Duc Tri Dang

Dr. Crawshaw

COMP 4991

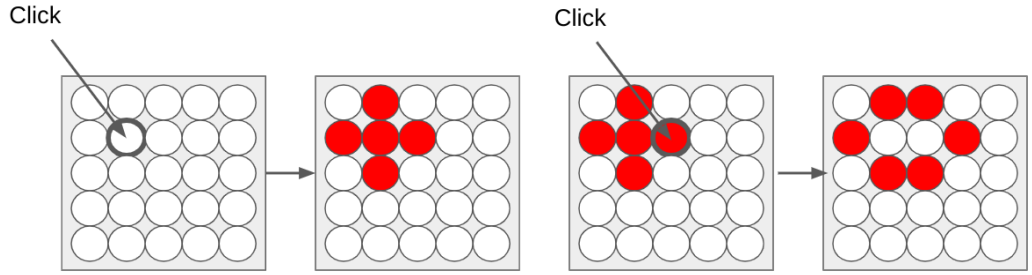
April 17, 2024

Computer Networking Final Project Design Document

1. **Introduction**

The topic was making a networked multi-user game using TCP.
The game I have chosen was inspired by a famous puzzle called Light Out Puzzle, then I have some changed on the idea of the game to make it more suitable to the context of the networked multi-user game as the final project of this course. The official name of the game is called Light Out Challenge.

The rules are simple. Each player would be given a board of 10x10 matrix full of lights, with 2 statuses, on and off. When a player adjusts any light, they switch the on/off status of that light and as well as its neighbours.



Picture Source: Baumstarck, P.G. "Programming Puzzle: Lights Toggle." *Medium*, The Startup, 5 Feb. 2023, medium.com/swlh/programming-puzzle-lights-toggle-f4d27bf3683e.

The mission of two players is work together to turn off out lights on the matrix. Note that, there is an interaction between the boards of two players, therefore, one player can be easily messed up the work of the other player and make the game more difficult to solve. More information will be provided in section 2.

2. **Product and instruction of usage**

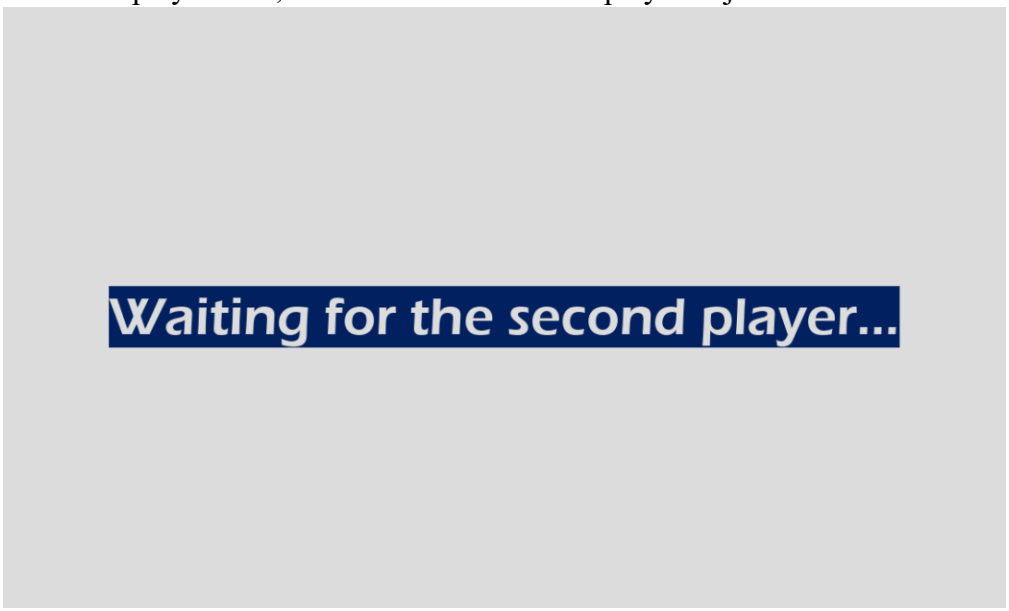
Step 1: Make sure that the using machine includes the *pygame* library. To install *pygame,* open your command prompt/terminal and type:
For MAC: *pip3 install pygame*
For Windows: *pip install pygame*
If it gives error for Windows, type in: py -3 -m pip install pygame

Step 2: To run the game, the user needs to run a server and two clients. This procedure can be done on either one machine or multiple machines as the user wants. Before running the game, the user should know the IP address of the machine that running the server. To get the IP address of a particular machine, type *ipconfig* on the command prompt of the machine running the server. After that, run the server and two clients.

