FT_TTT

Network, TCP/IP, Protocol, Game, and some other basic stuff

Summary:

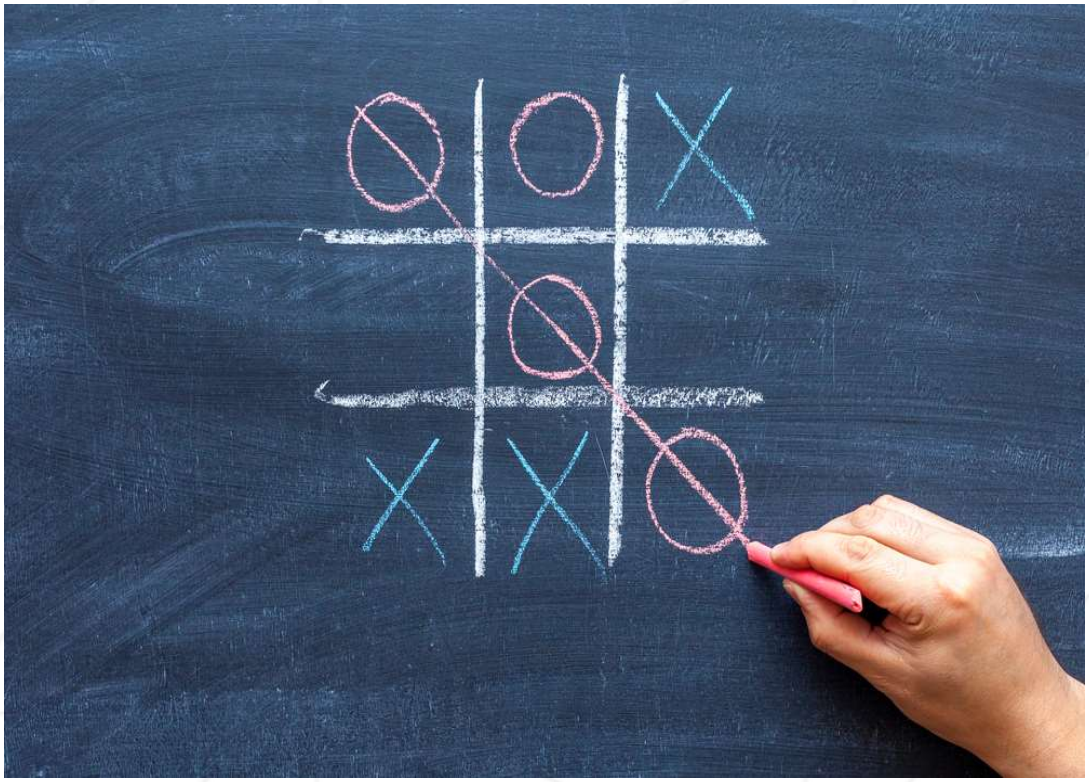
This project is about creating your game client, server.

Version: 0

Contents

Chapter I

Introduction



Tic-tac-toe (American English) is a paper-and-pencil game for two players who take turns marking the spaces in a three-by-three grid with X or O.
The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner.
It is a solved game, with a forced draw assuming best play from both players.

Chapter II

Common Instructions

• Your functions should not quit unexpectedly (segmentation fault, bus error, double free, etc) apart from undefined behaviors. If this happens, your project will be considered non functional and will receive a 0 during the evaluation.

• All heap allocated memory space must be properly freed when necessary. No leaks will be tolerated.

• If the subject requires it, you must submit a Makefile which will compile your source files to the required output with the flags -Wall, -Wextra and -Werror, use cc, and your Makefile must not relink.

• Your Makefile must at least contain the rules $(NAME), all, clean, fclean and re.

• To turn in bonuses to your project, you must include a rule bonus to your Makefile, which will add all the various headers, librairies or functions that are forbidden on the main part of the project. Bonuses must be in a different file _bonus.{c/h}. Mandatory and bonus part evaluation is done separately.

• If your project allows you to use your libft, you must copy its sources and its associated Makefile in a libft folder with its associated Makefile. Your project's Makefile must compile the library by using its Makefile, then compile the project.

• We encourage you to create test programs for your project even though this work won't have to be submitted and won't be graded. It will give you a chance to easily test your work and your peers' work. You will find those tests especially useful during your defence. Indeed, during defence, you are free to use your tests and/or the tests of the peer you are evaluating.

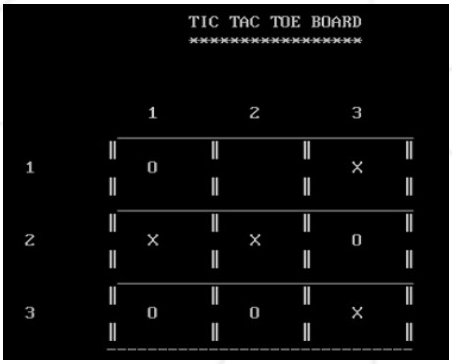
• Submit your work to your assigned git repository. Only the work in the git repository will be graded. If Deepthought is assigned to grade your work, it will be done after your peer-evaluations. If an error happens in any section of your work during Deepthought's grading, the evaluation will stop.

