

## **RELATÓRIO DO PROJETO:**

### **JOGO DA VELHA (3x3)**

#### **Nome e RGM:**

**Nome:** Slann Antonio Andrade Barbosa

**RGM:** 37120794

**Nome:** Ismael Victor Barros Machado

**RGM:** 38898900

**Nome:** Davi Barcia Viana

**RGM:** 38121379

**Nome:** Ana Laura de Moura Pereira

**RGM:** 38103397

**Nome:** Daniel Rozendo Lins

**RGM:** 38104075

#### **Introdução:**

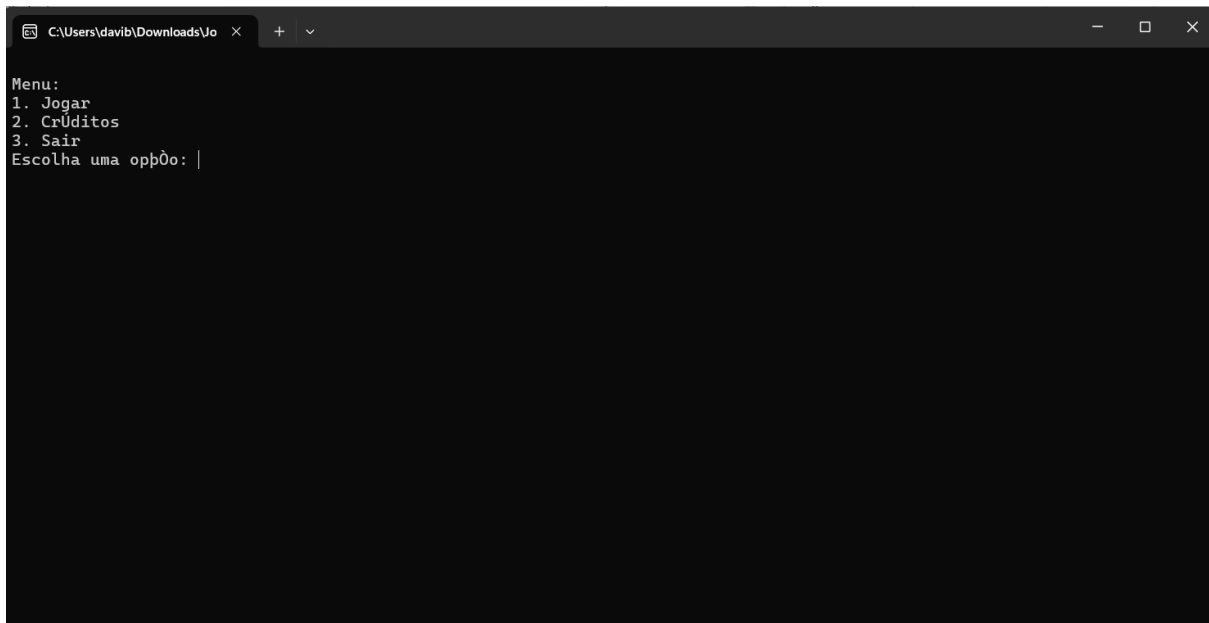
O jogo da velha é um jogo de tabuleiro simples e estratégico, jogado por duas pessoas, ou, nesse caso, por duas pessoas ou uma pessoa e o computador. O objetivo é alinhar três símbolos iguais em sequência, seja na horizontal, vertical ou diagonal, em uma grade de 3x3. **REGRAS:**

- Um jogador é escolhido para começar, seguido pelo adversário em turnos alternados, não sendo permitido jogadas do mesmo jogador em sequência.
- Cada jogador pode marcar apenas um espaço vazio por turno.
- Objetivo: ganhar o jogo formando uma linha de três símbolos iguais (X ou O) na horizontal, vertical ou diagonal.
- Empate: se todos os nove espaços forem preenchidos sem que nenhum jogador tenha formado uma linha, o jogo termina empatado.
- Não é permitido jogar em um espaço já ocupado.

