

Matthew Wallace

NETWORK SYSTEMS FOR GAME DEVELOPMENT (CMP303)

Observer mode with multiple clients

Transport layer protocol

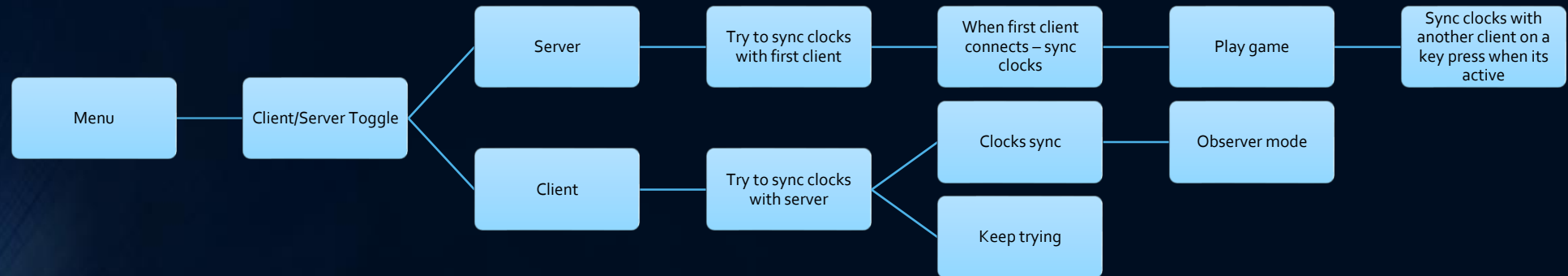
- Transport-layer protocol: UDP (Unreliable Datagram Communication)
- UDP socket in non-blocking mode (don't sit and wait for the packet to arrive)
- Network API: SFML Network Module
- Client – Server Architecture
 - The server holds the true state of the game
 - The client connects to the server to receive data and to display the game on its end

Why UDP?

- Observer mode doesn't need to be 100% accurate
- UDP is unreliable but guarantees no corruption of data
- UDP is not connection-based (one socket for multiple clients)
- UDP is faster and more lightweight than TCP (fewer requirements)
 - Send and receive to/from anyone at any time with the same socket
 - One socket allows broadcasting messages to multiple recipients, or even to an entire network