For server: *python server.py IP_ADDRESS* or *python3 server.py IP_ADDRESS*
For client: *python client.py IP_ADDRESS* or *python3 client.py IP_ADDRESS*

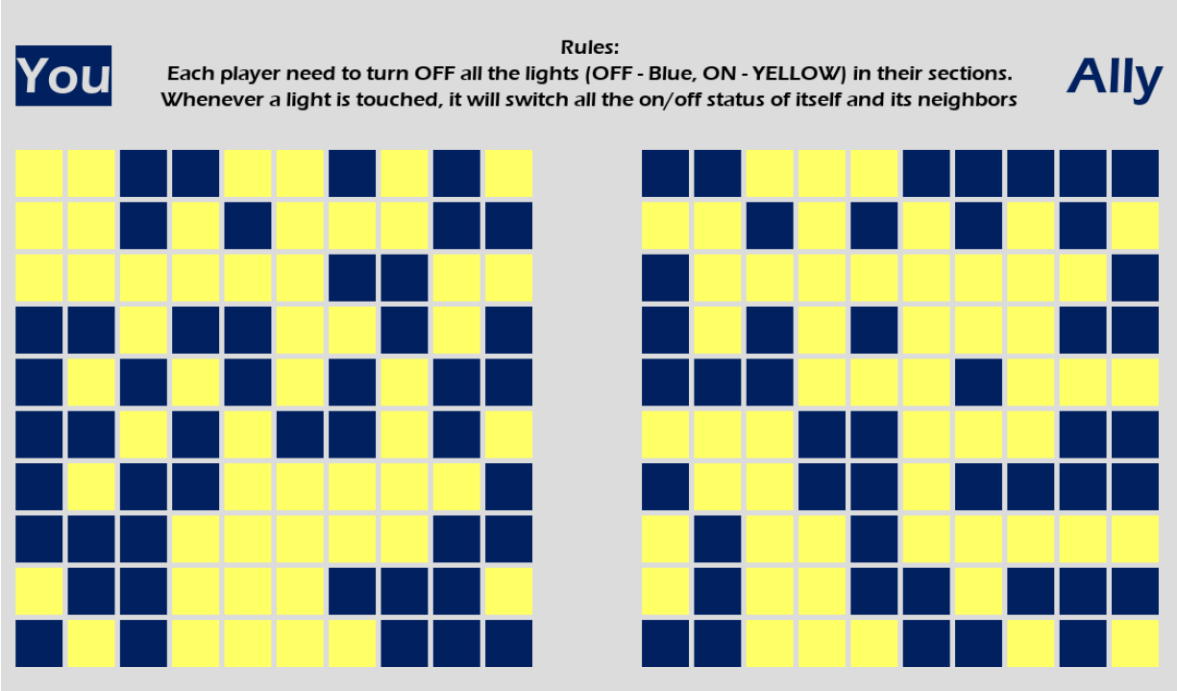
Step 3: The menu of the game should look like this:

After one player start, it will wait for the other player to join:

