Edrick Hansel Limantoro, Mikha Kelaya Wy, Roger Cashley

Muhammad Rausyan Fikri, S.Si, M.Eng, MIEEE, MAIAA.

Data Structure and Algorithm (COMP2424 – A)

November 25th, 2021

Final Project Proposal

Tic Tac Toe A.I. with Minimax Algorithm

1. Project Overview

The tic tac toe game is a famous simple game that is playable by two opposing players, one represents the character ‘X’ while the other ‘O’. With modern computing, players can now play the game against an “artificial intelligence”. The first iteration of the A.I. uses random slot selection and this makes the game “not challenging”, this gave birth to an algorithm called the minimax algorithm.

The minimax algorithm is a backtracking algorithm that optimizes the best next move for a player to take. Minimax only works for games with two players as each player will act as a minimizing player and a maximizing player. With tic tac toe, where there are only two players, minimax can be implemented for the computer to go against the opposing player. During the A.I.’s turn, minimax algorithm will generate and evaluate all possible move up to the depth given or when the algorithm reaches a solution that require less steps. These possible moves generated can be visualized and be travelled in a form of a tree data structure.

1. Project Structure

The project will be hosted on an online repository – GitHub, following a traditional C++ project structure:

/docs – Holds documents such as pictures, reports, videos, etc.

/includes - Contains all local include files (header files .h).

/lib – Contains the .cpp files for all the header files in the includes folder.

/src – Contains the main.cpp file for compiling and testing.

1. Implementation

This project (currently) has three main files, tictactoe.h, tictactoe.cpp, and main.cpp. This project will follow the object-oriented approach where the main file will only create an instance of the class and call the function explicitly.

tictactoe.h and tictactoe.cpp will hold the variables of the current instance of the game like the board, current player, current symbol, game mode, and game status.

main.cpp will be used for calling and testing the main program.

For the minimax algorithm function, we are still discussing whether to include the entire algorithm in the tictactoe.h or on a separate ai.h file.

Features such as storing the history of gameplay (using queue or linked list) or further optimization will be implemented if there is time.

1. References
2. GeeksforGeeks. (2021a, March 31). *Minimax Algorithm in Game Theory | Set 1 (Introduction)*. <https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/?ref=lbp>
3. GeeksforGeeks. (2021b, August 24). *Minimax Algorithm in Game Theory | Set 2 (Introduction to Evaluation Function)*. <https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-2-evaluation-function/>
4. GeeksforGeeks. (2021c, October 19). *Minimax Algorithm in Game Theory | Set 3 (Tic-Tac-Toe AI - Finding optimal move)*. <https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-3-tic-tac-toe-ai-finding-optimal-move/>
5. Lague, S. (2018, April 20). *Algorithms Explained – minimax and alpha-beta pruning*. YouTube. <https://www.youtube.com/watch?v=l-hh51ncgDI&feature=youtu.be>