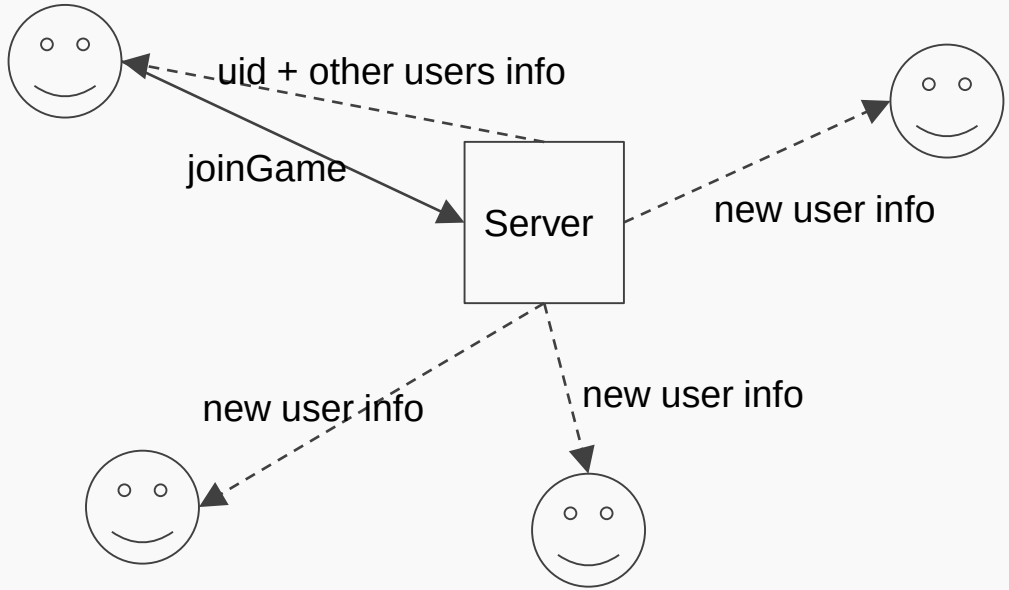
Discussion 7

Client-Server Interaction

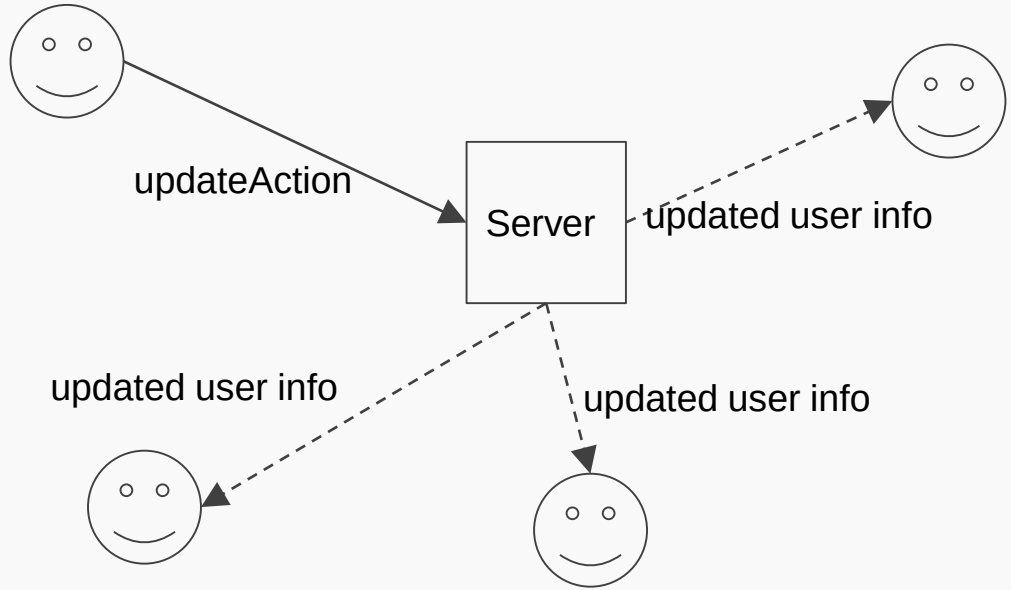
Client-Server

- Setup rpclib:
  - Download rpclib from here
  - Create a build folder under .../rpclib-master/
  - Use cmake-gui compile sln files in build folder:
    - Source code: ../rpclib-master/
    - Binaries: ../rpclib-master/build
    - Click “configure”, then click on “generate”
  - Build.../build/rpc.sln with “x64” + “Release” (or whatever mode you’ll be building your