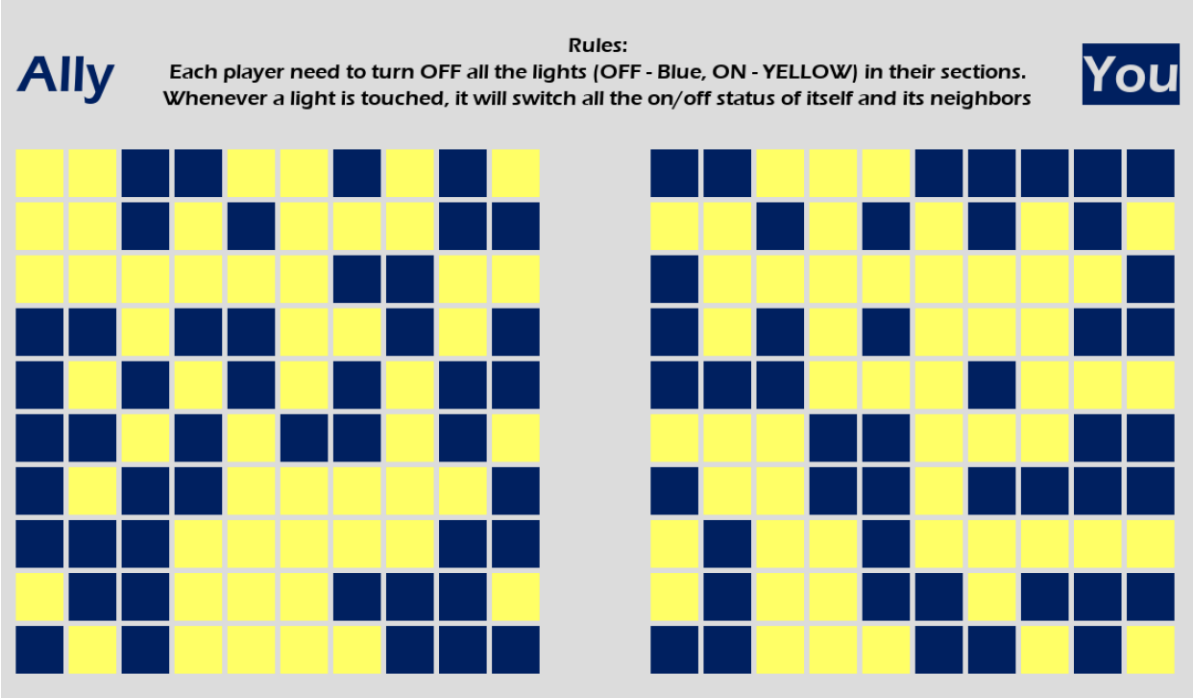


After both players connect, this is the display of the game. The players' goal is trying to turn off all the lights (make the grids turn dark), and the player just only access to the grid belong on their side (display on top of the grid). When a player hit on any light, it will switch the on/off stage of itself and its neighbors.

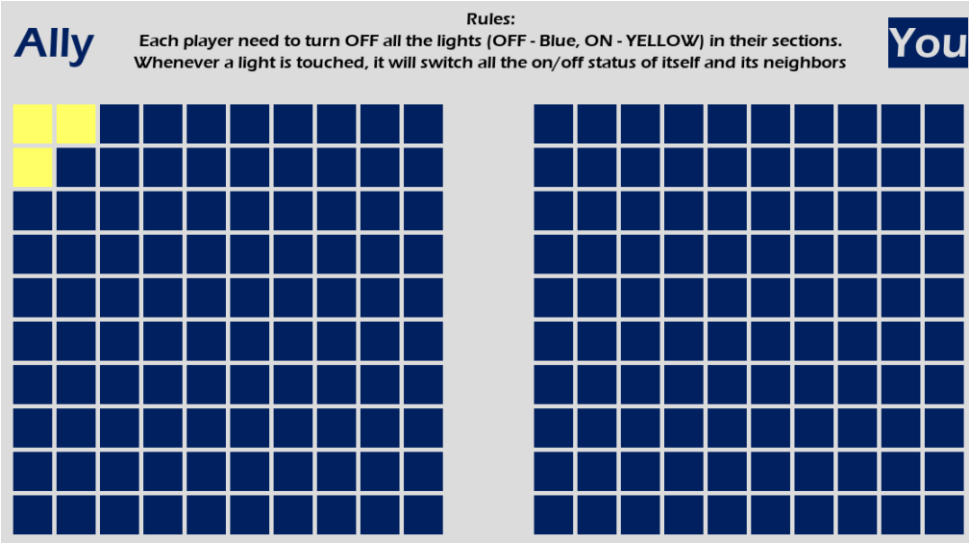
For player 1:



For player 2:



