Program Documentation

Task 1 – Single Player Game

Task 1 consistest of 2 classes a Client class that handels all the user input and the Server class that hosts the game and communicates with the client.

Client Class
The client class connects to the server and continues to send data until user quits the game. The user is able to play multiple rounds without having to reconnect. The Client class displays the recived messages from the server and requests for the user to input data. No game validation is done by the Client, it sends all input text to the server for validation and waits for the response.

Server Class
The server class waits for an incomming connection from a client and then enters a loop until the client exits the program.
The server class also logs all communication and Game data to serperate log files.
The server prompts the user for a name and code length before entering the guessing loop.
The client can exit the round at any time by pressing ‘f’.
After the client makes a guess the server responds with a hint if incorrect. Once all the guesses are up or the client guesses correctly the server exits the gueesing loop, sends the total amount of guesses and the asks the client if they wish to play another game.

Task 2 – Multiplayer Game

Task 2 consists of 6 classes. 4 main classes and 2 helper global classes.

Client Class
The client class is the same class used in task 1.

Server Class
The Server class has been refactored to work with multiple users. The main game function has been moved to the RoundThread Class and the client communication has been moved to the PlayerThread class. The server class handles new Clients connecting and add them to the lobbyQueue. Once there is 3 players the server takes the first 3 players and starts new thread to play the game.

RoundThread Class
This is where the game logic executes. For each player in the round this thread gets information from the PlayerThreads validates it and then send back the correct output. This class also creates the Code and notifies the PlayerThread class when the game is over.

PlayerThread Class
Each PlayerThread that is creates handels one client. This thread sends and recieves the game input and output to and from the client. When initially connected the client is prompted to enter a name. The PlayerThread stores this name as an attribute and also keeps count of the clients guesses. This thread stay open while the client wishes to continue playing the game.

Global Helper Classes
Both GameMsg and GuessResponse are global classes to allow the threads to communicate data between the RoundThread and PlayerThreads.