Chess Project

By Son Phi & Mohammed Elmi

Introductions:

Our project is a simulation of a chess game. Chess is a fun game and well known by everyone. It will help people to learn to how to play chess and practice as well. It also help us understand and put the concept of object oriented program in use.

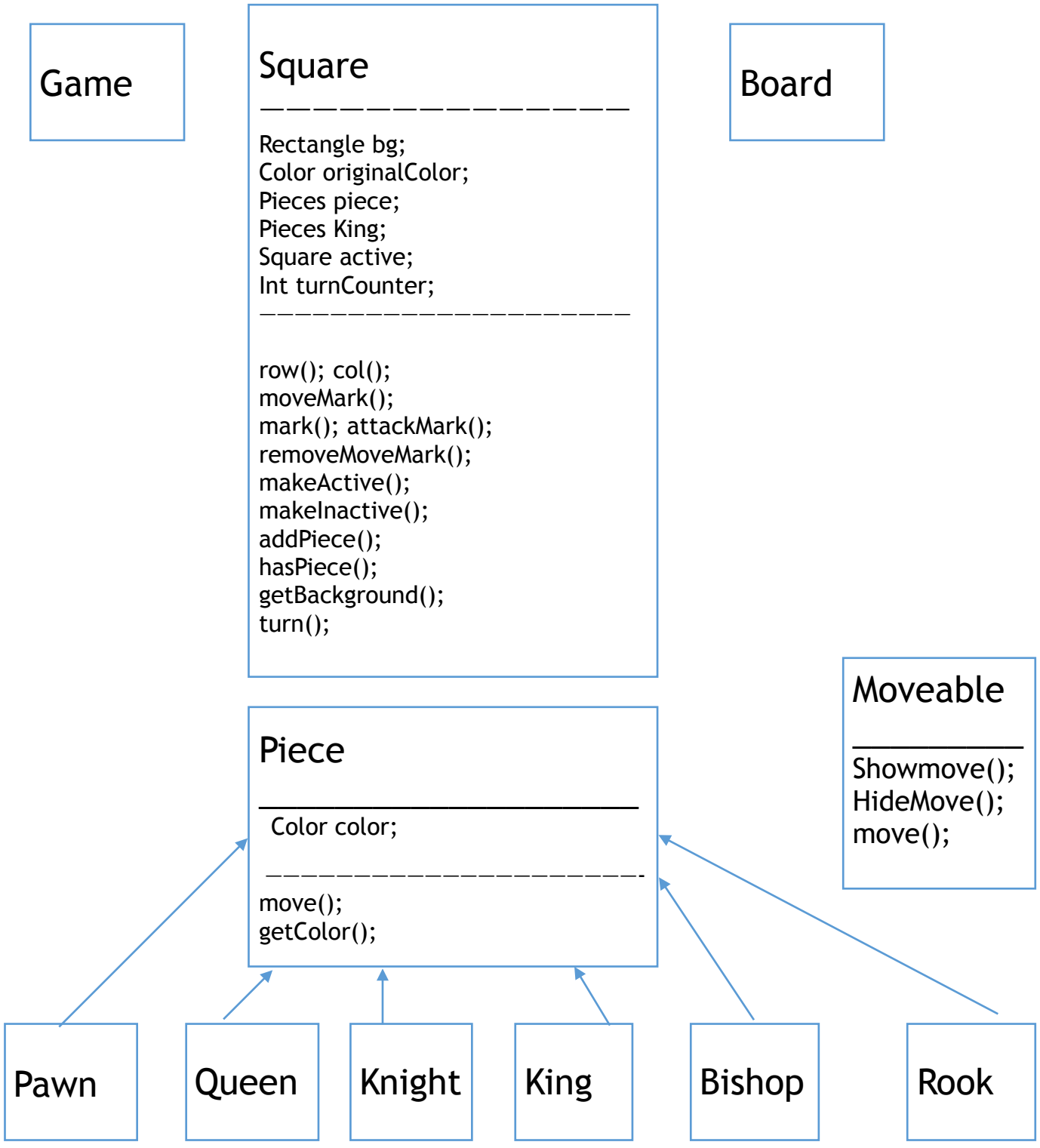
Mohammed in charge of creating the board and the GUI, Start screen and end screen

Son in charge of creating the logic and creating objects.