Chapter III

Exercise 00: I still love Tik-Tac-Toe

Exercise : 00
I still love Tik-Tac-Toe
Turn-in directory : ex00/
Files to turn in : Makefile, main.c, *.h , *.c
Forbidden functions : None

- You're going to make a simple tic-tac-toe in this exercise.
- Players must have O and X.
- O and X take turns starting the game.
- The position of x and y of the player you want to place must be in the range of 1 to 3.
- Out-of-range values cause errors and require re-entry.
- No player can be placed back in a position that has already been placed.
- If the same player is placed horizontally, vertically, diagonally, the game ends with the victory of the game.
- The input is shaped like this:
<player X pos, player Y pos>
- This is a simple Tik-Tac-Toe:



Chapter IV

Exercise 01: FT_TTT

Exercise : 01
FT_TTT
Turn-in directory : ex01/
Files to turn in : Makefile, main.c, *.h , *.c
Forbidden functions : None

It will implement a game server that handles remote play roles through pre-supplied "Tic Tac Toe" clients and defined protocols.

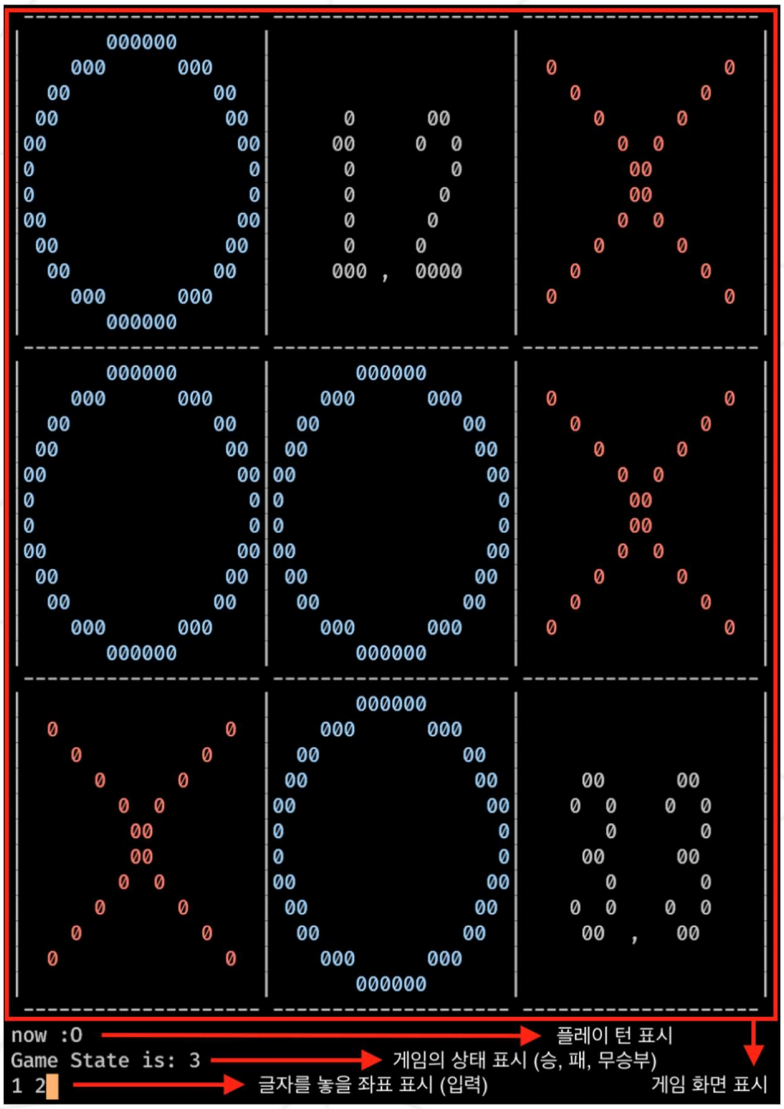
By configuring logic such as client access, game progress, and result processing through network communication and experiencing problems that may arise from network communication, you can experience the overall flow of network programming.

- What is a socket?
  - ✓ learn how to communicate
  - ✓ learn about socket event

- Server-Client Protocol
  - ✓ "- - S" means Start protocol.
  - ✓ "- - E" means End protocol.
  - ✓ "- - R" means When placed in a position that already exists.
  - ✓ "? ? O" means O is placed in [y, x] coordinates. ('?' is a number of from 1 to 3)
  - ✓ "? ? X" means X is placed in [y, x] coordinates. ('?' is a number of from 1 to 3)
  - ✓ "- - 1" means Player 1(O) Win
  - ✓ "- - 2" means Player 2(X) Win
  - ✓ "- - D" means Draw

- When the game begins, which clients will be given priority?
- To what extent will the server handle things like game logic and processing results notifications?
- While waiting for client input (block), will the game server configure seamless game progression regardless of client?
- If more than three clients are connected, what will be done to the third client?
- If the opponent's client is cut off during the game, what will happen to the victory and client-server connection?

- Wow! The nice client than you think. You can use it.



- something to think about
  - ✓ If the protocol in the Application Layer is variable rather than fixed, what logic should be configured?
  - ✓ What network options should be selected depending on the type and genre of the game?
    - SO_REUSERADDR, Nagle Algoritm, SO_LINGER, SO_KEEPALIVE
  - ✓ What issues does the port's TIME_WAIT state create for the game server?