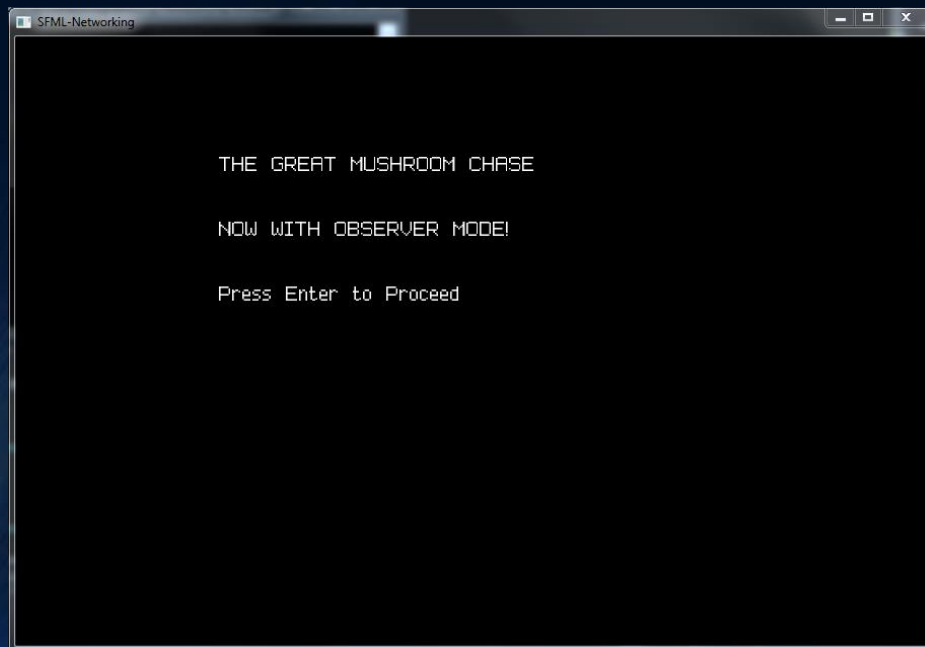
Client-Server state machine

Application-layer protocol

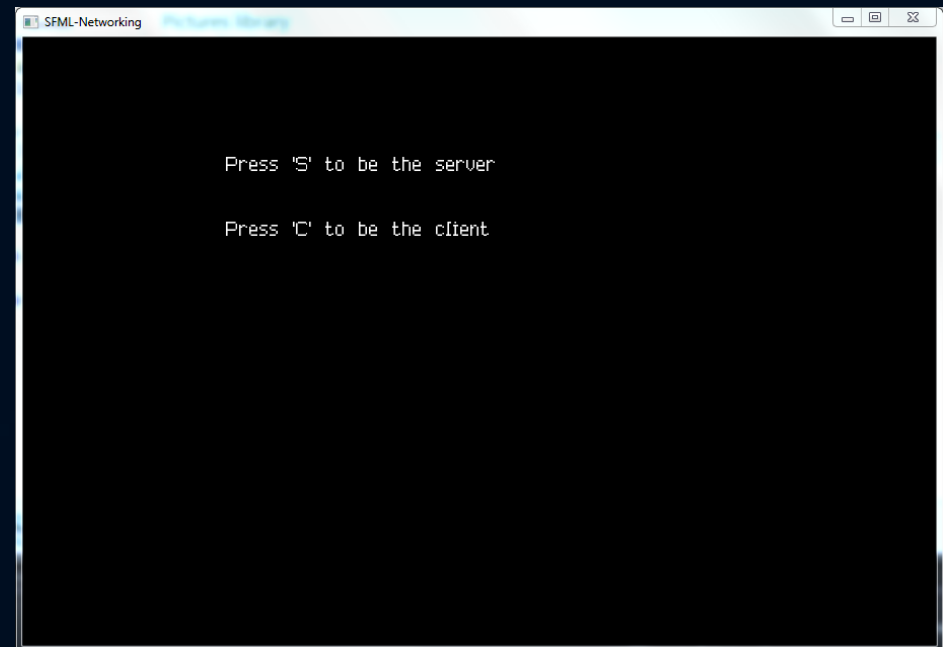


How it looks in game?

