ELEMENTO

PRINCIPLE OF OPERATING SYSTEM GAME PROJECT

Mark Gil T. Culaway
Ennis Rommel Del Rosario
James Bryan G. King
Jericho James B. Obiedo
Jerome Patrick R. Tayco

Introduction

This documentation outlines the structure and functionality of the game called “Elemento” implemented in C using sockets. The game allows two players to connect over a network and engage in a turn-based card game where the elements (water, fire, grass) and card values determine the outcome.

Game Description

Elemento is a turn-based card game which revolves around elements and numbers. Players can strategize on their deck of cards wherein each card has a numbered element. The game is rock-paper-scissors inspired wherein each element beats another element and players can enjoy the thrill of not knowing what card their opponent will play.

Prerequisites

Ensure that you have the following prerequisites installed:

- A C compiler (e.g., GCC)
- Basic understanding of socket programming concepts
- Two terminals for running the server and client

How to Compile

Compile the server and client programs separately using a C compiler.

Server Compilation:

bash

Copy code

```
gcc server.c -o server
```

Client Compilation:

bash

Copy code

gcc client.c -o client

How to Run

Start the server in one terminal:

bash

Copy code

./server <port_number>

Replace <port_number> with the desired port (e.g., 12345).
Connect clients in separate terminals:

bash

Copy code

./client <server_address> <port_number>

Replace <server_address> with the server's IP address or "127.0.0.1" for localhost, and <port_number> with the same port used for the server.
Follow on-screen prompts in both terminals to play the game.

Game Rules

- Players start with a randomized set of 5 cards where each card is assigned with an element (fire, water, or grass) and a number (1, 2, 3, 4, 5)
- Each player's turn involves choosing a card by specifying the element and card number.
- The game determines the winner of each turn based on the element, card values, and predefined rules.
- Players' health points (HP) decrease based on the game outcome.
- The game ends when all cards are used or when a player's HP reaches zero.

Predefined rules:

(1) There are three elements: Fire, Water, and Grass. Fire beats grass, grass beats water, and water beats fire
(2) If player 1 card element and numerical value beats player 2 card, player 2 takes value1 - value2 damage

(3) If player 1 card element beats player 2 card element but has lower numerical value, both players will not take damage
(4) If player 1 card element and number is the same with player 2's card, both will take 1 damage
(5) If player 1 card element is the same with player 2 card element but has greater numerical value, player 2 takes 1 damage

Server-Side Implementation (server.c)

Socket Initialization:
- The server creates a socket and binds it to a specified port.
- Listens for incoming connections and accepts the client connection.

Game Logic:
- Manages the turn-based game logic, receiving and sending data between players.

Communication:
- Uses socket communication to exchange game data between the server and clients.

Game End Conditions:
- Checks for conditions such as all cards used or a player's HP reaching zero to declare the game's end.

Client-Side Implementation (client.c)

Socket Initialization:
- Connects to the server using the provided IP address and port number.

Game Loop:
- Continuously prompts the user for input to play the game.

Communication:
- Sends and receives game data with the server using socket communication.

Game End Conditions:
- Handles and displays the game result when the game ends.

Important Functions and Structures

void shuffle(char *cards[], int n):
- Shuffles an array of cards randomly.

void game(int *A, int *B, char elem1[], char elem2[], int val1, int val2):
- Determines the outcome of a game turn based on the elements and card values.

void displayCards(char *cards[]):
- Displays the current set of cards.

void displayHP(int player1HP, int player2HP):
- Displays the current HP of both players.

typedef struct { int val; char elem[10]; int hp1; int hp2; } MyData;:

- Defines a structure for exchanging game data between server and client.

Team Contributions

Mark Gil Culaway

- Setup the game prerequisites
- Helped solve back-end coding problems
- Prepared and presented game presentation

Ennis Rommel Del Rosario

- Thought of the game idea
- Helped in fixing game mechanics issues
- QA tested the game and provided feedback

James Brian King

- Helped in UI design
- QA tested the game

Jericho James Obiedo

- Lead back-end and front-end developer
- Conducted meetings as leader
- Polished the game logic and mechanics
- Demonstrated game as server and explained the gameplay