Features:

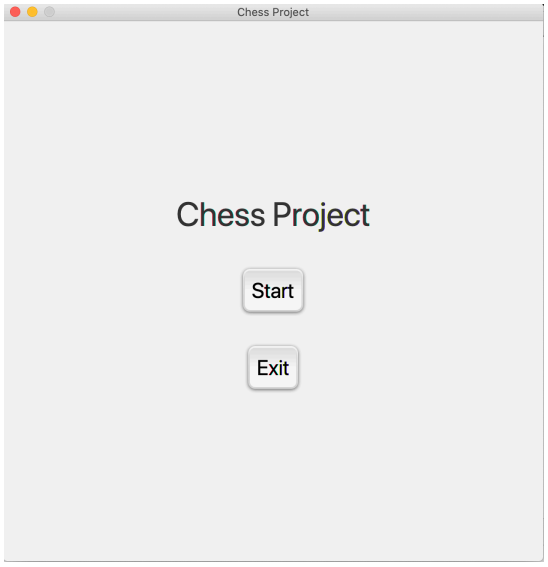
This also show the possible move the piece can make.

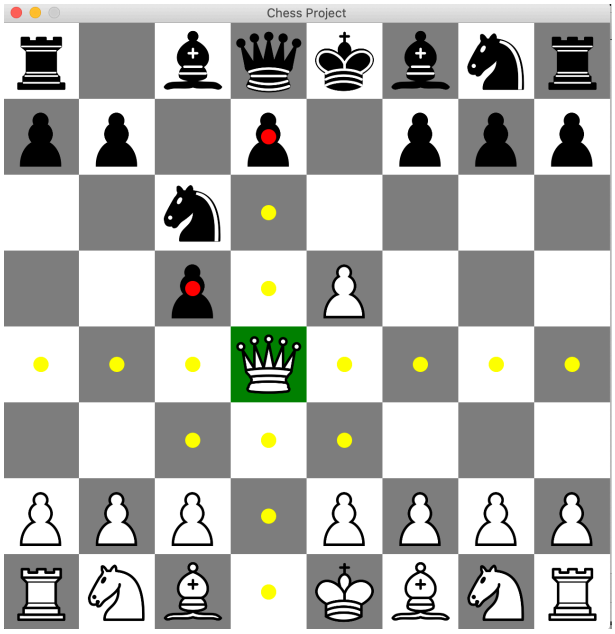
The Pawn can move forward to the unoccupied square immediately in front of it on the same line, or on its first move it can advance two squares along the same line. the pawn can only capture an opponent's piece on a square diagonally in front of it.
A bishop can move any number of squares diagonally, but cannot leap over other pieces.
The queen combines the power of a rook and bishop and can move any number of squares along a rank, file, or diagonal, but cannot leap over other pieces
A knight moves like "L"-shape: two squares vertically and one square horizontally, or two squares horizontally and one square vertically.) The knight is the only piece that can leap over other pieces.
A rook can move any number of squares along in a straight line, but cannot leap over other pieces.
The king moves one square in any direction. You win when you capture the other team’s King.

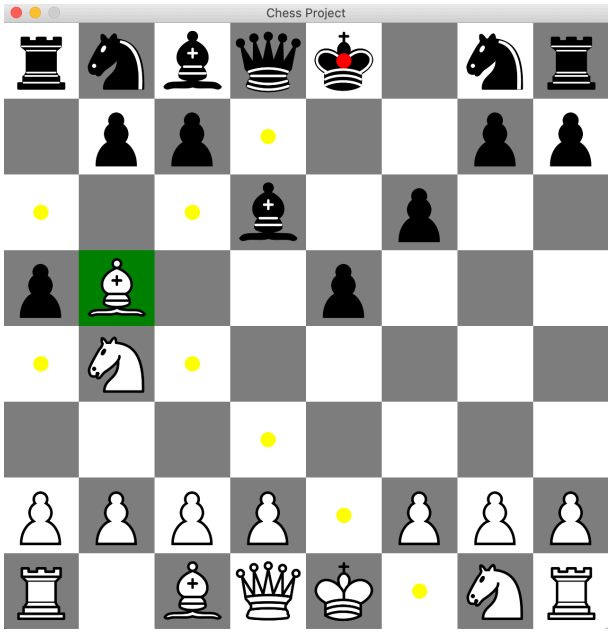

Game
Square
Rectangle bg;
Color originalColor;
Pieces piece;
Pieces King;
Square active;
Int turnCounter;
row(); col();
moveMark();
mark(); attackMark();
removeMoveMark();
makeActive();
makeInactive();
addPiece();
hasPiece();
getBackground();
turn();
Board
Moveable
Showmove();
HideMove();
move();
Piece
Color color;
move();
getColor();
Pawn
Queen
Knight
King
Bishop
Rook

