Purpose of files

Run.c contains the main method for the Tic Tac Toe game. Its purpose is to run the readSettings function that initialise the correct settings values for the game. If these are correctly read in the file then calls the user interface that plays the rest of the game and then frees the board.

CreateTable.c is responsible for reading the settings file and handling any error in the formatting or reading of this file. It sets the settings values to what is read from the file. This file is also responsible for freeing the board array.

UserInterace.c is in charge of dealing with the menu systems for the program. It deals with user input for the menu and passes all necessary data to the menu choices. It is also affected by the conditional formatting with SECRET and EDITOR being shown if they were compiled with the necessary flags.

GameFunc.c contains the bulk of the game logic. It is responsible for starting new games and running the methods to find if a game has ended and then printing the winner. This file also contains the method that prints the game board with the correct formatting. This file also has the newTurn function which reads the user input and inserts the player's move into the board if its fine to do so and then passes its information to the log function to add to the game logs.

WinCondition.c checks the board if to see if a player has won the game. It goes through horizontally, vertically, diagonally up and also diagonally down passing the array indexes to the checkWin function. The checkWin function takes in the index values and then counts how many in a row of each player there is. If this equals the number needed to win, the method outputs game end as true which stops the game.

LogFunc.c contains all of the methods that deal with the game logs and any additional linked list functions. This file deals with inserting a Game Entry struct into the log linked list with all of the data from the turns. It also contains functions that free the logs linked list. This file also deals with printing the linked list data in order to see the turns. One of the other functions of this file is to print the game data to a file. This means printing the current settings and all of the log data to a file which filename is dependent on the current time.

LinkedList.c is a generic file. This file contains methods that allow the abstract data type to exist. It contains all create, insert and remove functions that allow the adding of generic data to it and also contains methods that take in a function for printing and freeing that are specialised to the data that is stored in the linked list.

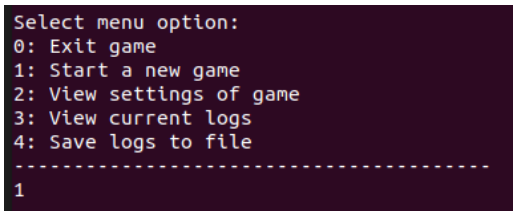
Implementation of the logs

My log file system was quite simple. I had a linked list that contained the game number, turn number, player and the co-ordinates. The linked list is generic so I had a file (LogFunc.c) that contained the dedicated functions that dealt with the data once it came out of the linked list. There is a function that printed out the turn data that was stored in the linked list, when printing if the turn count is equal to 1 a new game has started so it will then print the game number before the turns. Also one for freeing the struct in the linked list and also one that printed the linked list data to a file.

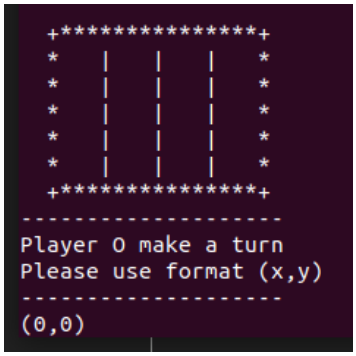
One of the problems I encountered while creating this was originally I had a linked list of linked lists. One linked list contained all of the games and the other had all of the turns with a game. I had some issues with freeing and decided to go for an approach that only needed one linked list. With this approach I had an issue with knowing when another game started so I added it to the struct which made it easy to print and write to a file.

How to use the program

To use this program you execute ./TicTacToe settings.txt (or a different filename)

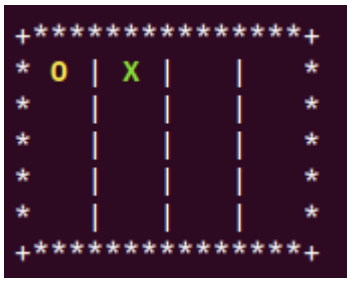


After running the program the following should appear if successful



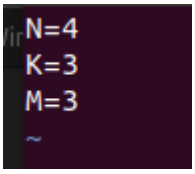
If you choose 1 (new game) you will be greeted with the board (see left) and to play a turn.

(0,0) will appear at the top right as shown in the image on the left.



(1,0) will result in the follow (on right).

The input file (settings file) should be formatted as the following. Where N is the height of the board, M is the width of the board and K is the number in a row in order to win.



The output file should look like the following. When a new game is created it will be under a new game hierarchy. Also note the file name is formatted with “MNK_<M>-<N>-<K>_<HOUR>-<MIN>_<DAY>-<MONTH>.log” (found at bottom of screenshot.