NETWORKING DOCUMENTATION

Jarrod Dowsey

Complex Game Systems Assessment

For this Assessment I have been required to create a checkers game with either an AI opponent or a Networked connection to another human player and I have opted to go with the Latter.

For the networking I am using the library Raknet which is a free to use networking library enabling me to create a connection between two peers and therefore giving them the opportunity to play checkers against each other. I can create this connection using one of two ways; The first way being a peer to peer connection where multiple computers connect to each other directly without the use of a server to transmit messages. The second way of networking is through the use of a server and client model that makes use of a server to send messages between clients.

I have chosen to use the Server and Client model in my checkers game. This model uses a packet and message system in which a client creates a packet and attaches a message to it to let the server know what it needs to do with the packet, in most cases it tells the server to read the packet and then pass it onto all other clients connected to the server.

Below I will explain how each part of this networking model works and connects with each other:

Server
The server in my checkers game needs to be created first so the clients have something to connect to. The server is the control centre of the networking, it takes in messages and packets from the clients and it then transmits all the packets to every other client that is connected to the server. Using the server client model means the server plays a vital part in communication and without it the clients would not be able to communicate with each other.

Clients
The Clients are created and connected to the server and are responsible for building packets and sending them up the server in order to transmit the move one player has made to the opposition player.

Packets
The packets are exactly what their name states, containers of information that are sent backwards and forwards between the clients and server. In my game these packets are built when a client makes a move. The data that I am transmitting into the packets is as follows:

- The ID of the board piece that has been changed
- Whether or not it is a green piece or purple piece that has been moved
- Whether or not the piece moved was a king piece
- Whether or not a board piece is still occupied or not

These values are written into a packet and then sent up to the server where they are read by the server and written back into another packet and sent to the clients that are connected. Once the packets have been received by the client the data inside is read and the values of the board pieces are set so both players boards are in the same state before the next client makes their move.

Messages

The messages used are very important, they inform the server and clients what exactly they need to do with the packet they have been sent. The messages that i have used are as follows:

- ID_SERVER_CLIENT_ID

This Message is used when a client connects the server and its purpose is to send an ID to the client so they can be identified by the server when later sending packets.

- ID_CLIENT_CREATE_OBJECT

This Message is used within the server and it creates an empty object and fills it with the data from the packet just sent from a client. Once that has happened it re packs it and sends it to all other clients connected.

- ID_SERVER_FULL_OBJECT_DATA

This message is used to tell the client the server has a full packet and it has been sent through to you. Once the client receives this message it knows to unpack the packet and change the values of the board pieces to match the ones in the packet.

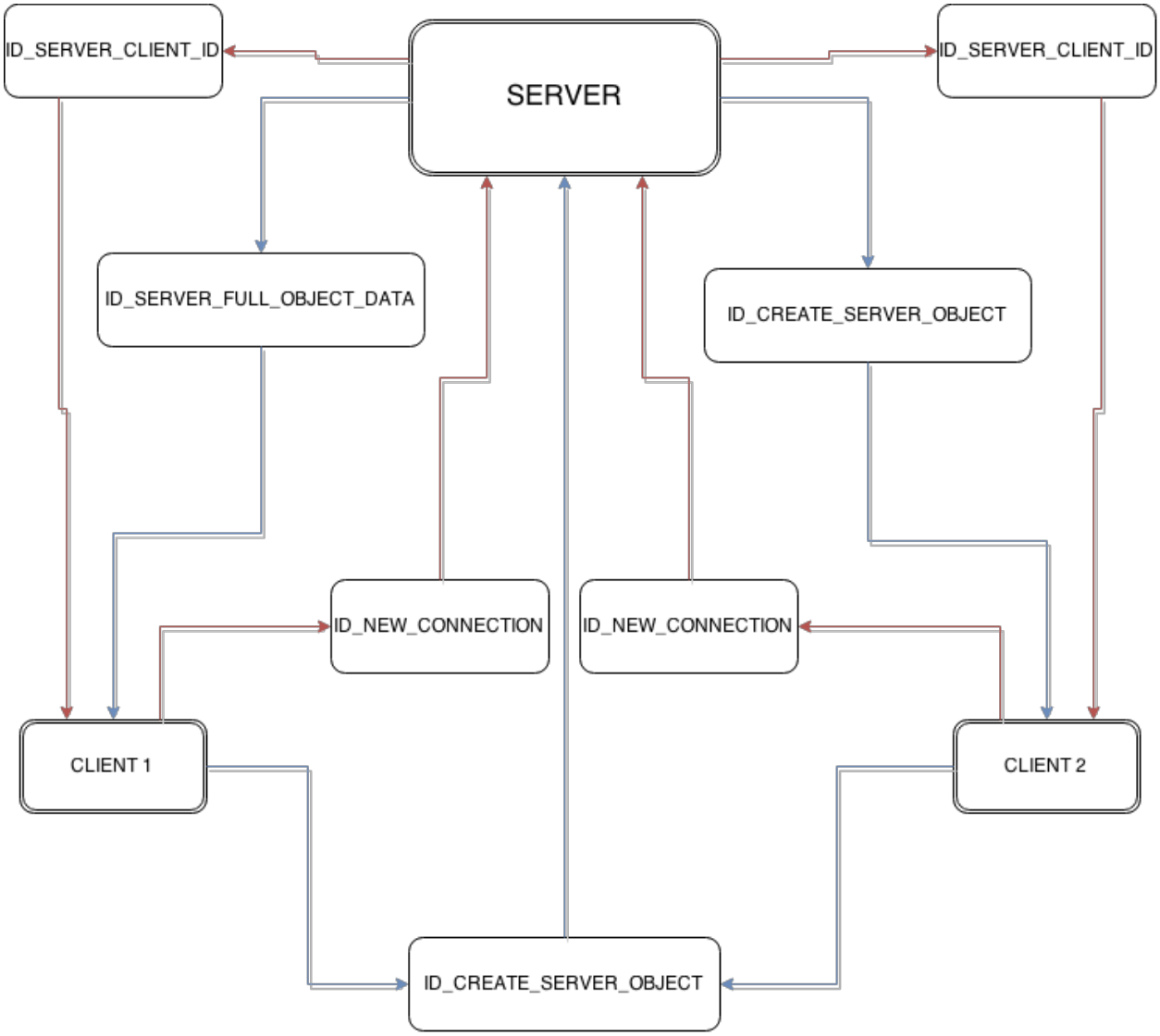
- ID_NEW_INCOMING_CONNECTION

This Message is within the server and its job is to connect the client that has sent this message and get ready to send the id_server_client_id message.

- ID_DISCONNECTION_NOTIFICATION

This message is received by the server when a client has disconnected and its job is to remove the data from that clients and reset the amount of clients connected to the server.

Below you will find a diagram that shows the communication between server and clients.


ID_SERVER_CLIENT_ID
SERVER
ID_SERVER_CLIENT_ID
ID_SERVER_FULL_OBJECT_DATA
ID_CREATE_SERVER_OBJECT
ID_NEW_CONNECTION
ID_NEW_CONNECTION
CLIENT 1
CLIENT 2
ID_CREATE_SERVER_OBJECT