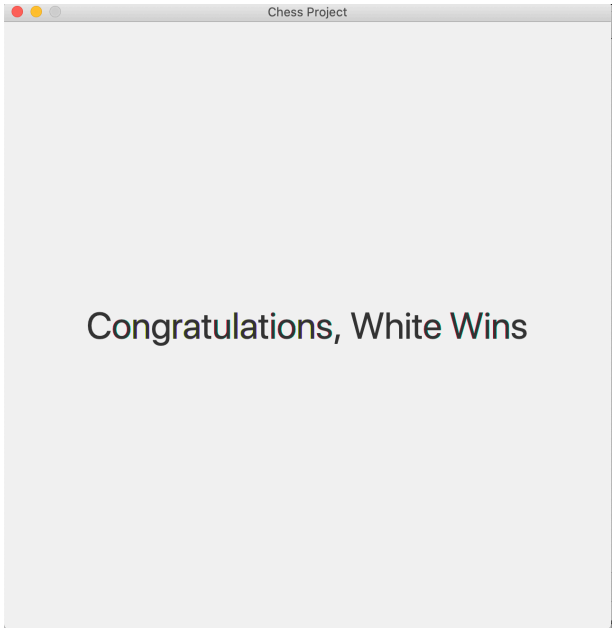
Design:

Screenshot:

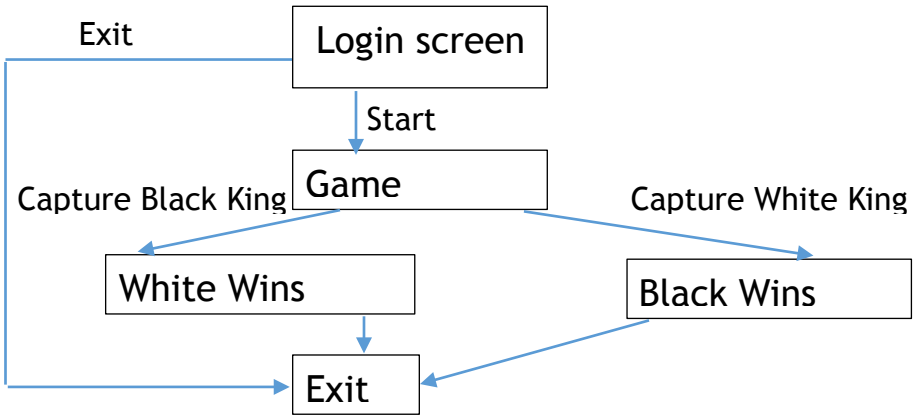








Flow Chart:



Source code:

