

COSC 465 - Computer Networks

Capstone Proposal

March 27th, 2024

Project Name: Net-Raiders

Project Objective:

Net-Raiders will be a networks themed multiplayer game for the web browser. Players compete to accumulate the highest ‘bandwidth’ — a virtual points system acting as their score — during a game session. There will be pickups spread across the map, which when collected can be used to boost one’s own ‘data transmission’, such as upgrading to a fiber optic network, or to hinder another player’s transmission, such as throwing a bandwidth bomb. The pickups and strategy in the game will be related to networks topics taught in the COSC 465, Computer Networks, and will seek to teach players about the tradeoffs of using different network protocols and mechanisms in a game.

To maximize accessibility, Net-Raiders is built for web browser gameplay. NR will be using a novel approach to browser networking, WebTransport, which utilizes UDP instead of TCP/IP for faster, more efficient data transmission. Usage of UDP has traditionally been impossible in browsers, limiting the capabilities of browser games. NR will use a client/server architecture, where nearly all gameplay computations, such as player position, health, and score, will be calculated on the server. Clients will simply tell the server what their inputs were, the server will compute the player state, and broadcast the players state to all clients.

Project Motivation:

This project is designed to provide future Networks students with a fun and engaging way to understand the various concepts taught in the course. We hope to encourage students to think about the development process of the game, including the network design considerations that were essential to its creation. Additionally, undertaking this project allows us to gain a deeper understanding of the

WebTransport protocol's functionality, acknowledging its critical role in the future of web real time networking.

Prior Artifacts:

- 1) Research indicates significant advantages in using video games as educational resources, as outlined in the study "[Application of the Educational Game to Enhance Student Learning](#)." The goal of this project is to assist both present and future students of Networks to grasp the principles of the subject matter.
- 2) The conference paper "[Aspects of Networking in Multiplayer Games](#)" does a great job at covering the fundamentals of various multiplayer Networking architectures. We will utilize the client/server architecture, however there are various other types of architectures, such as peer-to-peer or server-network models.
- 3) [WebTransport](#), a novel networking protocol developed by Google, utilizes UDP instead of the traditional TCP/IP. This change facilitates significantly faster communication between clients and servers.
- 4) [Protocol Buffers](#) are a binary serialization library similar to JSON but offer greater data efficiency. ProtoBufs are fast and practical for the frequent transmission of small chunks of data very frequently, like player health or position, between the client and server.

Project Plan:

1st milestone: Network Architecture

- 1) Get WebTransport working in JavaScript with a designated python server.
- 2) Setup basic infrastructure of network update loop on server and client, and ensure updates go from client->server-> all other clients.
- 3) Create a basic system of actual gameplay on Unity3D side. Game logic should be kept simple and to the point to avoid complexity.

2nd milestone: Playability

- 1) Have the main networks themes and educational parts of the game sorted out.
- 2) Have the main aesthetics of the game built out. Everything should be reactive, and should follow good UI/UX paradigms. There shouldn't be complex menus or features. When a player goes to the site, they should be one button click away from being in a match with other players.
- 3) Setup client side prediction. In the time between when a player sends an input and when they hear back from the server, there is always latency. Setup the ability for a client to 'predict' where they should go next based on an input. If the client is honest, their new position should be identical to the server's next computed game state. Keeping game logic simple and direct is important for this, so we can replicate game logic on the server side (python) and client side (C#). We don't need to be *super* strict in the 'authoritativeness' of our server, like a competitive FPS would be, but something 90-95% close to that level of authority is the goal.

ARTIFACTS TO DELIVER:

An educational, and fun, web game which utilizes WebTransport to achieve UDP multiplayer networking in the browser. The built out game artifacts should be well organized and documented, so that anyone could replicate, fork, or extend the project however they see fit, with a clear understanding of how the whole games infrastructure works. Again, this multiplayer game is built to be an educational tool for people trying to learn networks. Additionally, the site itself should be hosted on a proper server with TLS encryption (server hosting games static content won't be using QUIC so will need TLS).