Coursework Report

Alistair Lester
40322185@napier.ac.uk
Edinburgh Napier University - Module Title (SET09117)

Abstract

Keywords – java,checkers,program,ai,undo,redo

1 Introduction

The task was to create a checkers game in any language which allowed for human v human games and human vs ai. A good use of data structures must be used to achieve this. Undo and redo moves being supported too.

2 Design

The first task was first of all began to think about how to display the game board to the user , a multi-dimensional array was used to set up a theoretical game board which is easy to understand and didn't waste any time on looking good. The game board was set up in a different class and package to make it easier to call it. It was then displayed it along with some logical appendix's by using some loops and a string array of letters .

From there the users were asked for the coordinates of the piece they wanted to move on the board and where they wanted to move it. A class was created for the users move data in the model for to allow cleaner data and easier calling. They were stored in a sub sting as one of the inputs is a letter and then ran through a validation function to swap the users letter into a number that the move function can read and use, a switch was used on the user data sub string. Then movement checking function which used a switch statement to check which direction logically the users piece was being moved.

Once the user had performed their move i used a function to end their turn and begin the next players turn and ask them to move again.

All the moves were checked in functions to see if they were illegal , could jump, space was occupied or could become a king. The king function changed the type of the piece depending on what what type it was and which x coordinate it got to, once it became a king you could move it anywhere. If the space was occupied the function would tell the user what type of checker the space was occupied by. the illegal move checker used booleans and a lot of if and else's to check that the piece was moving in a legal direction and also only one space at a time unless they jumped. The jump function checked if the coordinates behind the piece that the user moved to and then called the remove a piece function. The remove function would call the game board and set the coordinates position to 0 making it an empty space for checkers to move on.

The model, game board, move and piece classes all had there own variables and they all had getters and setters and to strings . This object orientated layout was very helpful in implementing the checkers game as it allowed for a few different types of the same data in the main program.

3 Enhancements

If more time were allowed more time there would have been more time spent on the ai to make it more fleshed out and make some more decision based moves, so it chooses good moves occasionally rather than just random. More time would be spent making a scoring system for moves taken to rank players for winning with the least amount of moves. A Gui even though not required for this project would have been nice for the users eyes, one would have been developed using swing as there seems to be a lot of nice swing tutorials (which would involve me learning how to use swing). A gui of some for would allow use of a mouse listener in java and support movement with the move moves and clicks. A proper user interface would have gone down well with users and have room for more options for what to show the user and maybe even different type of checkers games !

4 Critical evaluation

Move could have been handled better , its a big mess of code that is hard to maintain as its hard to read. Storing the data for move should have been stored in a deque it just makes a lot more sense for undo and redo and makes it simple.

Handling the users input could have also been done a lot better as it was passed around many functions and changed alot before actaully making the move.

5 Personal Evaluation

I squandered a lot of time as i was scared of the vastness of this project . I spent to much thinking about it all at once instead of breaking it up into manageable chunks. This caused me to get quite sad and waste even more time being upset about my incapability to know how to get everything working.

I relied on some Friends to explain and show me how some of the things were done, this was a good thing as it made me less scared.

I feel my code is rather sloppy and would have like to rename and change some of the variables.

I have learned how to make a console board and what limitations to put on it to get a functional game going. I took some time to relearn the controller/model method of doing java projects which was beneficial, as it helped my code look a bit neater.

Learning how to move about a 2dimensional array with just console commands was a good thing to learn as a vast amount of time was spend trying to Google how to do it and most of the results were using awt or swing to move.

A lot of what a i relearned was working with array and starting at zero. when i laid out my board i wasn't thinking at first in terms of zero to seven i was laying out a board of one to eight . This took a while to relearn and implement, i made a lot of mistakes along the way where my checker's where laid out wrong and took a while to get them right.

For my first university project i have not done to well but i have learned things simply cannot be left to the last minute. As much as this was a step up from what i am used to work wise i should have risen to the occasion rather than running away.