//————————————————————Board————————————————————

```
package board;
import java.util.ArrayList;
import javafx.scene.Group;
import javafx.scene.paint.Color;
import pieces.Bishop;
import pieces.King;
import pieces.Knight;
import pieces.Pawn;
import pieces.Queen;
import pieces.Rook;
```

```
public class ChessBoard extends Group {

	public static ArrayList<ArrayList<Square>> all_squares = new
ArrayList<ArrayList<Square>>();

	public ChessBoard() {

		for (int row = 0; row < 8; row++) {
			ArrayList<Square> arrRow = new ArrayList<Square>();
			for (int col = 0; col < 8; col++) {

				Color bg = Color.GRAY;
				if ((row + col) % 2 == 0) {
					bg = Color.WHITE;
				}

				Square s = new Square(bg);
				s.setTranslateX(col * Square.SIZE);
				s.setTranslateY(row * Square.SIZE);
				this.getChildren().add(s);
				arrRow.add(s);

				// PAWNS
				if (row == 1) {
					s.addPiece(new Pawn(Color.BLACK));
				}
```

```
if (row == 6) {
        s.addPiece(new Pawn(Color.WHITE));
}

// ROOKS
if (row == 0) {
        if (col == 0 || col == 7) {
                s.addPiece(new Rook(Color.BLACK));
        }
}

if (row == 7) {
        if (col == 0 || col == 7) {
                s.addPiece(new Rook(Color.WHITE));
        }
}

// KNIGHTS
if (row == 0) {
        if (col == 1 || col == 6) {
                s.addPiece(new Knight(Color.BLACK));
        }
}

if (row == 7) {
```

```
        if (col == 1 || col == 6) {
                s.addPiece(new Knight(Color.WHITE));
        }
}

// BISHOPS
if (row == 0) {
        if (col == 2 || col == 5) {
                s.addPiece(new Bishop(Color.BLACK));
        }
}

if (row == 7) {
        if (col == 2 || col == 5) {
                s.addPiece(new Bishop(Color.WHITE));
        }
}

// QUEENS
if (row == 0) {
        if (col == 3) {
                s.addPiece(new Queen(Color.BLACK));
        }
}

if (row == 7) {
```

```
                        if (col == 3) {
                            s.addPiece(new Queen(Color.WHITE));
                        }
                    }

                    // KINGS
                    if (row == 0) {
                        if (col == 4) {
                            s.addPiece(new King(Color.BLACK));
                        }
                    }

                    if (row == 7) {
                        if (col == 4) {
                            s.addPiece(new King(Color.WHITE));
                        }
                    }
                }
                all_squares.add(arrRow);
            }
        }

        public static Square getSquare(int x, int y) {
            return all_squares.get(y).get(x);
        }
    }
```

```
//-----------------------------------------Square--------------------------------
package board;
import java.util.ArrayList;
import game.game;
import javafx.scene.Group;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Rectangle;
import pieces.King;
import pieces.Piece;

public class Square extends Group {

	public static final double SIZE = 80;
	private Rectangle bg;
	private Color originalColor;
	private Piece piece;
	public static Square active;
	private static int turnCounter = 0;
	public static ArrayList<Square> marked = new ArrayList<Square>();

	public Square(Color c) {
		originalColor = c;
		bg = new Rectangle(SIZE, SIZE, c);
		this.getChildren().add(bg);
```

```
            this.setOnMouseClicked(event -> {

                if (marked.contains(this)) {
                    Piece p = active.piece;
                    if (this.piece instanceof King) {
                        if (this.piece.getColor() == Color.WHITE) {
                            game.blackwin();


                        }
                        if (this.piece.getColor() == Color.BLACK) {
                            game.whitewin();
                            System.out.println("Congratulations!
You won in "+ Square.turnCounter + " turns!");
                        }
                        return;
                    }
                    active.piece = null;
                    active.makeInactive();
                    turnCounter++;
                    System.out.println("Turn #"+ turnCounter);
                    this.addPiece(p);
                    p.move();
                    return;
                }

                if (!hasPiece()) {
```

```
                return;
            }

            if (hasPiece()) {
                if (turnCounter % 2 == 0 && piece.getColor() ==
Color.BLACK) { // White
                    return;
                }
                if (turnCounter % 2 == 1 && piece.getColor() ==
Color.WHITE) { // Black
                    return;
                }
                makeActive();
                @SuppressWarnings("unused")
                int row = row();
