

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO AMAZONAS – UFAM
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA – ICET
CAMPUS UNIVERSITÁRIO MOYSÉS BENARRÓS ISRAEL
BACHARELADO EM ENGENHARIA DE SOFTWARE

TRABALHO PRÁTICO

ITACOATIARA-AM
2020

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO AMAZONAS – UFAM
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA – ICET
CAMPUS UNIVERSITÁRIO MOYSÉS BENARRÓS ISRAEL
BACHARELADO EM ENGENHARIA DE SOFTWARE

GABRIEL FONSECA FERREIRA
MATRÍCULA: 21955310

TRABALHO PRÁTICO

Trabalho referente ao assunto de Introdução à Programação II, apresentado, como requisito para obtenção de nota parcial, do segundo período do segundo semestre de 2020, solicitado pelo professor Alternei Brito, ministrante da disciplina de Introdução à Programação II, no Curso de Bacharelado em Engenharia de Software, do Instituto de Ciências Exatas e Tecnologia – ICET.

ITACOATIARA-AM
2020

1. INTRODUÇÃO

Jogo da velha é muito conhecido mundialmente, mas se engana quem acha que ele surgiu recentemente. Há registros dele em escavações feitas no templo de Kurna, no Egito datadas no século 14. Não apenas foi encontrado registros do jogo da velha nessa região como também na China antiga, na América pré-colombiana e no Império Romano. No entanto, foi na Inglaterra do século 19 que esse jogo se popularizou e ganhou esse nome. Quando as mulheres inglesas se reuniam na hora do chá para bordar tinha aquelas mais velhas que não conseguiam mais fazer este ofício. Muito dessas senhoras já apresentavam problemas de vista e não enxergavam o suficiente para conseguir bordar. Portanto, a solução para conseguirem um novo passatempo foi jogar o jogo de velha. E é por isso que ele recebe este nome: porque era jogado por velhas.

Existem muitas formas de jogar e várias técnicas abordadas para conseguir ganhar, o objetivo do mesmo é preencher ou as linhas diagonais ou as horizontais ou as verticais com um mesmo símbolo (X ou O) e impedir que seu adversário faça isso primeiro que você. Com a finalidade de exercitar o pensamento lógico esse passatempo tem alguns truques que ajudam na hora da partida.

Com a tecnologia, o jogo de tabuleiro físico foi perdendo espaço dando abertura para novas formas de se jogar, seja de forma digital, on-line, jogando com um computador, entre outras. Assim este presente trabalho vem para mostrar uma das formas empregadas para se implementar este jogo.

2. OBJETIVOS

O objetivo deste trabalho é criar um programa em C para simular o jogo da velha, na qual utilizando das regras originais, implementá-las no espaço virtual com as seguintes propostas:

Utilize o conceito de modularização para as funções do jogo e do programa, além das estruturas de dados, *arrays* ou matrizes para representar o tabuleiro. Para a simulação, considere que haverá dois jogadores. Caso deseje, um deles pode ser o computador. Especifique a vez de cada jogador. Crie um menu para ser exibido ao final de cada partida, pedindo para o usuário encerrar ou continuar no jogo. Utilize as funções para verificar se um jogador venceu o jogo, ou seja, se foi preenchida com um símbolo, caractere ou número uma das linhas, colunas ou diagonais. Os jogadores podem jogar quantas vezes quiserem, desde que informem a opção de prosseguir no menu. Ao escolherem finalizar o jogo, um relatório deve ser exibido, informando os seguintes dados:

- Quantidade de partidas jogadas
- Quantas partidas o jogador 1 venceu
- Quantas partidas o jogador 2 venceu
- A porcentagem de vitórias cada jogador

3. IMPLEMENTAÇÃO

Inicialmente foram declaradas as variáveis, como mostra a Figura 1:

```
81 int main(){  
82     setlocale(LC_ALL, "Portuguese");  
83     int op, i=0, j=0, lin=0, col=0, vezJogador=0, jogadas=0, j1=0, j2=0, empate=0;  
84     char jogo[3][3] = {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '};
```

Figura 1. Inicial

Em torno da matriz “jogo” que se rodeia todo o processo de simulação do jogo da velha, na qual foi inicializada com espaços vazios.

