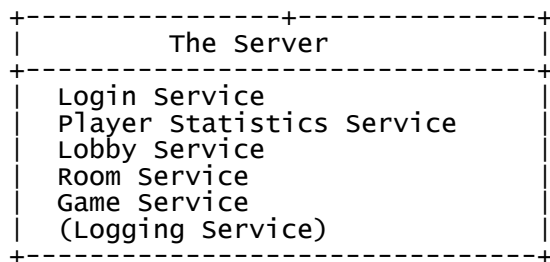


architecture

We will use the client-server model.

The Server



Login Service:

keeping track of online players. Log player in. Create new player. Log player out.

Player Statistics Service:

keeps track of a record for each client. This record includes player name, client ID, number of game played, and wins. The service will include updating client's record and retrieving client's record. This service is private and can only be used by the server itself.

Lobby Service:

will be in charge of keeping a record of all the rooms, listing the players online, listing the rooms, displaying players' statistics

Game Lobby/Room Service:

creating a room, listing players in the game room and preparing game and starting the game.

In the case the client choose to load a game from his/her hard drive, this local map will be sent to the server and the server will send the map to all the other clients.

Game Service:

serve as a message broker for the all the clients in a game. The server will keep a map that is in sync with the latest map of the clients. It will communicate with client using Remote Procedure Call.

If a client had made a game move, it would send a message to the server, the server would then update itself's map and then push the update to all the other clients. The update will only succeed if the sender received a confirmation from the recipient, if not succeed, it will block and wait until timeout, and the client will be considered disconnected from the server.

The Game Service will also be in charge of notifying the client of the turn change, the game win and ending the game.

Only the server can generate trees and gold at the end of each turn.

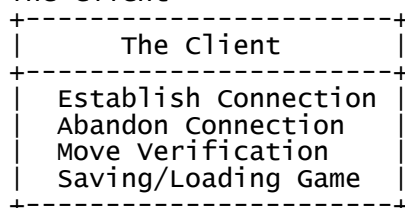
Logging service:

keeping a log of the current game. This service is also private.

Storage:

player stats should be permanent and will be stored in file or a database. rooms and games should be in RAM, and are destroyed when nobody remains in the room/game.

The Client



Establish Connection:

when a client is started, it will automatically establish a connection with the server.

architecture

Abandon Connection:

when the client exits, it will disconnect from the server.

Move Verification:

verify each game move is valid before making it.

Saving/Loading Game:

each client can save game, game will be saved in local drive. And when in the game room, The host(whoever created the room) will be able to load the game from his/her hard drive. Before the game started, the loaded game will be sent to the server, and the server will send the game to all other clients in the room.

Storage:

currently playing game is stored in RAM. And gameSave will be in hard disk.