



INF212

ALGORITHMS AND PROGRAMMING

PROJECT-2

Tic-Tac-Toe Game

The deadline is Tuesday, April 16, 2024, at 23:59.

Projects that are not delivered on time are not accepted.

Upload the project to the Project 2 assignment section of the INF212 class.

The questions can be asked to course lecturer Doç. Dr. Tuba GÖZEL and teaching assistant Murat Enes HATİPOĞLU.

Upload your files as followings:

- After uploading your report and source files, you must use **TURN IN** button (**GÖNDER** butonu) in the system, otherwise it will be a **draft** of your version and will not be accepted.
- MS Teams "**INF212**" class is the directory for your files.
- Do not create a folder or RAR or whatever, just **UPLOAD your source codes**.
- Report (*.docx) and source files (all code as C file *.c) must be named as "INF212_P2_ID**XXX**.docx" and "INF212_P2_ID**XXX**_main.c, respectively.

PROJECT 2 – Tic-Tac-Toe Game

Tic Tac Toe is a classic two-player game played on a 3x3 grid. Players take turns marking the cells with their chosen symbol, typically X's or O's, until one player forms a line of three symbols in a row, either horizontally, vertically, or diagonally.

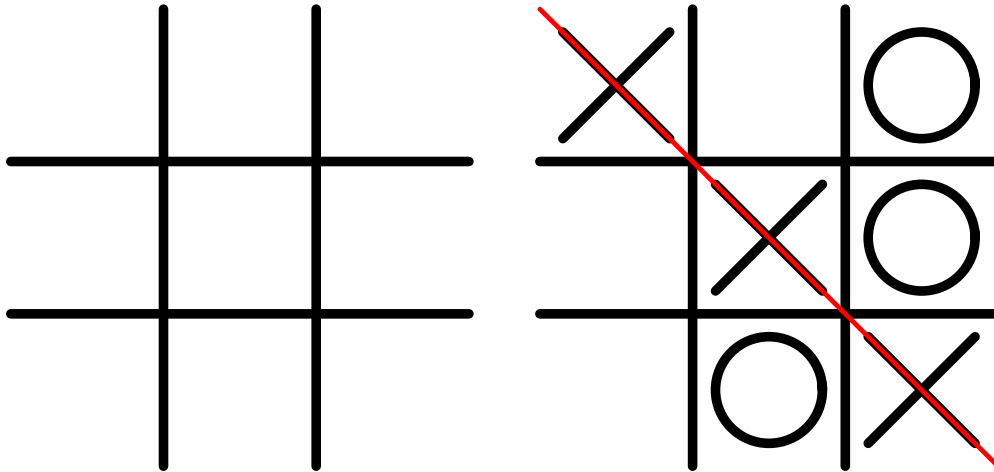


Figure 1: The Tic-Tac-Toe Game: Empty Game Board (on left), A winning situation (on right)

Write a C program that plays the Tic-Tac-Toe game. The program should allow two players to take turns placing their chosen symbol, either X's or O's, on a **3x3 grid and 9x9 grid**.

The program should prompt the user to enter the grid position they want to place their symbol and update the board accordingly.

That is, players must enter specific numbers that correspond to the position on the grid, and the program should prompt them for their move. The first player should use circles, while the second should use crosses. After each successful move, the program should alternate the players.

The program should update the grid after each move and to determine the winner, check if any player has formed a line of three symbols in any direction. If a player has won, the program should announce the winner and end the game.

As part of this project, you should practice programming fundamentals such as user input, conditionals, loops, and arrays. You can also add features such as a menu screen, a scoreboard, or an option to play against the computer.

Overall, this assignment will provide you with a fun and challenging opportunity to practice their C programming skills while also creating a classic game enjoyed by people of all ages.

Hints and Instructions:

To write a Tic-Tac-Toe game, you will need basic functions such as the following. These functions will form the cornerstones of your game.