MENU STATE

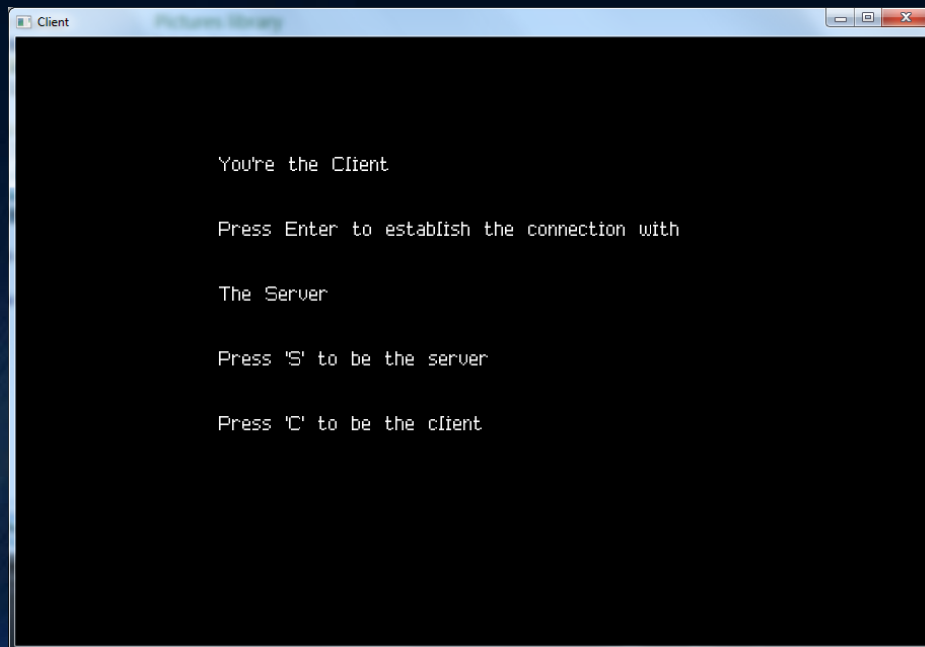


NETWORK STATE

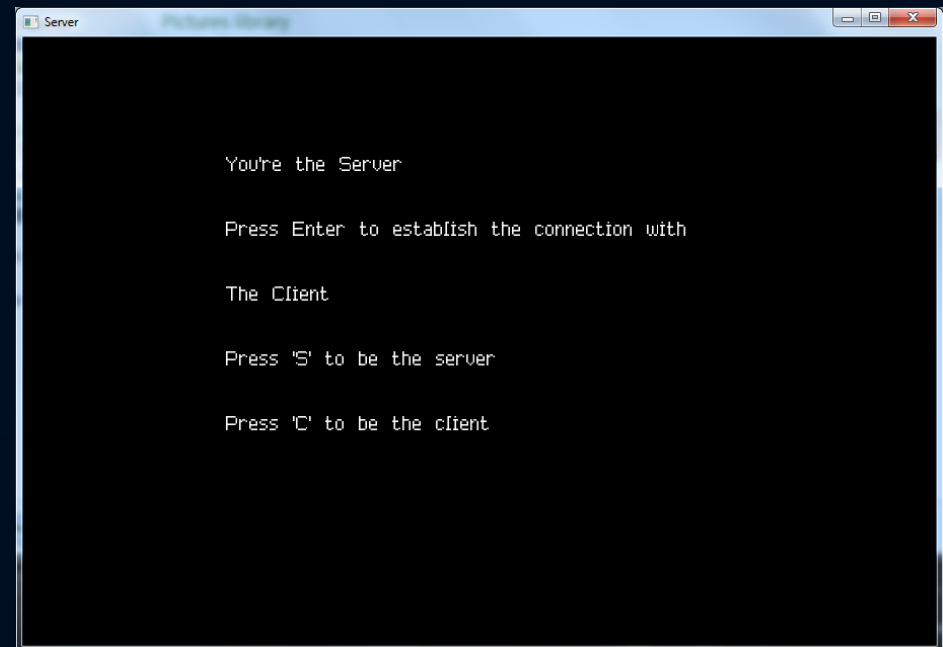


How it looks in game?

