JOGO INTERATIVO ONLINE

Giovane Hashinokuti Iwamoto, Paulo Henrique Mendonça Leite

Universidade Federal de Mato Grosso do Sul - UFMS

giovane.iwamoto@ufms.br, paulo.h.leite@ufms.br

Resumo. *Relatório de um aplicativo peer-to-peer híbrido para representar uma rede de jogos online, composta de um servidor e vários jogadores (usuários). A interação entre servidor e usuários é ao estilo cliente-servidor e a interação entre jogadores é ao estilo P2P.*

1. CLIENT

Após a inicialização do Servidor de Autenticação e Informação (SAI), os usuários podem se conectar a este e realizar o cadastro, login ou finalização do aplicativo.

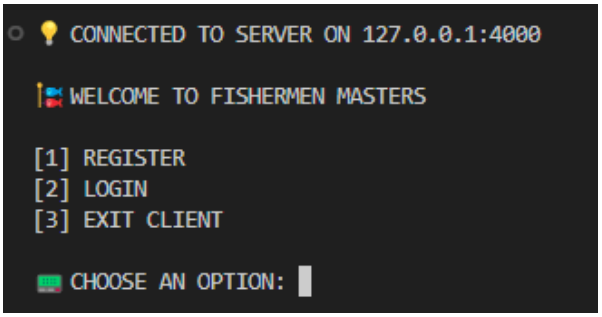


Figura 1 - Tela inicial

Na sessão de cadastro, o usuário informa o nome e senha e estes são salvos na base de dados do servidor. Existem tratamentos de exceções para tentativas de registrar um usuário que já está cadastrado e em casos de o input não for preenchido.

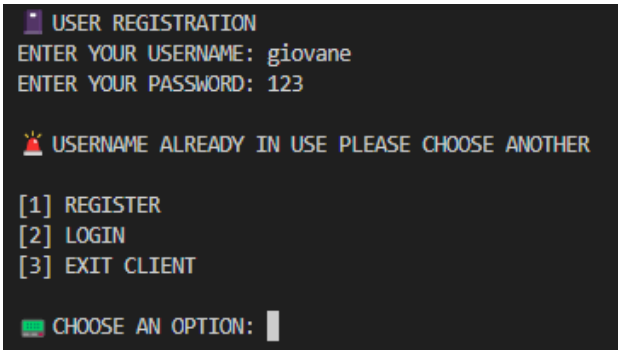


Figura 2 - Tela de cadastro

Durante o login existe também um sistema de autenticação e validação de existência do usuário, senha correta, se já está logado na aplicação e outros casos de inputs não desejados. Após o usuário ser autenticado, o lobby com diversas funcionalidades é apresentado.

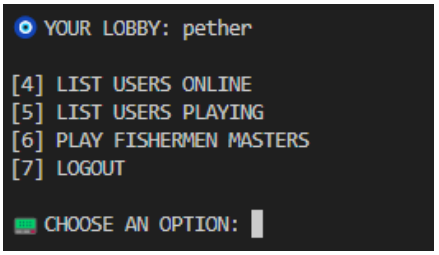


Figura 3 – Lobby

A opção de listar todos os usuários onlines mostra os users que estão atualmente logados e conectados com o servidor principal e que não estão atualmente vinculados a alguma partida, sendo assim, estão no lobby. A listagem contém a relação entre usuários e seus respectivos status, ip e porta. No retorno da lista o próprio user que fez a solicitação nunca é listado.

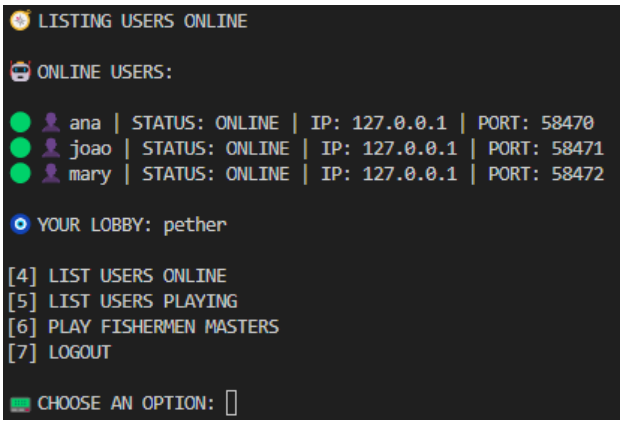


Figura 4 - Listagem usuários online

Quando solicitado a listagem de usuários que estão atualmente jogando a exibição é quase semelhante à listagem anterior quanto às informações de retorno porém com o diferencial de que é exibido a relação entre os oponentes.

