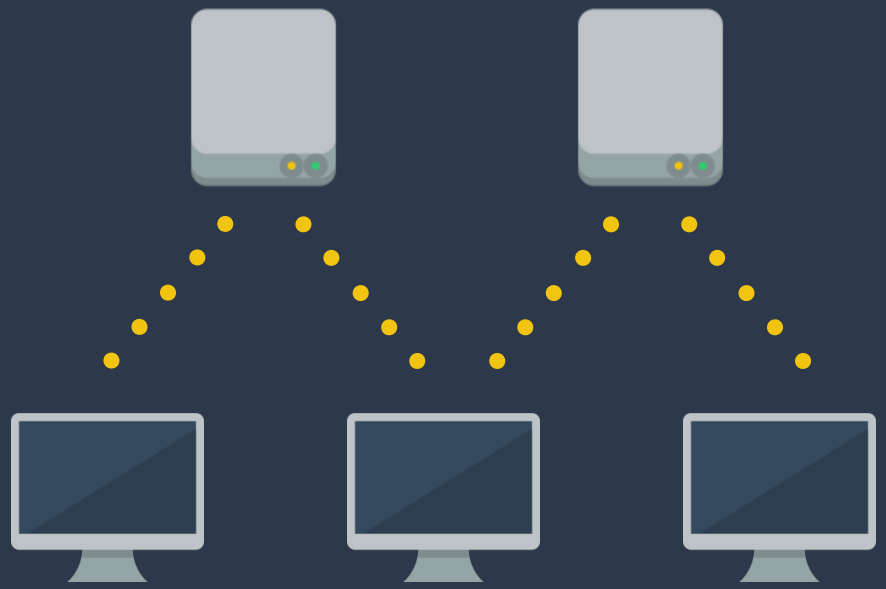
Who Shot First?

Implementing fundamental networking

Networking Concept

Why do we need networking?



- Networking allows for support of multiplayer, instant messaging, and so much more!
- Netcodes come in all shapes and sizes, different developers will use different implementations.
- Different game genres will also need different protocols, i.e. turn based versus First Person Shooter.

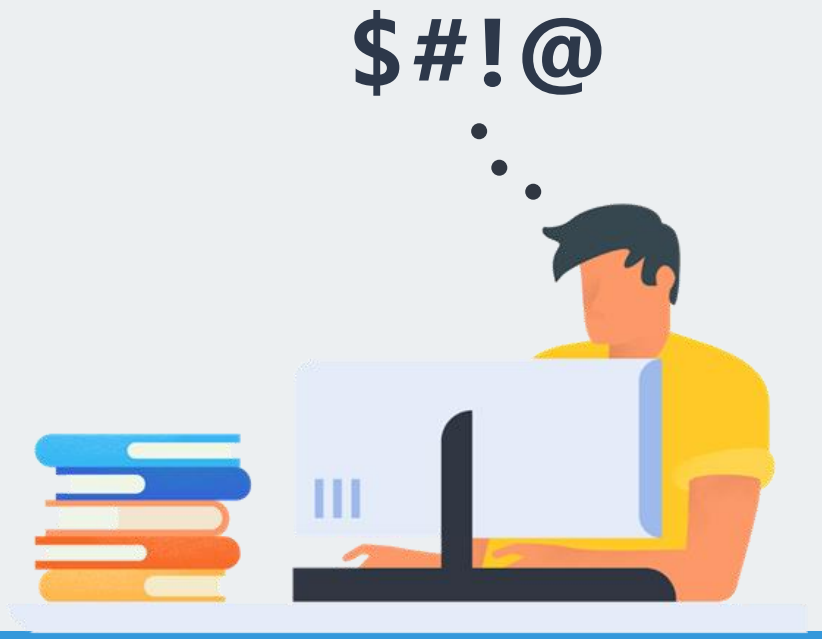
More Networking

Why is networking so hard?

Experiencing **LAG**? No problem!

Just fix one or more of the below causes:

- U.S. broadband service
- Internet traffic load
- Weather interference
- Distance to modem
- Ethernet cable length
- Computer LAN settings
- Firewall configuration
- Cheap wireless adapters
- Bad network programmers
- . . .



Demonstration

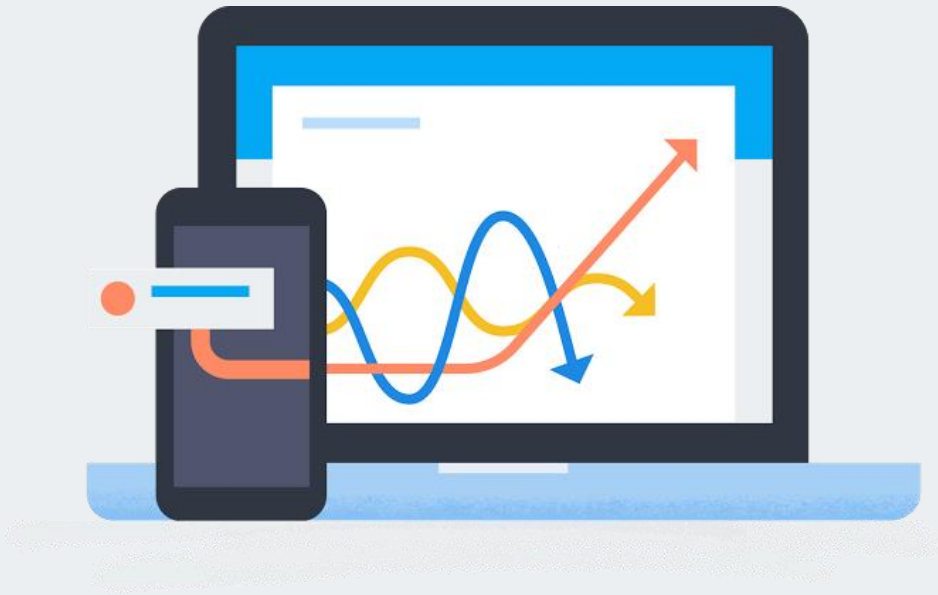
How awesome is it?