CLIENT TOGGLE

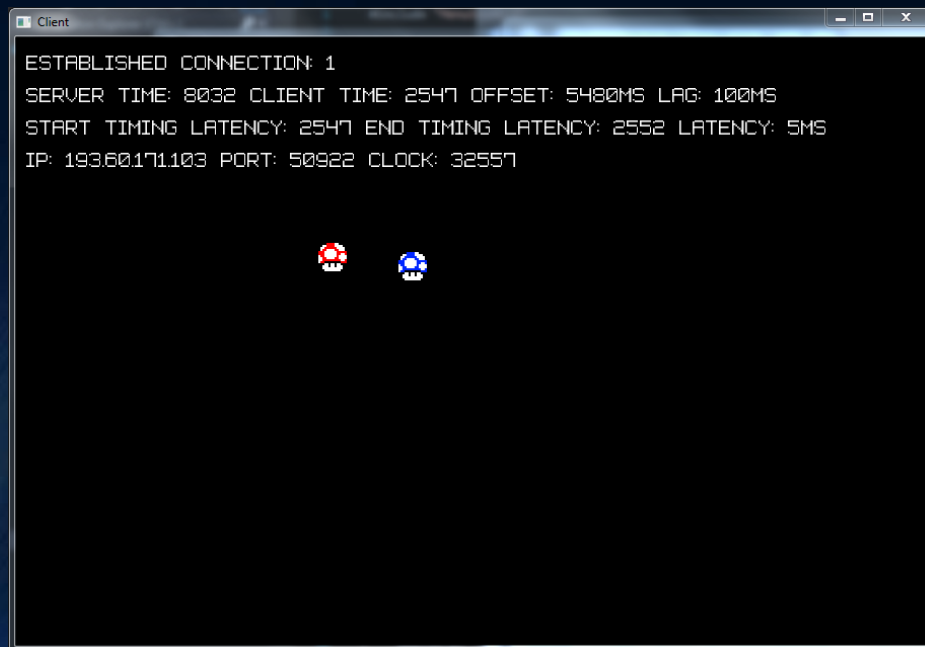


SERVER TOGGLE

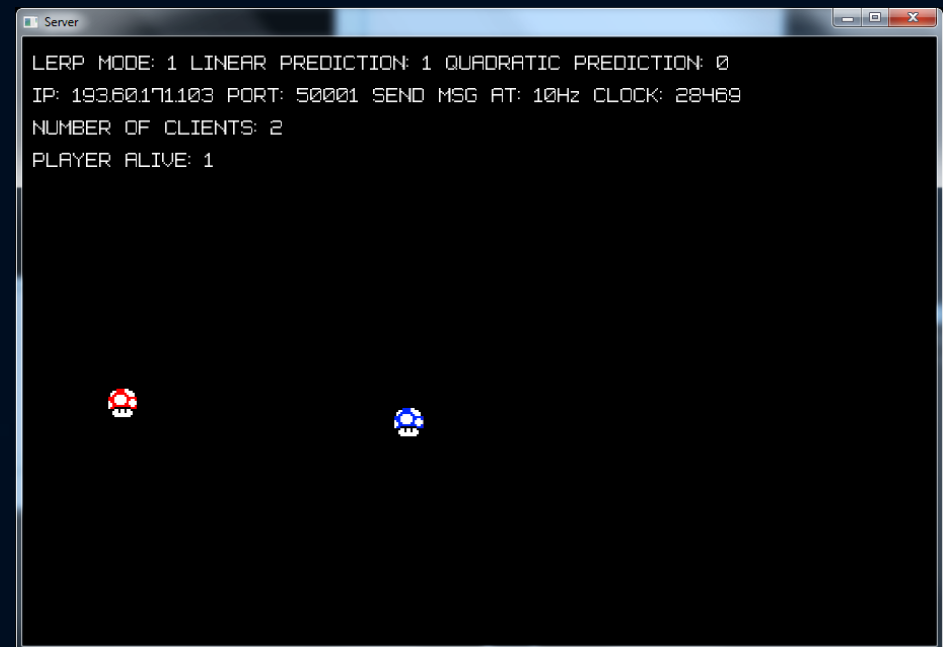


How it looks in game?

