



CENTRO UNIVERSITÁRIO DE JOÃO PESSOA
CENTRO DE TECNOLOGIA
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
TÉCNICAS DE DESENVOLVIMENTO DE ALGORITMOS

LISTA DE INTEGRANTES:

Bruna Siqueira Correia - ***RGM: 31075371***
Leon Denis Silvestre Lucena - ***RGM: 31292704***
Matheus de Moura Aguiar Quintanilha - ***RGM: 31802508***
Murilo Vicente Araujo Batista Ribeiro - ***RGM: 31088783***

INTRODUÇÃO

O jogo escolhido foi o Jogo da Velha, que consiste num desenho composto por três linhas e três colunas, sendo baseado por turnos, onde cada jogador coloca sua marca no local que deseja jogar, marcando com um “X” ou um “O”. O objetivo do jogo é obter uma sequência de três símbolos iguais, seja em linha horizontal, vertical, diagonal principal ou diagonal secundária, enquanto o adversário faz o mesmo. Quando um dos jogadores fecha uma linha, coluna ou diagonal, ganha o jogo. Se nenhum dos jogadores fecha nenhuma linha, coluna ou diagonal termina com empate, chamado popularmente de “deu velha”, e pode-se iniciar uma nova partida.

RESULTADO

Para estrutura do jogo tínhamos a possibilidade de escolher uma matriz 3x3 ou 4x4 e optamos por uma estrutura de matriz 3 por 3, gerando assim um tabuleiro clássico de jogo da velha. Para gerar este tabuleiro foi necessário criar uma função para inicializar toda a matriz, outra para preencher a matriz com espaços em branco (**figura 01**) e outra para organizar o visual do jogo para o usuário (**figura 02**).

Figura 01: Função de inicializar e imprimir a matriz.

```
12  /* Preenche a matriz do Tabuleiro */
13  void inicializarMatriz() {
14      for (l = 0; l < 3; l++) {
15          for (c = 0; c < 3; c++)
16              matrizJogo[l][c] = ' ';
17      }
18  }
19
20  /* Imprime a matriz do Tabuleiro no formato do jogo da velha */
21  void imprimir() {
22      printf("\n\n\t 0   1   2\n\n");
23      for(l = 0; l < 3; l++) {
24          for(c = 0; c < 3; c++) {
25              if(c == 0)
26                  printf("\t");
27              printf(" %c ", matrizJogo[l][c]);
28              if(c < 2)
29                  printf("|");
30              if(c == 2)
31                  printf(" %d", l);
32          }
33          printf("\n");
34          if(l < 2)
35              printf("\t-----\n");
36      }
37  }
38
```

Fonte: Dados dos Autores.

Figura 02: Visual do Tabuleiro do Jogo da Velha.

```

      0   1   2
      |   |
-----
      |   |
-----
      |   |
      0
      1
      2
Jogador X digite a linha: _
```

Fonte: Dados dos Autores.

Após a elaboração e estruturação do tabuleiro, realizamos a estruturação das funções que delimitam as condições de vitória, na qual primeiramente pensamos em fazer apenas 4 funções, uma para as vitórias em linha, outra para colunas, diagonal principal e diagonal secundária, no entanto tivemos uma dificuldade para declarar as condições de linha e coluna e chamá-las no início do jogo, assim optamos por criar mais 2 funções uma para marcar a vitória por linha e outra para marcar a vitória por

coluna na chamada dentro da função jogo (explicada posteriormente). Portanto foi necessário formular 6 funções para checar as condições de vitória (ganhouPorLinha, ganhouPorLinhas, ganhouPorColuna, ganhouPorColunas, ganhouPorDiagPrin e ganhouPorDiagSec) e foi criada para checar se existem campos da matriz vazios para marcar a condição de empate (quantVazias).

Para entrada de dados foi criada uma função chamada lerCoordenadas, nesta função solicitamos que o usuário indique uma linha e uma coluna para marcar a posição da jogada no tabuleiro, importante falar que dentro desta função temos a chamada de uma outra que checa se a coordenada digitada é válida (cordenadaValida), nesta função de checagem temos o tratamento de erro que checa se o usuário preencheu alguma coordenada já digitada ou se preencheu uma coordenada fora da matriz 3x3, caso o usuário preencha alguma coordenada invalida ele irá solicitar que preencha novamente outra linha ou coluna.

Por fim temos a função jogar, nela estruturamos a sequência de repetição do jogo dentro de um do-while, nele chamamos a função imprimir e depois rodamos um if e else, para cair na condição do if é necessário que o valor jogador seja 1 (atribuído inicialmente na função o "int jogador = 1"), ou seja, sempre que iniciar ele cairá na condição do if, dentro do if ele chama a função de lerCoordenadas para o jogador1, checa se alguma condição de vitória bateu e soma 1 ao valor de jogador, assim na próxima repetição do while o jogador terá o valor de 2 e cairá no else, dentro do else rodamos as mesmas coisas para o jogador2, sendo que nele atribuímos novamente o valor de jogador como 1 para que no próximo turno ele volte para o if (**figura 03**).

