Grupo: Maria Heloiza, Zafnny Carvalho, Leonardo Saraiva

Curso(s)/Área(s): Ciência da Computação 2 º Período B

Relatório: Projeto de Técnicas e Desenvolvimento de Algoritmos

Relatório: Jogo da Velha - Implementação em C

1. Introdução: Descrição do Jogo e Regras

O jogo desenvolvido é o **Jogo da Velha** é um jogo de tabuleiro para dois jogadores onde o objetivo é alinhar três símbolos (X ou O) de forma contínua, seja na horizontal, vertical ou diagonal.

- **Regras do Jogo:**
    1. O jogo é jogado em um tabuleiro 3x3, onde cada posição pode estar vazia ou ocupada por um "X" ou um "O".
    2. O jogador "X" sempre começa, e os jogadores alternam suas jogadas.
    3. O vencedor é aquele que conseguir alinhar três símbolos consecutivos (na linha, coluna ou diagonal).
    4. Se o tabuleiro estiver cheio e não houver vencedor, o jogo termina em **empate**.
    5. O jogo oferece a opção de jogar contra o computador, onde o computador faz jogadas aleatórias.

Além disso, o jogo mantém um **ranking** de vitórias, permitindo que o jogador registre suas vitórias e visualize um histórico.

2. Resultados: Descrição Geral do Jogo

- **Estrutura de Dados:**
  - A estrutura Jogador armazena o nome do jogador e o número de vitórias.
  - O tabuleiro é representado por uma matriz 3x3 de caracteres (char), onde os espaços vazios são representados por ' ' (espaço em branco), e os símbolos dos jogadores são representados por 'X' ou 'O'.
  -
- **Funções principais do jogo:**
  - **criarTabuleiro:** Cria o tabuleiro 3x3, alocando memória dinamicamente.
  - **exibirTabuleiro:** Exibe o estado atual do tabuleiro no console.
  - **verificarVencedor:** Verifica se algum jogador conseguiu formar uma linha, coluna ou diagonal com três símbolos consecutivos.
  - **tabuleiroCheio:** Verifica se o tabuleiro está completamente preenchido, indicando empate.
  - **jogarJogador:** Permite ao jogador humano inserir sua jogada (linha e coluna).
  - **jogarComputador:** Faz a jogada do computador, escolhendo uma posição aleatória disponível.
  - **salvarRanking e exibirRanking:** Permite salvar e exibir o ranking de vitórias dos jogadores.

3. Exemplificação de Código Fonte

Abaixo estão os trechos de código principais que implementam a lógica do jogo.

Criação do Tabuleiro:

```
char** criarTabuleiro() {
    char** tabuleiro = (char**)malloc(3 * sizeof(char*));
    for (int i = 0; i < 3; i++)
        tabuleiro[i] = (char*)malloc(3 * sizeof(char));

    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            tabuleiro[i][j] = ' ';  // Posicoes vazias do tabuleiro

    return tabuleiro;
}
```

Exibição do Tabuleiro:

```
void exibirTabuleiro(char** tabuleiro) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf(" %c ", tabuleiro[i][j]);
            if (j < 2) printf("|");
        }
        printf("\n");
        if (i < 2) printf("---+---+---\n");
    }
}
```

Verificação de Vencedor:

```
int verificarVencedor(char** tabuleiro, char jogador) {
    for (int i = 0; i < 3; i++) {
        if (tabuleiro[i][0] == jogador && tabuleiro[i][1] == jogador && tabuleiro[i][2] ==
jogador) return 1;
        if (tabuleiro[0][i] == jogador && tabuleiro[1][i] == jogador && tabuleiro[2][i] ==
jogador) return 1;
```

```
    }
    if (tabuleiro[0][0] == jogador && tabuleiro[1][1] == jogador && tabuleiro[2][2] ==
jogador) return 1;
    if (tabuleiro[0][2] == jogador && tabuleiro[1][1] == jogador && tabuleiro[2][0] ==
jogador) return 1;

    return 0;
}
```

Jogada do Computador (aleatória):

```
void jogarComputador(char** tabuleiro) {
    int linha, coluna;
    linha = rand() % 3;
    coluna = rand() % 3;

    while (tabuleiro[linha][coluna] != ' ') {
        linha = rand() % 3;
        coluna = rand() % 3;
    }

    tabuleiro[linha][coluna] = 'O';  // Coloca a jogada do computador
}
```

Dificuldades Encontradas e Soluções Implementadas

1. **Problema com a verificação de vitória**: Inicialmente, a verificação de vitória estava apenas comparando a linha e a coluna diretamente. No entanto, era necessário garantir que as duas diagonais também fossem verificadas.
    - **Solução**: Implementamos uma função que verifica as duas diagonais, além das linhas e colunas.
2. **Controle de jogadas no computador**: O computador, ao jogar, inicialmente não conseguia realizar jogadas inteligentes. Ele apenas escolhia uma posição aleatória.

