14 May

NWEN241

**main task will be to create a menu to the games.**

**Task 1**

create a menu which will interconnect all the games you previously created (guess my number, tic-tac-toe and K game).

do not put any requirements on the format of the menu itself

following criteria:

* When you run the program, the main menu should be showed to the user with a welcome note and some instructions/options how to use it.
* Except the options to step into the specific game menu, there should be an option to end the program.
* For navigation, either the letters of the alphabet (e.g. A, B, C, etc.) or numbers (e.g. 1, 2, 3, etc.) should be used. When **other characters are used for navigation (e.g. white space, tabular, !, ?, $, etc.), the program should handle such a situation without any problem while the user should be asked to re-enter the choice**.
* In the specific game menu (e.g. tic-tac-toe), the user should have the options:

1. to run the game and
2. to exit (back) into the main menu.

* Each working game (either won or lost) should ask the user to *start the game from the beginning or end it.* When ‘end the game’ was selected, the program should return to the menu of the game (not the main menu).
* *we are going to evaluate only the menu and how you connect the games using it.*

**Task 2 (50 marks) – Multi File Programming**

1. Your task is to ensure *multi file programming style* while you complete this assignment.
2. **You will *achieve this with the appropriate makefile.***
3. Each game should be in one source file (**with the associated header file**)
4. *all the source (code) together with the source file for the menu (containing* ***main()****)*
5. should compile into **ONE** binary executable
6. which will contain the menu with all the games.
7. You should make sure to include the right header files (if necessary).
8. You should also avoid circular includes (note that included header files often include other header files).

**Challenge (20 marks)**

* Your challenge is to extend all the games with a function for saving the actual state.
* The game state should be saved into separate files which will be created (if not existing) in the folder from *where the program is started.*
* If the files already exist, you should overwrite them with new contents.