`void initialize_Game_Board(...)` : This function ***initializes*** the Tic-Tac-Toe game board for ***a 3x3 grid and 9x9 grid, whatever grid size is desired***, as seen below. (The example below is for 3x3.)

```

      1   |   2   |   3
      -----
      4   |   5   |   6
      -----
      7   |   8   |   9
  
```

`int update_Game_Board(...)` : This function is responsible for ***modifying*** the cells on the board. If an invalid position is entered, a ***warning message*** will be displayed.

Example: Update the third cell in the given board on player 2 move.

```

      |   |
      -----
      | O |
      -----
      |   |
  
```

Player 1 Move

```

      |   | X
      -----
      | O |
      -----
      |   | .
  
```

Player 2 move, update the
cell 3 with X

`void show_Game_Board(...)` : This function is used to ***print*** the game board after every play of each player.

```

      O |   |
      -----
      |   |
      -----
      |   |
  
```

Player 1's turn

```

      O |   |
      -----
      | X |
      -----
      |   |
  
```

Player 2's turn

`int check_for_Win(...)`: This function used to **find the winner** of the game. It checks on each row, column, and diagonal. If the same sign is present in any one of these places, then **print the winner**.

`void start_Game(...)` : This function **starts** the Tic-Tac-Toe game for 3x3 and 9x9 game board.

The above mandatory functions are function prototypes above `int main()` in `main.c` file. You should implement these functions below `int main()` in `main.c` file and use them in it `main()` in `main.c` file. Function arguments are not given to you as a hint, you must decide what they should be. You can also create additional functions if you need.

The project you create should be able to play tic tac toe game on a 3x3 or 9x9 board according to the user's choice. If you only do one of them, you will receive half of the total points.

Also, you must use all the structures you have seen in the lab until the 2nd project. It is also mandatory to use pointers when dealing with arrays, otherwise you will lose points.

Besides, if you want to improve your game even more, you can use the following additional functions (If you want or can do it, **optional**.)

`void Menu(...)`

`void show_Game_Rules(...)`

`void write_Score_To_File(...)`

`void show_ScoreBoard(...)`

You can use “switch case” or “if” for `void Menu` function.

`void show_Game_Rules` function should print the rules of the game to the terminal. You can write the game rules in a text file, read this text file with File I/O operation and print it to the terminal. You can set a score for each game win and in the `void write_Score_To_File` function, print the winner's score to a file after each game using File I/O operation. Then you can print the contents of this file where you save the scores to the terminal with the help of the `void show_ScoreBoard` function.

Remember that File IO operations are the topic for the following weeks; so, you can try to do this without using File/IO (maybe by storing the score and the rule string in arrays.) However, you can create the menu function with the knowledge you have learnt.

These optional tasks will give you a bonus if you have minor imperfections in your working code.

As above, function arguments are not given to you as a hint, you must decide what they should be.

Also, you must use all the structures you have seen in the lab until the 2nd project. It is also mandatory to use pointers when dealing with arrays, otherwise you will lose points.

Important notice 1: You can use examples from the internet to help you complete your project, but please **avoid plagiarism. Projects found to be an exact copy of another student's work or source code on internet or directly copy of AI generated source code will be given a grade of 0. The necessary disciplinary action defined in the regulations will be taken for the owner of this homework.**

Important notice 2: Please include comment lines to explain what you are doing in your code.

Example:

```
/*  
  
This code has written by Mahmut Tuncer  
  
This function does this and aims to do that using this.  
  
*/  
  
or  
  
// this loop does this work  
  
// this is defined for this work.
```

Instructions for submission

- *Do the project task in a single .c file (main.c file). You can use as many functions as you want.*
- *Before you upload your task to the relevant assignment, please pay attention to these file naming conventions:*

*For main: INF212_P2_IDXXX_main.c
Example: INF212_P2_121024005065_main.c*

- *If your Student ID has lower than 12 digit, please add zero to left side of your Student ID until reaching 12 digits.*

Example: INF212_P2_000171053011_main.c

- *Prepare your project report in .docx format according to the template we have shared with you.*
- *Before you upload your report the relevant assignment, please pay attention to these file naming conventions:*

*INF212_P2_IDXXX.docx
INF212_P2_121024005065.docx*

You must upload 2 files, .c and .docx files, to the relevant assignment.

Good Luck!