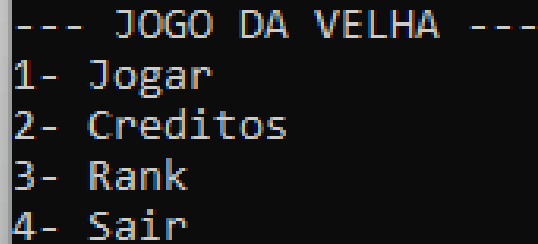
Figura 03: Parte da Função de Jogar que checa condições de vitória ou empate.

```
void jogar() {  
  
    int jogador = 1, vitoriaDeX = 0, vitoriaDe0 = 0;;  
    char jogador1 = 'X', jogador2 = '0';  
  
    do {  
        imprimir();  
        if (jogador == 1) {  
            lerCoordenadas(jogador1);  
            jogador++;  
            vitoriaDeX += ganhouPorLinhas(jogador1);  
            vitoriaDeX += ganhouPorColunas(jogador1);  
            vitoriaDeX += ganhouPorDiagPrin(jogador1);  
            vitoriaDeX += ganhouPorDiagSec(jogador1);  
        } else {  
            lerCoordenadas(jogador2);  
            jogador = 1;  
            vitoriaDe0 += ganhouPorLinhas(jogador2);  
            vitoriaDe0 += ganhouPorColunas(jogador2);  
            vitoriaDe0 += ganhouPorDiagPrin(jogador2);  
            vitoriaDe0 += ganhouPorDiagSec(jogador2);  
        }  
        system("CLS");  
    } while (vitoriaDeX == 0 && vitoriaDe0 == 0 && quantVazias() > 0);  
}
```

Fonte: Códigos dos Autores.

Ainda na função jogar, a condição de parada do while é caso o valor de vitoriaDeX ou de vitoriaDe0 for diferente de 0 ou se quantVazias for maior que 0. Ao temos a vitória de um jogador chamamos novamente a função de imprimir, mostramos quem foi o jogador que ganhou e mostramos como está o rank. Para finalizar o programa, dentro do nosso Main temos um switch que roda um menu onde o usuário poderá escolher entre 1- Jogar, 2- Créditos, 3- Rank e 4- Sair (**figura 04**).

Figura 04: Menu de escolha ao iniciar o programa.

A screenshot of a terminal window with a black background and white text. The text displays a menu for a game titled 'JOGO DA VELHA'. The menu options are numbered 1 through 4: '1- Jogar', '2- Creditos', '3- Rank', and '4- Sair'. The title 'JOGO DA VELHA' is centered and flanked by three dashes on each side.

```
--- JOGO DA VELHA ---  
1- Jogar  
2- Creditos  
3- Rank  
4- Sair
```

Fonte: Códigos dos Autores.

APÊNDICE

APÊNDICE 01: Código fonte do Jogo da Velha

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
/*-----> JOGO DA VELHA <-----*/

/* Variaveis de uso global */
char matrizJogo[3][3]; // Gera a matriz do Jogo sendo 3x3.
int l, c; // l = Linha | c = Coluna.
int quantidade_1 = 0; // variavel para o rank
int quantidade_2 = 0; // variavel para o rank

/* Preenche a matriz do Tabuleiro */
void inicializarMatriz() {
    for (l = 0; l < 3; l++) {
        for (c = 0; c < 3; c++)
            matrizJogo[l][c] = ' ';
    }
}

/* Imprime a matriz do Tabuleiro no formato do jogo da velha */
void imprimir() {
    printf("\n\n\t 0   1   2\n\n");
    for(l = 0; l < 3; l++) {
        for(c = 0; c < 3; c++) {
            if(c == 0)
                printf("\t");
            printf(" %c ", matrizJogo[l][c]);
            if(c < 2)
                printf("|");
            if(c == 2)
                printf(" %d", l);
        }
        printf("\n");
        if(l < 2)
            printf("\t-----\n");
    }
}

/* Funções que Definirem condições de Vitoria */

/* Ganhou por linha - Jogador 'c' na Linha 'l' coluna fixa */
int ganhouPorLinha(int l, char c) {
    if (matrizJogo[l][0] == c && matrizJogo[l][1] == c && matrizJogo[l][2] == c)
        return 1;
    else
        return 0;
}
```

```

}

/* Marca vitoria linhas - Jogador 'c' Nas Linhas */
int ganhouPorLinhas(char c) {
    int ganhou = 0;
    for (l = 0; l < 3; l++)
    {
        ganhou = ganhou + ganhouPorLinha(l, c);
    }
    return ganhou;
}

/* Ganhou por Coluna - Jogador 'j' Na Coluna 'c' Linha fixa */
int ganhouPorColuna(int c, char j) {
    if (matrizJogo[0][c] == j && matrizJogo[1][c] == j && matrizJogo[2][c] == j)
        return 1;
    else
        return 0;
}

/*
/* Marca vitoria coluna - Jogador 'j' Nas Colunas */
int ganhouPorColunas(char j) {
    int ganhou = 0;
    for (c = 0; c < 3; c++)
    {
        ganhou += ganhouPorColuna(c, j);
    }
    return ganhou;
}

