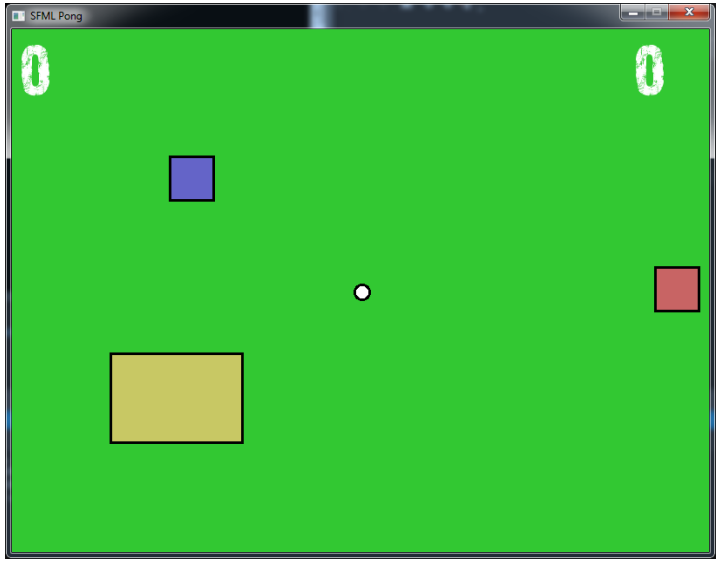
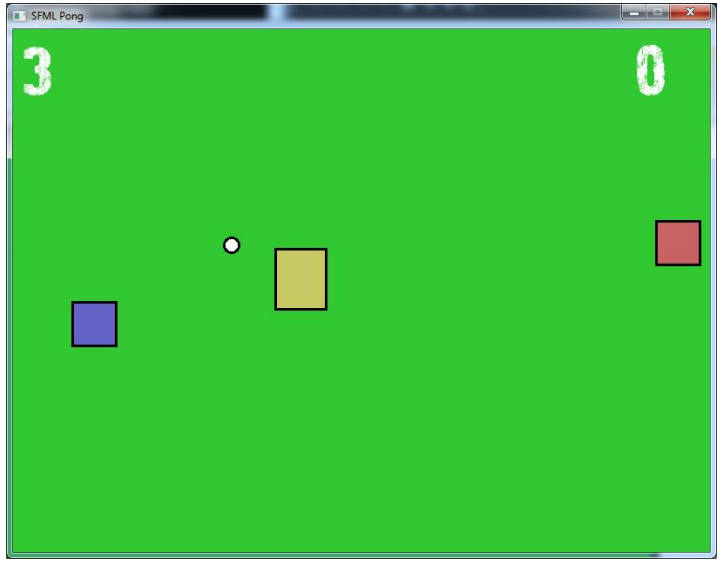
Network Systems for Game Development

GREG BALBIRNIE - 1500405

The Game

- Two players race each other to collect ten dots.
- The winner is whoever collects the ten dots first.





Peer to Peer vs Client-Server

	Advantages	Disadvantages
Peer to Peer	• Less latency	• More ambiguity of object positions • Potentially more connections per player
Client-Server	• Holds master game world so less ambiguity	• Higher latency

I chose client-server

- Master game world can handle positions of dots and barriers.
- Higher latency is dealt with using prediction.

Application layer protocol design

- Send function sends a packet with data and an identifier.
- Receive checks the identifier to see what the message will contain and then deals with the input appropriately.
- Set to non blocking mode to allow the game to continue when nothing is received.
- Client uses three states. Inactive, Lobby and Playing.
  - Inactive – Before anything has happened.
  - Lobby – When waiting for the game to begin.
  - Playing – The main game state.

Transport layer protocol

- I chose UDP over TCP.
- No need to set up a connection first. Can send and receive easily.
- Won't get split up across the network so both sides will receive all necessary information.
- UDP is faster – Messages can be sent more frequently.
- UDP is less reliable – Messages are more likely to be lost but that's not a problem.

Networking API

I have used the SFML Sockets API.

- The game was made in SFML as it was simple and could create a graphical interface.
- SFML Sockets is made to work with SFML projects.
- The packet class made sending different kinds of messages simple.

Prediction/Interpolation

- I have used the linear prediction model.
- Takes less time than the quadratic model to make up for errors.
- Simple game, so doesn't need anything more complex.
- Used a deque to store previous positions for easy access to members.

Critical Evaluation

Positive

- Sending an identification string at the start of the packet made sending and receiving more flexible.
- Waiting for both clients to send a message ensured that all systems could communicate before the game started.

Negative

- The sent time variable is sometimes sent incorrectly leading to a null predicted position.
- Predicted positions are created less frequently than they should be making movement stutter.