                @SuppressWarnings("unused")
                int col = col();
            } else {
                if (active != null) {
                    active.makeInactive();
                }
                mark(this.piece.getColor());
            }
        });
    }
```

```
    private int col() {
        int y = row();
        for (int i = 0; i < 8; i++) {
            if (ChessBoard.all_squares.get(y).get(i) == this) {
                return i;
            }
        }
        return -1;
    }

    public int row() {
        for (int i = 0; i < 8; i++) {
            if (ChessBoard.all_squares.get(i).contains(this)) {
                return i;
            }
        }
        return -1;
    }

    public void moveMark() {
        Circle cir = new Circle(SIZE / 2, SIZE / 2, SIZE / 10,
Color.YELLOW);
        this.getChildren().add(cir);
        marked.add(this);
    }
```

```
	public void mark(Color c) {
		if (!this.hasPiece()) {
			moveMark();
		} else {
			if (this.piece.getColor() != c) {
				attackMark();
			}
		}
	}

	public void attackMark() {
		Circle cir = new Circle(SIZE / 2, SIZE / 2, SIZE / 10,
Color.RED);
		this.getChildren().add(cir);
		marked.add(this);
	}

	private void removeMoveMark() {
		for (Square square : marked) {
			square.getChildren().remove(square.getChildren().size()
- 1);
		}
		marked.clear();
	}

	private void makeActive() {
```

```
        if (active != null) {
                active.makeInactive();
        }
        active = this;
        this.piece.showMove(col(), row());
        this.getBackground().setFill(Color.GREEN);
}

private void makeInactive() {
        removeMoveMark();
        active = null;
        this.getBackground().setFill(originalColor);
}

public void addPiece(Piece p) {
        if (hasPiece()) {
                this.getChildren().remove(piece);
        }
        this.piece = p;
        this.getChildren().add(p);
}

public boolean hasPiece() {
        return this.piece != null;
}
```

```
	public Rectangle getBackground() {
		return this.bg;
	}

	public Boolean turn(int i) {
		boolean whiteTurn = true;
		if (i % 2 == 0) {
			return whiteTurn = false;
		} else {
			return whiteTurn;
		}
	}
}
//--------------------Game--------------------------------
package game;
import board.ChessBoard;
import board.Square;
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.image.Image;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;
import javafx.stage.Stage;
```

```
public class game extends Application {

	public static Stage window;
	Scene menuScene, gameScene;

	private static Scene blackwinScreen;
	private static Scene whitewinScreen;


	public static void main(String[] args) {
		launch();
	}

	public void start(Stage primaryStage) throws Exception {

		window = primaryStage;

		// Menu----------------------------------------------------

		Label gameName = new Label("Chess Project");
		Button startButton = new Button("Start");
		Button exitButton = new Button("Exit");

		String style1 = "-fx-font-size: 3em;";
```

```
        String style2 = "-fx-background-color: linear-gradient(#f2f2f2, #d6d6d6), linear-gradient(#fcfcfc 0%, #d9d9d9 20%, #d6d6d6 100%), "
                                        + "linear-gradient(#dddddd 0%, #f6f6f6 50%); -fx-background-radius: 8,7,6; -fx-background-insets: 0,1,2; -fx-text-fill: black;"
                                        + "-fx-effect: dropshadow( three-pass-box , rgba(0,0,0,0.6) , 5, 0.0 , 0 , 1 ); -fx-font-size: 1.9em; -fx-padding: 10px;";

        gameName.setStyle(style1);
        startButton.setStyle(style2);
        exitButton.setStyle(style2);

        VBox menuVBox = new VBox(40);
        menuVBox.setAlignment(Pos.CENTER);

        menuVBox.getChildren().addAll(gameName, startButton, exitButton);

        Scene menuScene = new Scene(menuVBox, Square.SIZE * 8 - 10, Square.SIZE * 8 - 10);

        // Game----------------------------------------------------

        ChessBoard board = new ChessBoard();
        Scene gameScene = new Scene(board, Square.SIZE * 8, Square.SIZE * 8);
```

```
        //White Win----------------------------------------------------

        Label whitewin = new Label("Congratulations, White Wins");

        whitewin.setStyle(style1);

        VBox winVBox = new VBox(40);
        winVBox.setAlignment(Pos.CENTER);

        winVBox.getChildren().add(whitewin);

        whitewinScreen = new Scene(winVBox, Square.SIZE * 8,
Square.SIZE * 8, Color.BLACK);

        startButton.setOnAction(e ->
primaryStage.setScene(gameScene));
        exitButton.setOnAction(e -> System.exit(0));

        // Black Win----------------------------------------------------

        Label blackwin = new Label("Congratulations, Black Wins");

        String style3 = "-fx-font-size: 3em;";