- **Solução**: A solução foi deixar o computador escolher posições aleatórias dentro do tabuleiro, mas garantindo que a posição fosse vazia, o que foi feito com um laço while.

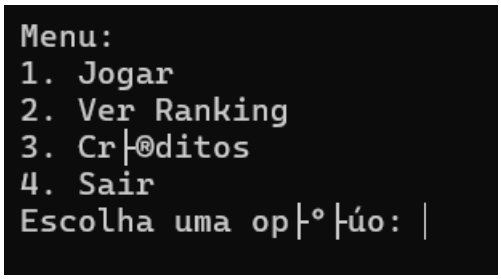
Funcionalidades Implementadas

- **Menu de Opções**: O menu inicial permite ao jogador escolher entre jogar, ver o ranking, visualizar os créditos ou sair do jogo.
- **Jogo com Computador**: Após o jogador escolher a opção "Jogar", o jogo se inicia. O jogador faz sua jogada, depois o computador faz a dele. O jogo continua até alguém vencer ou empatar.
- **Ranking**: O número de vitórias do jogador é armazenado em um arquivo de texto chamado "ranking.txt". A cada vitória, o jogador tem seu nome e número de vitórias registrados nesse arquivo.

5. Demonstrativo das Funcionalidades Implementadas

Prints da tela do funcionamento do jogo:

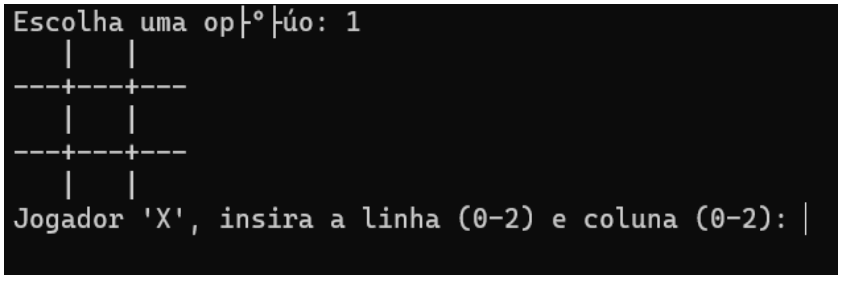
1. **Tela Inicial - Menu Principal:**



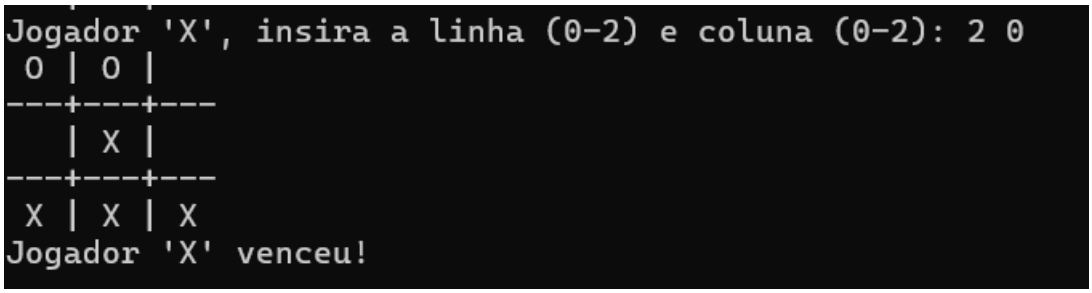
1. Jogar
2. Ver Ranking
3. Créditos
4. Sair

Escolha uma opção: 1

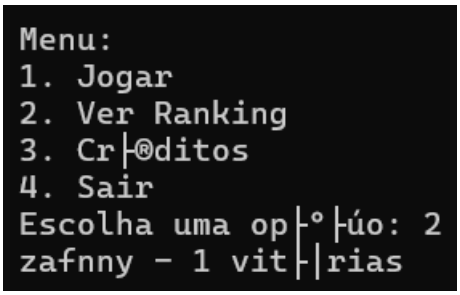
2. **Jogo em andamento:**



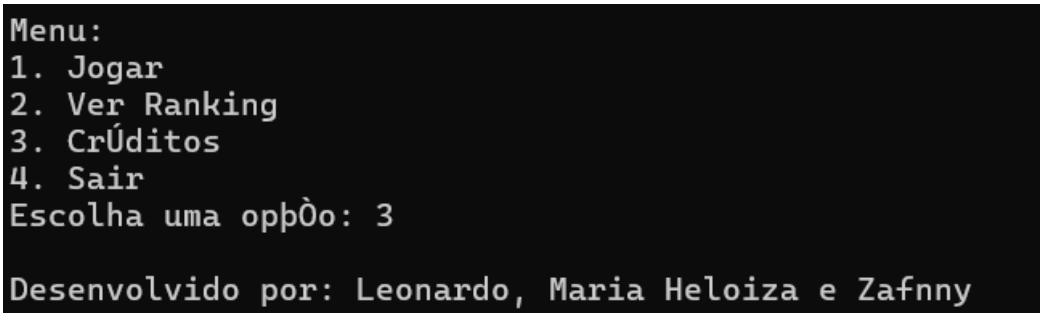
3. **Fim do Jogo - Vencedor:**



4. **Ranking: Opção 2**



5. **Créditos: Opção 3**



6. Apêndice - Código Fonte

Este é o código completo, com as principais funcionalidades de jogo, gerenciamento de ranking, e interação com o jogador e o computador.

