Algorithms and Data Structures Coursework Report

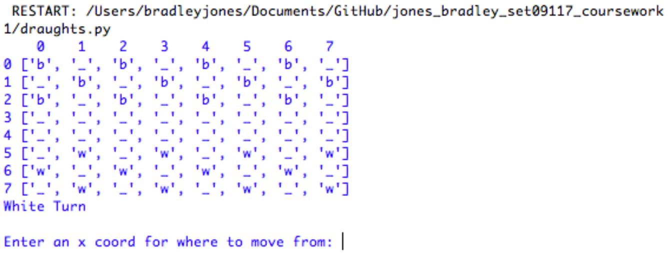
Bradley Jones
40167459@napier.ac.uk
Edinburgh Napier University - Algorithms and Data Structures (SET09117)

Keywords – Algorithms, Data, Structures, Edinburgh, Napier, SET09117, Draughts, King, report

1 Introduction

The objective of the coursework was to demonstrate an understanding of both theory and practise in relation to Algorithms and Data Structures. The task was to implement a Draughts game, that allowed for player versus player play or player versus the computer play or computer versus computer play. The game should record game history, for example the saving of moves, allowing the user to record and replay games, undo and redo features. Finally, your game should implement an algorithm that enables the computer to choose which moves to make during their own turn, i.e. a simple AI player. The choice of algorithm for the AI player may be from the literature, or of the developer's own design and this should all be done using a coding language of the developer's choice.

Overview In this case, the developer's choice of programming language was Python. The features that were included in this version of the draughts game were, the basic rules, for example turns and movement ensuring the pieces could only be moved as according to the rules of draughts and finally, 'kinging' was also implemented.



```
 RESTART: /Users/bradleyjones/Documents/GitHub/jones_bradley_set09117_coursework
1/draughts.py
    0    1    2    3    4    5    6    7
0 ['b', '_', 'b', '_', 'b', '_', 'b', '_']
1 ['_', 'b', '_', 'b', '_', 'b', '_', 'b']
2 ['b', '_', 'b', '_', 'b', '_', 'b', '_']
3 ['_', '_', '_', '_', '_', '_', '_', '_']
4 ['_', '_', '_', '_', '_', '_', '_', '_']
5 ['_', 'w', '_', 'w', '_', 'w', '_', 'w']
6 ['w', '_', 'w', '_', 'w', '_', 'w', '_']
7 ['_', 'w', '_', 'w', '_', 'w', '_', 'w']
White Turn

Enter an x coord for where to move from: |
```

Figure 1: **Draughts.png** - A screenshot of Draughts.py running

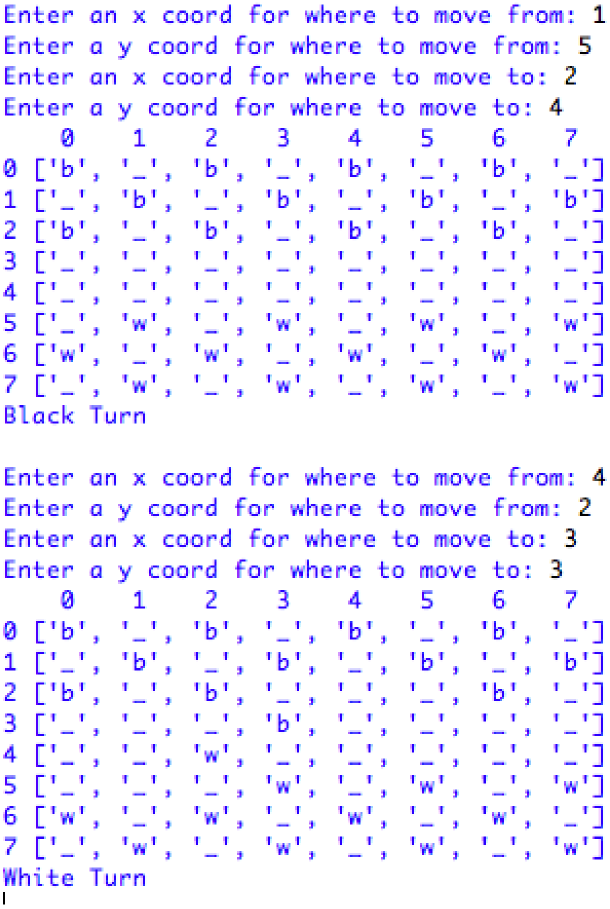
2 Design

List of Lists The primary data structure used in the project was a list of lists. This was chosen because it stepped through the steps one by one, allowing the user to chose coordinates.

