Mrs. Terri Kelly 821 HN Advanced Programming Student: Michael Bobrowski
Initial Final Project Description

Chess – Two players play chess using a mouse to interact with a GUI

1. Classes
    a. ChessGame – Contains every piece and the array the contains the positions of the pieces in players and id as well as the Board setup, Frame setup, and an Engine. ChessGame also controls the layout of the JFrame;
        i. Libraries
            1. java.awt.Color
            2. java.awt.GridLayout
            3. java.awt.BorderLayout
            4. javax.swing.JPanel
            5. javax.swing.border.Border
        ii. fields
            1. Frame frame
            2. Engine engine
            3. Board chessBoard
            4. Int[][] gameArray
            5. Int[][] idArray
            6. Piece pawnOne1
            7. Piece pawnOne2
            8. Piece pawnOne3
            9. Piece pawnOne4
            10. Piece pawnOne5
            11. Piece pawnOne6
            12. Piece pawnOne7
            13. Piece pawnOne8
            14. Piece rookOne1
            15. Piece rookOne2
            16. Piece knightOne1
            17. Piece knightOne2
            18. Piece bishopOne1
            19. Piece bishopOne2
            20. Piece queenOne
            21. Piece kingOne
            22. Piece pawnTwo1
            23. Piece pawnTwo2
            24. Piece pawnTwo3
            25. Piece pawnTwo4
            26. Piece pawnTwo5
            27. Piece pawnTwo6
            28. Piece pawnTwo7
            29. Piece pawnTwo8
            30. Piece rookTwo1
            31. Piece rookTwo2
            32. Piece knightTwo1
            33. Piece knightTwo2
            34. Piece bishopTwo1
            35. Piece bishopTwo2
            36. Piece queenTwo

37. Piece kingTwo

iii. constructor headers
1. public ChessGame() – instantiates the chessBoard, engine, frame, gameArray, and idArray. Makes a painter JPanel and a bar JPanel to add to the frame. This also calls the methods to create the pieces and fill the arrays.

iv. getters – none

v. setters – none

vi. private methods
1. createPiecesPlayerOne – instantiates the player one pieces
2. createPiecesPlayerTwo – instantiates the player two pieces
3. fillGameArray – fills the gameArray
4. fillIdArray – fills the idArray

vii. public methods – none

b. Frame – creates a JFrame with a mouse listener and mouse motion listener and controls the piece movement

i. Libraries
1. javax.swing.JFrame
2. java.awt.event.MouseEvent
3. java.awt.event.MouseListener
4. java.awt.event.MouseMotionListener
5. java.awt.event.ActionEvent
6. java.awt.event.ActionListener

ii. fields
1. int x
2. int y
3. int startRow
4. int startColumn
5. int endRow
6. int endColumn
7. int turn
8. boolean firstTurn
9. boolean validMove
10. boolean inCheck
11. int pieceToMove
12. int pieceAt
13. Piece piece
14. Piece affectedPiece
15. ChessGame game
16. Engine engine
17. Int[][] gameArray
18. Int[][] idArray

iii. constructor headers
1. public Frame(String title, Engine engine, ChessGame game) – sets the title of the JFrame with title and sets the values of game, engine, gameArray, and id Array. This also calls the setup method

iv. getters – none

v. setters – none

vi. private methods
1. getColumn – gets the column that the click on the board was on

2. getRow – gets the row that the click on the board was on

vii. public methods
1. setup – sets the characteristics of the JFrame and creates the mouse listener and mouse listener

c. Board – Contains the characteristics of the chess board including rows, columns, primaryColor, secondaryColor, playerOneColor, and playerTwoColor
i. Libraries
1. java.awt.Color
ii. fields
1. int rows
2. int columns
3. Color primaryColor
4. Color secondaryColor
5. Color playerOneColor
6. Color playerTwoColor
iii. Constructor headers
1. public Board(int rows, int columns, Color primaryColor, Color secondaryColor, Color playerOneColor, Color playerTwoColor) – sets the amount of rows and columns, as well as the colors that will be used for painting on the frame
iv. Getters
1. getRows – returns the number of rows
2. getColumns – returns the number of columns
3. getPrimaryColor – returns the primary board color
4. getSecondaryColor – returns the secondary board color
5. getPlayerOneColor – returns the color of player one's pieces
6. getPlayerTwoColor – returns the color of player two's pieces
v. setters
1. setRows – sets the number of rows
2. setColumns – sets the number of columns
3. setPrimaryColor – sets the primary board color
4. setSecondaryColor – sets the secondary board color
5. setPlayerOneColor – sets player one's piece color
6. setPlayerTwoColor – sets player two's piece color
vi. private methods – none
vii. public methods
1. toString – returns all the characteristics of the board in the formatted string

d. Piece – controls the characteristics of the pieces including type, row, column, value, player, whether or not they are alive, and the piece
i. Libraries – none
ii. Fields
1. String type
2. int row
3. int column
4. int value
5. int player
6. boolean alive
7. int piece
iii. public Constructor headers