Possui também uma espécie de reinício, que ocorre a cada partida nova, onde as variáveis voltam ao estado inicial, como na Figura 2:

```
jogadas = 0;  
vezJogador = 0;  
for(i=0; i<3; i++){  
    for(j=0; j<3; j++){  
        jogo[i][j] = ' ';  
    }  
}
```

Figura 2. Reinício

Após isso se desenrola todo o processo de alojamento dos sinais do jogo (X ou O), fazendo a verificação das posições na matriz, que para facilitar os leigos, foram utilizadas a partir do 1 ao 3, não do 0 ao 2 como é de costume da programação, ou seja, ele escolhe a partir da primeira linha (1) e vai até a terceira linha (3) e com as colunas se representam da mesma forma, as Figuras 3 e 4 representam esta explicação:

```
C:\Users\Sandra e Gabriel\Docum

|  | 
-----
|  | 
-----
|  | 

Vez do jogador 1 (X)

Digite a linha: 1

Digite a coluna: 1
```

Figura 3. Escolhida linha 1 e coluna 1 para se colocar o X

```
C:\Users\Sandra e Gabriel\Do

X |  | 
-----
|  | 
-----
|  | 

Vez do jogador 2 (O)

Digite a linha: _
```

Figura 4. X já inserido na primeira posição

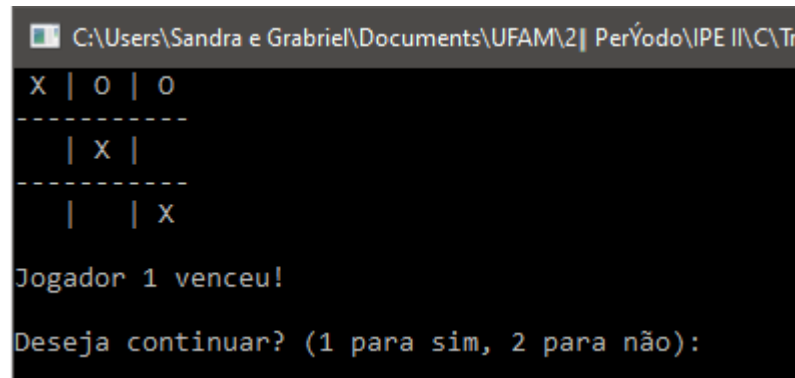
Caso um dos jogadores tente usar um local já preenchido, ele voltará a tentar novamente sua jogada.

Depois das verificações das variáveis, são verificadas com uma função, se um dos jogadores já ganhou a partida, como mostra a Figura 5:

```
44 int verifica(char jogo[3][3],int jogadas){
45     if(jogo[0][0] == 'X' && jogo[0][1] == 'X' && jogo[0][2] == 'X'){
46         jogadas = -1;
47     }else if(jogo[1][0] == 'X' && jogo[1][1] == 'X' && jogo[1][2] == 'X'){
48         jogadas = -1;
49     }else if(jogo[2][0] == 'X' && jogo[2][1] == 'X' && jogo[2][2] == 'X'){
50         jogadas = -1;
51     }else if(jogo[0][0] == 'X' && jogo[1][0] == 'X' && jogo[2][0] == 'X'){
52         jogadas = -1;
53     }else if(jogo[0][1] == 'X' && jogo[1][1] == 'X' && jogo[2][1] == 'X'){
54         jogadas = -1;
```

Figura 5. Verificação de ganhador

Assim, ele retorna um valor para indicar o ganhador correspondente ou indica se é empate e após isso é mostrada uma mensagem se desejamos continuar o jogo ou finalizar as partidas, como mostra a Figura 6:



```
C:\Users\Sandra e Gabriel\Documents\UFAM\2º Período\IPE II\C\Tr
X | O | O
-----
  | X |
-----
  |  | X

Jogador 1 venceu!

Deseja continuar? (1 para sim, 2 para não):
```

Figura 6. Mostra quem ganhou ou se houve empate e a mensagem de continuação

Caso digite 1, o jogo reinicia, se não, o jogo finaliza e assim os relatórios são feitos, em uma função especial, como mostra a Figura 7:

```
16 void relatorio(int j1, int j2, int empate, FILE *arq){
17     float jog1=0.0, jog2=0.0, emp=0.0;
18     jog1 = (j1/partidas)*100;
19     jog2 = (j2/partidas)*100;
20     emp = (empate/partidas)*100;
21     printf("-----");
22     printf("\n\t\tRELATÓRIO DAS PARTIDAS");
23     printf("\n\nQuantidade de partidas jogadas: %.0f partidas",partidas);
24     printf("\n\nQuantidade de partidas vencidas pelo jogador 1: %d partida(s)",j1);
25     printf("\n\nQuantidade de partidas vencidas pelo jogador 2: %d partida(s)",j2);
26     printf("\n\nQuantidade de empates: %d empate(s)",empate);
27     printf("\n\nPorcentagem de ganhos do jogador 1: %.0f por cento",jog1);
28     printf("\n\nPorcentagem de ganhos do jogador 2: %.0f por cento",jog2);
29     printf("\n\nPorcentagem de empates: %.0f por cento",emp);
30     printf("\n-----");
```

