

Trabalho 1

Jogo de tabuleiro e Sockets

Índice

Índice	1
Objetivo	1
Características	2
Tabuleiro	2
Jogadores	3
Casas	3
Indicador de turnos	4
Controles	4
Pendências	4
Regras do jogo	4
Pré-requisito	4
Durante o jogo	4
Critério de vitória	5
Implementação	5
game.py	5
run.py	5
server.py	5
Conclusão	5

Objetivo

- Desenvolvimento de um jogo de tabuleiro que funcione com no mínimo 2 usuários em rede e utilizando *sockets*;
- O trabalho deve utilizar *sockets* para o envio de mensagens entre clientes e servidor;

- O usuário deve informar o IP e a porta para conexão, esses campos não devem ser estáticos;
- Definir as regras do jogo;
- Descrever como será feita a troca de mensagens entre os clientes e o servidor;
- Explicar como o jogo foi implementado.

Características

- ✓ Multijogador
- ✓ O jogo deve suportar no mínimo 2 jogadores em rede para cada partida;
- ✓ O jogador deverá possuir um nickname (ou representar um personagem) no jogo;
- ✓ Não deve ser permitido partidas de apenas 1 jogador;
- ✓ O jogo deve ter um caminho a seguir;
- ✓ Deve haver casas com surpresas (bônus e ônus);
- ✓ Deve haver um dado (virtual ou não) para que o jogador possa pular de casa;
- X Ao fim do jogo, a conexão de rede deve ser desfeita, de forma que seja possível voltar para o menu do jogo e jogar novamente.

Tabuleiro

O tabuleiro virtual consiste de um caminho linear composto por 26 casas (ou posições), dois círculos representando os jogadores, um indicador de turnos no topo da tela e um botão “Jogar” no canto superior direito.

T1 - Marco Goedert



Turno: Adversario :(



Jogar



Jogadores

Por definição, o jogador preto sempre será o usuário adversário e o jogador amarelo sempre será o usuário local. O primeiro jogador (player 1) e o segundo jogador (player 2) são definidos por ordem de chegada pelo servidor e compartilhados por ambos os jogadores. Entretanto, a identificação visual dos players é feita no lado do cliente.

Casas

As casas do tabuleiro são divididas em duas categorias: casas neutras e casas especiais. As casas neutras são identificadas visualmente pela cor cinza e não possuem nenhum efeito diferencial. As casas especiais possuem duas subcategorias: as casas positivas (na cor verde) faz o jogador avançar 1 casa; as casas negativas (na cor vermelha) faz o jogador retroceder 1 casa.

Indicador de turnos

No topo do tabuleiro, um indicador de turno indica para o usuário se é a vez dele jogar ou se deve esperar a jogada do usuário adversário. Caso seja a vez do adversário, uma mensagem “Turno: Adversário :(“ é exibida. Caso contrário, a mensagem “Turno: Você :)” é exibida.

Controles

No canto superior direito da tela, um botão “Jogar” é exibido para que o jogador que está no seu turno de jogada possa arremessar o dado e avançar as n casas resultantes. Se não for a vez do usuário e este clicar no botão, é exibido a mensagem “Aguarde o seu turno para jogar!”. Também é possível se desconectar do jogo através da tecla *Escape* (ESC).

Pendências

Para um melhor entendimento do usuário sobre o status do jogo, abaixo do indicador de turnos também há um indicador geral de status do game. Originalmente o propósito do indicador era indicar qual valor cada jogador rolou no dado ou se o jogador venceu/perdeu a partida. O indicador está em grande parte já implementado, porém devido algum bug ainda desconhecido, não foi possível exibir esta parte neste documento.

Regras do jogo

O jogo de tabuleiro foi desenvolvido pensando em regras simples e intuitivas, assemelhando-se a um jogo de tabuleiro tradicional.

Pré-requisito

Para que o jogo possa começar, devem estar conectados ao servidor exatamente dois jogadores de forma simultânea. Quando ambos usuários estiverem conectados, o jogo é desbloqueado.

Durante o jogo

A ordem das jogadas é definida por ordem de chegada, ou seja, o primeiro jogador a se conectar é o primeiro a jogar. Uma jogada é feita quando o usuário que está em seu turno de jogada clicar no botão “Jogar”, que se encontra no canto superior direito da tela. Em seguida, o botão aciona o dado virtual do jogo, que seleciona de forma aleatória um número entre 1 e 6 e move o personagem pelo tabuleiro. Se o personagem cair sobre uma casa especial, o movimento de avançar ou retroceder é feito automaticamente. Cada jogador pode realizar apenas uma jogada por vez.

Critério de vitória

O jogador que chegar ou ultrapassar a última casa do tabuleiro (≥ 26) primeiro vence a partida.

Implementação

O projeto foi implementado com a linguagem de programação Python e a clássica biblioteca de jogos PyGame. Ao total, três arquivos constituem a lógica do jogo: *run.py*, *game.py* e *server.py*.

run.py

Responsável pela conexão do cliente com o servidor e pela inicialização da interface gráfica do jogo chamando *game.py*.

game.py

Responsável por toda interface gráfica visualizada pelos jogadores. Possui as classes:

- *Player()*: responsável por manipular dados relacionados aos jogadores no lado do cliente
- *Information()*: responsável por manipular e exibir o status do jogo para os jogadores (indicador de turnos)
- *Canvas()* e *Board()*: responsáveis pela criação e gestão do tabuleiro
- *Position()*: responsável pela criação das casas/posições do tabuleiro

server.py

Responsável por estabelecer a conexão com os jogadores e a sincronização das jogadas. Possui a classe *Player()* para manipulação dos dados dos jogadores no lado do servidor e um método `__main__` que gerencia os turnos do jogo e a posição dos jogadores.

Conclusão

A abordagem escolhida no desenvolvimento do jogo acabou resultando em problemas na sincronização dos turnos e, por consequência, na experiência final do jogador. Porém, o resultado obtido foi próximo do desejado e o estudo aplicado foi além da área de redes, como também a área de interface gráfica com a biblioteca PyGame, trazendo assim uma visão muito mais tácita do que é lidar com redes em aplicações com mais de um usuário, simultaneamente.

Portanto, acredito que nos próximos trabalhos soluções mais robustas e interessantes irão surgir, já lidar com redes pode parecer algo complexo à primeira vista, porém se mostra mais acessível com o tempo.