

UNIVERSIDADE FEDERAL DO AMAZONAS

CONCEIÇÃO NEVES SILVA - 21952453

FELIPE DE VASCONCELOS ASSUNÇÃO - 21955313

**INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA II – ENSINO REMOTO
EMERGENCIAL**

ITACOATIARA

2020

CONCEIÇÃO NEVES SILVA

FELIPE DE VASCONCELOS ASSUNÇÃO

**INTRODUÇÃO A PROGRAMAÇÃO ESTRUTURADA II – ENSINO REMOTO
EMERGENCIAL**

Relatório técnico para obtenção de nota parcial em busca de aprovação na disciplina de Introdução à Programação Estruturada II, em bacharelado em Engenharia de Software, do Instituto de Ciências Instituto de Ciências Exatas e Tecnologia, da Universidade Federal do Amazonas.

PROF. ALTERNEI BRITO

ITACOATIARA

2020

1. Introdução

O jogo da velha é um dos passatempos mais conhecidos e divertidos que se tem lembrança. Além de ser simples e rápido, este jogo ajuda a melhorar muito a sua capacidade lógica. Nesse sentido, o “Jogo da Velha” foi aplicado como trabalho prático da disciplina de Introdução à Programação II.

Portanto, o presente documento tem como finalidade apresentar um algoritmo que represente o jogo da velha, o algoritmo foi desenvolvido em linguagem C utilizando vetores e funções e estruturas de repetição.

2. Objetivo

O objetivo do trabalho é apresentar um jogo da velha que ofereça ao usuário jogar inúmeras vezes até o mesmo desejar encerrar o jogo. O algoritmo deve apresentar a quantidade de partidas jogadas, a quantidade de vitórias do jogador 1 e jogador 2. O algoritmo oferece a quantidade de vitórias em porcentagem de ambos os jogadores.

3. Detalhes da implementação

a. Estruturas de dados utilizadas

- Função;
- Vetores;
- Operador ternário ou condição;
- Laços de repetição.

b. Bibliotecas:

#include <stdio.h>

É a biblioteca padrão de entrada e saída utilizada no cabeçalho de um algoritmo. O objetivo dessa biblioteca é realizar leitura de dados digitados no teclado e exibição de informações na tela do programa de computador. Também possui numerosas definições de constantes, variáveis e tipos.

#include <stdlib.h>

É um arquivo cabeçalho da biblioteca de propósito geral padrão da linguagem de programação C. Ela possui funções envolvendo alocação de memória, controle de processos, conversões e outras. Ela foi usada porque foi declarada variáveis constantes no algoritmo, assim como também o uso de vetores.

#include <locale.h>

A linguagem C utiliza o arquivo de cabeçalho `locale.h` para implementar a chamada “localização”. O uso desta biblioteca garante a utilização de caracteres como o “ç” e acentuação em palavras que não pertencem à língua inglesa, ela foi usada no algoritmo para exibir de forma correta as frases corretamente acentuadas da língua portuguesa, assim como também o uso de letras padrão do idioma português.

c. Funções:

void preencher();

Visto que o tabuleiro já se inicia com o vetor preenchido, essa função tem como finalidade preencher novamente o vetor, substituindo todos os caracteres que nele se encontram após a primeira partida. Ou seja, o jogo se inicia e é dado o fim da primeira partida, logo a função “**preencher**” irá alocar novos valores no vetor com a finalidade de limpar o tabuleiro.

void tabuleiro();

Efetivamente é uma função do tipo void, sendo a responsável por imprimir para o usuário o tabuleiro do jogo, sempre que for realizada uma “chamada” da mesma. É uma função sem parâmetros.

int vencedor();

A função **int vencedor();** tem como propósito verificar o vencedor da partida, se houve empate durante a partida ou se a mesma ainda não chegou ao fim. Essa função também tem como finalidade verificar todas as ocasiões de elementos repetidos nas linhas e colunas (uma de cada vez) e nas diagonais da tabela.

Por ser uma função do tipo inteira ela gera um valor de retorno. Para isso, a função verifica se há alguma linha, coluna ou diagonal com valores iguais, caso isso ocorra, será gerado um *return* de valor 1. Caso a função constate que todos os valores encontram-se preenchidos de forma diferente do tabuleiro inicial, a mesma irá gerar um *return* do valor 0, ou seja, retorna 0 ao constatar empate. Todavia, visto que nenhum dos casos anteriores ocorra, a função gera um *return* do valor -1 para continuação da partida até que a mesma finalize com empate ou vitória de um dos dois jogadores. Vale ressaltar que essa função também não possui parâmetros.

char jogarNovamente();

O objetivo dessa função é perguntar ao usuário e armazenar o desejo do mesmo de continuar com a partida ou não.

void estatisticas();

Essa função é responsável por exibir o histórico de partidas, incluindo os vencedores, a quantidade de vitórias de ambos os jogadores 1 e 2. A porcentagem de vitórias, o número de empates e quantidade de partidas jogadas pelos jogadores. Essa função não retorna nenhum valor.

int main();

A função principal é a main(), nela estão reunidos as operações lógicas de jogo, tais como: verificar se as posições do vetor estão preenchidas. Visto que o usuário escolhe no vetor uma posição para marcar sua jogada, nesse sentido a condição **if, else if, else** inicial da função main, verifica a posição marcada pelo usuário e substitui na casa do vetor para X ou O. Todavia, o código desenvolvido está com comentários para melhor entendimento.

4. Recursos Principais

Para o desenvolvimento deste algoritmo, é de suma importância o conhecimento de vetores ou matrizes, pois é possível desenvolver o jogo-da-velha tanto com matrizes como também com vetores que por sinal foi utilizado para a elaboração deste trabalho. É necessário o conhecimento em funções e estruturas de dados, além da lógica de programação.

5. Ideia Geral

A ideia geral deste algoritmo é utilizar todo o conhecimento ministrado durante a presente disciplina de IPE II e aplicar em prática para esse trabalho final. Entretanto, a ideia de utilizar vetor é devido a vontade de utilizar o mínimo de recursos disponível para provar que é possível desenvolver um jogo da velha sem o uso de struct, matrizes, e ponteiros, usando apenas vetores, função, laços de repetição, condicionais ou função ternária (o que tem a mesma finalidade das condicionais).