Mrs. Terri Kelly

821 HN Advanced Programming Student: Michael Bobrowski
Initial Final Project Description

JChess – The user will run the program and choose the colors that they want for the game using JColorChoosers that can be accessed with a button. The player will request a move using a JTextField that will then be validated or denied by the game engine. The game will end when one of the kings is in checkmate, which will be detected by the engine class.

rewrite 2 as just prose – no technical

1. Classes
   a. Main – The class will run the other classes of the project as well as contain the basic setup characteristics for the JFrame.
      i. Libraries
         1. javax.swing.JFrame
         2. java.awt.Container
         3. java.awt.Color
         4. ~~java.awt.event.KeyEvent~~
         5. java.awt.event.KeyListener

.text of textfield

~~Use after update~~

use a button for turn

      ii. Fields
         1. JFrame frame
         2. Container game
         3. DrawingPanel drawingPanel – instantiating an instance of Drawing Panel Class to add to the frame

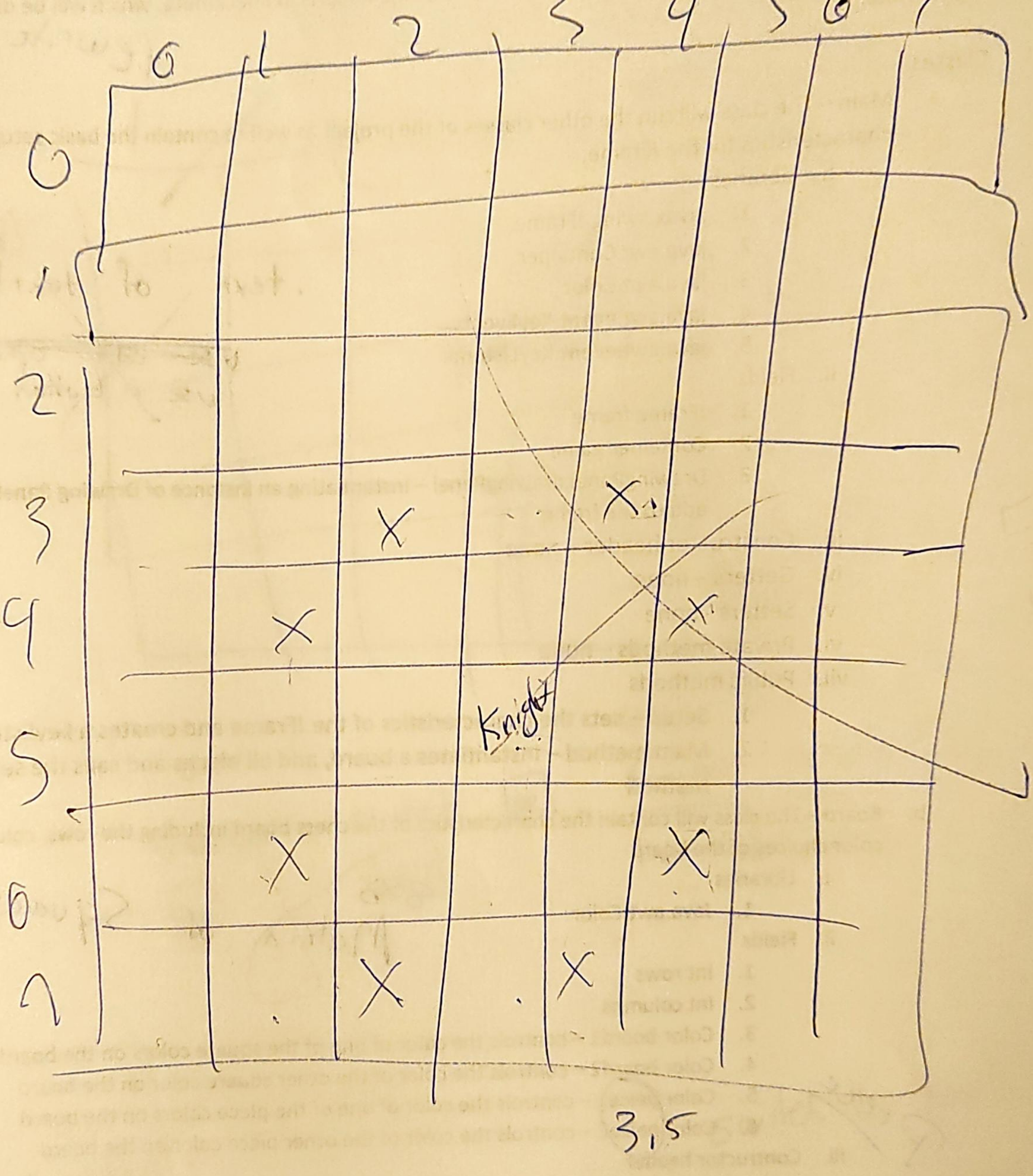
      iii. Constructor header – none
      iv. Getters – none
      v. Setters - none
      vi. Private methods – none
      vii. Public methods
         1. Setup – sets the characteristics of the JFrame and creates a keylistener
         2. Main method – instantiates a board, and all pieces and calls the setup method
   b. Board – The class will contain the characteristics of the chess board including the rows, columns, and color choices of the board
      i. Libraries
         1. Java.awt.Color

