Coursework Report

Lachlan T Austin-Robb
40223531@napier.ac.uk
Edinburgh Napier University - Algorithms
Data Structures (SET09117)

1 Introduction

The aim of the project was to make a functioning checkers game, that allowed for both human versus human play as well as human versus computer play. It should make use of the most appropriate data structures throughout. As the choice of language was left open, for this project the use of Python was decided upon.

2 Design

The first step was to decide on a data structure to represent the board. For this a dictionary was used, with the key being the number of the square and the value representing whether or not that the particular square contains a game piece. While having each of the board squares numbered from one to sixty four allows for simpler operations for moving the game pieces it is not ideal for the player entering moves. Therefore a second dictionary was constructed, this one with the key being the name of the square, e.g G8 or C3, and the values being the squares number. This means that when a user enters a move, they can enter it in the more user friendly fashion that players are used to and then it can be converted into the squares numbers for use in other operations throughout the program.

when the program is run the first thing to happen is the user is presented with a choice, to enter the number of players, if the player enters 1 then it will begin a human versus computer game, where as if the player had entered 2 then a human versus human game would begin. This would also set the number of players variable to either 1 or 2 so that when the method for player twos turn began it would either auto select a move or ask the human player 2 to enter a move.

Once the user has selected the number of players and the initial starting board has been printed, player one enters their first move. by entering the name of the square for the piece they wish to move followed by where they wish to move it to, for example f1 e2. this input is the split into two variable for the start location and the end location and then each is converted from the squares name into the squares number. It will then check if this is a valid move by ensuring the start location contains one of that players pieces and that the end location is an empty square. If the player entered a move to jump one of the opposing players pieces it will also confirm that and opposing player piece is in the middle of the start and end locations. this is done as the difference between the start and end square numbers is always the same for any move type, for instance player one moving one square diagonally to the right the end square will always be the start number plus nine. similarly when jumping to the right end will be start+18 and it can check that start+9 is an opposing piece.

When a player selects a jump move to take an opposing players piece it will automatically check the squares surrounding the end square for possible other jumps and if possible continue jumping opposing player pieces automatically until it lands on a square where it is unable to do so.

A user can also enter undo and a move input rather than a move. When this is done it cause the board to revert back to its previous state at the beginning of that players last turn. reversing the action made by both players in the interim.

After a turn there is a method that checks how many of each type of piece is left on the board. when one of these numbers equals zero the game will end and the player with remaining pieces declared the winner. Then a play again prompt will be displayed, if the user enters yes the game will restart alternatively if the user enters no the game will quit.

3 Enhancements

One major enhancement that could be made is the inclusion of a gui rather than simply printing the game in the command line. Not only would this add to the game by making it look nicer it would also remove one of the main issues, printing the board after each move, this simply looks bad.

Another enhancement that could be mad is improvements to the ai. Currently in a human versus ai game, the ai will simply make the first available move that it can find. Upgrading this so that it is able to react to the players moves with something more resembling of strategy would make for a more engaging game.

4 Critical Evaluation

While the game is functional, the way it prints to the command line does cause some issues.Most notably is the fact that it reprints the board after every move. While this is alright when a human player makes a simple move, when the ai moves, or either player makes a move that results in multiple automatic jumps, it causes multiple boards to be printed in

rather quick succession and this can at times result in some confusion as to what pieces have moved where.

5 Personal Evaluation

During this project I learned that i need to improve my time management skills. Especially in circumstances involving having to work on multiple different projects at once. I also could stand to do some more planning before starting the coding. I feel more time spent in the planning phase would produce neater more organised code, with methods that are more concise and work together more effectively.