CLIENT

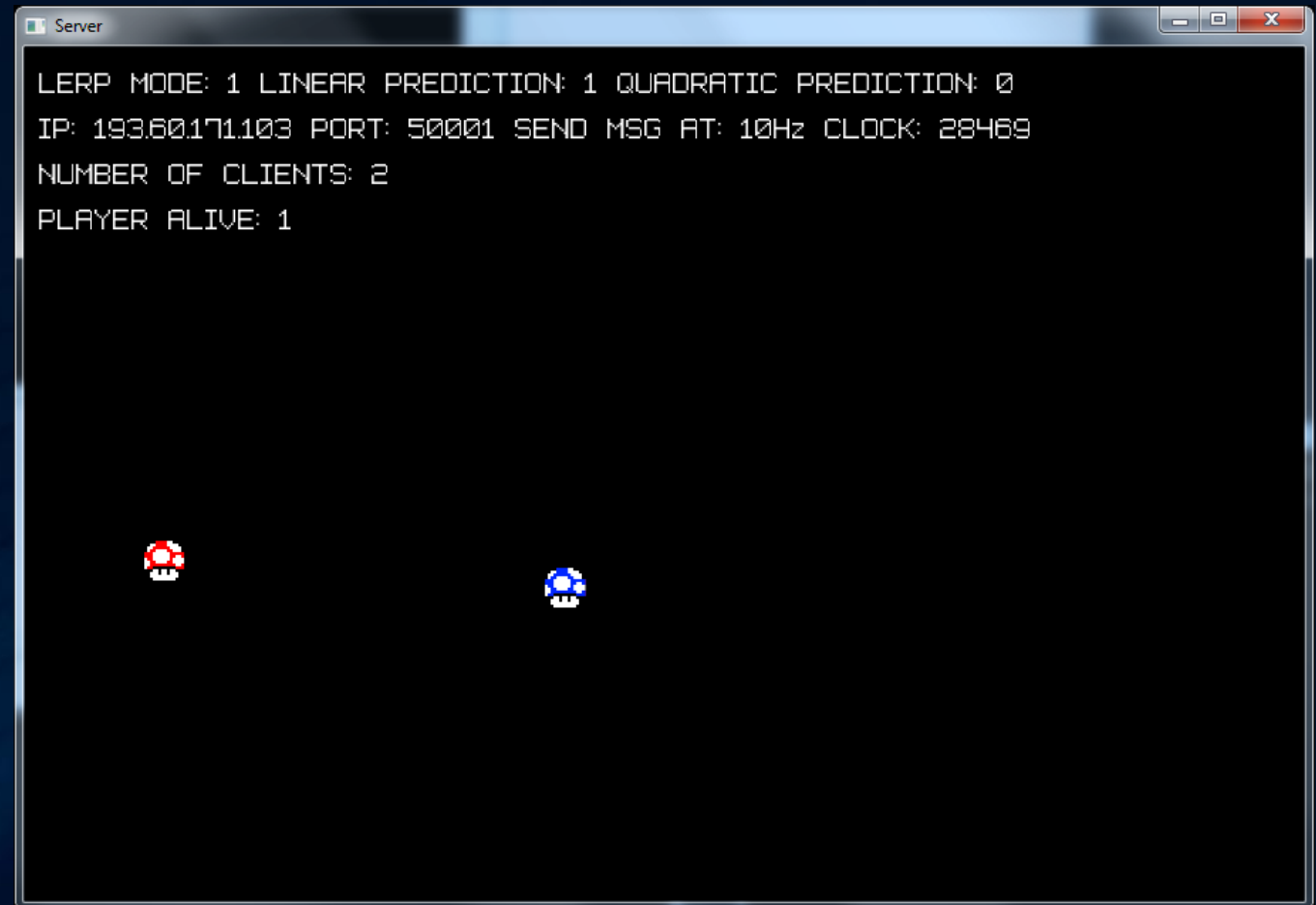


SERVER



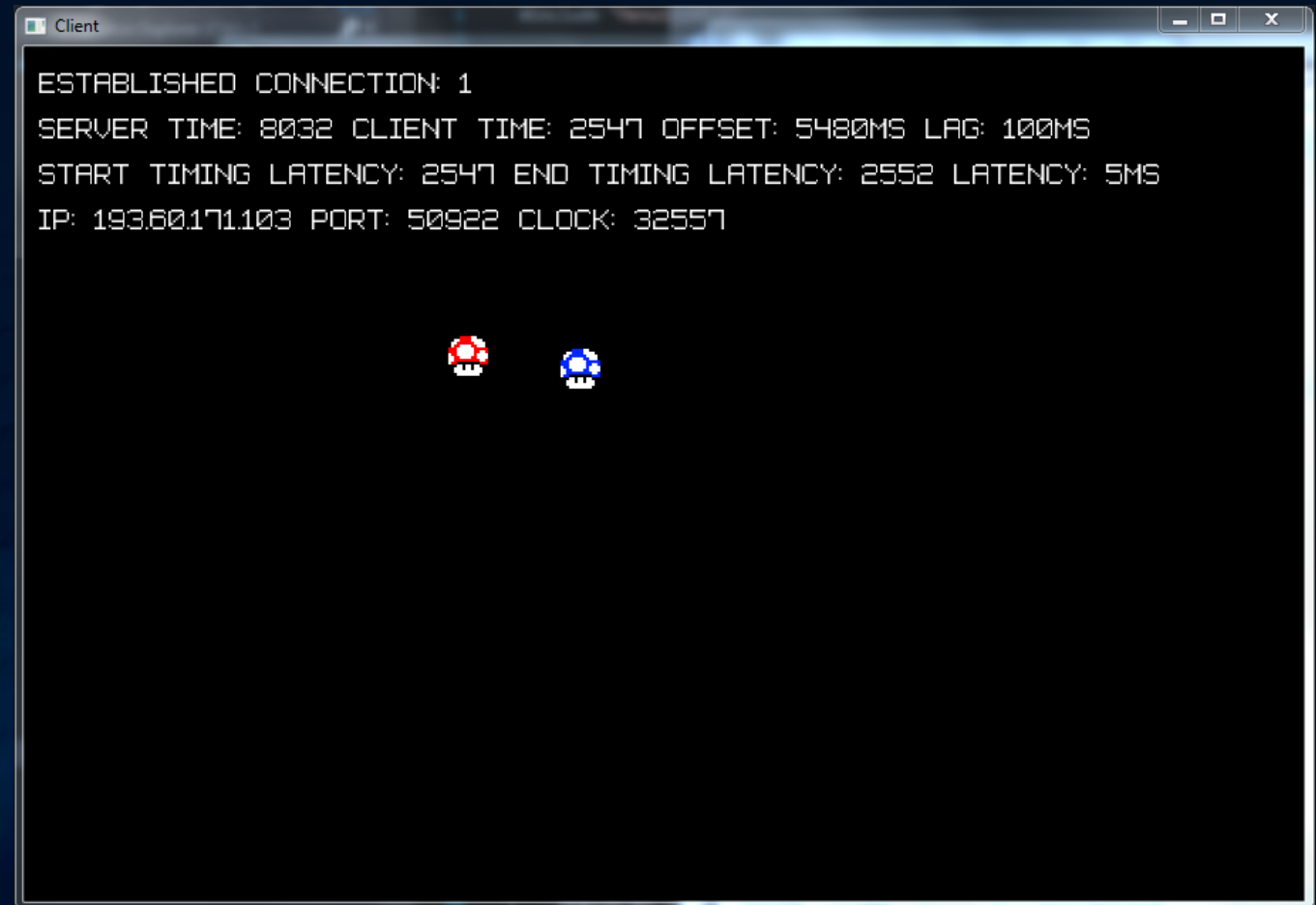
Server

- Display: toggle of lerp mode, linear and quadratic prediction; server's IP, port and clock; message sending rate (Hz), number of connected clients, player's state
- Message sending rate can be set to: 10Hz, 12Hz, 15Hz, 20Hz, 30Hz or 60Hz