8x8 Matrix of Squares

      ii. Fields
         1. int rows
         2. int columns
         3. Color board1 – controls the color of one of the square colors on the board
         4. Color board2 – controls the color of the other square color on the board
         5. Color peice1 – controls the color of one of the piece colors on the board
         6. Color peice2 – controls the color of the other piece color on the board

Piece

      iii. Contructor header
         1. Board(int rows, int columns, Color board1, Color board2, Color piece1, Color peice2)
      iv. Getters
         1. getRows
         2. getColumns
         3. getBoard1 – gets the board1 color
         4. getBoard2 – gets the board2 color
         5. getPeice1 – gets the piece1 color
         6. getPeice2 – gets the peice2 color
      v. Setters



(3,4)

array of 32 pieces

1. setRows
2. setColumns
3. setBoard1 – sets the board1 color
4. setBoard2 – sets the board2 color
5. setPeice1 – sets the piece1 color
6. setPeice2 – sets the piece1 color

vi. Private methods – none

vii. Public methods

1. toString – returns the attributes of the board

c. Piece – the attributes of all the pieces in the game 2 JBUTTON > SQUARE on Board

i. Libraries

1. java.awt.Color

ii. Fieds

1. String type – the type of piece (pawn, rook, etc.)
2. int[][] coords – the current coordinates of the piece
3. int value – the points value of the piece
4. Color color – the color of the piece(white or black only – independent of the color choosing)

iii. Constructor header

1. public Piece(String type, int[][] coords, int value, Color color) ~~unicode~~ constant

iv. Getters

1. getType – gets the type of peice
2. getCoords – gets the current coordinates of the piece
3. getValue – gets the value of the piece
4. getColor – gets the color of the piece

v. Setters

1. setType – sets the type of piece
2. setCoords – sets the coorinates of the piece
3. setValue – sets the point value of the peice
4. setColor – sets the color of the piece