1. Chat room and messaging
2. Entity creation and movement
3. Lag simulation and analysis

Implementation

What are the features?

1. Simple peer-to-peer chat messaging in game entered through terminal.
2. User Datagram Protocol network for real-time player synchronization.
3. Lag simulation that portrays input delays across the network.



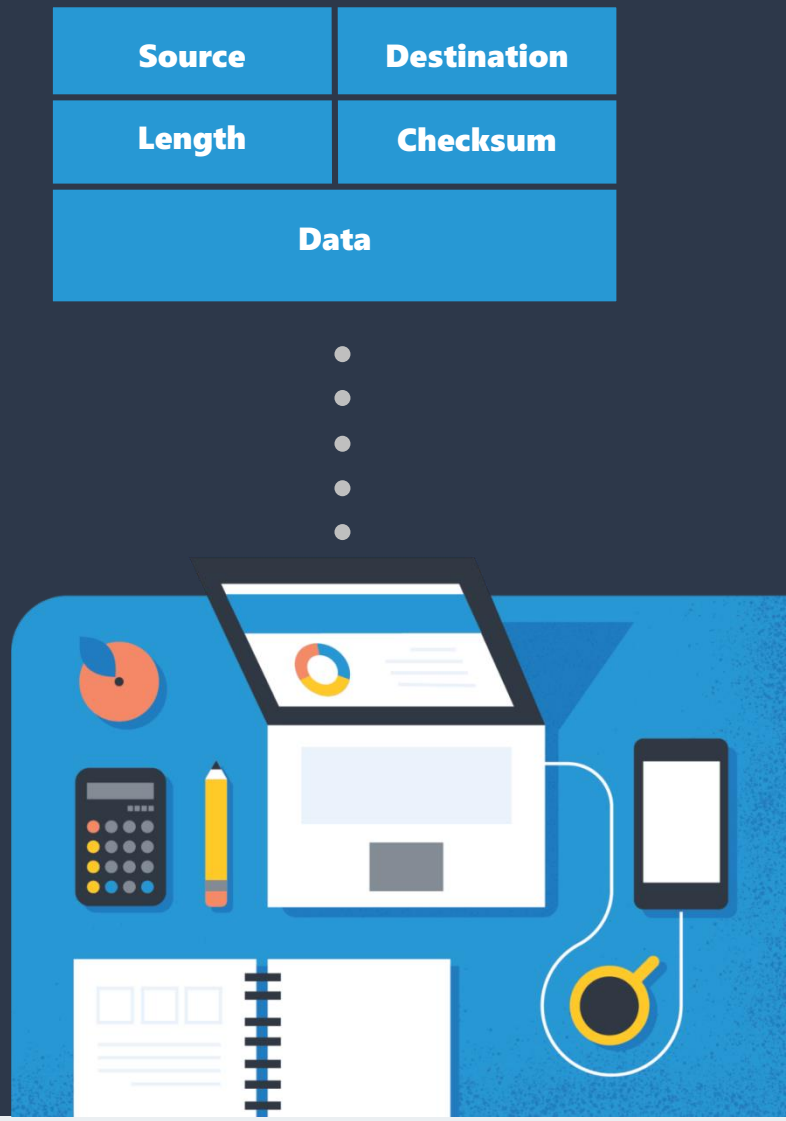
Behind the Scene

How does it work?

Client/Server run unique threads to constantly receive input through socket at specified port.

Each packet sent is 1024 bytes, the first four bytes represent a data type, i.e. movement, add player, etc.

Lag simulation is done through a thread sleeper method that sleeps at specified min threshold.



Challenges

What were the difficulties?

Implementation	Integration	Testing
Implementing the UDP server and client was challenging when data was lost at times. Wireshark sniffing definitely helped.	Integrating with the game engine probably took the longest time, since Visual Studio's errors were unhelpful, and including certain libraries across projects was tricky.	Testing lag simulation and the implementation overall was easier. Changes were made quickly if problems occurred.

Future

Can it be more awesome?

1. Non peer-to-peer network implementation
2. Predictive movement and trace back on faulty detection
3. A generally better looking interface and UI widgets
4. Projectile and hitscan transmission over the network

Before You Quit

Who really shot first?

Thanks!

QUESTIONS?

TEAM

Matthew Bu

RESOURCES

Star Wars GIF

TERM

Game Architecture
Spring 2017