#### **Resultados:**

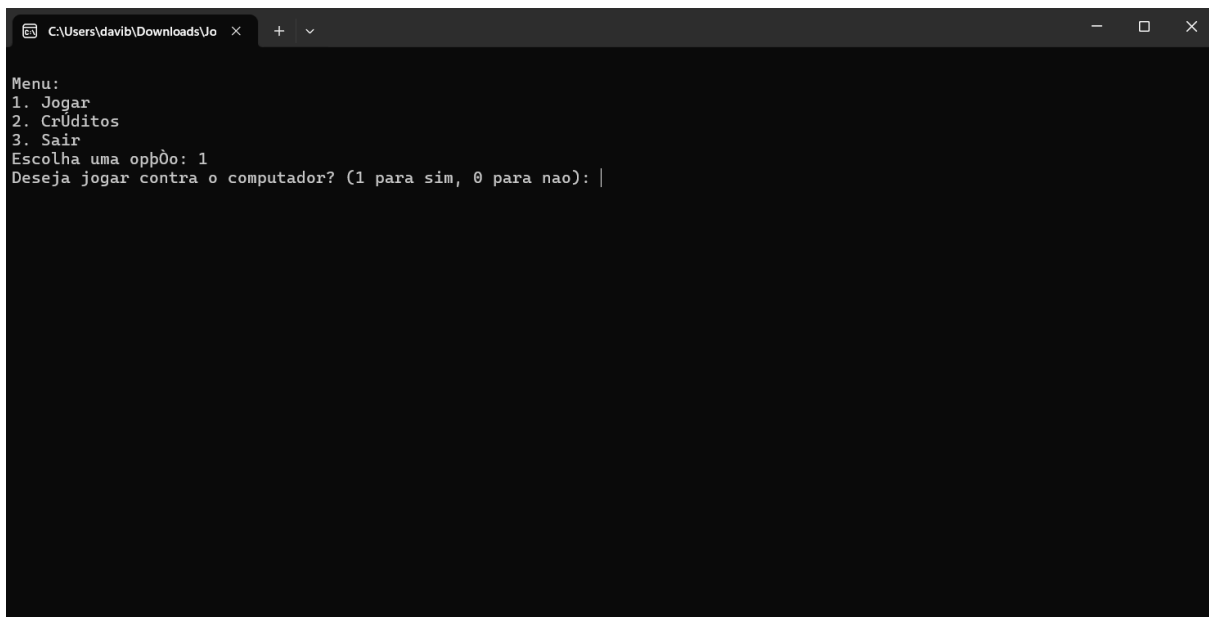
Jogo feito na linguagem C, com funções para iniciar e exibir o jogo na tela, verificação de vitórias e empates, e a função que permite que o computador faça as jogadas no lugar do segundo jogador. Além disso, o algoritmo começa pedindo para que o usuário escolha entre 3 opções: Jogar, Créditos e Sair (1, 2 e 3 respectivamente, com tratamento de

erro caso seja digitado outro número). Dessa maneira:



```
C:\Users\davib\Downloads\Jo x + v
Menu:
1. Jogar
2. CrUditos
3. Sair
Escolha uma opção: |
```

Após o usuário escolher a opção 1, o programa perguntará se deseja jogar contra o computador ou não.



```
C:\Users\davib\Downloads\Jo x + v
Menu:
1. Jogar
2. CrUditos
3. Sair
Escolha uma opção: 1
Deseja jogar contra o computador? (1 para sim, 0 para nao): |
```

Em caso de afirmativa, o nome do jogador 1 será solicitado e o jogo começa. Em caso de negativa, o programa irá pedir o nome dos jogadores 1 e 2.

```
C:\Users\davib\Downloads\Jo x + v
Menu:
1. Jogar
2. Créditos
3. Sair
Escolha uma opção: 1
Deseja jogar contra o computador? (1 para sim, 0 para nao): 1
Informe o nome do Jogador 1 (X): Davi
|
|
---+---+---
| |
---+---+---
| |
| |

Davi (X), informe a linha e a coluna (0, 1, 2): |
```

Após isso, o jogo fluirá normalmente, com a posição que o computador joga sendo determinada aleatoriamente. Além disso, se o jogador jogar em posição inválida ou posição já jogada, exibirá uma mensagem de erro.