1. Piece(String type, int row, int column, int value, int player, boolean alive, int piece) – sets the location of the piece, its name, its value, whether or not it is alive, and its id number

iv. Getters
1. getType – returns the name of the piece
2. getRow – returns the current row of the piece
3. getColumn – returns the current column of the piece
4. getValue – returns the value of the piece
5. getPlayer – returns the player affiliation of the piece
6. getAlive – returns the life status of the piece
7. getPiece – returns the id number of the piece

v. Setters
1. setType – sets the name of the piece
2. setRow – sets the row of the piece
3. setColumn – sets the column of the piece
4. setValue – sets the value of the piece
5. setPlayer – sets the player of the piece
6. setAlive – sets the life status of the piece
7. setPiece – sets the id number of the pieces

vi. private methods – none

vii. public methods
1. toString – returns the characteristics of a piece in a formatted string

e. Painter – paints on the JFrame

i. Libraries
1. java.awt.Dimension
2. java.awt.Graphics
3. java.awt.Graphics2D
4. java.awt.Color
5. java.awt.Font
6. java.awt.Rectangle
7. java.awt.Polygon
8. java.awt.geom.Arc2D
9. java.awt.geom.Ellipse2D
10. javax.swing.JPanel

ii. Fields
1. ChessGame chess
2. Frame frame
3. Board chessBoard

iii. Constructor headers
1. public Painter(ChessGame chess) – instantiated chess, frame, and chessBoard, and created a Dimension to set the size of the JPanel

iv. Getters – none

v. Setters – none

vi. Private methods
1. drawPawn – draws a pawn
2. drawRook – draws a rook
3. drawKnight – draws a knight
4. drawBishop – draws a bishop
5. drawQueen – draws a queen

6. drawKing – draws a king

vii. public methods

1. paintComponent – paints the chessBoard and all of the pieces onto the JFrame

f. ColorButtonPanel – makes a JPanel with a JColorChooser

i. Libraries

1. java.awt.Color
2. java.awt.event.ActionEvent
3. java.awt.event.ActionListener
4. java.awt.Component
5. java.awt.Font
6. javax.swing.JPanel
7. javax.swing.JColorChooser
8. javax.swing.JButton

ii. Fields

1. Board board
2. Color color
3. Color currentChoice

iii. Constructor headers

1. ColorButtonPanel(String title, Color color, int type, Board board) – sets the title, background color, and type of the JColorChooser

iv. Getters – none

v. Setters – none

vi. Private methods – none

vii. Public methods

1. createButton – creates a JButton

g. SideBar – creates a JPanel with JButtons that make JColorChoosers for color selection

i. Libraries

1. java.awt.Color
2. java.awt.Dimension
3. javax.swing.Box
4. javax.swing.BoxLayout
5. javax.swing.JPanel

ii. Fields

1. ChessGame chessBoard
2. Board board
3. Color boardPrimary
4. Color boardSecondary

iii. Constructor headers

1. public SideBar(ChessGame game) – instantiates chessBoard, board, boardPrimary, and boardSecondary and adds the JPanels with JButtons for color customization

iv. Getters – none

v. Setters – none

vi. Private methods – none

vii. Public methods – none

h. Engine – contains all of the rules for the game of Chess

i. Libraries – none

ii. Fields

1. ChessGame main
2. Int[][] gameArray

iii. Constructor headers
    1. public Engine(ChessGame game) – instantiates main and gameArray
iv. Getters – none
v. Setters – none
vi. Private methods – none
vii. Public methods
    1. verifyPawn – verifies a move by a pawn
    2. verifyRook – verifies a move by a rook
    3. verifyKnight – verifies a move by a knight
    4. verifyBishop – verifies a move by a bishop
    5. verifyQueen – verifies a move by the queen
    6. verifyKing – verifies a move by the king
    7. inCheckWhite – checks if white has put the black king in check
    8. inCheckBlack – checks if black has put the white king in check

2. Approach – bulleted descriptions of each of these
    a. Setup
        i. Build the JFrame
        ii. Create the ChessGame class
        iii. Create the Board class
        iv. Create the Piece class
        v. Create the ColorButtonPanel class
        vi. Create the SideBar class
        vii. Create the Painter class
        viii. Create the Engine class
        ix. Add movement mechanics using the mouse listener in frame
    b. Run of Program
        i. Create the JFrame
        ii. Add the JPanels to the JFrame
        iii. The GUI will handle the rest
    c. Ending
        i. The JFrame exits on close
3. Research – list the area and how you intend to use the area
    a. JButton – control the colors by using a JButton that opens a JColorChooser
    b. JColorChooser – control the colors by using a JColorChooser to read in user input on what color they are choosing
    c. GridLayout – control how JPanels are placed on the JFrame
    d. BorderLayout – control how JPanels are placed on the JFrame
4. Help Requested – list the areas where you are hoping that the instructor will provide quidance
    a. (question answered) How can I get all of the methods in all of the classes to work with each other to run the game of Chess
5. Citations
    a. https://docs.oracle.com/javase/7/docs/api/javax/swing/JButton.html
    b. https://docs.oracle.com/javase/7/docs/api/javax/swing/JColorChooser.html
    c. https://docs.oracle.com/javase/7/docs/api/java/awt/GridLayout.html
    d. https://docs.oracle.com/javase/7/docs/api/java/awt/BorderLayout.html
    e. https://docs.oracle.com/javase/tutorial/uiswing/components/colorchooser.html
    f. https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html