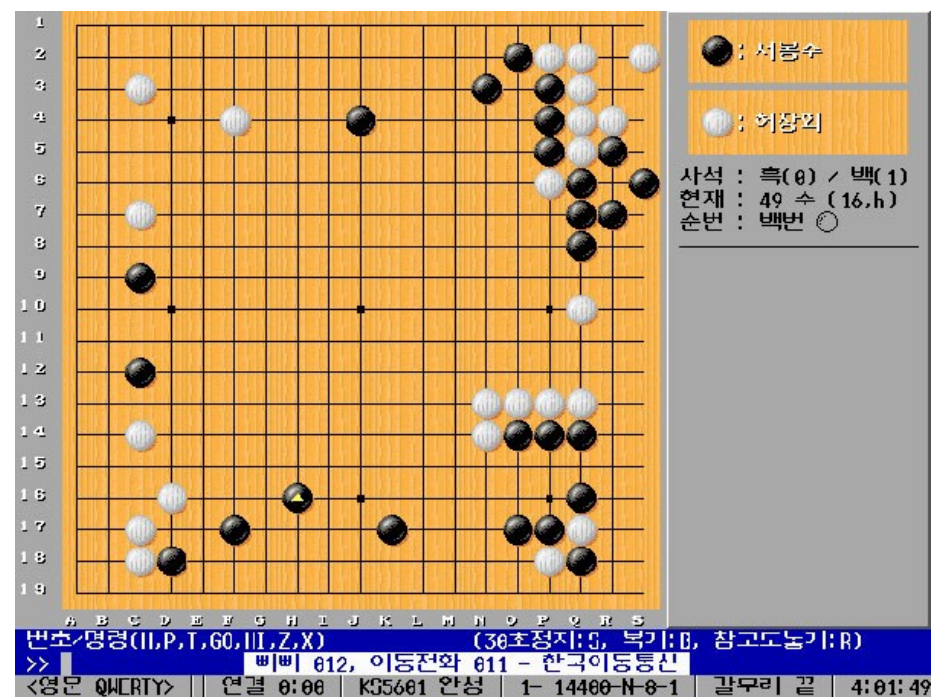
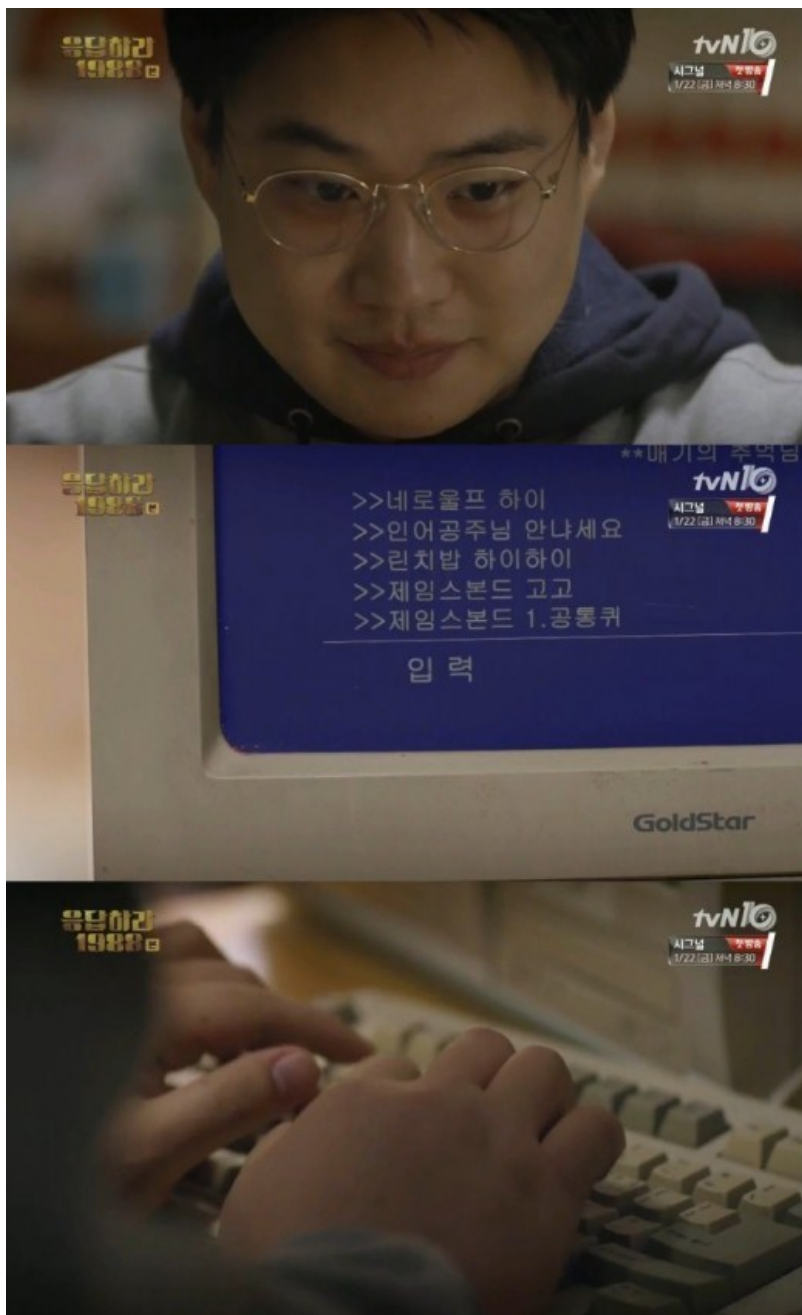