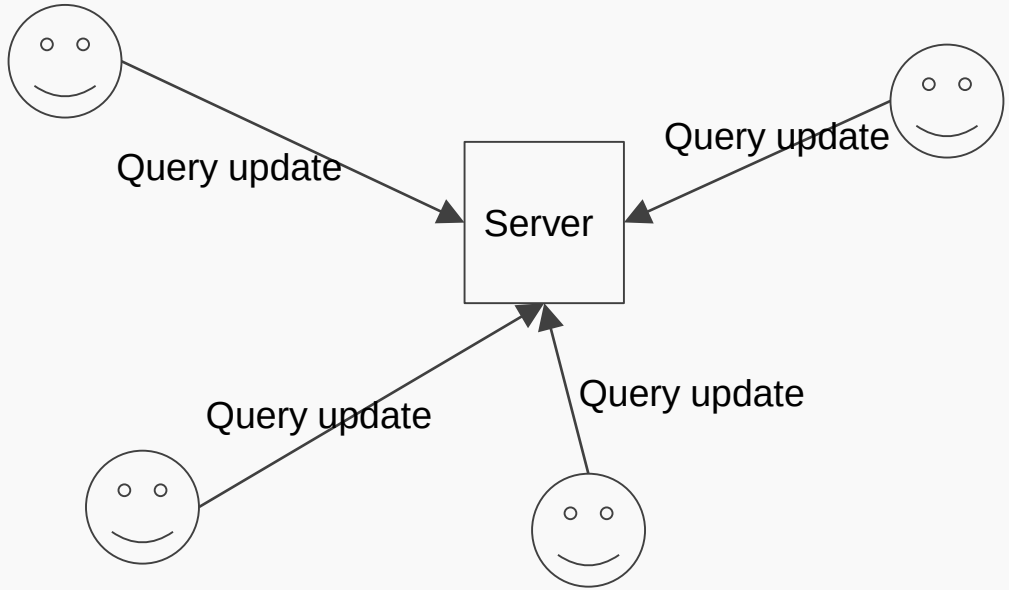
Client-Server



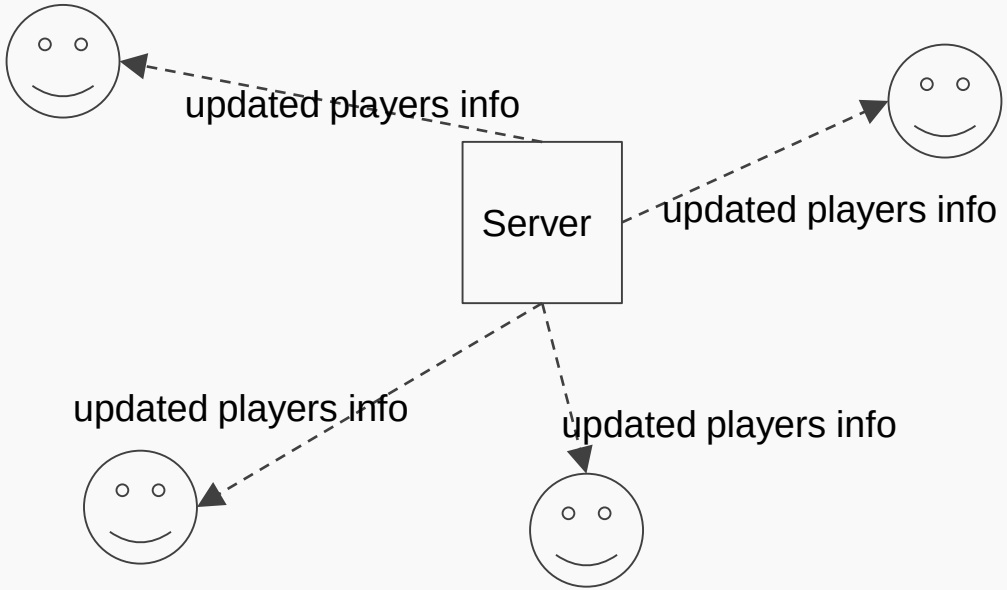
Client-Server



Client-Server



Client-Server



Server

The server has its own update loop where it will:

- Handle user queries
- Run physics/game logic
- Send updated scene information to each client

```
int joinGame(){
    id = -1;
    ...
    return id;
}
```

```
int main(){
    rpc::server srv(8080);
    srv.bind("joinGame", &joinGame);
    srv.run();
    return 0;
}
```

Things to consider:
Assign an unique user id for each player.

Store a list of current player information with timestamps.

Set a maximum number of players.

Handles player look-up for p-2-p actions.

Consider interpolation if needed.

Consider using timestamp to see if server needs to send all updates to the client.

Client

The client sends queries:

- Renders the scene from data returned by the server:
  e.g. other players' location, orientation, etc.
- Sends input information: e.g. shooting an enemy etc.

```
int main() {
        rpc::client c(8080);
        uid = c.call("joinGame",
...).as<int>();
        ... [render loop]
        c.call("updateGame", ...)
        ... [render loop end]
        return 0;
```

Things to consider:

You may want to have the following classes:
Enemy, Player, Network, Game

Consider store the last updated timestamp from the server so that the client can send it and let the server decide if all updates are needed for this client.

Helpful References

1. Here is a good reference on how to integrate a server-client structure to your openGL game with SDL library. Using rpclib would be using a very similar logic:
   https://www.youtube.com/watch?v=iJfC4-yNnzY

2. Read through rpclib reference to see what other client functions are available:
   http://rpclib.net/reference/