/* Ganhou por diagonal - Jogador 'c' Na Diagonal Principal */
int ganhouPorDiagPrin(char c) {
    if (matrizJogo[0][0] == c && matrizJogo[1][1] == c && matrizJogo[2][2] == c)
        return 1;
    else
        return 0;
}

/* Ganhou por diagonal - Jogador 'c' Na Diagonal Secundaria */
int ganhouPorDiagSec(char c) {
    if (matrizJogo[0][2] == c && matrizJogo[1][1] == c && matrizJogo[2][0] == c)
        return 1;
    else
        return 0;
}

/* Fim das Funções de condição de vitoria */

/* Função para validar uma jogada l e c */
int cordenadaValida(int l, int c) {
    if (l >= 0 && l < 3 && c >= 0 && c < 3 && matrizJogo[l][c] == ' ')
        return 1;
    else

```

```

        return 0;
    }

/* Função Para Pedir ao Jogador as Coordenadas para jogar*/
void lerCoordenadas(char j) {
    int linha, coluna;

    printf("\nJogador %c digite a linha: ", j);
    scanf("%d", &linha);
    printf("\nJogador %c digite a coluna: ", j);
    scanf("%d", &coluna);

    while (cordenadaValida(linha, coluna) == 0) {
        printf("\nCoordenadas invalidas!\n\nJogador %c digite outra linha: ", j);
        scanf("%d", &linha);
        printf("\nJogador %c digite outra coluna: ", j);
        scanf("%d", &coluna);
    }
    matrizJogo[linha][coluna] = j;
}

/* Funcao Que Checa Posicoes Vazias para caso o jogo dê velha */
int quantVazias() {
    int quantidade = 0;

    for (l = 0; l < 3; l++) {
        for (c = 0; c < 3; c++)
            if (matrizJogo[l][c] == ' ')
                quantidade++;
    }
    return quantidade;
}

/* Funcao Para Iniciar Jogo da Velha */
void jogar() {

    int jogador = 1, vitoriaDeX = 0, vitoriaDe0 = 0;;
    char jogador1 = 'X', jogador2 = '0';

    do {
        imprimir();
        if (jogador == 1) {
            lerCoordenadas(jogador1);
            jogador++;
            vitoriaDeX += ganhouPorLinhas(jogador1);
            vitoriaDeX += ganhouPorColunas(jogador1);
            vitoriaDeX += ganhouPorDiagPrin(jogador1);
            vitoriaDeX += ganhouPorDiagSec(jogador1);
        } else {
            lerCoordenadas(jogador2);
            jogador = 1;
            vitoriaDe0 += ganhouPorLinhas(jogador2);

```



```

        vitoriaDe0 += ganhouPorColunas(jogador2);
        vitoriaDe0 += ganhouPorDiagPrin(jogador2);
        vitoriaDe0 += ganhouPorDiagSec(jogador2);
    }
    system("CLS");
} while (vitoriaDeX == 0 && vitoriaDe0 == 0 && quantVazias() > 0);

imprimir();

if (vitoriaDe0 == 1) {
    printf("\nParabens Jogador 2. Voce venceu!!!\n");
    quantidade_2 += 1;
} else if (vitoriaDeX == 1) {
    printf("\nParabens Jogador 1. Voce venceu!!!\n");
    quantidade_1 += 1;

} else {
    printf("\nIsh Parece que Deu velha!!!\n");
}

printf("O Jogador 1 ganhou %d vezes\n", quantidade_1);
printf("O jogador 2 ganhou %d vezes", quantidade_2);

}

/* Menu para selecionar no Switch */
int selecaoswitch() {
    int escolhido;

    printf("--- JOGO DA VELHA ---");
    printf("\n1- Jogar");
    printf("\n2- Creditos");
    printf("\n3- Rank");
    printf("\n4- Sair\n\n");
    scanf("%d", &escolhido);
    return escolhido;
}

/* Main */
int main() {

    setlocale(LC_ALL, "Portuguese");

    int menu;

    do {
        system("CLS");
        switch (selecaoswitch()) {
            case 1:
                do {
                    system("CLS");
                    inicializarMatriz();

```

```

        jogar(&quantidade_1, &quantidade_2);
        printf("\nDigite 1 para jogar novamente ou outro numero para sair: ");
        scanf("%d", &menu);
    } while (menu == 1);
    break;
case 2:
    system("CLS");
    printf("--- CREDITOS ---\n\n");
    printf("Leon Denis Silvestre de Lucena\n");
    printf("Matheus de Moura Aguiar Quintanilha\n");
    printf("Bruna Siqueira Correia\n");
    printf("Murilo Vicente Araujo Batista Ribeiro\n");
    break;
case 3:
    system("CLS");
    printf("--- R A N k ---\n\n");
    printf("O jogador 1 ganhou %d\n", quantidade_1);
    printf("O jogador 2 ganhou %d\n", quantidade_2);
    break;
case 4:
    exit(0);
    break;
default:
    printf("Opção invalida, digite um numero referente ao menu!");
}
printf("\nDigite 1 para retornar ao menu: ");
scanf("%d", &menu);
} while (menu == 1);

return 0;
}

```