vi. Private methods – none

vii. Public methods

1. toString – returns the attributes of the peice in a String

d. Engine – the game engine that plays the game

i. Libraries – none

ii. Fields – none

iii. Constructor Header – none

iv. Getters – none

v. Setters – none

vi. Private Methods – none

vii. Public Methods

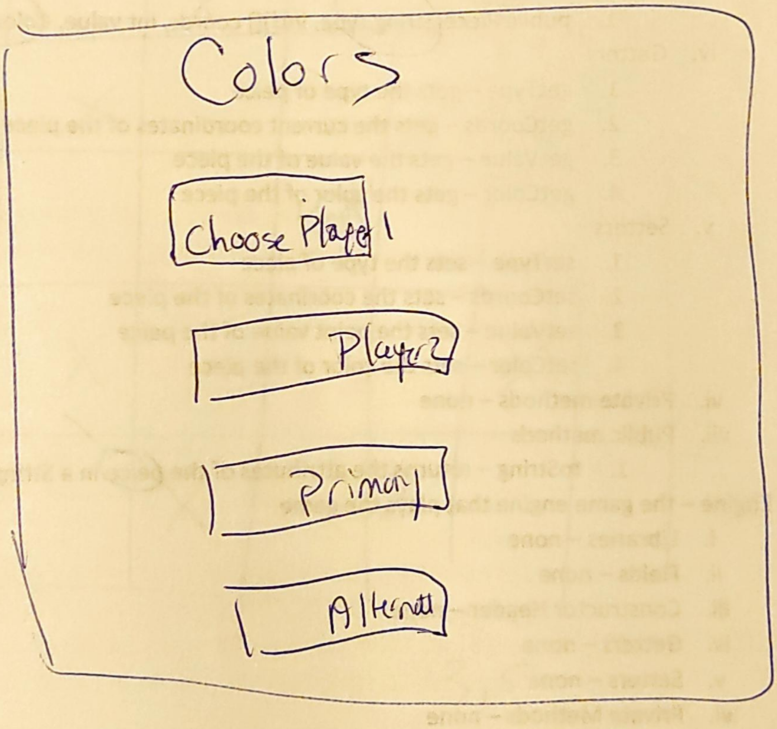
1. inCheckWhite – checks if the white king is in check
2. inCheckBlack – checks if the black king is in check
3. mateWhite – checks if the white king is in checkmate
4. mateBlack – checks if the black king is in checkmate
5. legalMove – checks if the requested move is legal according to the rules of chess

Multiple methods probably 1 for each piece type

e. Painter – draws all of the graphics

i. Libraries

1. java.awt.Graphics



Change background of button to color chosen

2. java.awt.Graphics2D
3. java.awt.Color
4. java.awt.Font
5. java.awt.event.ActionEvent
6. java.awt.event.ActionPerformed
7. javax.swing.JPanel
8. javax.swing.Timer

ii. Fields – none

iii. Constructor – none

iv. Getters – none

v. Setters – none

vi. Private methods – none

vii. Public methods

1. drawPawn – draws a pawn at desired coordinates
2. drawRook – draws a rook at desired coordinates
3. drawKnight – draws a knight at desired coordinates
4. drawBishop – draws a bishop at desired coordinates
5. drawKing – draws a king at desired coordinates
6. drawQueen – draws a queen at desired coordinates
7. drawBoard – draws a board according to the instantiated board in Main
8. paintComponent – handles everything else graphics related
9. Painter – makes different buttons

f. ButtonCreator – a template to make different buttons

2 panel

i. Libraries

1. java.awt.Color
2. java.awt.Component
3. java.awt.Font
4. javax.swing.JButton
5. javax.swing.JLabel
6. javax.swing.JTextField
7. javax.swing.JColorChooser

ii. Fields

1. String FONT – constant for font that is going to be used
2. Color BACKGROUND – constant for the background color of the buttons
3. Color FOREGROUND – constant for the foreground color of the buttons
4. Font standardFont – font that will be used for all buttons

iii. Constructor – none

iv. Getters – none

v. Setters – none

vi. Private methods – none

vii. Public methods

1. createLabel – creates a JLabel
2. createField – creates a JTextField
3. createButton – creates a JButton
4. createColorChooser – creates a JColorChooser

