NETWORK SYSTEM FOR GAME DEVELOPMENT

SCHOOL OF ARTS MEDIA AND COMPUTER GAMES

COMPUTER GAMES APPLICATIONS DEVELOPMENT

ANDREW PETERS

1502220

NETWORK ARCHITECTURE

- Peer to Peer
- Only a few players
- Wanted to make them equal
- First player starts the server and other clients join it but have the same rights
- No traffic
- Doesn’t affect the other players if one crashes

WHY NOT USE UDP

- UDP packets can be lost, sent in the wrong order, can be duplicated
- Each client will have to bind to different port number
- If a necessary packet of information is lost you need to find a way to detect this and resend it such as if a player dies or the game ends
- The packets are important to reach the other clients as it is only a 2 player game so loss of packets could be detrimental

TCP CONNECTION

Why choose this?

- 3 way handshake means a reliable connection
- Server just needs to listen to port and client will connect to it
- Packets are put into a queue in order to be read
- Doesn’t send data too quickly for the connection to handle

SOCKETS API

SFML

- Builds on the socket API
- Allows us to easily listen in to ports
- Sending packets is made very simple
- Makes building a game around the libraries very efficient without taking away too much of the control surrounding sockets and packets

ASYNCHRONOUS I/O

- Sfml uses a function called setblocking
- When we set it false it goes of and does the calls in the background
- This allows the program to continue and not wait to see if it has been received
- This has been set up for when the game window opens

PREDICTION TECHNIQUES

Linear Prediction

- The game checks to see if the player has moved
- If it has then it will check the last 2 known values of the opponent
- It will continue along this path until it receives another packet stating otherwise
- Still makes sharp turns but that is fine because player can only move left right up and down

HOW THE PROGRAM FUNCTIONS

- Won’t work well if there is more than 2 players and it would slow down because of the tcp sending packets
- If there is not enough data tcp might not send the packet
- With more players if packets are lost tcp will ask for them again and this could slow down the whole network

QUESTIONS?