Relatório do Trabalho 1 da Disciplina de Organização de Computadores

Nomes: Rodrigo Appelt, Rodrigo Schmidt Becker

1. Introdução

Neste trabalho, foi desenvolvido, em linguagem de programação Assembly, um programa que simula uma partida de jogo da velha entre o usuário e o computador, contando com uma inteligência artificial para ditar as escolhas do último.

1. Documentação

A função *main*, a parte mais importante do jogo, se localiza nas linhas 535-608, nas últimas linhas do arquivo. Ela começa com preparações, como zerar variáveis de contador, o campo, e o vencedor. Depois vem um loop *while*, que coleta inputs da máquina e do usuário em turnos alternados, após fazendo verificações de validade, empate e vitória.

As funções relacionadas com a inteligência artificial do jogo estão localizadas nas linhas 399-531. A IA é feita de 3 regras, a primeira sendo sempre ganhar quando tiver 2 casas seguidas, a segunda sendo impedir a vitória do oponente caso ele tenha 2 casas seguidas. A terceira e última regra é jogar aleatoriamente caso nenhuma das regras anteriores tenham sido detectadas.

A função que detecta vitória de qualquer um dos jogadores está localizada nas linhas 346-396, sendo composta de um simples loop que verifica todas linhas, colunas e as diagonais, procurando por 3 casas iguais.

A simples função que coloca uma marcação no tabuleiro de acordo com o jogador atual está localizada nas linhas 320-342. Ela é formada basicamente por um *if* que coloca o caractere ‘X’ ou o caractere ‘O’ de acordo com o contador de jogo atual.

A função “valido”, localizada nas linhas 259-280, verifica se um par de coordenadas (linha e coluna) estão dentro do campo e já não foram escolhidas anteriormente.

A função designada para imprimir o campo e suas marcações, está escrita nas linhas 198-256. Ela é responsável por imprimir guias das numerações de linhas e colunas, as divisões do campo e as marcações dos jogadores que já foram postas no tabuleiro.

Nas linhas 162 até 193, estão localizados defines(*.eqv)*, *strings* para exibição e as variáveis de controle.

Finalmente, nas linhas 4 até 148 temos inúmeras definições de macros, que simplificaram extremamente o desenvolvimento do jogo.

1. Conclusão

Concluímos que, ao desenvolver um projeto em Assembly, obtivemos conhecimentos mais assertivos acerca do funcionamento de computadores devido à natureza de baixo nível da linguagem, assim aprimorando nossas habilidades como programadores.

1. Bibliografia

* PATTERSON, David A.; HENNESSY, John L. **Organização e projeto de computadores: a interface hardware/software**. 5. ed. [S.l.]: Elsevier, 2017.
* Página de ajuda do software Mars, versão 4.5.
* MCQUAIN, W. D. (2014). *MIPS Assembly Overview* [PowerPoint slides]. <https://courses.cs.vt.edu/cs2506/Fall2014/Notes/L04.MIPSAssemblyOverview.pdf>