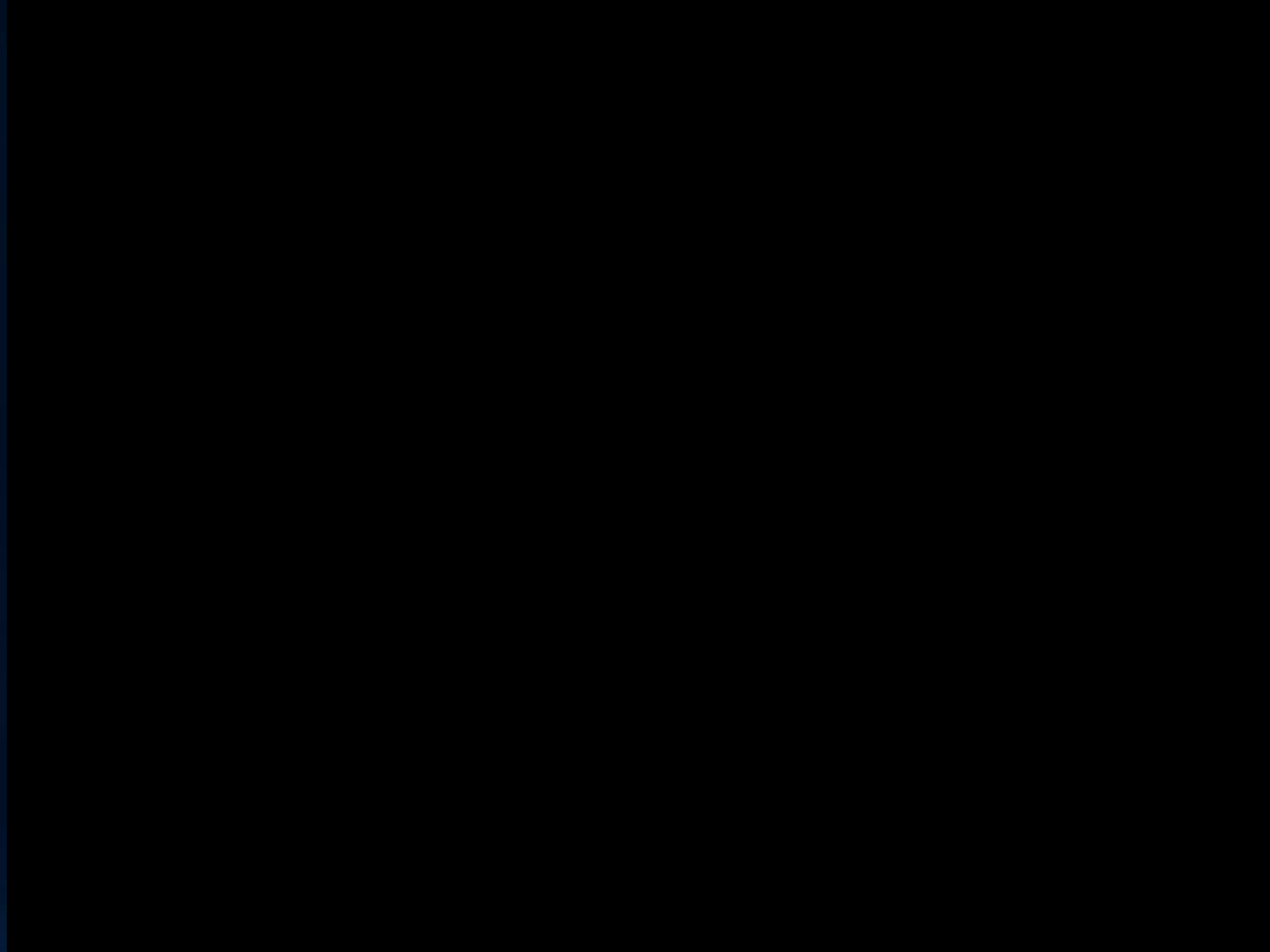


Client

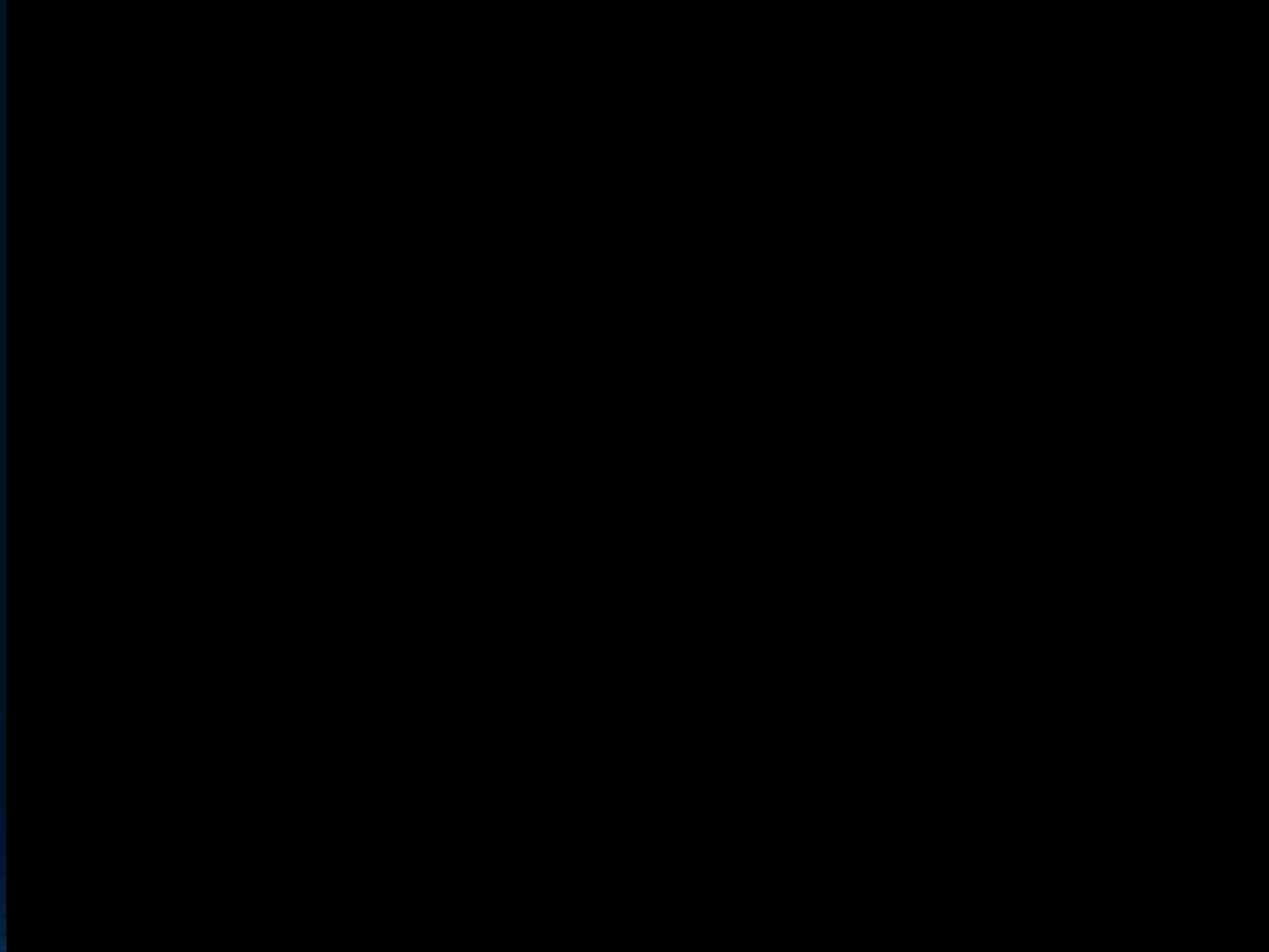
- Display: connection state, client's IP address, port, clock
- Server's and client's time at the beginning of syncing the clocks
- $\text{Offset} = \text{server time} + (1/2 \text{ latency}) - \text{client time}$
- $\text{Latency} = \text{start timing latency when sent msg to server} - \text{end timing latency when received msg from server}$



