Project - 2

Abdullah Nawaz – 202108536  
 Rufus John Kurian – 202107326  
Sarim Toqeer – 202008545  
Mohammed Faheem Ali Zaidi – 202109963

CMPS-405 B03

**Contribution Percentages**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Group Member** | **Contribution Percentage** |
|  |
| 1 | Abdullah Nawaz | 25.00% |  |
| 2 | Rufus John Kurian | 25.00% |  |
| 3 | Sarim Toqeer | 25.00% |  |
| 4 | Mohammed Faheem Ali Zaidi | 25.00% |  |

**Summary**

* We designed a multiplayer player game. We created a Server class which is responsible for handling client connections, managing games, players, and tickets.
* We implemented methods for creating, joining, and starting games concurrently, managing multiple games to process at a time with their multiple rounds.
* We implemented methods for handling player timeouts, reading and writing tickets to file storage, and managing game states.
* Our project includes client class, which is responsible for connecting to the server, handling user input, and displaying game messages.
* The game features a functionality for new and old players to join the game and provides nicknames as well input guesses during the game.
* Develop game mechanics such as multiple rounds, round winners based on player guesses, updating points of the players, and ending the game.
* The game has a feature to discourage players from choosing 0 all the time by the end of the game in which both players have only 1 point left, and one player chooses 0.

**Challenges:**

* One of the main challenges we faced was the chat system, the server regularly sends a message to each client and expects a reply; if no reply is received, the player is timed out and dropped from their game, if any.
* Implementing chat system, ping system was interfering with rest of the project.
* Concurrent development and coordination among team members to ensure consistent understanding and implementation of the codebase, especially when different team members have different approaches or interpretations of the requirements.
* First, we were facing issues to enable games to process concurrently, first we started with a single player, later with the use of multi-threading we were able to play multiplayers on server.
* Dependency management and ensuring that changes made by one team member do not break functionality or compatibility with other parts of the codebase.
* Testing and debugging were quite an issue in the beginning, however we ensured that changes made to fix one part of the code do not introduce new bugs or regressions in other parts of the application. Debugging issues caused by conflicting changes or unexpected interactions between components can be time-consuming.

**Issues:**

Ping System

Chat System

**Contributions:**

|  |  |  |
| --- | --- | --- |
| Member | Date | Description |
| Sarim | 14/4/2024 | Initialised all the classes with comments |
| Sarim | 14/4/2024 | Started communication with one player using sockets |
| Abdullah | 16/4/2024 | restructure Project  Server Player Game Ticket Client |
| Abdullah | 16/4/2024 | Connection Establishment Implement Ticket Class (store id and nickname) Client class connects with server and new users provide nickname, whereas returning users provide a ticket In the server, handle clients differently based on if they are a new player or returning player. |
| Abdullah | 16/4/2024 | Handle Tickets store tickets in a file load them to the array each time the server starts keep the file updated |
| Abdullah | 28/4/2024 | Start Implementing Player Class  Move code from Server to Player Thread Each Player Thread is associated with a Client Player Thread will communicate with the Client Using Placeholder methods from the Game right now Player Thread will call methods in the Game Class |
| Abdullah | 28/4/2024 | Game with one player working  Game logic implemented in Game.java Working with one Player (right now player always gets point deducted so that game isn't infinite) NEXT STEP is making the game work with multiple Clients and then having Multiple Games Working |
| Faheem | 1/5/2024 | Multiplayer Support |
| Faheem | 2/5/2024 | Enhanced Multiplayer, round end info broadcast, round end functions |
| Rufus | 4/5/2024 | Modified the Game class to discourage players from selecting 0 |
| Rufus | 4/5/2024 | Update Player.java |
| Rufus | 4/5/2024 | Updated server and client classes |
| Rufus | 4/5/2024 | Added Leaderboard |
| Faheem | 4/5/2024 | Input condition and printing |
| Sarim | 5/5/2024 | Developed the spectation of the game by looser players |
| Abdullah | 5/5/2024 | Get IP address and port number from the command line |
| Rufus | 5/5/2024 | Pinging Implementation |
| Abdullah | 5/5/2024 | worked on the ping system but it wasn’t working |
| Sarim | 5/5/2024 | Worked on debugging and final report and sample testing |

**Github Link:**

<https://github.com/abdullah-an2108536/OS-Game>

**References:**

* <https://queue.qa/cmps405/labs/>
* <https://www.geeksforgeeks.org/simple-chat-application-using-sockets-in-java/>