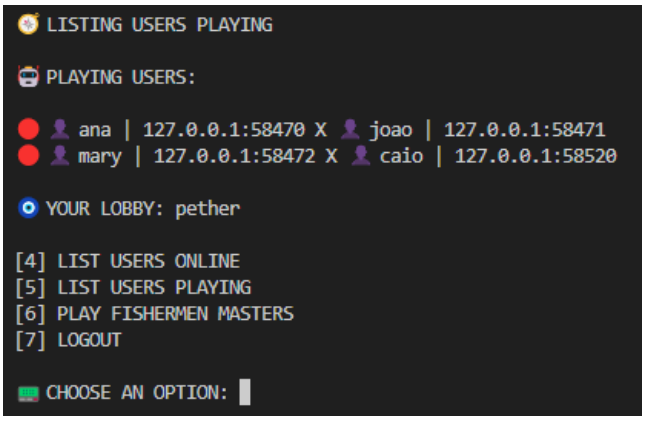


Figura 5 - Listagem usuários jogando

Para iniciar um jogo, o usuário pode desafiar um oponente que esteja online ou atualmente dentro de outra partida. Nesta opção existem tratamentos em casos de que o adversário esteja offline, não esteja cadastrado no banco de dados do servidor ou que o nome inserido seja o do próprio user. Caso desafie um usuário que esteja disponível para jogar contra, uma notificação será enviada e o client

aguarda pela resposta do convidado que pode ser do tipo aceitar ou recusar o convite. Na situação em que o oponente aceita o desafio o jogo se inicia, caso contrário retorna ao lobby.

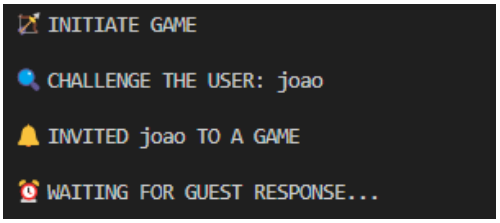


Figura 6 - Convidar jogador

Figura 7 - Convite recusado

De volta ao client, caso o usuário opte por realizar o comando de logout, este se torna offline e realiza a desconexão com o servidor SAI. Por conta própria, o server também é capaz de observar a desconexão inesperada dos usuários que anteriormente estavam conectados e atualizar os seus status devidamente.

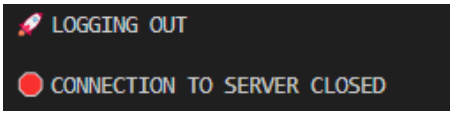


Figura 8 - Logout usuário

2. SERVER SAI

O SAI Server fica responsável de gerir dados das conexões feitas entre os clientes, para isso foi definido que guardaremos hosts, portas, usuários online, usuários jogando. Com esses dados sendo guardados pelo SAI, temos toda a informação necessária para fazer o peer-to-peer híbrido. Ele não interfere na conexão entre os clientes quando ela já está formada, nesse momento toda a lógica fica a critério do cliente.

Em relação ao armazenamento de dados, a aplicação gera dois arquivos, o primeiro, database.txt, guarda as seguintes informações:

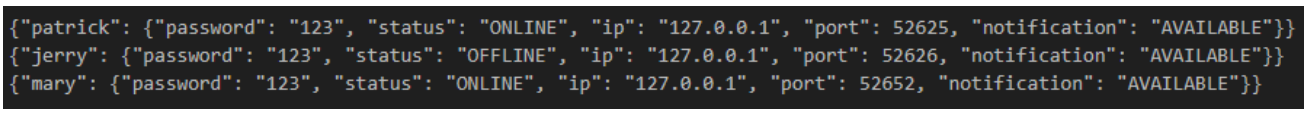


Figura 9 - Arquivo database.txt

Como podemos ver, é guardado as informações: usuario, password, status, ip e porta, notification, com essas informações o SAI Server consegue fazer o intermédio entre os clientes, inclusive sabendo em tempo real, os status dos seus usuários, caso estejam, offline, online ou playing, por exemplo. Não há limite para usuários nesse arquivo, com exceção da memória não exceder o espaço em disco. Um dos dados guardados é notification é definido como disponível

ou ocupado e serve para não interromper os usuários enquanto eles estão recebendo um convite de conexão.