Open-source Software Lab

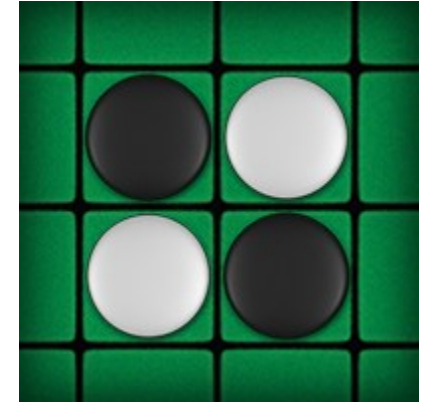
Project 3. Reversi

1 June 2023

Shin Hong



Overview



- Make a two-person network game of Reversi with text-based user interface
 - use the Ncurses library for TUI
 - use socket for network communication
- Follow programming requirements
 - write your program in multiple source code files
 - use Makefile
 - collaborate with GitHub
- Each team must submit the Github repo and a demo video by 9 PM, Mon 19 June
 - work on the repo while keeping it private, and turn the repo to public at the submission time

Reversi

- A turn-based board game with an 8x8 square board, and black and white stones
- Game rule <https://cardgames.io/reversi>
 - At the start of the game there are four pieces on the board, two white and two black.
 - Each player is to make a horizontal, vertical or diagonal line of pieces, where your pieces surround the other player's pieces. Then, the surrounded opponent pieces are will be flipped over to your color
 - You must try to capture opponent pieces and flip them over, so they turn into your color.

Background

- NCURSES for Text-based UI
 - <https://tldp.org/HOWTO/NCURSES-Programming-HOWTO/>
 - <https://github.com/hongshin/LearningC/tree/master/reversi/ncurses-examples>
- Simplest Chat Server-Client possibly found on this planet
 - <https://github.com/hongshin/LearningC/tree/master/reversi/socket-examples>

Program Design

- Your program must build a single executable `reversi` which can work as a server (White stone), or a client (Black stone) depending on command-line option
 - Server (White stone):
`$/reversi -server 8888`
 - Client (Black stone):
`$/reversi -client 127.0.0.1 8888`
- The player at the client-side first draws a Black stone, and then, the player at the server-side draws a White stone and take turns alternatively.

Programming Requirements

- Each team member must make more than 5 commits, more than 2 pull requests, and more than 2 issues in the GitHub repositories
- You must write commit messages properly
 - Commit message guideline
<https://gist.github.com/robertpainsi/b632364184e70900af4ab688decf6f53>
- Your program must be composed of two or more source code files
 - Makefile must be properly written to generate an executable from the source code files

Evaluation

- Your result will be evaluated according to the following criteria:
 - Text user interface must be valid and user-friendly
 - The game rules must be correctly policed
 - The communication protocol must be clearly explained in the video
 - The video clearly evidences that the program works correctly
 - The source code is clean and properly maintained using Git
- Good demo videos will be recognized

Instructions

- Open chat for Q&A: <https://open.kakao.com/o/gII0Vjbf>
- Demo video
 - No more than 5 minutes.
 - Each member must take a part in the video.
 - Upload your demo video to YouTube, and write down the URL on the submission message
 - You must use English in recording a video demo
 - Your demo video may be shared in the class, especially if it is to be recognized.
- Submission
 - HDLMS
 - One team, one submission
 - No late submission will be accepted