

NetWork Project

1801815, Games Application Development



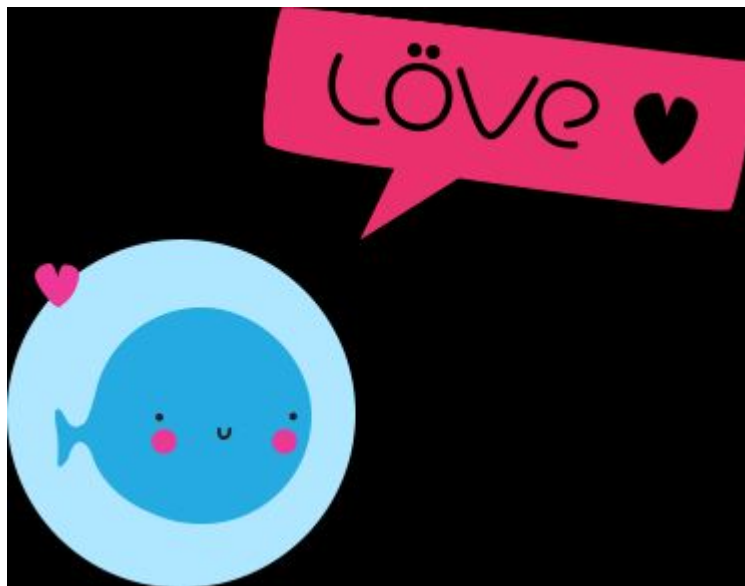


The game





The FrameWork





Protocols:

Transport layer protocol: UDP

- real time game
- gameplay is more important than reliability

Application layer protocol:

- message ip and port
- a message with a sequence of strings and numbers is sent and dissected in order, first part is always the type

message = "update_health 4 60"

- timeout is zero(time of waiting messages, avoids blocking)



Architecture

Client-Server was chosen:

- the server runs every client logic
- client only get world state and sends input
- real time, easier to organize, helps with desync, less work for client, safer for avoiding griefing



Code Structure Server

Update function:

- receive_client_data(), player id, input(shoot, etc)
- update logic
- check collisions and send health changes
- send object death
- world state: object positions and directions



Code Structure Client

Client has a copy of every game object for update purposes

Renders for the most part

Sends:

- connection request, waits for response

- input: movement and shooting

Waits for server: position, health, death



Interpolation

Server receives directions

Interpolates positions

Press “I” to toggle interpolation



Discussion

The game is very simple, hard to feel techniques (ex.interpolation)

Making the logic centralised worked fine

Game can be scaled for more player easily

Setting up server on personal computer is annoying (for other to access and play it online)

Debugging is hard in some situations

Lua library is very useful to work with strings in network(regex matches, formatting)