O segundo arquivo é o game.log, ele é responsável por guardar informações dos estados em que os usuários estão no aplicativo, tais como quando o usuário faz o registro, quando se faz o login na aplicação, fica inativo, quando estão jogando, quando desconectam, entre vários outros. Além disso, é guardado a data e a hora em que o evento ocorreu.

Figura 10 - Arquivo game.log

A seguir listamos algumas das funções que o SAI Server tem, de forma mais técnica:

- load_users_from_file: carrega o SAI Server com os usuários do arquivo database.txt.
- save_users_to_file: salva os dados do usuário para o arquivo database.txt.
- get_user_address: pega o ip e porta do usuário.
- start: abre o socket e começa a ouvir os clientes.
- handle_client: lida com um pedido de conexão vindo do cliente.
- add_user_connection: vincula a conexão com o client.
- remove_user_connection: remove a conexão realizada com o usuário.
- get_user_connection: guarda a conexão com o usuário.
- handle_message: o servidor trata os comandos recebidos pelo usuário.
- register_user: função que registra o usuário.
- login_user: função que loga o usuário na aplicação.
- send_online_users: retorna ao usuário a relação de users online.
- send_playing_users: retorna ao usuário a relação de users em partida.
- initiate_game: verifica se o host e guest conseguem começar uma conexão para jogar.
- start_game: efetivamente muda os status dos usuários para jogando.
- send_guest_conn_port: envia para o usuário convidado que porta ele deve se conectar.
- game_over: muda os status do jogador para online.
- set_invite_status_available: permite que usuário lide com convites.
- log_event: adiciona eventos ao arquivo game.log.
- stdout_event: adiciona eventos na saída padrão do terminal.
- generate_unique_token: gera um token único para a partida.

3. JOGO

O jogo ‘Fishermen Masters’ é uma disputa entre pescadores onde o mapa de pesca lembra bastante o estilo dos jogos de batalha naval, os quais usam sistemas de coordenadas para se remeter a um ponto de escolha. Com o usuário logado vá para a opção 6 para iniciar a escolha do oponente.

Então você poderá escolher um jogador para desafiar, mas antes disso liste os usuários online ou jogando para saber o nome deles, então digite o nome do usuário:

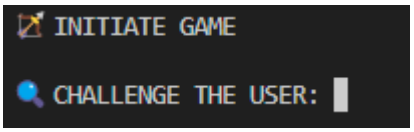
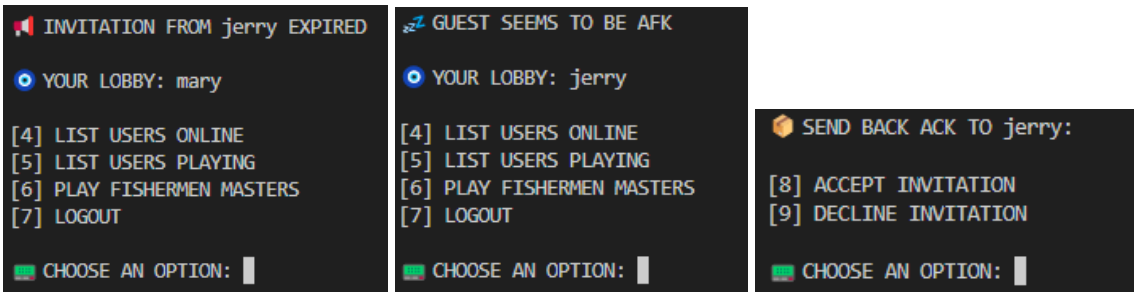


Figura 12 - Escolha de usuário a ser desafiado.

Basta esperar pela resposta do usuário desafiado. Se atente ao fato que existe um tempo de resposta que deve ser respeitado para aceitar o convite, caso contrário o convite seja expirado, o hospedeiro recebe a mensagem “*GUEST SEEMS TO BE AFK*” e o convidado recebe “*INVITATION FROM HOST EXPIRED*”.



Figuras 13 - Tela de resposta para convite expirado para guest e host respectivamente e tela de envio de resposta do convite.

Aceito o convite, a tela inicial do jogo é gerada para ambos e assim dá-se início a partida.