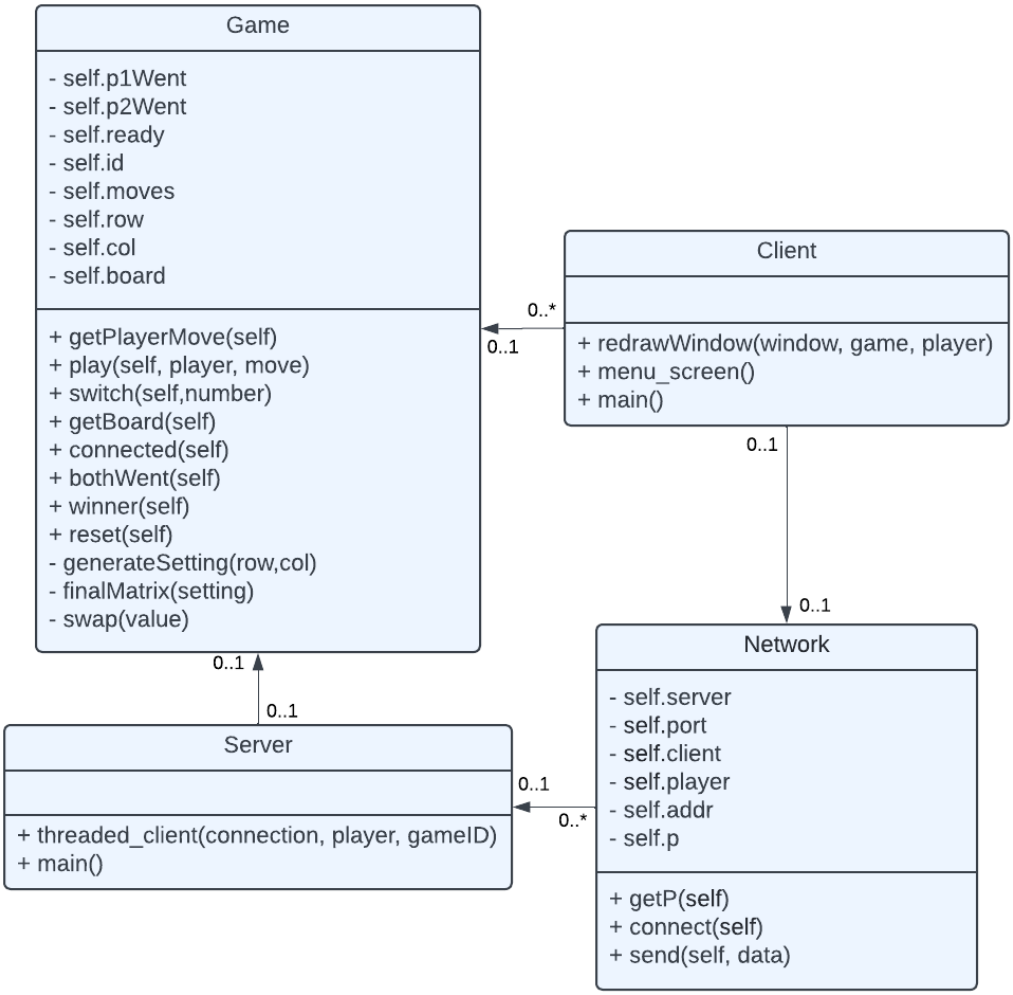
After player reach the last stage, which should like this:



After player 1 hit the top-left corner, both players win the game. Player can click anywhere to go out of the playing stage.

3. Procedure

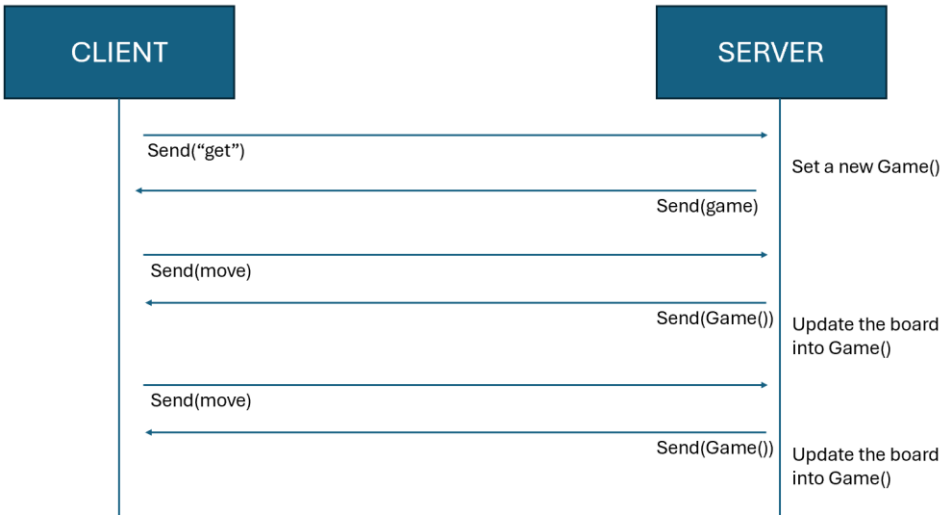
The game was made using Python and build on TCP server and TCP clients.



The idea was to run the client and receive player input. When players click on the screen, send the string "get" to the server via the Network class, and when the server receive the "get", they will start the Game and set that player to be ready. After both players are ready, the Game begins, and the player's moves are recorded after that. The moves data is sent to the Game class and is set as a string "row col", then the Game class will update the move by extracting the row and col by using the Python method split(), then change them to an integer. Inside the game class, there is a board representing both players, and it will get updated after every data is sent to the server. To be more specific, after the Game gets updated, the Game will send back the Game to the client to display it. (Shown in the graphics below). When receiving input, the client will need to check. After the grid has been solved, display the winning state and quit if any player clicks.

