Jason Agbobli – 36802020

Kwabena Koranteng Obeng – 60442022

**Data Structures Final Project Report**

**Game AI: Tic-Tac-Toe**

**Introduction**

This game AI will be implemented using a game tree for the popular game tic-tac-toe or knots and crosses. This game is normally referred to as ‘x and o’ in Ghana. Tic-tac-toe is a common strategy game that can be played against an AI or a human opponent. This game has been around for a long time and can be implemented using a game tree. A game tree is a directed graph whose nodes are positions and whose edges are moves in a game. This game is an example of a zero-sum game with two players in which one player gains as a result of the other’s losses. This problem is interesting because it helps build the player’s cognitive skills and helps to think on your feet. This helps improve your reaction time.

**Background**

This game tree will be implemented using the minimax algorithm and searched using breadth-first traversal.

Breadth-first traversal traverses a graph in a breadth ward motion and uses a queue to remember to get the next vertex to start a search, when a dead end occurs in any iteration(“Data Structure - Breadth First Traversal - Tutorialspoint,” n.d.). It is often performed because a computer is unable to explore a complete game tree in a limited amount of time. The computer, therefore, considers all moves, then responses to those moves, going as deep as computational time allows(“Data Structures and Algorithms in Java, 6th Edition [Book],” 2014).

The minimax algorithm that tells you which move to make in a game. It looks forward through the game tree, looking for the best strategy to win the game. It was originally formulated for two-player zero-sum game theory, which covers both cases where players make moves in turns and those where they make moves simultaneously(“Minimax,” 2019).

**Approach**

**Key modules**

Obtaining and defining the coordinates:

We defined the coordinates of the board namely, X and Y coordinates by creating a new class called ‘Coordinates’. This will give us the coordinates we will use to enter values.

Having defined the coordinates, there are nine titles on a tic-tac-toe board which we created using a board class. Thus, the filling of the squares is based on the coordinates and values entered.

Create a board:

Made of nine titles using a multi-dimensional array which was 3x3:

We also defined a class called ‘Board’. The board has nine tiles which is made by using a multi-dimensional array of length and width of 3 and will take input and helps us to determine whether either X or O has won. We created a scanner object to take input from user. We defined various methods to take input, play at a particular coordinate, display free or available coordinates and display the board to the user.

A game tree was used because it can list and predict the possible outcomes to inform our AI. An array-based list was used to store available coordinates on the board. A toString method was added to the coordinate class because the program was displaying addresses to the user when asking for available coordinates to play.

**References**

Data Structure - Breadth First Traversal - Tutorialspoint. (n.d.). Retrieved November 17, 2019, from https://www.tutorialspoint.com/data\_structures\_algorithms/breadth\_first\_traversal.htm

Data Structures and Algorithms in Java, 6th Edition [Book]. (n.d.). Retrieved November 17, 2019, from https://www.oreilly.com/library/view/data-structures-and/9781118771334/

Minimax. (2019). In *Wikipedia*. Retrieved from https://en.wikipedia.org/w/index.php?title=Minimax&oldid=925373360