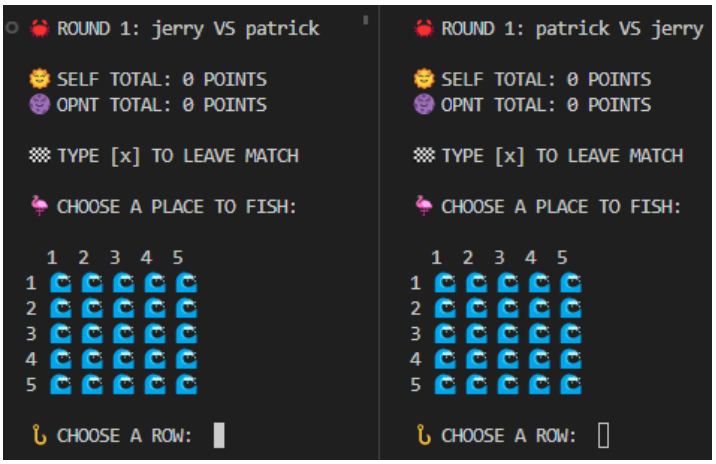


Figura 14 - Tela de início da partida entre jogadores.

Cada jogador tem 25 escolhas para fazer, onde em cada rodada cada um deverá submeter sua opção através de uma linha e coluna válida, no momento em que ambos enviarem as respostas serão

mostrados todos os animais de cada coordenada do oceano e o que cada jogador pescou ao escolher uma posição. Com base no animal pescado a pontuação é atribuída ao total.

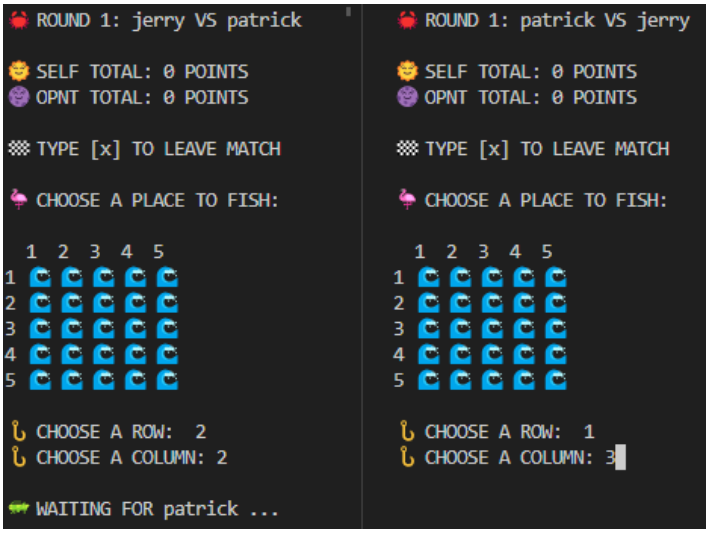


Figura 15 - Jogadores host e guest com suas escolhas feitas.

Figura 16 - Gabarito do oceano de cada player.

Cada animal tem uma pontuação pré determinada e uma raridade de ser gerado em cada coordenada do oceano. O jogador tem sua pontuação total como a soma do valor de cada animal fisgado ao decorrer das 5 rodadas.

Segue abaixo a listagem de cada animal com suas respectivas pontuações e raridade:

Animal	Pontuação	Chance de encontrar
SHARK	100	5%
SQUID	80	10%
LOBSTER	60	20%

FISH	50	30%
SHRIMP	30	35%

Tabela 1 - Tabela de pontuação dos animais e sua chance de ser encontrado.

Após as cinco rodadas é mostrado as informações da partida e a conexão peer-to-peer entre os jogadores se encerra, retornando assim ao menu principal.

