



Blackjack

Documentação Técnica

Visão Geral

Este código implementa um jogo simples de Blackjack (21) em Assembly RISC-V, onde o jogador compete contra o dealer (computador). O programa controla o sorteio das cartas, calcula os valores da mão do jogador e do dealer, e decide o vencedor de cada rodada. O jogo mantém o placar de vitórias de ambos.

Registradores e Cartas

No código, os registradores são usados principalmente para controlar o fluxo do jogo e armazenar temporariamente os valores das cartas e pontuações: `s0` e `s1` guardam as somas das mãos do jogador e do dealer, `s2` a `s5` armazenam as duas primeiras cartas de cada um, e registradores temporários (`t0`–`t6`) são usados para cálculos e manipulação de variáveis na memória. Apesar de existirem áreas reservadas com `.space` para `player_hand` e `dealer_hand`, que poderiam armazenar todas as cartas sorteadas durante o jogo, o código não utiliza essas regiões efetivamente; em vez disso, as cartas são mantidas apenas nos registradores, e as somas das mãos são atualizadas diretamente, sem salvar os valores individuais das cartas em memória.

Funções

main

Exibe a mensagem de boas-vindas e inicia o fluxo principal do jogo.

play_game

Mostra o total de cartas e o placar atual, depois executa uma rodada de Blackjack.

show_cards_and_score

Exibe o número de cartas restantes no baralho e o placar de vitórias do jogador e do dealer.

play_round

Distribui as cartas iniciais para jogador e dealer, exibe as mãos e inicia o turno do jogador.

player_turn

Permite que o jogador escolha comprar cartas ("Hit") ou parar ("Stand") enquanto não ultrapassar 21 pontos.

ask_input

Solicita e lê a escolha do jogador entre "Hit" e "Stand", repetindo em caso de entrada inválida.

player_hit_loop

Executa o sorteio de uma nova carta para o jogador, atualiza sua soma e retorna para o turno do jogador.

check_result

Inicia o turno do dealer, que compra cartas até atingir pelo menos 17 pontos.

dealer_loop

Realiza o processo de compra automática de cartas pelo dealer enquanto sua soma for menor que 17.

analyze_result

Compara as pontuações do jogador e do dealer, determina o vencedor, atualiza o placar e exibe o resultado.

is_draw

Exibe mensagem de empate e pergunta se o jogador quer jogar novamente.

dealer_wins

Atualiza o placar com vitória do dealer, exibe mensagem de derrota do jogador e pergunta se quer jogar novamente.

player_wins

Atualiza o placar com vitória do jogador, exibe mensagem de vitória e pergunta se quer jogar novamente.

ask_play_again

Pergunta ao jogador se deseja iniciar uma nova partida ou encerrar o jogo.

draw_card

Sorteia uma carta aleatória entre 1 e 12, garantindo que não ultrapasse o limite de 4 unidades por carta, e atualiza o total de cartas restantes.

Fluxograma

Início do programa



Exibe mensagem de boas-vindas



→ Chama função `play_game`



→ Mostra total de cartas restantes e placar de vitórias



→ Chama `play_round`



→ Sorteia 2 cartas para o jogador (soma em s0)

→ Sorteia 2 cartas para o dealer (soma em s1)



→ Exibe cartas do jogador e uma do dealer



→ Entra no turno do jogador



[Loop] Jogador escolhe:

→ 1 (Hit): compra carta e soma em s0 → repete

→ 2 (Stand) ou estoura (s0 > 21): sai do loop



→ Dealer joga:

[Loop] Compra carta até atingir pelo menos 17 (s1 < 17)



→ Exibe mão final do dealer



→ Verifica resultado:

→ Vitória do jogador? Incrementa `player_score`

→ Vitória do dealer? Incrementa `dealer_score`

→ Empate? Nada muda



→ Pergunta se deseja jogar novamente

→ Sim: reinicia em `play_game`

→ Não: chama `exit` e encerra o programa