PROJECT 02 – TICTACTOE GAME

1 Introduction

This assignment involves implementing a popular game called TicTacToe, also known as Caro. It offers a direct chance for head-to-head competition between 2 players "x" and "o" who takes turns marking the spaces in a 3x3 grid (which can be extended later). The winner is the first one who places three of their marks in a diagonal, horizontal or vertical row before another could do the same. The game will be a forced draw/tie if none of them can succeed before the grid becomes full-filled. A running example can be found in google page with "tictactoe" as the searching term.

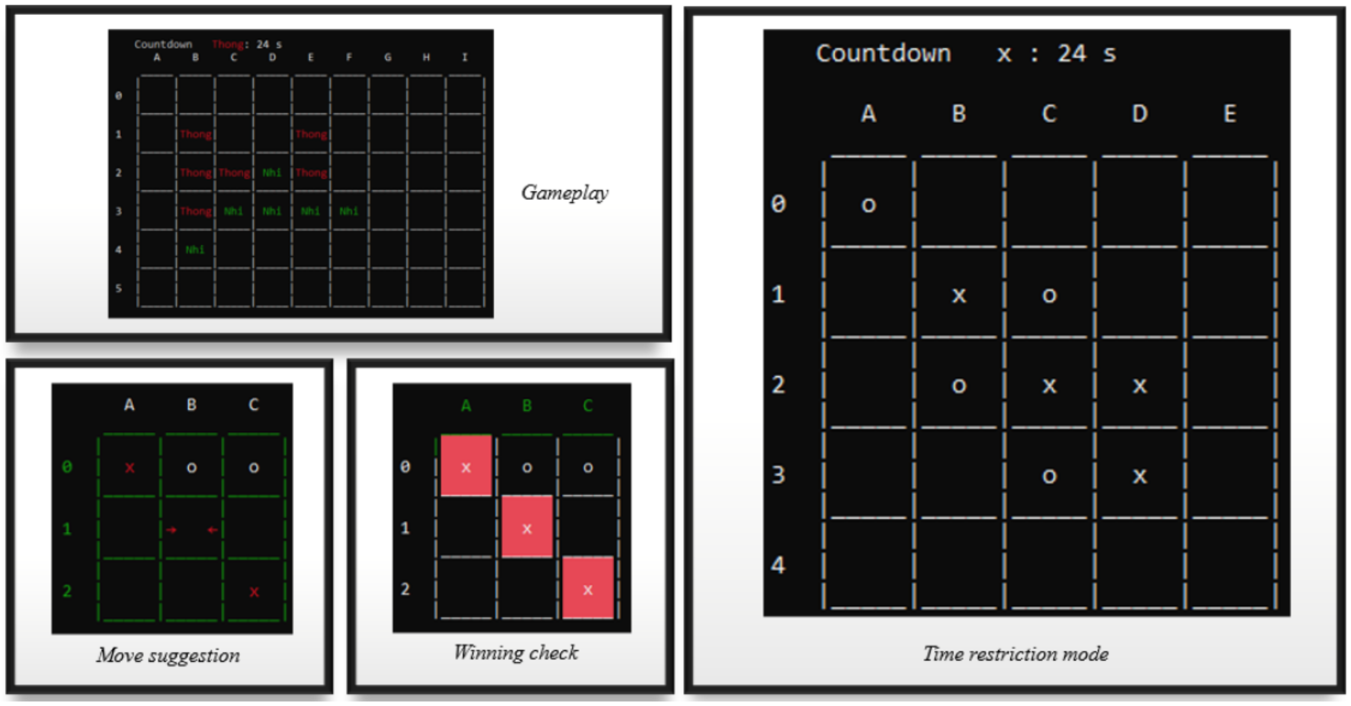

Figure 1: The Tictactoe game

2 Requirements

2.1 Pro This section is an upgraded version of the basic TicTacToe game. It requires the array theory in both single and multi dimension to design a better and well-structured program. The result has to consist of user multiple choice for grid (3x3, 5x5, 7x7) as well as the statistical outcome.

Requirements:

- Grid represented using 2-dimension char array.
- Print board status (after every moves).
- Switch between players after each move.
- Check if any of players is the winner in three conditions: diagonal, horizontal or vertical row; or the forced tie/draw match
- Check if the current move is available and acceptable.
- Print the statictical outcome after one match. The outcome should involve: result of the match, the number of moves, etc.
- Game Setting:
  - User can choose the grid size grid size (3x3, 5x5, 7x7).
  - `[Optional]` Change the color of the board background.
  - `[Optional]` User can change the player icons (default are ‘x’ and ‘o’)
  - `[Optional]` Turn off/on sound and background music

2.2 Expert In this section, you are required to implement a full-option of TicTacToe game using data structure `struct` and `file`. That means you will again write another program that satisfies requirements a to h from Pro part above (i and j are also welcome) following some specific rules below:

- Redefine the board grid using `struct`.
- Rewrite all checking conditions for winner based on the new grid.
- Account creation, account deletion, game log in, achievements, etc. for individual players.
- Design a cursor for players to move around the grid using the keyboard.
- Save game / Move replay.
- Move suggestion.
- Time restriction mode.
- Player vs Computer mode.

3 How to submit

You will do all of your work in a single executed code, called `tictactoe.cpp`. In addition, you are required to make a report in what you have done. So, your submission **ID.zip**, which is uploaded throught Moodle, need to contain the following files:

- A `tictactoe.cpp` code file
- A `tictactoe.pdf` report

  2 files above must be zip in **ID.zip** (all other formats are not allowed)

The Report Your report should include all information below:

a) Your name, ID and class

b) Your project structure

c) Any other remarks about your design and implementation

d) All links and books related to your work must be mentioned. If you discuss with your classmates or high-level mentor, their name should be listed too

e) **DO NOT** insert you source code in the report

Assessment

- Submission with wrong regulation will result in 0 point
- Your program need to be operable or another 0 point
- Plagiarism and Cheating will result in a 0 point for the entire course and will be subject to appropriate referral to the Management Board for further action