

Trabalho 2 de Redes

Victor Ribeiro Garcia - GRR20203954

Luiz Fernando Giongo dos Santos - GRR20203965

- Como executar:
 - Abra 4 terminais diferentes e execute um comando por terminal nessa ordem:
 - `python3 game.py 3`
 - `python3 game.py 2`
 - `python3 game.py 1`
 - `python3 game.py 0`
- Para testar:
 - Se quiser testar a eliminação do jogador mais facilmente pode alterar a variável global `player_score`, que seria inicializada normalmente com 12 de score para cada jogador, com um valor menor.
 - Se quiser testar a passagem de dealer para outro jogador mais facilmente pode alterar a variável global `current_round`, que é inicializada normalmente com 1 de round inicial, com um valor maior (jogadores receberão menos cartas para jogar).

1) Rede em Anel e Passagem de Bastão:

Os endereços IP e portas de cada nó (jogador) da rede são configurados em uma lista. O índice do nó atual é passado como argumento na linha de comando, permitindo a execução do mesmo código em diferentes máquinas.

O socket é configurado para usar o protocolo UDP (DGRAM). Cada nó cria um socket e o associa ao seu endereço IP e porta correspondente.

O envio de mensagens é feito na função `send_message` (utiliza `sock.sendto`), essa função codifica a mensagem em formato de string e a envia para o nó de destino especificado.

A recepção de mensagens é gerenciada pela função `receive_message`, essa função escuta continuamente por mensagens e, ao receber uma, a decodifica e a encaminha para o tratamento adequado utilizando a função `handle_message` que interpreta a mensagem recebida e toma as ações apropriadas.

O controle do bastão é feito pela função `pass_token`, que verifica se quem chamou a função tem o bastão no momento, retira o bastão de quem chamou a função e envia o bastão para o próximo jogador. Esse mecanismo de passagem de bastão é crucial para manter a sincronização e ordem correta das ações no jogo.

2) O jogo Fodinha:

O funcionamento do jogo pode ser dividido em várias etapas principais:

1. Distribuição de Cartas: a distribuição de cartas é feita pelo dealer. As cartas são embaralhadas e distribuídas de acordo com a rodada atual. Cada jogador recebe o número de cartas igual a 14 menos o round atual (`current_round`), por exemplo, na rodada 1, cada participante recebe 13 cartas.
2. Apostas: cada jogador faz a aposta do número de rodadas que acredita que irá ganhar começando pelo dealer. O jogador faz sua aposta quando possui o bastão e envia sua aposta por mensagem para o próximo jogador. Quando todos jogadores fizeram suas apostas, a mensagem contendo todas as apostas é transmitida jogador por jogador até chegar no dealer.
3. Jogadas: os jogadores jogam suas cartas em turnos, o jogador que detém o bastão realiza sua jogada (escolhe uma carta que recebeu por turno) e envia a sua jogada por mensagem para o próximo jogador. Quando todos jogadores fizeram suas jogadas naquele turno, a mensagem contendo todas as jogadas é transmitida jogador por jogador até chegar no dealer.
4. Cálculo de Resultados Parciais: Após todas as cartas serem jogadas no turno, o resultado é calculado por cada jogador (possuem o valor de todas as cartas jogadas no turno). O jogador com a carta mais alta vence a rodada (a ordem das cartas vai de 3, 2, Ás, K, J, Q, 7, 6, 5 até 4 e os naipes vão de Paus, Copas, Espadas até Ouros na ordem de carta mais forte), e os resultados são contabilizados na função `calculate_results`.
5. Cálculo de Resultados: Ao final da rodada (quando todos jogadores jogaram todas cartas que tinham na mão), os jogadores que não cumprirem exatamente suas apostas (ganharam menos ou menos rodadas do que apostaram) são penalizados (aposta menos a quantidade de turnos ganhos). O cálculo da penalidade de cada jogador é feito na função `accounting_results`.
6. Atualização de Vidas: O jogador que não cumpriu com sua aposta perde vidas (`score`). O cálculo na função `give_score` é feito de acordo com seu `score_atual` menos a penalidade que recebeu e, então, o `score` dos jogadores é atualizado.
7. Verificação de Eliminação: Os jogadores são eliminados se seus `scores` (vidas) caírem a zero, quem faz a verificação é a função `check_for_elimination`. Se algum jogador for eliminado, o jogo termina.
8. Atualização de Rodadas: Ao final de cada rodada, o dealer é rotacionado (passa o bastão para o próximo jogador no anel) e uma nova rodada é iniciada.