        blackwin.setStyle(style3);
```

```
            VBox blackwinVBox = new VBox(40);
            blackwinVBox.setAlignment(Pos.CENTER);


            blackwinVBox.getChildren().add(blackwin);


            blackwinScreen = new Scene(blackwinVBox, Square.SIZE * 8,
Square.SIZE * 8, Color.BLACK);


            window.getIcons().add(new Image("file:/images/
WHITE_KING.png"));
            window.setTitle("Chess Projec");
            window.setResizable(false);
            window.setScene(menuScene);
            window.show();
      }


      public static void blackwin() {
            window.setScene(blackwinScreen);
      }


      public static void whitewin() {
            window.setScene(whitewinScreen);
      }
}
//--------------------Bishop-------------------------------
package pieces;
```

```
import board.ChessBoard;
import board.Square;
import javafx.scene.paint.Color;

public class Bishop extends Piece {

	public Bishop(Color c) {
		super(c);
	}

	public void showMove(int x, int y) {

		// Up Left
		for (int i = 1; i < 8; i++) {
			if (x - i < 0 || y - i < 0) {
				break;
			}
			Square s1 = ChessBoard.getSquare(x - i, y - i);
			if (!s1.hasPiece()) {
				s1.mark(getColor());
			} else {
				s1.mark(getColor());
				break;
			}
		}
```

```
// Up Right
for (int i = 1; i < 8; i++) {
	if (x + i > 7 || y - i < 0) {
		break;
	}
	Square s1 = ChessBoard.getSquare(x + i, y - i);
	if (!s1.hasPiece()) {
		s1.mark(getColor());
	} else {
		s1.mark(getColor());
		break;
	}
}

// Down Left
for (int i = 1; i < 8; i++) {
	if (x + i > 7 || y + i > 7) {
		break;
	}
	Square s1 = ChessBoard.getSquare(x + i, y + i);
	if (!s1.hasPiece()) {
		s1.mark(getColor());
	} else {
		s1.mark(getColor());
		break;
	}
```

```
            }

            // Down Right
            for (int i = 1; i < 8; i++) {
                if (x - i < 0 || y + i > 7) {
                    break;
                }
                Square s1 = ChessBoard.getSquare(x - i, y + i);
                if (!s1.hasPiece()) {
                    s1.mark(getColor());
                } else {
                    s1.mark(getColor());
                    break;
                }
            }
        }

        public void hideMove() {

        }
}
//——————————King——————————————
package pieces;

import board.ChessBoard;
import board.Square;
```

```
import javafx.scene.paint.Color;

public class King extends Piece {

	public King(Color c) {
		super(c);
	}

	public void showMove(int x, int y) {

		// Left
		for (int i = 1; i < 2; i++) {
			if (x - i < 0) {
				break;
			}
			Square s1 = ChessBoard.getSquare(x - i, y);
			if (!s1.hasPiece()) {
				s1.mark(getColor());
			} else {
				s1.mark(getColor());
				break;
			}
		}

		// Right
		for (int i = 1; i < 2; i++) {
```

```
        if (x + i > 7) {
                break;
        }
        Square s1 = ChessBoard.getSquare(x + i, y);
        if (!s1.hasPiece()) {
                s1.mark(getColor());
        } else {
                s1.mark(getColor());
                break;
        }
}

// Up
for (int i = 1; i < 2; i++) {
        if (y - i < 0) {
                break;
        }
        Square s1 = ChessBoard.getSquare(x, y - i);
        if (!s1.hasPiece()) {
                s1.mark(getColor());
        } else {
                s1.mark(getColor());
                break;
        }
}
```

```
// Down
for (int i = 1; i < 2; i++) {
	if (y + i > 7) {
		break;
	}
	Square s1 = ChessBoard.getSquare(x, y + i);
	if (!s1.hasPiece()) {
		s1.mark(getColor());
	} else {
		s1.mark(getColor());
		break;
	}
}

// Up Left
for (int i = 1; i < 2; i++) {
	if (x - i < 0 || y - i < 0) {
		break;
	}
	Square s1 = ChessBoard.getSquare(x - i, y - i);
	if (!s1.hasPiece()) {
		s1.mark(getColor());
	} else {
		s1.mark(getColor());
		break;
	}
```

```
}

// Up Right
for (int i = 1; i < 2; i++) {
	if (x + i > 7 || y - i < 0) {
		break;
	}
	Square s1 = ChessBoard.getSquare(x + i, y - i);
	if (!s1.hasPiece()) {
		s1.mark(getColor());
	} else {
		s1.mark(getColor());
		break;
	}
}

// Down Left
for (int i = 1; i < 2; i++) {
	if (x + i > 7 || y + i > 7) {
		break;
	}
	Square s1 = ChessBoard.getSquare(x + i, y + i);
	if (!s1.hasPiece()) {
		s1.mark(getColor());
	} else {
		s1.mark(getColor());
```

```
                        break;
                }
            }

            // Down Right
            for (int i = 1; i < 2; i++) {
                if (x - i < 0 || y + i > 7) {
                        break;
                }
                Square s1 = ChessBoard.getSquare(x - i, y + i);
                if (!s1.hasPiece()) {
                        s1.mark(getColor());
                } else {
                        s1.mark(getColor());
                        break;
                }
            }
        }