Figura 7. Relatório 1

Neste caso optei para além do que é pedido, mostrar também a quantidade e porcentagem dos empates, bem como gravar em um arquivo externo o relatório ao final. O exemplo de relatório é mostrado na Figura 8:

```
C:\Users\Sandra e Gabriel\Documents\UFAM\2| PerÝodo\IPE II\C\TrabalhoFinalGabriel\JogoDaVelha.exe

-----
                        RELATÓRIO DAS PARTIDAS
-----
Quantidade de partidas jogadas: 1 partidas
Quantidade de partidas vencidas pelo jogador 1: 1 partida(s)
Quantidade de partidas vencidas pelo jogador 2: 0 partida(s)
Quantidade de empates: 0 empate(s)
Porcentagem de ganhos do jogador 1: 100 por cento
Porcentagem de ganhos do jogador 2: 0 por cento
Porcentagem de empates: 0 por cento
-----

Process exited after 15.02 seconds with return value 0
Pressione qualquer tecla para continuar. . . █
```

Figura 8. Relatório 2

A cada partida, é mostrado o tabuleiro atualizado com uma função específica, como mostra a Figura 9:

```
7 void tabuleiro(char jogo[3][3]){
8     system("cls");
9     printf(" %c | %c | %c \n",jogo[0][0],jogo[0][1],jogo[0][2]);
10    printf("-----\n");
11    printf(" %c | %c | %c \n",jogo[1][0],jogo[1][1],jogo[1][2]);
12    printf("-----\n");
13    printf(" %c | %c | %c \n",jogo[2][0],jogo[2][1],jogo[2][2]);
14 }
```

Figura 9. Tabuleiro

Além disso, existe uma função específica que mostra um rápido tutorial do jogo, como mostra a Figura 10:

```
C:\Users\Sandra e Gabriel\Documents\UFAM\2| PerÝodo\IPE II\C\Trabalho
Deseja ver o tutorial? (1 para sim, 2 para não): █
```

Figura 10. Tutorial 1

Caso o usuário digite 1 (um), irá ver o tutorial. Como mostra a Figura 11, caso digite 2 (dois), irá direto ao jogo:


```
C:\Users\Sandra e Gabriel\Documents\UFAM\2 | Período\IPE II\C\TrabalhoFinalGabriel\TrabalhoFinalGabriel\Jog

 1  2  3
1  |  |
-----
2  |  |
-----
3  |  |

ESTE É O TABULEIRO

Você digita uma coordenada (linha e coluna) e aparece seu respectivo símbolo

Por exemplo vamos adicionar o X na primeira posição

Digite a linha: 1
Digite a coluna: 1
Logo após isso o tabuleiro fica deste jeito:

 1  2  3
1 X |  |
-----
2  |  |
-----
3  |  |

Ganha quem preencher a mesma coluna, linha ou diagonal com o mesmo símbolo!

Digite algo para retornar ao jogo (ou digite 1 para rever o tutorial):
```

Figura 11. Tutorial 2

Após a leitura do tutorial, ele dá a opção de ir ao jogo ou rever o tutorial.

A função do tutorial está representada na Figura 12:

```
8 char tutorial(){
9     char op;
10    system("cls");
11    printf(" 1  2  3 \n");
12    printf("1  |  | \n");
13    printf("-----\n");
14    printf("2  |  | \n");
15    printf("-----\n");
16    printf("3  |  | \n");
17    Sleep(1500);
18    printf("\n\nESTE É O TABULEIRO");
19    Sleep(2000);
20    printf("\n\nVocê digita uma coordenada (linha e coluna) e aparece seu respectivo símbolo");
21    Sleep(2000);
22    printf("\n\nPor exemplo vamos adicionar o X na primeira posição");
23    printf("\n\nDigite a linha: 1");
24    printf("\n\nDigite a coluna: 1");
25    Sleep(1500);
26    printf("\n\nLogo após isso o tabuleiro fica deste jeito:\n\n");
27    printf(" 1  2  3 \n");
28    printf("1 X |  | \n");
29    printf("-----\n");
30    printf("2  |  | \n");
31    printf("-----\n");
32    printf("3  |  | \n");
```

Figura 12. Função tutorial

O jogo em si foi implementado com sucesso em seu jeito mais simples, existem outras formas de se implementar e esta foi a escolhida. Ele respeita as regras originais do jogo e conclui com seu objetivo.