

# O Jogo da velha e da moça

Trabalho 1 de CI067 - Oficina de Computação

2º Semestre - 2018

Data de Entrega: 23/09/2018

O jogo da velha e da moça é um jogo de estratégia que deverá acontecer entre dois jogadores. Um dos jogadores aquele que visualiza a velha na Figura 1, terá 32 peças com a letra *v* e o outro jogador aquele que visualiza a moça terá também 32 peças com a letra *m*.



Figura 1: Velha ou moça?

Os jogadores se alternam colocando suas peças em um tabuleiro 8x8. O jogo inicia com 4 peças já colocadas no centro do tabuleiro, sendo duas da moça e duas da velha em posições opostas diagonalmente, conforme ilustrado na Figura 2. Para uma jogada ser considerada válida é preciso que haja pelo menos uma linha reta (horizontal, vertical ou diagonal) sem espaços vazios entre a posição onde será colocada a peça e outra peça com a mesma letra e que haja pelo menos uma peça do adversário entre elas. Essas peças do adversário se transformaram em moça ou velha, ou seja, passarão a conter a mesma letra de quem estiver jogando. A Figura 3 mostra com pontinhos as possibilidades para a próxima jogada para o jogador que tem a letra *m*. O objetivo do jogo é ter o maior número de peças com a sua letra quando a última jogada possível for efetuada. O jogo termina quando nenhum jogador puder fazer movimentos.

## Instruções para implementação

Desenvolver um programa estruturado e modularizado em linguagem C que permita a dois usuários jogarem o jogo da Velha e da Moça em uma janela do terminal. O jogo poderá ser salvo em um arquivo para continuar posteriormente. O programa inicia recebendo os nomes dos jogadores (estes nomes não devem conter espaços em branco) e também a escolha de cada um. Após estas escolhas, o programa apresentará o quadro da Figura 2 na tela do monitor.

	A	B	C	D	E	F	G	H
1								
2								
3								
4				v	m			
5				m	v			
6								
7								
8								

< Nome1- Velha > 30 peças

< [Nome2] - Moça > 30 peças

Figura 2: Posição inicial do jogo

Utilize caracteres básicos para mostrar o tabuleiro. Embaixo do tabuleiro devem aparecer os nomes dos jogadores e quantas peças ele ainda possui. Veja que inicialmente cada um tem 30 peças. O nome do jogador da vez deverá ser apresentado entre [ ]. O jogador que fará o primeiro movimento deverá ser a moça. Você pode colocar outros elementos para auxiliar a visualização, ou facilitar a implementação mas é necessário apresentar pelo menos todos os elementos na Figura 2.

	A	B	C	D	E	F	G	H
1								
2								
3					.			
4			.	m	v	.		
5			.	v	m			
6		.	v	m	m			
7		v	.	v	m			
8			.	.				

< Nome1- Velha > 22 peças

< [Nome2] - Moça > 22 peças

Figura 3: Possibilidades de jogada para a moça.

A cada nova jogada, o programa solicitará ao jogador que indique a posição no tabuleiro onde deseja colocar a nova peça, por exemplo C5. O programa deverá validar a jogada conforme a regra de colocação das peças e se a jogada for válida, trocar as letras das peças do adversário para correspondentes letras do jogador (por exemplo D5 passa a conter *m*) e decrementar o número de peças disponíveis para a próxima jogada. O programa deverá rejeitar qualquer jogada inválida (por exemplo, colocar em uma posição isolada, ou ocupada por outra peça, etc.). Por exemplo na Figura 3 não é possível para a moça colocar uma peça na posição E8, porque não tem nenhuma peça “v” ao lado acima ou na diagonal entre a posição E8 e outra peça “m”. No entanto é possível colocar a peça “m” nos pontos marcados com “.”.

O programa deve ainda verificar se não é mais possível algum movimento válido que indica final de jogo. Neste caso deve parabenizar o jogador vencedor (com o maior número de peças) e fornecer a opção de reiniciar (R) o jogo com os mesmos jogadores, começar novo jogo (N) ou finalizar (X), ou ainda ler (L) de um arquivo um jogo que foi interrompido anteriormente. Neste caso o nome do arquivo a ser lido terá o mesmo nome do jogador1 concatenado com o nome do jogador2.

## Critérios de Correção

O trabalho vale 100 pontos e seguirá os seguintes critérios e valores.

1) funcionalidade, (70) ou seja: a inicialização, a jogabilidade, a validação correta de uma jogada, o efeito produzido corretamente para a jogada, e a finalização correta do jogo. O armazenamento e recuperação do jogo.

2) a estruturação e organização do código (10) e a legibilidade (5) conforme material passado em aula. Utilize funções, declare suas funções, comentários adequados, arquivos .h e .c

3) defesa do trabalho (15): respostas corretas a perguntas feitas no dia da apresentação.

Caso o programa enviado não compile, a nota será 0. Teste muito bem o seu programa e certifique-se que ele está funcionando corretamente. Preferivelmente guarde versões dele, pois se acontecer algum imprevisto, uma versão anterior poderá ser utilizada.

Caso o programa compile, mas não execute, o código será avaliado pela sua legibilidade e organização. Retire todos os *warnings*.

Caso o programa execute, porém falhe durante a execução por um erro de programação, então o programa será avaliado individualmente e será dada uma nota relativa aos requisitos atendidos.

Durante a apresentação do trabalho serão feitas perguntas sobre como o aluno implementou o programa. Caso o aluno não saiba responder, serão descontados pontos por pergunta errada.

Quaisquer funcionalidades extras que não tenham sido pedidas no enunciado são opcionais e não contribuem para a nota.

## Forma de Entrega

O trabalho consiste em arquivos com código-fonte .h e .c na linguagem C. Não serão aceitos arquivos .doc ou .pdf. Você deve criar um arquivo com extensão .tar e utilizar o programa entrega, assim como os exercícios entregues em sala, com o comando abaixo.

```
> /home/html/inf/silvia/entrega/bin/entrega_oficinac 20 arquivo.tar
```

Os arquivos devem ser entregues até às 23:59 de domingo, dia 23/09. Trabalhos entregues depois do dia 23 terão 10 pontos descontados por dia de atraso.

O programa entregue será apresentado pelo próprio aluno em sala de aula no dia 27/09. Durante a apresentação serão feitas perguntas sobre a implementação.