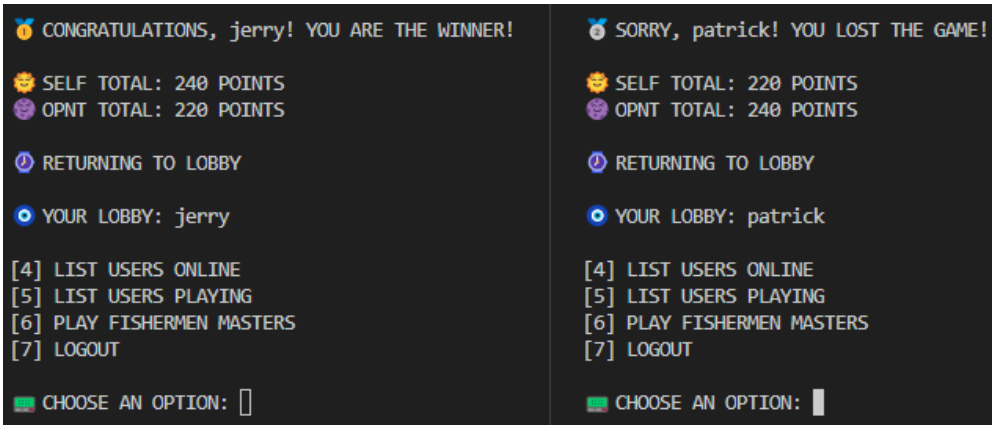


Figura 17 - Tela de vitória e derrota.

Assim como mencionado anteriormente no processo de convite de usuários, é possível também receber notificações durante uma partida de usuários externos. A lógica segue sendo a mesma, porém caso aceito um convite, o encerramento da partida em andamento se dá por imediato e se inicia uma nova com o remetente. Caso o usuário recuse o convite, o progresso da partida atual decorre normalmente.

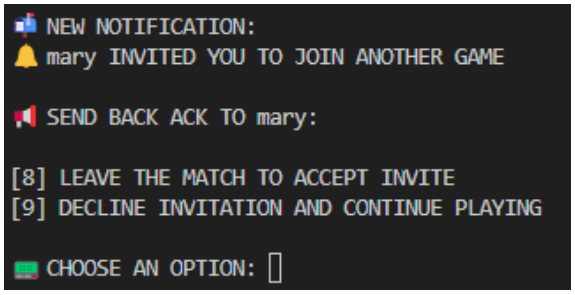


Figura 18 - Convite externo durante uma partida.

4. SOLICITAÇÕES IMPLEMENTADAS

- ☑ Listar usuários online.
- ☑ Listar usuários jogando.
- ☑ Interação sem interferência do SAI.
- ☑ Encerrar conexão ao terminar o jogo.
- ☑ Receber solicitações para iniciar jogos com outros usuários.

- ☑ Jogador consegue encerrar a partida a qualquer momento.
- ☑ Servidor armazena usuários online e jogando.
- ☑ Arquivo log com registro de eventos marcando a data e hora.
- ☑ Jogo interativo com dois participantes.
- ☐ Jogo interativo com mais de dois jogadores.
- ☑ Desenvolvido em Python.
- ☑ Servidor não participa do jogo e é concorrente.
- ☑ O usuário avisa em caso de desconexão, mas o servidor também verifica se algum usuário "morreu" e remove da lista de online.
- ☑ O usuário verifica se o outro jogador ficou AFK durante a interação, altera seu status de online para AFK e avisa o servidor.

EXTRAS: Tratamento de autenticação para login e registro de usuários com mensagens de erro para diversos casos como usuário ja ativo no sistema, username já cadastrado, senha incorreta, input vazio entre outros. Tratamento para banco de dados em que o servidor avisa se o arquivo não existe e cria um automaticamente para inicialização e retorna em casos de o arquivo estiver vazio. Interface do usuário na saída padrão é sempre limpa e com mensagens claras e coesas quanto à interação com o Client para obter-se uma melhor sensação de simultaneidade durante a execução. Tratamento de stdin para todos os casos permitindo que o usuário possa inserir apenas os comandos permitidos. Mensagens personalizadas de desconexão e de unhandled exceptions com o servidor SAI. Durante a execução do game o jogador pode optar por deixar uma partida, retornando ao lobby e encerrando a conexão com o oponente, assim, é atribuído ao usuário que deixou a partida como derrotado e ao adversário como vencedor automaticamente.

4. COMO RODAR O FISHERMEN MASTERS

É importante considerar que a interface exige uma grande dependência dos emojis presentes em seu sistema operacional e no próprio shell para uma melhor experiência. Recomendamos utilizar terminais que suportam esses tipos de caracteres como o do VSCode.

O projeto foi construído em Python e é facilmente executável. Primeiro é necessário rodar o SAI Server no terminal utilizando o comando:

python sai_server.py

Agora será possível rodar os clientes, em um novo terminal rode o seguinte script localizado na raiz do projeto, é possível rodar ele em vários terminais diferentes, cada um sendo uma conexão possível:

python user_client.py

Script para limpar o banco de dados e o arquivo game.log, caso seja necessário:

python clean_script.py