        public void hideMove() {

        }
}
//----------------------Knight----------------------------
package pieces;
```

```
import board.ChessBoard;
import board.Square;
import javafx.scene.paint.Color;

public class Knight extends Piece {

	public Knight(Color c) {
		super(c);
	}

	public void showMove(int x, int y) {

		// Up Right
		if (x + 1 < 8 && y + 2 < 8) {
			Square s1 = ChessBoard.getSquare(x + 1, y + 2);
			s1.moveMark();
		}

		// Right Up
		if (x + 2 < 8 && y + 1 < 8) {
			Square s1 = ChessBoard.getSquare(x + 2, y + 1);
			s1.moveMark();
		}

		// Right Down
		if (x + 2 < 8 && y - 1 >= 0) {
```

```
        Square s1 = ChessBoard.getSquare(x + 2, y - 1);
        s1.moveMark();
}

// Down Right
if (x + 1 < 8 && y - 2 >= 0) {
        Square s1 = ChessBoard.getSquare(x + 1, y - 2);
        s1.moveMark();
}

// Down Left
if (x - 1 >= 0 && y - 2 >= 0) {
        Square s1 = ChessBoard.getSquare(x - 1, y - 2);
        s1.moveMark();
}

// Left Down
if (x - 2 >= 0 && y - 1 >= 0) {
        Square s1 = ChessBoard.getSquare(x - 2, y - 1);
        s1.moveMark();
}

// Left Up
if (x - 2 >= 0 && y + 1 < 8) {
        Square s1 = ChessBoard.getSquare(x - 2, y + 1);
        s1.moveMark();
```

```
            }

            // Up Left
            if (x - 1 >= 0 && y + 2 < 8) {
                Square s1 = ChessBoard.getSquare(x - 1, y + 2);
                s1.moveMark();
            }
        }

        public void hideMove() {

        }
}
//———-Movealble————————-
package pieces;

public interface Moveable {

        void showMove(int x, int y);

        void hideMove();

        void move();

}
package pieces;
```

```
import board.ChessBoard;
import board.Square;
import javafx.scene.paint.Color;

public class Pawn extends Piece {

	private boolean hasMoved = false;

	public Pawn(Color c) {
		super(c);
	}

	public void showMove(int x, int y) {
		if (this.getColor() == Color.WHITE) {
			Square s1 = ChessBoard.getSquare(x, y - 1);
			if (!s1.hasPiece()) {
				s1.moveMark();
			} else {
				return;
			}
			if (!hasMoved) {
				Square s2 = ChessBoard.getSquare(x, y - 2);
				if (!s2.hasPiece()) {
					s2.moveMark();
				} else {
```

```
                        return;
                    }
                }
            } else {
                Square s1 = ChessBoard.getSquare(x, y + 1);
                if (!s1.hasPiece()) {
                    s1.moveMark();
                }
                if (!hasMoved) {
                    Square s2 = ChessBoard.getSquare(x, y + 2);
                    if (!s2.hasPiece()) {
                        s2.moveMark();
                    }
                }
            }
                // White Right Up
                if (this.getColor() == Color.WHITE) {
                    Square s3 = ChessBoard.getSquare(x + 1, y - 1);
                    if (s3.hasPiece() && this.getColor() != Color.BLACK)
{
                        s3.attackMark();
                    }

                    // White Left Up
                    Square s5 = ChessBoard.getSquare(x - 1, y - 1);
```

```
                if (s5.hasPiece() && this.getColor() != Color.BLACK)
{
                        s5.attackMark();
                } else {
                        return;
                }
        }

        // Black Right Down
        if (this.getColor() == Color.BLACK) {
                Square s4 = ChessBoard.getSquare(x + 1, y + 1);
                if (s4.hasPiece() && this.getColor() != Color.WHITE)
{
                        s4.attackMark();
                }

                // Black Left Down
                Square s3 = ChessBoard.getSquare(x - 1, y + 1);
                if (s3.hasPiece() && this.getColor() != Color.WHITE)
{
                        s3.attackMark();
                } else {
                        return;
                }
        }
```

```
	}

	public void changePawnToQueen() {

	}

	public void move() {
		hasMoved = true;
	}

	public void hideMove() {

	}
}

//——————————————Piece——————————
package pieces;

import board.Square;
import javafx.scene.Group;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.paint.Color;

public abstract class Piece extends Group implements Moveable {
```

```
	private Color color;

	public Piece(Color c) {

		this.color = c;

		String COLOR = "WHITE";
		if (c == Color.BLACK) {
			COLOR = "BLACK";
		}

		Image img = new Image("/images/" + COLOR + "_" +
this.getClass().getSimpleName().toUpperCase() + ".png");
		ImageView iv = new ImageView(img);
		iv.setFitWidth(Square.SIZE);
		iv.setFitHeight(Square.SIZE);
		this.getChildren().add(iv);
	}

	public void move() {

	}

	public Color getColor() {
		return this.color;
	}
```

```
}
package pieces;

import board.ChessBoard;
import board.Square;
import javafx.scene.paint.Color;

