission.

Phase 2 Grading Criteria:

- GUI functionality: 30% (static game board display)
- Piece movement and capture logic: 30% (run time events handling)
- Endgame notification and winner declaration: 10% (run time events handling)
- Extra features implementation: 20% (minimum of two extra features implemented)
- Code organization, documentation, and submission quality: 10%