Quando um jogador, ou o computador, alinhar 3 símbolos iguais em uma linha, coluna ou diagonal, ocorrerá vitória e o jogo voltará ao menu.

```
C:\Users\davib\Downloads\Jo x + v
| |
Davi (X), informe a linha e a coluna (0, 1, 2): 0 1
X | X |
---+---+---
| O |
---+---+---
| |

Computador jogou na posição: (2, 0)
X | X |
---+---+---
| O |
---+---+---
O | |

Davi (X), informe a linha e a coluna (0, 1, 2): 0 2
X | X | X
---+---+---
| O |
---+---+---
O | |

Davi venceu!

Menu:
1. Jogar
2. Créditos
3. Sair
Escolha uma opção: |
```

Na hora de fazer o computador escolher a posição, houve uma certa dificuldade de criar o código para isso, mas foi resolvido utilizando a função `rand()` e `srand`, no qual ele escolhe a posição aleatoriamente, independentemente da posição escolhida pelo usuário.

**Código fonte:**

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

// Variaveis globais para armazenar os nomes dos jogadores
char nomeJogador1[50];
char nomeJogador2[50];

// Estrutura para armazenar dados do jogador
typedef struct {
    char nome[50];
} jogador;

// Inicializa o tabuleiro do jogo, preenchendo cada posicao com um espaco vazio (' ').
void iniciarjogo(char tabuleiro[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            tabuleiro[i][j] = ' ';
        }
    }
}

// Funcao que exibe o tabuleiro na tela
void exibirjogo(char tabuleiro[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf(" %c ", tabuleiro[i][j]);
            if (j < 3 - 1) printf("|");
        }
        printf("\n");
        if (i < 3 - 1) {
            printf("----+----+---\n");
        }
    }
    printf("\n");
}

// Funcao para verificar se ha um vencedor, em linhas, colunas e diagonais
int verificarvencedor(char tabuleiro[3][3]) {
    // Verificar linhas e colunas
    for (int i = 0; i < 3; i++) {

        if (tabuleiro[i][0] == tabuleiro[i][1] && tabuleiro[i][1] == tabuleiro[i][2] &&
            tabuleiro[i][0] != ' ')
            return 1; // Vencedor na linha
    }
}

```

```

        if (tabuleiro[0][i] == tabuleiro[1][i] && tabuleiro[1][i] == tabuleiro[2][i] &&
tabuleiro[0][i] != ' ')
            return 1; // Vencedor na coluna
    }

    // Verificar diagonais
    if (tabuleiro[0][0] == tabuleiro[1][1] && tabuleiro[1][1] == tabuleiro[2][2] &&
tabuleiro[0][0] != ' ')
        return 1; // Vencedor na diagonal principal
    if (tabuleiro[0][2] == tabuleiro[1][1] && tabuleiro[1][1] == tabuleiro[2][0] &&
tabuleiro[0][2] != ' ')
        return 1; // Vencedor na diagonal secund?ria

    return 0; // Nenhum vencedor
}

// Fun???o para verificar se o tabuleiro esta cheio e se nao ha mais jogadas possiveis,
indicando um empate.
int verificarempate(char tabuleiro[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if (tabuleiro[i][j] == ' ') {
                return 0; // Ainda ha espacos vazios
            }
        }
    }
    return 1; // O tabuleiro esta cheio
}

// Permite que o computador escolha uma posicao aleatoria no tabuleiro para jogar,
garantindo que a posicao escolhida esteja vazia.
void jogarcomputador(char tabuleiro[3][3]) {
    int linha, coluna;
    do {
        // Gera uma linha e coluna aleatoria
        linha = rand() % 3;
        coluna = rand() % 3;
    }
    while (tabuleiro[linha][coluna] != ' '); // Verifica se a posicao esta vazia

