Keith Feeney

15015556 – Introduction to AI – 2018/9 –BSHCSD4

CA3 Document

STRUCTURES FORKNOWLEDGE REPRESENTATION & LOGICALREASONING SYSTEM

Contents

[Introduction 2](#_Toc530043365)

[Knowledge Representation 2](#_Toc530043366)

[Logical Reasoning 2](#_Toc530043367)

[Conclusion 3](#_Toc530043368)

# Introduction

In this report, we will look at the knowledge representation of the AI in chess games. We will look at what the Square and Stack classes are used for in the base coding provided on Moodle, as well as possible uses for the Move class.

After this, we will discuss three Artificial Intelligence strategies that can be used in creating a Chess game. The first strategy discusses if a piece is moved at random, and is a valid move. The second strategy discussed using a tree search with Minimax and how it would work. The third and final strategy discusses the Move Visualisation, where the Artificial Intelligence would map out which moves it would be able to take, provided it’s a valid move.

# Knowledge Representation

In the base code for Part 2, there are Square and Move classes. The Square class is used to get the X, Y and the piece name of one square on the board. If there is no piece present on the square on the board, the piece name remains blank. The Move class is used with the Square class and is used to get the starting square on the board and the landing square on the board, when a piece is moved. Stack is used with both Square and Move. Stack contains the amount of moves a piece can make and where a piece is in the way. For example, a Stack would be when the Queen piece can move anywhere on the board. If the Queen wants to go to a Square, and there’s a piece in the way, it will not be a valid move.  
  
The above is one way of how Move could be used, but another way could be Move Visualisation. What this does is it looks at all possible moves that a piece can make. This is done in the base code for CA4, where the square is outlined in green for where the piece can move to. If the square is outlined in red, this is because it cannot or should not move to that square, possible due to a piece already present in that location. (Hartikka, 2017)

# Logical Reasoning

For AI1, a method of logical reasoning for this AI would be that the AI would move a piece at random. This would be a control test to see if random moves would actually be logically viable. This would move a valid piece[[1]](#footnote-1) a random number of places in a valid move, where possible. It would be suspected that the AI would not be too challenging and would be defeated quite quickly. If a human were to play a Chess-master, without knowing the game sufficiently. (fryedk, et al., 2011)

For AI2, the way the AI would play using a search tree and the Minimax algorithm. The Minimax algorithm is used to decide the best move for the AI. There are two users in Minimax, the maximiser and minimiser. The maximiser tries to get the highest score while the minimiser does the opposite. (Hartikka, 2017) So, in our game of chess, we can say that the white player (the AI) is going to be the maximiser, while the black player (human) is the minimser. Each piece on the board will be given a numeric score with the pawns having the less impact (so, in the case of the white pieces, these could be a score of 10) and the King having the most impact (again, with the white pieces, this could have a score of 1000).

For AI3, Move Visualisation would be a good strategy to work with. Move Visualisation would be able to see all the available moves a piece could make. The way it would move would be to look at the possible moves and see which one is avoiding capture but also able to capture an opponent’s piece. This would be decided on how many of the AI pieces are left. The higher the number, to more risk it would be able to take. But with less pieces, the less risk it would take and it would move away from other pieces more. (Hartikka, 2017)

# Conclusion

This report looked at the knowledge representation of the AI in chess games. It discussed what Square and Stack classes are as well at looked at the Move classes where multiple uses could be completed.

Next, it looked at some Artificial Strategies that would be able to be used when creating an AI player in Chess. The first one discussed moving pieces at random, where it would be a valid move. The next strategy looked at using Minimax and tree searching for the AI to play. The final strategy looked at using the Move Visualisation, where the AI player could look at all possible valid moves it could take.

# References

fryedk, DarkArmy60, Rvfvs, HGMuller, et al. 2011. *This is what happens when you play random moves.* [Online]   
Available at: https://www.chess.com/forum/view/general/this-is-what-happens-when-you-play-random-moves  
[Accessed 15 November 2018].

Hartikka, L., 2017. *A step-by-step guide to building a simple chess AI.* [Online]   
Available at: https://medium.freecodecamp.org/simple-chess-ai-step-by-step-1d55a9266977  
[Accessed 15 November 2018].

1. “Valid piece” means that for example at the start of the game, only a pawn can be moved. The AI must select a pawn, but would do so at random. If, for example, a bishop was enclosed by pieces, it would not be a valid piece [↑](#footnote-ref-1)