To send the Game() class, I used the Pickle built-in library to send an object-oriented class, along with the socket library. The pygame library was used to display the game's graphics. I

also successfully answered my question on the first report on the IP address, so I will also ask the user to include their IP address on the command line argument using library sys.



4. **Testing**

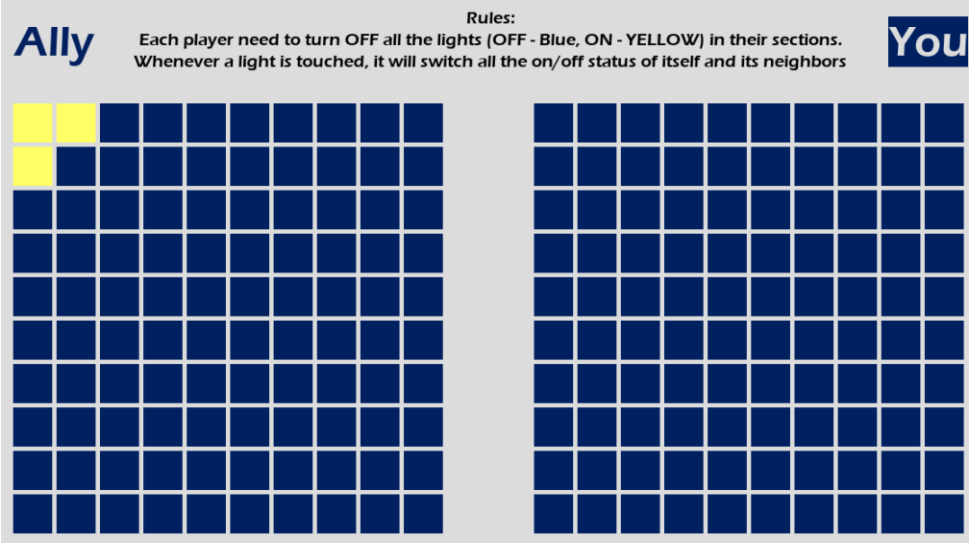
Test process #1: When the game is run, play as player 1 and click on the grid on player 1 portion, then test if the game sends data to the Game class and change the grid for both players. The test passed as expected.

Test process #2: When the game is run, play as player 1 and click on the grid on player 2 portion, and the game should not send any data to the server. The test passed as expected

Test process #3: Force the game to the winning state and check if the transition between playing state and winning state. The test passed as expected

```
self.board = [[0 for i in range(self.col)] for j in range(self.row)]
borad = self.board
borad[0][0] = 1
borad[0][1] = 1
borad[1][0] = 1
```

The result is going to look like this:



After that, player 1 finish off and it reach the final state:

Note that the game can have multiple matches happened as a same time, thanks to having a list to store game's ID.

5. Conclusion and reflection

In conclusion, it was an interesting experience in my opinion that I had a chance to apply the material I learnt during class on something I like. Making games is my most favorite thing to do in computer science area, and this is the hardest game I have ever made, may be this is the first time I made an online two players game. I had 12 hours stuck on the sending data part of the game, and it would be an unforgettable memory that I will have after this class.

References:

Baumstarck, P.G. “Programming Puzzle: Lights Toggle.” *Medium*, The Startup, 5 Feb. 2023, medium.com/swlh/programming-puzzle-lights-toggle-f4d27bf3683e.

“Built-in Exceptions.” *Python Documentation*, docs.python.org/3/library/exceptions.html#Exception. Accessed 18 Apr. 2024.

Flincher, Jon. “PyGame: A Primer on Game Programming in Python.” *Real Python*, Real Python, 18 July 2023, realpython.com/pygame-a-primer/.

Ghosh, Saurabh. “How to Develop a Two Player Pong Game Over Network in Python and Pygame.” *Medium*, Predict, 16 Jan. 2023, medium.com/predict/how-to-develop-a-two-player-pong-game-over-network-in-python-and-pygame-fb8752d8200a.

Mastromatteo, Davide. “The Python Pickle Module: How to Persist Objects in Python.” *Real Python*, Real Python, 31 July 2023, realpython.com/python-pickle-module/.

“Pygame Front Page.” *Pygame Front Page - Pygame v2.6.0 Documentation*, www.pygame.org/docs/. Accessed 27 Feb. 2024.