

Service Name and Motivation

The name came from inspiration of the classic "Snake" game, as once very popular on classic Nokia phones and game consoles. I wanted to reimplement the Snake game in terminal. Also, the name shorts to SIT to allude to the host university name (initially the map was supposed to be modelled after campus, but this was dropped due to lack of time).

Proposed system

The service is meant to allow collecting real-time user input from multiple client terminals, and modify server-side game state parameters.

As mentioned, inspired by Snake; However due to lack of time and manpower, no GUI - just status messages.

This still allows for remote control of server from many clients.

Used tools, source code

Please see `README.md` . Code was explained during presentation and in-code comments.

Key Features

- Server uses `fork()` , socket-handling and pipe to make multiple client connections possible
- Data sent between server and client in structures
- Utilizes `ncurses` .

Comments

What did I learn with this particular project

- Incorporated `Makefile` to automate testing.
- Automatic random port assignment.

- How to use ncurses to capture keyboard input.
- Transmitting C structures via send() and recv().
- Send gamestate between parent and children of server using pipes.
- How to use zlib's Adler32 checksum.
- How to timestamp structure. (These 2 were supposed to be checked to prevent cheating and calculate ping, but not enough time to actually implement that).

What did I learn in general on the course

- Network communication principles (don't remember last time I had class on TCP)
- Low-level file management
- Operating Systems POSIX C programming (which now sometimes confuses me with Embedded C).

Main difficulties

- Pipe logic (it's confusing at times)
- Synchronization of data and input (has to make sure client and server child process transmit in same order in both programs)
- Managing this project with other project and report deadlines, which is why I eventually failed to fix the code and prepare better documentation.