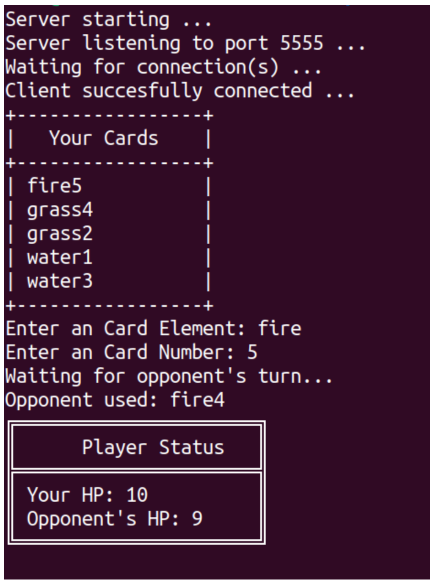
Jerome Patric Tayco

- Helped in the UI design
- Demonstrated the game as client
- Helped solve back-end coding problems

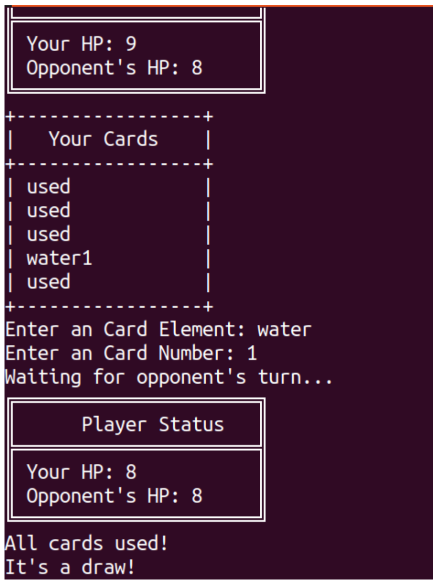
Conclusion

The Elemental game allows two players to engage in a strategic card game over a network. Understanding the game rules and implementing the server-client architecture ensures a seamless gaming experience.

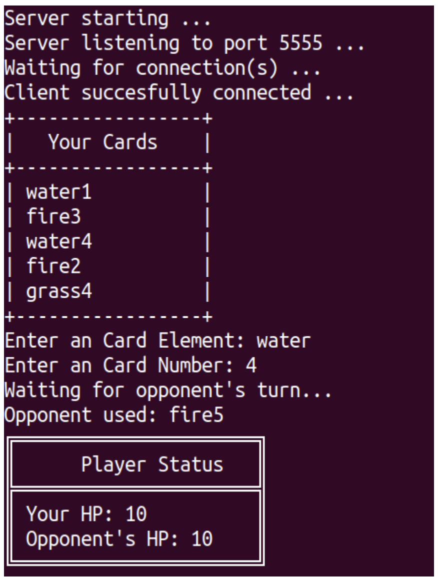
Sample Runs



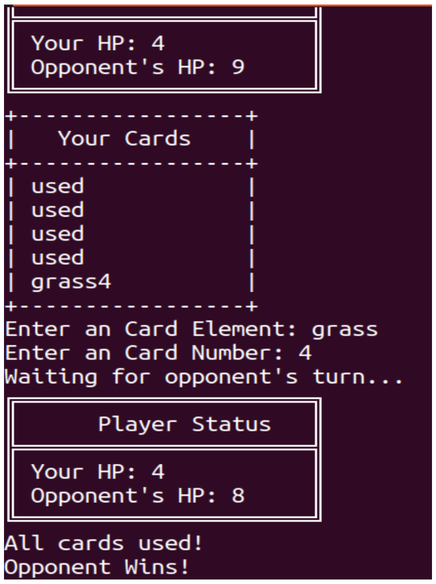
```
Server starting ...
Server listening to port 5555 ...
Waiting for connection(s) ...
Client succesfully connected ...
+----------------+
|   Your Cards   |
+----------------+
| fire5          |
| grass4         |
| grass2         |
| water1         |
| water3         |
+----------------+
Enter an Card Element: fire
Enter an Card Number: 5
Waiting for opponent's turn...
Opponent used: fire4
╔═══════════════════════╗
║     Player Status     ║
╠═══════════════════════╣
║ Your HP: 10           ║
║ Opponent's HP: 9      ║
╚═══════════════════════╝
```