Python Python was chosen as the programming language because it is a general-purpose language, which means it can be used to build just about anything. Python is a high level language, so it reads like pseudo-code.

Appearance Due to the fact the program has to run in the command line, the overall appearance of the program is very simplistic. Using underscores to represent empty tiles and lowercase "b" and "w" represent singular black and white Draughts respectively, with capital "B" and "W" representing King Draughts.

Moving One aspect of the design that worked as intended was the movement, as shown in the screenshot. Draughts would move in the diagonals and were limited by error handling the rules of English Draughts what limited where the pieces could move to.

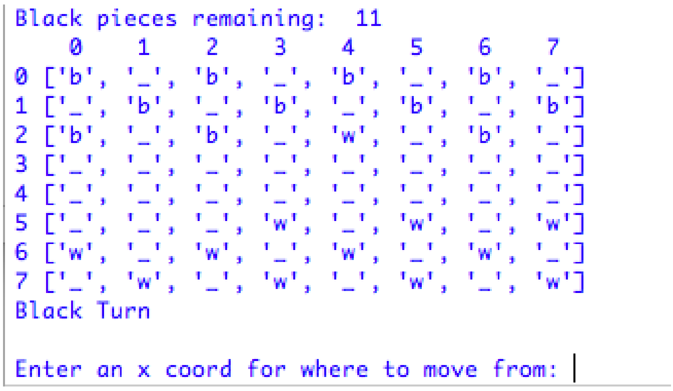


```
Enter an x coord for where to move from: 1
Enter a y coord for where to move from: 5
Enter an x coord for where to move to: 2
Enter a y coord for where to move to: 4
    0    1    2    3    4    5    6    7
0 ['b', '_', 'b', '_', 'b', '_', 'b', '_']
1 ['_', 'b', '_', 'b', '_', 'b', '_', 'b']
2 ['b', '_', 'b', '_', 'b', '_', 'b', '_']
3 ['_', '_', '_', '_', '_', '_', '_', '_']
4 ['_', '_', '_', '_', '_', '_', '_', '_']
5 ['_', 'w', '_', 'w', '_', 'w', '_', 'w']
6 ['w', '_', 'w', '_', 'w', '_', 'w', '_']
7 ['_', 'w', '_', 'w', '_', 'w', '_', 'w']
Black Turn

Enter an x coord for where to move from: 4
Enter a y coord for where to move from: 2
Enter an x coord for where to move to: 3
Enter a y coord for where to move to: 3
    0    1    2    3    4    5    6    7
0 ['b', '_', 'b', '_', 'b', '_', 'b', '_']
1 ['_', 'b', '_', 'b', '_', 'b', '_', 'b']
2 ['b', '_', 'b', '_', '_', '_', 'b', '_']
3 ['_', '_', '_', 'b', '_', '_', '_', '_']
4 ['_', '_', 'w', '_', '_', '_', '_', '_']
5 ['_', '_', '_', 'w', '_', 'w', '_', 'w']
6 ['w', '_', 'w', '_', 'w', '_', 'w', '_']
7 ['_', 'w', '_', 'w', '_', 'w', '_', 'w']
White Turn
```

Figure 2: **Moving.png** - A screenshot of the movement working

Capturing Another aspect of the design what was implemented in the Draughts game was the capturing and taking of pieces. This feature also had additional error handling implemented, as to prevent players from capturing their own pieces.

