Coursework Report

**Venetsia Krasteva**

[**40313507@live.napier.ac.uk**](mailto:40313507@live.napier.ac.uk)

**Edinburgh Napier University - Algorithms and Data Structures (SET08122)**

## Introduction

The task is to create a text-based Tic-Tac-Toe game using the C programming language. The focus should be on the data structures and algorithms needed to implement the game.

## Design

The software firstly opens a menu with switch.

In the Multiplayer option I have used stacks in order to track the movement so the function undo is enabled. The program pushes every move, which is not an undo function, into the stack and when undo is chosen it pops the move out of the stack. The board is a character array an each number represents a square on the board. Player one is CROSS and Player two is NOUGHT. The Multiplayer option runs on choices (values) made by the user. The history is stored in a file created by the user when he selects the option. A file is created and on every move, the value inputted, is added to the file. When game is done you can run the Replay option from the Menu and the user has to input the name of the history so it can be replayed.

In the Single Player vs AI the board is 25 squares. The program uses ENUMs to define NOUGHTS, CROSSES, BORDER, EMPTY, HUMANWIN, COMPWIN and DRAW. Enums limit you to the required set of inputs whereas even if you use constant strings you still can use other String not part of your logic. This helps the program not to make a mistake to enter something out of the domain. It also improved program readability. In order for us to play in 9 squares we convert the 25 to 9 specific. The program uses four directions so it can find if there are any three on a roll or on a cross. For the easy option the computers’ move is based on a random choice. In the Unbeatable option the program uses the MinMax algorithm to find the best move. It is used to find the optimal move for player. Every board state has a value associated with it. The function evaluates all the best available moves and returns the best one depending on if it’s the minimizer or the maximizer. The function recalls itself until it gets the best move and maximize the score.

In Replay Multiple player game if reads the file and every line is read as a different input so the choices are made.

## Enhancements

* Add a redo function to multiplayer.
* Add undo, redo and history for single player vs AI
* Think about different board shapes to play the game and try to implement them
* Add switch players

## Critical Evaluation

* Replay is working well on Multiplayer. – it is working as it should, it is replaying the game but did not work on Single Player vs AI
* Switching players is not added at all and was not working
* Undo function is working good but could not use the same method for Single Player

## Personal Evaluation

I have learned so much with this coursework. I have remembered stuff that I forgot from last year from all the language switching. Definitely feel more comfortable writing C language programs. Every difficulty I had was fixed by the help of online tutorials and YouTube videos. The most difficult one understood the MinMax algorithm. But it was very interesting. Had difficulties figuring out how to do the undo function as well. Went through the lectures and practicals and figure I could try Stack. Had problems with pointers and passing by value or with reference. Took me some time to figure it out. Had problems with taking the input from user in program to write to file so it could be used later as an input. Had different ideas how to do it but it all came down to have a file saved and loaded afterwards.

## References

<https://www.google.com/search?q=how+to+do+a+makefile+in+c&rlz=1C1GCEA_enBG768BG768&oq=how+to+do+a+makefile+in+c&aqs=chrome..69i57j0l5.5888j1j1&sourceid=chrome&ie=UTF-8>

<https://www.google.com/search?q=minimax+algorithm&rlz=1C1GCEA_enBG768BG768&oq=minmax+al&aqs=chrome.1.69i57j0l5.2807j1j4&sourceid=chrome&ie=UTF-8>

<https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/>

<https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-3-tic-tac-toe-ai-finding-optimal-move/>

<https://stackoverflow.com/questions/2229498/passing-by-reference-in-c>

<https://stackoverflow.com/questions/2425167/use-of-exit-function>

<https://medium.freecodecamp.org/how-to-make-your-tic-tac-toe-game-unbeatable-by-using-the-minimax-algorithm-9d690bad4b37>

<https://towardsdatascience.com/tic-tac-toe-creating-unbeatable-ai-with-minimax-algorithm-8af9e52c1e7d>

<https://hackernoon.com/i-created-an-ai-that-beats-me-at-tic-tac-toe-3ea6ba22cd71>