Checkmate! With GUI

In this in-class activity, we will create a simple **chess game** with GUI.

There are 3 classes (ChessBoardUI, ButtonListener, Main).

[Rule of Chess]

1. A chess board is a board of fixed size, **8x8**, where each square initially contains an whitespace (“ ”).
2. Chess pieces include King, Queen, Rook, Bishop, Knight, and Pawn, totaling six types.

   For this activity, we attempt to implement only the Pawn class. (You may create other piece classes if you want!)

[Code structure and implementation details] – DO NOT MODIFY Main.java

1. ChessBoardUI

- Constructor for ChessBoardUI
    - Set the title of the window to "Chess Game"
    - Ensure the application exits when the window is closed
    - Set the layout of the window to an 8x8 grid, suitable for a chessboard
    - Set the size of the window to 600x600 pixels
    - Make the window visible
    - **Hint**: use setTitle, setDefaultCloseOperation, setLayout, setSize, setVisible
    -
- initializeBoard() method - Method to initialize the chess board with pieces
    - Initialize the board with empty spaces (" ")
    - Place pawns on the board (Assign the string to the board in the same way as in activity 5)
    - Place other pieces on the board
        - Please refer to the execution result

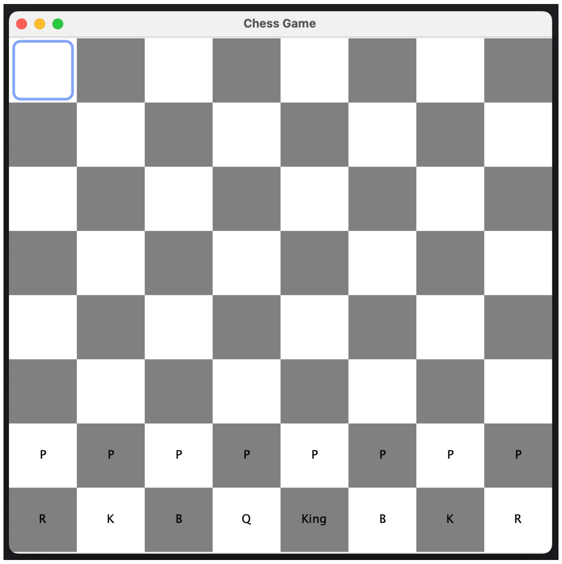
- createUI() method – Method to create the chessboard UI
    - Create the chess board UI
        - Set the color of the squares
        - Add ActionListener to each button

- updateUI() method - Method to update the UI based on the current state of the board
    - set text (board) on the board (squares)
    - **Hint**: use setText

2. ButtonListener

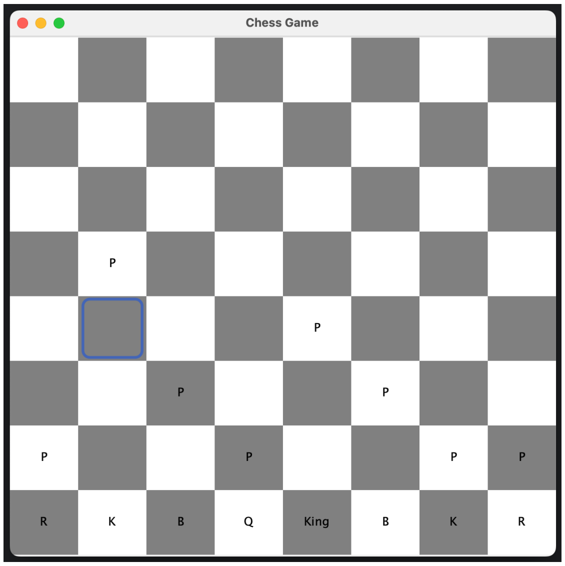
- ButtonListener should implement ActionListener.
- Constructor for ButtonListener
    - Initializes the **x** and **y** values
- actionPerformed method - Called when the button is clicked
    - **Piece movement logic**
        - Implement the logic to move a piece (only pawn "P")
        - In this case, if the pawn ("P") is clicked, the pawn piece will be moved
            - Checks if the pawn can move one square forward, and if possible, moves it.
    - **Update UI**: After moving the piece, updates the chessboard UI. (use updateUI method of ChessBoardUI class)

[Output Result of Main.java]



P	P	P	P	P	P	P	P
R	K	B	Q	King	B	K	R

< initialized ChessBoardUI >



	P						
				P			
		P			P		
P			P			P	P
R	K	B	Q	King	B	K	R

< After moving pawn pieces>

[While doing the activity, refer to and utilize the followings:]

- **Practice reading the java documentation.**
- **importing external libraries and modules**
- **inheritance**
- **implementation**
- **static**