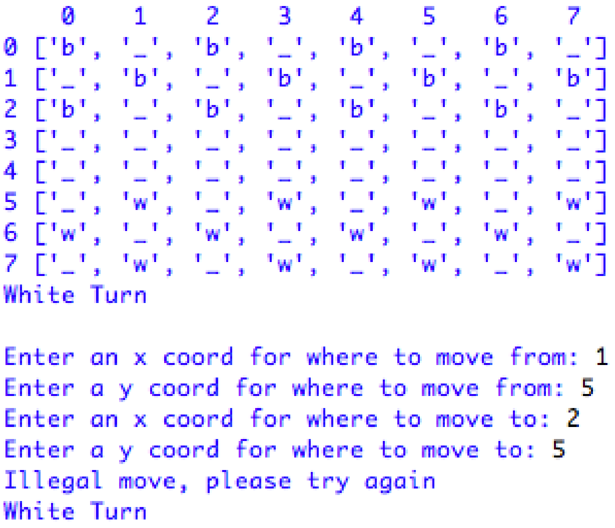


```
Black pieces remaining:  11
    0    1    2    3    4    5    6    7
0 ['b', '_', 'b', '_', 'b', '_', 'b', '_']
1 ['_', 'b', '_', 'b', '_', 'b', '_', 'b']
2 ['b', '_', 'b', '_', 'w', '_', 'b', '_']
3 ['_', '_', '_', '_', '_', '_', '_', '_']
4 ['_', '_', '_', '_', '_', '_', '_', '_']
5 ['_', '_', '_', 'w', '_', 'w', '_', 'w']
6 ['w', '_', 'w', '_', 'w', '_', 'w', '_']
7 ['_', 'w', '_', 'w', '_', 'w', '_', 'w']
Black Turn

Enter an x coord for where to move from: 
```

Figure 3: **Capturing.png** - A screenshot of the capturing function

Error Handling Another feature that was included in the program was error handling. This prevented the Draughts from moving solely on a horizontal or vertical axis. It also prevent the Draughts from moving too far forward, or backwards. However, it did allow for the Kings to move as intented by the rules.



```
    0    1    2    3    4    5    6    7
0 ['b', '_', 'b', '_', 'b', '_', 'b', '_']
1 ['_', 'b', '_', 'b', '_', 'b', '_', 'b']
2 ['b', '_', 'b', '_', 'b', '_', 'b', '_']
3 ['_', '_', '_', '_', '_', '_', '_', '_']
4 ['_', '_', '_', '_', '_', '_', '_', '_']
5 ['_', 'w', '_', 'w', '_', 'w', '_', 'w']
6 ['w', '_', 'w', '_', 'w', '_', 'w', '_']
7 ['_', 'w', '_', 'w', '_', 'w', '_', 'w']
White Turn

Enter an x coord for where to move from: 1
Enter a y coord for where to move from: 5
Enter an x coord for where to move to: 2
Enter a y coord for where to move to: 5
Illegal move, please try again
White Turn
```

Figure 4: **Error Handling** - A screenshot showing the error handlings

3 Enhancements

Some of the features that would have been implemented if there was more time available to work on the project are undo/redo, recording of play and AI.

Undo/Redo If there was more time available to work to work on the project would have been to get the undo/redo function working as intended. This is because during the development of the program, the undo/redo function would not run correctly, sometimes it would fail to change the positioning of pieces or would change between white and black turns unintentionally.

3.1 Code Listing

Listing 1: The attempted Undo Function in Python

```
#Asks the user if they would like to undo the move
#      undo_answer =  print(input("Would you like to undo this move? Y/N: "))

      #Starts the undo function, if yes is selected
     if undo_answer == y or undo_answer == Y:
         grid[start_y][start_x] = WD
         grid[end_y][grid_x] = EC
         print_board(grid)
         return move(value_package, grid, wpc, bpc)

      #Starts the undo function, if no is selected
     if undo_answer == n or undo_answer == N:
     return move(value_package, grid, wpc, bpc)
```

Recording of Play Another feature that would have been implemented if there was more time would have been the recording of play. This would have allowed the player to save games and replay them at a later time, however I was unable to implement this feature.

AI The final feature that would have been implemented if there was more time is AI. A simple AI would have been implemented that would have allowed for the player to play against it, or to have the AI play against itself.

4 Critical Evaluation

Data Structures The use of Data Structures in the Draughts game is definitely the weakest aspect, and despite being functional and working as intended, it definitely could use improvement. The Data Structures in the current program are rather bare bones and could use improvement.

Python Python was well implemented in the development of the Draughts game, however, a different language, for example, Java may have been better suited to the development of the Draughts game.

5 Personal Evaluation

During the course of this coursework, my grasp of python have improved. I faced a few challenges during my efforts to create a functioning Draughts game, like learning how to adapt the information I learned in the labs and lectures to the coursework. Overall, I feel like I have performed well but should have spent more time on the coursework and more time should have been spent on it when it was first issued, however, due to circumstances that were outwith my control, finding time to work on the project was rather difficult.