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

typedef struct {
    char nome[50];
```

```
    int vitorias;
} Jogador;

char** criarTabuleiro() {
    char** tabuleiro = (char**)malloc(3 * sizeof(char*));
    for (int i = 0; i < 3; i++)
        tabuleiro[i] = (char*)malloc(3 * sizeof(char));
    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            tabuleiro[i][j] = ' ';
    return tabuleiro;
}

void exibirTabuleiro(char** tabuleiro) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf(" %c ", tabuleiro[i][j]);
            if (j < 2) printf("|");
        }
        printf("\n");
        if (i < 2) printf("---+---+---\n");
    }
}

int verificarVencedor(char** tabuleiro, char jogador) {
    for (int i = 0; i < 3; i++) {
        if (tabuleiro[i][0] == jogador && tabuleiro[i][1] == jogador && tabuleiro[i][2] ==
jogador) return 1;
        if (tabuleiro[0][i] == jogador && tabuleiro[1][i] == jogador && tabuleiro[2][i] ==
jogador) return 1;
    }
```

```
    if (tabuleiro[0][0] == jogador && tabuleiro[1][1] == jogador && tabuleiro[2][2] ==
jogador) return 1;
    if (tabuleiro[0][2] == jogador && tabuleiro[1][1] == jogador && tabuleiro[2][0] ==
jogador) return 1;
    return 0;
}

int tabuleiroCheio(char** tabuleiro) {
    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            if (tabuleiro[i][j] == ' ') return 0;
    return 1;
}

void jogarJogador(char** tabuleiro, char jogador) {
    int linha, coluna;
    printf("Jogador '%c', insira a linha (0-2) e coluna (0-2): ", jogador);
    scanf("%d %d", &linha, &coluna);
    tabuleiro[linha][coluna] = jogador;
}

void jogarComputador(char** tabuleiro) {
    int linha, coluna;
    linha = rand() % 3;
    coluna = rand() % 3;
    while (tabuleiro[linha][coluna] != ' ') {
        linha = rand() % 3;
        coluna = rand() % 3;
    }
    tabuleiro[linha][coluna] = 'O';
}
```

```
void salvarRanking(Jogador jogador) {
    FILE* arquivo = fopen("ranking.txt", "a");
    fprintf(arquivo, "%s - %d vitórias\n", jogador.nome, jogador.vitorias);
    fclose(arquivo);
}

void exibirRanking() {
    FILE* arquivo = fopen("ranking.txt", "r");
    char linha[100];
    while (fgets(linha, sizeof(linha), arquivo)) {
        printf("%s", linha);
    }
    fclose(arquivo);
}

int main() {
    srand(time(NULL));
    Jogador jogador;
    char** tabuleiro;
    int opcao;
    jogador.vitorias = 0;

    printf("Digite seu nome: ");
    fgets(jogador.nome, 50, stdin);

    do {
        printf("\nMenu:\n1. Jogar\n2. Ver Ranking\n3. Créditos\n4. Sair\nEscolha uma opção: ");
        scanf("%d", &opcao);
```

```
switch (opcao) {
case 1:
    tabuleiro = criarTabuleiro();
    int vencedor = 0;
    while (!vencedor && !tabuleiroCheio(tabuleiro)) {
        exibirTabuleiro(tabuleiro);
        jogarJogador(tabuleiro, 'X');
        if (verificarVencedor(tabuleiro, 'X')) {
            vencedor = 1;
            jogador.vitorias++;
        } else if (!tabuleiroCheio(tabuleiro)) {
            jogarComputador(tabuleiro);
            if (verificarVencedor(tabuleiro, 'O')) {
                vencedor = 1;
            }
        }
    }
    if (vencedor) {
        printf("Jogador '%c' venceu!\n", vencedor == 1 ? 'X' : 'O');
    } else {
        printf("Empate!\n");
    }
    salvarRanking(jogador);
    break;
case 2:
    exibirRanking();
    break;
case 3:
    printf("Desenvolvido por Leonardo\n");
    break;
case 4:
```

```
            printf("Saindo do jogo.\n");
        default:
            printf("Opção inválida.\n");
        }
    } while (opcao != 4);

    return 0;
}
```