Program Documentation

Classes:

Client: Client class is used to for connecting to server and play the game. All the communication done with server by users is via client class. It helps user to enter the number, send response to server, and close if the user is not interested to play anymore. Following are the methods and objects used in this class. Timer object is created to receive keep alive messages from server.

register () → This method is used to register client details like client name to the server. Also stores the name given by server.

receiveMulticastMessageToClient () → This method is to receive multicast message sent from the server.

guessInput () → This method is to make sure that client only enters number between 0-12 to play the game or 'e' to forfeit.

exitCondition () → This function is used for asking a client when they finish their game if they want to play again or quit the game. Following are the import file that used in the client:

```
import java.io.*; ← to use any IO packages. For example, File, IO exception
import java.net.DatagramPacket; ← Datagram packet is used for the receiving   message from server
import java.net.InetAddress; ← Inetaddress used for connecting the common client server address for udp multicast connection.
import java.net.MulticastSocket; ;← udp Multicast socket for creating a listening for server connection.
import java.net.Socket; ← for creating socket connection fr server and client
import java.net.SocketException; ← For any socket exception generated and to be handled
import java.net.SocketTimeoutException;← For any socket exception generated and to be handled
import java.util.Timer; ←For Keep alive message receiving
import java.util.TimerTask; ←Storing the task of receiving message sent by server.
```

Client Handler: Client handler class is used to implement threads for clients and all the communication from server to client is done via client handler class. Logs are also written in this class about the activity of the users. We have also used a comparator to sort the players array list to display the ranking functionality in a round. Client handler requires Server.java to involve it. Following are the methods and objects used in this class.

Constructor → Stores the connection of the client with server and save the client name as the thread name and create DataInputStream and DataOutputStream object for sending and receiving request from client to server.

run()→ Run method calls the game method as soon as thread starts.

game() → It is called inside run method and it is used for starting and handling the game server. It is responsible for communication with the clients. If client wants to play again, it allows server to add the client to player queue.

writeGameLog () → exports gaming log to gamingLog.log

writeCommunicationLog () → exports communication log to communicationLog.log

Comparator → To sort the arraylist in order of guess counts so that Users can be ranked based on their guess count.

Following are the import file that used in the ClientHandler:

```
import java.io.*; ← to use any IO packages. For example, File, IO exception
import java.net.InetAddress; ← Inetaddress used for logs to store client address.
import java.net.Socket;← To store the client server connection
```

Program Documentation

```
import java.net.SocketException; ← For any socket exception generated and to be handled
import java.net.SocketTimeoutException; ;← For any socket exception generated and to be
handled
import java.text.DateFormat;← For logs to store the date time
import java.text.SimpleDateFormat;← helps us to store the date time according to simple
format
import java.util.Calendar;← for capturing the date time and changing date time format.
import java.util.Comparator;←For sort the number guess count from low to high
import java.util.Date;← for storing date.
```

Server: Server class is used for hosting game server. Firstly, it registers client in the start, create threads of client who are supposed to play in the current round and put other users in the waiting lobby. It will display the ranking of the users after the round is finished and exports ranking logs to gaming logs. Moreover, this class is also used for sending multicast messages to clients. Following are the methods and objects used in this class.

Constructor → It creates a server socket. Firstly, it adds at max 6 client in a queue and then start the game until all players exit from the game

Add players () → It used for adding new client in the queue of client handler so that they could play the game later.

sendMulticastMessageToClient () → This function is used to send UDP multicast message to all clients.

resultOfRound () → This generates result of a game in a string type.

Timer () → To implement keep alive messages to clients on every 30 seconds.

lobbyQueue () → This is used to create a playing lobby and waiting lobby for players to either play or wait. Waiting players can play once all players in playing queue finish their game. This function is basically a lobby management function.

Following are the import file that used in the server:

```
import java.io.*; ← to use any IO packages. For example, File, IO exception
import java.net.DatagramPacket; ← Datagram packet is used for the sending   message to
client throw udp connection
import java.net.InetAddress; ← Inetaddress used for connecting the common client server
address for udp multicast connection.
import java.net.MulticastSocket;← udp Multicast socket for udp socket connection all
clients.
import java.net.ServerSocket;← to create socket for server side
import java.net.Socket; ← for socket connection between server and client
import java.net.SocketException; ← For any socket exception generated and to be handled
import java.net.SocketTimeoutException;←For any socket timeout generated and to be handled
import java.net.UnknownHostException; ← For any Unknown Host issue found and to be handled
import java.util.ArrayList; ← For playing lobby and waiting lobby in queue for game.
import java.util.Collections; ←For ranking the player by number of guess count in
ascending order.
import java.util.LinkedList;← For Queue data structure
import java.util.Queue; ← Queue data structure is used for adding players for the game.
import java.util.Random;← For generating different number from 0-12 in each round of game.
import java.util.Timer; ←For Keep alive message sending
import java.util.TimerTask;← Storing the task of sending message for sending to all
client.
```