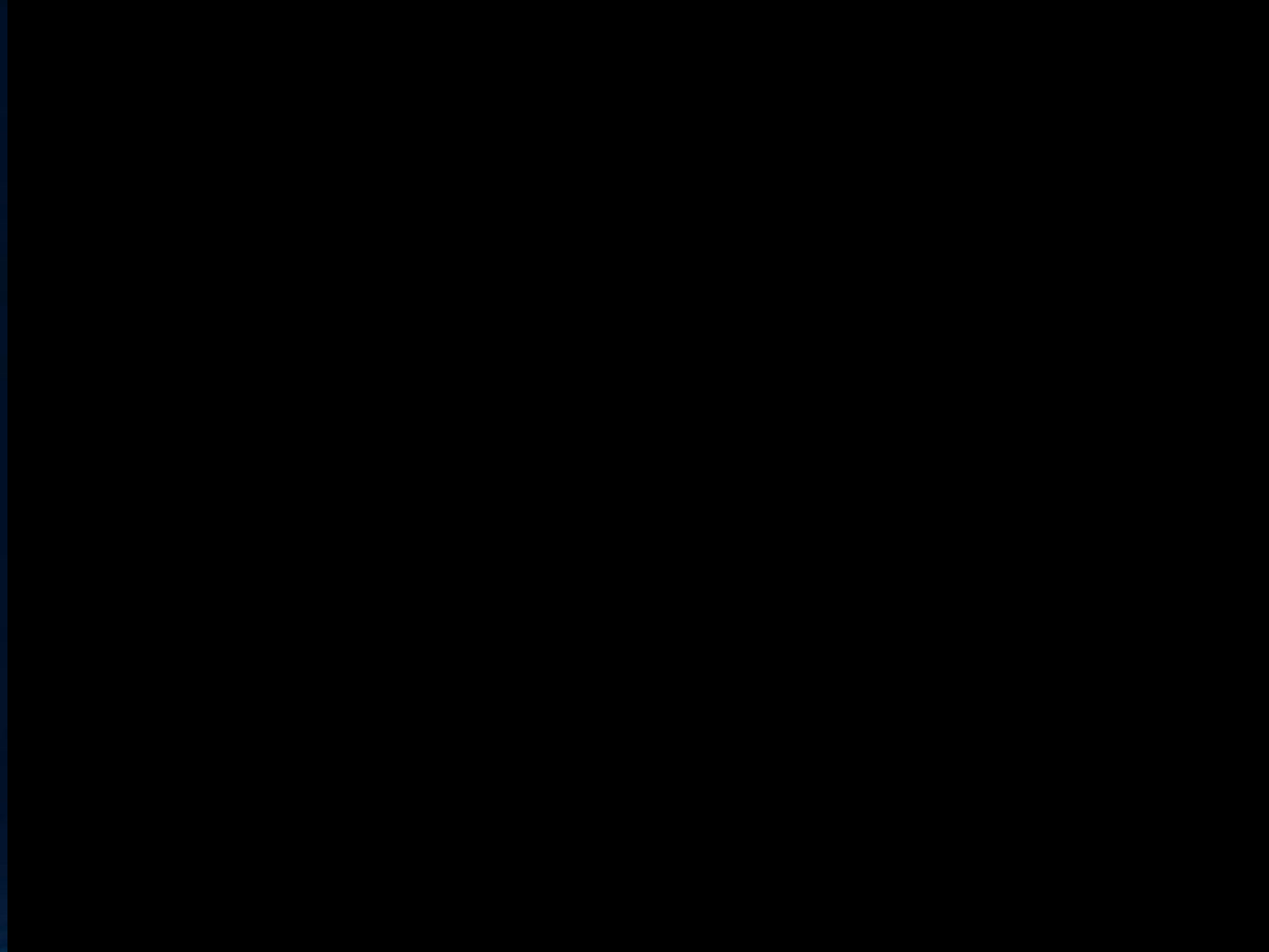
Testing
3 second lag



Testing Linear/Quadratic prediction w/ and /wo Interpolation



Testing
3 clients



Possible improvements

- Event based i/o
 - Pause event
- Multiple clients improvements:
 - Every client can join at any time without a button click on the server's side
 - Queue of client's waiting to join

Thanks!

MATTHEW WALLACE
NETWORK SYSTEMS FOR GAME DEVELOPMENT (CMP303)