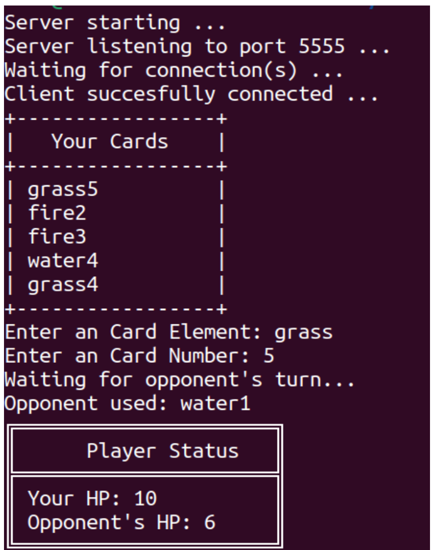
```
║ Your HP: 9            ║
║ Opponent's HP: 8      ║
╚═══════════════════════╝
+----------------+
|   Your Cards   |
+----------------+
| used           |
| used           |
| used           |
| water1         |
| used           |
+----------------+
Enter an Card Element: water
Enter an Card Number: 1
Waiting for opponent's turn...
╔═══════════════════════╗
║     Player Status     ║
╠═══════════════════════╣
║ Your HP: 8            ║
║ Opponent's HP: 8      ║
╚═══════════════════════╝
All cards used!
It's a draw!
```



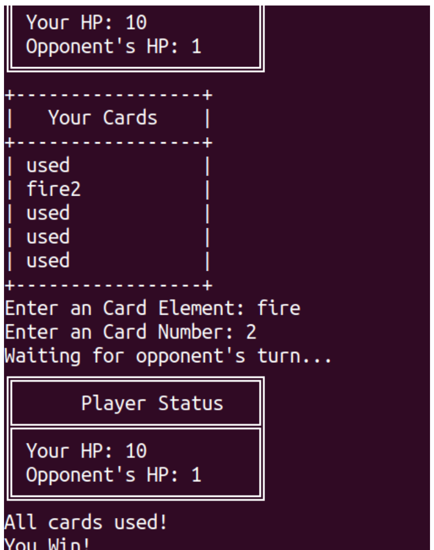
```
Server starting ...
Server listening to port 5555 ...
Waiting for connection(s) ...
Client succesfully connected ...
+----------------+
|   Your Cards   |
+----------------+
| water1         |
| fire3          |
| water4         |
| fire2          |
| grass4         |
+----------------+
Enter an Card Element: water
Enter an Card Number: 4
Waiting for opponent's turn...
Opponent used: fire5
╔═══════════════════════╗
║     Player Status     ║
╠═══════════════════════╣
║ Your HP: 10           ║
║ Opponent's HP: 10     ║
╚═══════════════════════╝
```



```
║ Your HP: 4            ║
║ Opponent's HP: 9      ║
╚═══════════════════════╝
+----------------+
|   Your Cards   |
+----------------+
| used           |
| used           |
| used           |
| used           |
| grass4         |
+----------------+
Enter an Card Element: grass
Enter an Card Number: 4
Waiting for opponent's turn...
╔═══════════════════════╗
║     Player Status     ║
╠═══════════════════════╣
║ Your HP: 4            ║
║ Opponent's HP: 8      ║
╚═══════════════════════╝
All cards used!
Opponent Wins!
```



```
Server starting ...
Server listening to port 5555 ...
Waiting for connection(s) ...
Client succesfully connected ...
+----------------+
|   Your Cards   |
+----------------+
| grass5         |
| fire2          |
| fire3          |
| water4         |
| grass4         |
+----------------+
Enter an Card Element: grass
Enter an Card Number: 5
Waiting for opponent's turn...
Opponent used: water1
╔═══════════════════════╗
║     Player Status     ║
╠═══════════════════════╣
║ Your HP: 10           ║
║ Opponent's HP: 6      ║
```



```
║ Your HP: 10           ║
║ Opponent's HP: 1      ║
╚═══════════════════════╝
+----------------+
|   Your Cards   |
+----------------+
| used           |
| fire2          |
| used           |
| used           |
| used           |
+----------------+
Enter an Card Element: fire
Enter an Card Number: 2
Waiting for opponent's turn...
╔═══════════════════════╗
║     Player Status     ║
╠═══════════════════════╣
║ Your HP: 10           ║
║ Opponent's HP: 1      ║
╚═══════════════════════╝
All cards used!
You Win!
```