public class Queen extends Piece {

	public Queen(Color c) {
		super(c);
	}

	public void showMove(int x, int y) {

		// Left
		for (int i = 1; i < 8; i++) {
			if (x - i < 0) {
				break;
			}
			Square s1 = ChessBoard.getSquare(x - i, y);
			if (!s1.hasPiece()) {
				s1.mark(getColor());
			} else {
				s1.mark(getColor());
				break;
```

```
        }
    }

    // Right
    for (int i = 1; i < 8; i++) {
        if (x + i > 7) {
            break;
        }
        Square s1 = ChessBoard.getSquare(x + i, y);
        if (!s1.hasPiece()) {
            s1.mark(getColor());
        } else {
            s1.mark(getColor());
            break;
        }
    }

    // Up
    for (int i = 1; i < 8; i++) {
        if (y - i < 0) {
            break;
        }
        Square s1 = ChessBoard.getSquare(x, y - i);
        if (!s1.hasPiece()) {
            s1.mark(getColor());
        } else {
```

```
                s1.mark(getColor());
                break;
        }
}

// Down
for (int i = 1; i < 8; i++) {
        if (y + i > 7) {
                break;
        }
        Square s1 = ChessBoard.getSquare(x, y + i);
        if (!s1.hasPiece()) {
                s1.mark(getColor());
        } else {
                s1.mark(getColor());
                break;
        }
}

// Up Left
for (int i = 1; i < 8; i++) {
        if (x - i < 0 || y - i < 0) {
                break;
        }
        Square s1 = ChessBoard.getSquare(x - i, y - i);
        if (!s1.hasPiece()) {
```

```
                s1.mark(getColor());
        } else {
                s1.mark(getColor());
                break;
        }
}

// Up Right
for (int i = 1; i < 8; i++) {
        if (x + i > 7 || y - i < 0) {
                break;
        }
        Square s1 = ChessBoard.getSquare(x + i, y - i);
        if (!s1.hasPiece()) {
                s1.mark(getColor());
        } else {
                s1.mark(getColor());
                break;
        }
}

// Down Left
for (int i = 1; i < 8; i++) {
        if (x + i > 7 || y + i > 7) {
                break;
        }
```

```
                Square s1 = ChessBoard.getSquare(x + i, y + i);
                if (!s1.hasPiece()) {
                        s1.mark(getColor());
                } else {
                        s1.mark(getColor());
                        break;
                }
        }

        // Down Right
        for (int i = 1; i < 8; i++) {
                if (x - i < 0 || y + i > 7) {
                        break;
                }
                Square s1 = ChessBoard.getSquare(x - i, y + i);
                if (!s1.hasPiece()) {
                        s1.mark(getColor());
                } else {
                        s1.mark(getColor());
                        break;
                }
        }
}

public void hideMove() {
```

```
	}
}
package pieces;

import board.ChessBoard;
import board.Square;
import javafx.scene.paint.Color;

public class Rook extends Piece {

	@SuppressWarnings("unused")
	private boolean hasMoved = false;

	public Rook(Color c) {
		super(c);
	}

	public void showMove(int x, int y) {

		// Left
		for (int i = 1; i < 8; i++) {
			if (x - i < 0) {
				break;
			}
			Square s1 = ChessBoard.getSquare(x - i, y);
			if (!s1.hasPiece()) {
```

```
                s1.mark(getColor());
        } else {
                s1.mark(getColor());
                break;
        }
}

// Right
for (int i = 1; i < 8; i++) {
        if (x + i > 7) {
                break;
        }
        Square s1 = ChessBoard.getSquare(x + i, y);
        if (!s1.hasPiece()) {
                s1.mark(getColor());
        } else {
                s1.mark(getColor());
                break;
        }
}

// Up
for (int i = 1; i < 8; i++) {
        if (y - i < 0) {
                break;
        }
```

```
                Square s1 = ChessBoard.getSquare(x, y - i);
                if (!s1.hasPiece()) {
                        s1.mark(getColor());
                } else {
                        s1.mark(getColor());
                        break;
                }
        }

        // Down
        for (int i = 1; i < 8; i++) {
                if (y + i > 7) {
                        break;
                }
                Square s1 = ChessBoard.getSquare(x, y + i);
                if (!s1.hasPiece()) {
                        s1.mark(getColor());
                } else {
                        s1.mark(getColor());
                        break;
                }
        }
}

public void move() {
        hasMoved = true;
```

```
        }

        public void hideMove() {

        }
}
```

Reference:

https://commons.wikimedia.org/wiki/Category:SVG_chess_pieces
https://www.youtube.com/watch?v=h8fSdSUKttk&t=24s
https://www.youtube.com/watch?v=yBsWza2039o