

B242L Coursework 2

University of Portsmouth, UK
Petri Tuononen
24/05/2010

Table of Contents

1. Most notable changes from the text based version.....	3
2. AWT classes used.....	3
3. Solving problems.....	3
3.1 Problem 1: How to draw a chess board.....	3
3.2 Problem 2: How to draw pieces.....	3
3.3 Problem 3: How to select piece and destination square with a mouse click.....	3
3.4 Problem 4: How to move pieces.....	4
3.5 Problem 5: How to highlight selected piece and it's legal moves.....	4
3.6 Problem 6: How to implement different modes.....	4
4. Mouse key functionalities.....	4
5. Game modes.....	4
5.1 Human VS Human.....	4
5.2 Human VS AI.....	5
5.3 AI VS AI.....	5
6. Bug fixing.....	5
7. Testing.....	5

Text based chess game (coursework 1) modified to a graphical version.

1. Most notable changes from the text based version

Not that many changes were needed to make a graphical version of the chess game. Game class's main method became defunct and so did Board class's board array variable and the method that was used to draw the text based chess game.

New Gui class was created. Its main function is to draw board layout in paintComponent method and after that draw pieces on top of the board. The basic idea was that every time a piece is moved to one location to another, repaint method is called and board and pieces are drawn again. That's the basic high level idea behind the graphical version.

2. AWT classes used

JFrame is the main window and that contains JMenuBar and JPanel. It was a good idea to use JPanel as a container for chess graphics, because it helps to acquire correct coordinates easily.

3. Solving problems

3.1 Problem 1: How to draw a chess board

Chess board is an 8x8 array where every other square has a different background colour. The sequence is similar in every other line. Basically Graphics object was used to draw rectangles of different colour according to previous description.

3.2 Problem 2: How to draw pieces

Every piece has to be drawn sequentially therefore loop statement is used. Next piece is returned from the arraylist and its colour and type is checked and the correct image chosen according to those values. To be able to draw piece images to the correct place some calculations have to be done. Method notationToPieceLoc returns correct top-left square coordinates when chess notation is given as a parameter. Piece images are exactly same size as the squares and when the image is drawn to specified square's top-left corner the actual piece picture is in the middle of the square. In drawPieces method every piece is given this same treatment inside a loop.

3.3 Problem 3: How to select piece and destination square with a mouse click

When clicking some spot on the frame with a mouse, the coordinates can be acquired from the MouseEvent object. Then the coordinates must be mapped into a specific square on the chess board. This chess game has 400x400 pixel board and one square is 50x50

pixels and board has 8x8 squares. For example top-left square is $0 < x < 50$ and $0 < y < 50$. A method was created that maps a clicked coordinate to chess notation. Then it was easy to just use an old method that returns the right Piece object when chess notation is given as a parameter.

3.4 Problem 4: How to move pieces

When a player clicks a square it is first checked that it actually contains player's own piece. If it does it is selected and then given an opportunity to select a destination square. Then after a second mouse click the selected destination square is checked if it is a legal move. If it's legal movePiece method gets a go ahead. Method changes the pieces coordinates and possibly removes a piece that was captured and move is documented in the history of moves. The method also checks if own pawn is reached the enemy's end and if so the pawn is promoted to queen. After that repaint method is called that draws the board and all pieces again and all the changes are made visible.

3.5 Problem 5: How to highlight selected piece and it's legal moves

Chess game has a functionality that highlights the selected piece and then highlights all the legal moves in different colour that makes playing the game easier for novice players. This functionality is implemented in paintComponent method inside Gui class. Every time paintComponent is called it checks if the piece is selected and if it is then it's background color is changed and all possible legal moves are highlighted with different colour as well. After that all the pieces are drawn on top of the board so that highlighting doesn't make the pieces invisible.

3.6 Problem 6: How to implement different modes

There's three different modes and one of them is always chosen. User can change the mode from the menubar or use F1, F2 and F3 buttons. After changing the mode it would be better to right click once to repaint the board. Every time user left clicks on the board the mode is first checked and actions made according to that.

4. Mouse key functionalities

Left mouse button is used to select piece/square and right mouse button to cancel the selection and then it's possible for a human player to select other piece to move.

5. Game modes

Chess game features three different gaming modes. They are human versus human, human versus computer and computer versus computer.

5.1 Human VS Human

In the beginning white side always starts first. Player first selects the piece to move by

clicking the left mouse button and then left clicks the square where he/she wants to move the piece. As the piece is clicked it can be seen that the piece is selected when it has blue background. Possible moves are identified with orange background. This helps the player to choose quick where to move the piece and it's also a great help for beginners. When the first human player has made a turn then it's the other player's turn. Tip: Pressing a right mouse button cancels the piece selection and it's then possible to choose other piece instead.

5.2 Human VS AI

Human is the white side and it's also the starting side. As the human player moves his/her piece similarly than in Human VS Human mode, computer moves it's piece as soon as human player has moved his/her.

5.3 AI VS AI

Pressing a mouse button makes both computer players to move at the same time one turn at a time.

6. Bug fixing

In the text based version there was a bug. When both computer players had a piece under attack, deadlock could occur so that they both just skipped a turn, if no good moves were in sight. Now it's corrected so that they always move some piece somewhere even in tight situations. A lot of testing has helped to make this bug free.

7. Testing

Especially computer versus computer mode needed a lot of testing. I followed 20 consecutive computer versus computer games and they all ended nicely. It's common that games end in a situation where another player has no real changes to win and needs to resign.