2. Approach – bulleted descriptions of each of these

a. Setup

i. Build the JFrame

ii. Create the ButtonCreator class

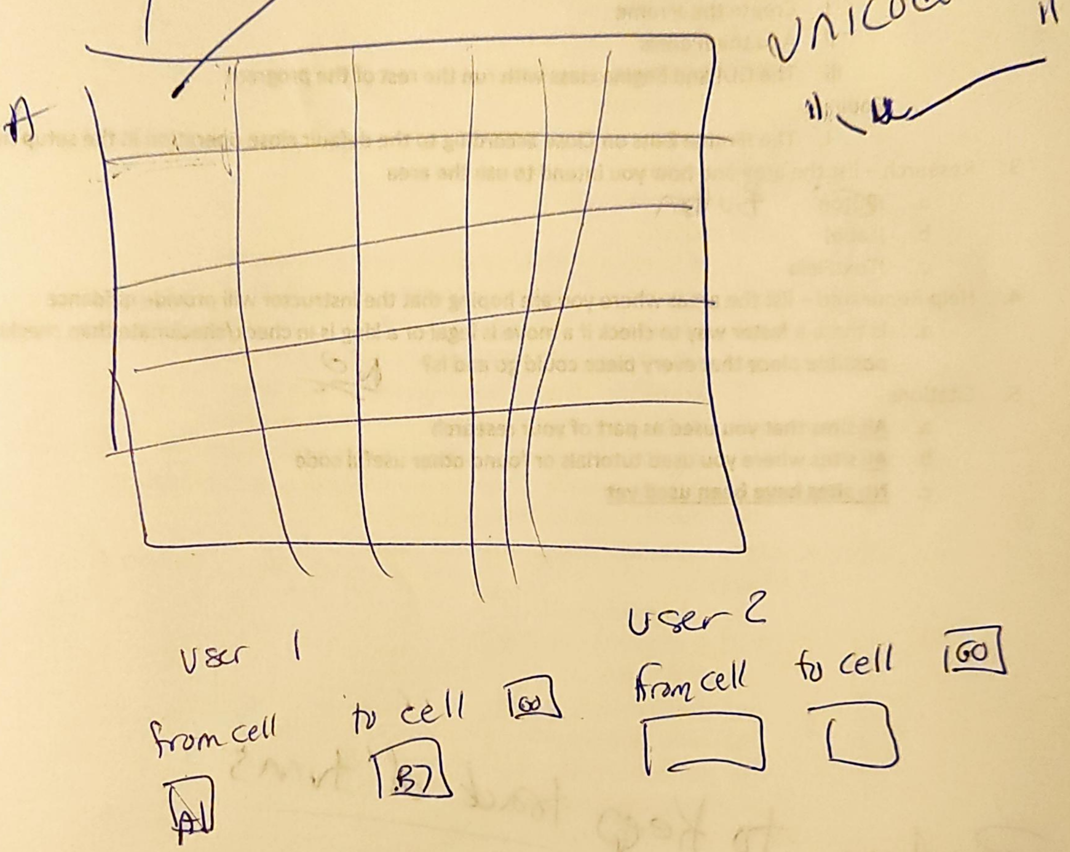
        iii. Create the Board Class
        iv. Create the Piece Class
        v. Create the Create the Painter Class
        vi. Build and test the Engine Class
    b. Run of Program
        i. Create the JFrame
        ii. Add the JPanels
        iii. The GUI and Engine class with run the rest of the program
    c. Ending
        i. The JFrame Exits on Close according to the default close operation in the setup method

3. Research – list the area and how you intend to use the area
    a. JButon Button
    b. JLabel
    c. JTextField
4. Help Requested – list the areas where you are hoping that the instructor will provide quidance
    a. Is there a faster way to check if a move is legal or a king is in check/checkmate than checking every possible place that every piece could go and is? NO
5. Citations
    a. **All** sites that you used as part of your research
    b. **All** sites where you used tutorials or found other useful code
    c. **No sites have been used yet**

Boolean to Keep track of turns

Buttons w/ unicode chess pice

set text to unicode



Make constants

WHITE_ROOK = "\u—";

switch (piece)

case WHITE-ROOK: ______