

snek - Network Challenges

Utilized UDP and TCP

Clients use TCP for sending packets to the server. The server broadcasts UDP messages to clients. These broadcasts are repeated twice to prevent packet loss.

Synchronized time steps

In our game, when a client wants to move, they send a movement packet to the server. But they wait for the progress packet from the server to move the snakes. And the server sends the progress packet when it receives a movement packet from every client. This is like an ACK mechanism.

Small packets

Since movement packets are sent so frequently we wanted to make sure that they're small. In our system, players send only their snake id and their movement direction to the server. To achieve that we're running a game engine both on the client and the server independently.

Packet loss control mechanism

Messages from the server are incrementally indexed and each player keeps track of that to check if they missed any. If so, the player leaves the game. We needed this because each client runs their game engine independently and a lost message can mess it up.

Timeout mechanism

Since the server waits for each snake to send their movement update, a laggy player can slow down the entire game. So, we implemented a timeout mechanism. Let's see it in the diagram below, After every progress packet sent, the server starts a timer for each snake (in our case, 2 seconds). If a snake doesn't send a movement request before the timer stops, the server kicks that snake from the game to make sure the game runs smoothly for other players.