    // Marca a posicao com 'O' (simbolo do computador)
    tabuleiro[linha][coluna] = 'O';
    printf("Computador jogou na posi???o: (%d, %d)\n", linha, coluna);
}

// Fun???o para exibir credits
void exibircredits() {
    printf("Desenvolvedor: Davi, Daniel, Slann, Ismael, Ana\n");
}

```

```

printf("Vers?o: 1.0\n");
printf("Obrigado por jogar!\n");
}

int main() {
    int opcao;
    jogador jogador1, jogador2; // Declaracao das variaveis do tipo jogador

    // Inicializa o gerador de numeros aleatorios
    srand(time(NULL));

    do{
        printf("\nMenu:\n");
        printf("1. Jogar\n");
        printf("2. Cr?ditos\n");
        printf("3. Sair\n");
        printf("Escolha uma op??o: ");
        scanf("%d", &opcao);

        switch (opcao) {
            case 1: {
                char tabuleiro[3][3];
                iniciarjogo(tabuleiro); // Inicializar o tabuleiro antes de
comecar o jogo
                int jogador = 1; // 1 para jogador 'X' e 2 para jogador 2
ou computador
                char simbolo; // 'X' para (Jogador 1) e 'O' para (Jogador
2 ou computador)
                int vencedor = 0;
                int jogarcomputadorouNao; // Determina se o
computador jogar ou nao
                int linha, coluna;

                // Pergunta se o jogador 2 sera controlado por uma
pessoa ou pelo computador
                printf("Deseja jogar contra o computador? (1 para sim,
0 para nao): ");
                scanf("%d", &jogarcomputadorouNao);

                // Solicita os nomes dos jogadores com base na
escolha
                if (jogarcomputadorouNao) {
                    printf("Informe o nome do Jogador 1 (X): ");
                    scanf("%s", jogador1.nome);
                    strcpy(jogador2.nome, "Computador");
                }
                else {
                    printf("Informe o nome do Jogador 1 (X): ");

```

```

scanf("%s", jogador1.nome);
printf("Informe o nome do Jogador 2 (O): ");
scanf("%s", jogador2.nome);
}

// Loop do jogo
while (!vencedor && !verificarempate(tabuleiro)) {
    exibirjogo(tabuleiro);
    if (jogador == 1) {
        simbolo = 'X';
        printf("%s (X), informe a linha e a coluna (0, 1, 2): ",
jogador1.nome);

        scanf("%d %d", &linha, &coluna);
    }
    else {
        if (jogarcomputadorouNao) {
            jogarcomputador(tabuleiro);
            vencedor =
verificarvencedor(tabuleiro);

            if (vencedor ||
verificarempate(tabuleiro)) break; {

                jogador = 1;
                continue;
            }
        }
        else {
            simbolo = 'O';
            printf("%s (O), informe a linha e a coluna (0, 1, 2):
", jogador2.nome);

            scanf("%d %d", &linha, &coluna);
        }
    }

    if (linha >= 0 && linha < 3 && coluna >= 0 && coluna <
3 && tabuleiro[linha][coluna] == ' ') {
        tabuleiro[linha][coluna] = simbolo;
        vencedor = verificarvencedor(tabuleiro);
        if (!vencedor) {
            jogador = (jogador == 1) ? 2 : 1; //
Alternar jogador apenas se nao houver vencedor
        }
    }
    else {
        printf("Posicao invalida ou ja ocupada. Tente
novamente.\n");
    }
}
}

```

```

        exibirjogo(tabuleiro);

        // Exibir o resultado do jogo
        if (vencedor) {
            printf("%s venceu!\n", (jogador == 2) ? jogador2.nome
: jogador1.nome);
        }
        else {
            printf("Terminou empate!\n");
        }
        break;

    }
    case 2:
        exibircreditos();
        break;
    case 3:
        printf("Saindo do jogo. Obrigado por jogar!\n");
        break;
    default:
        printf("Op??o inv?lida. Tente novamente.\n");
        break;
    }
} while (opcao != 3);

return 0;
}

```