UNIPÊ
Centro Universitário
de João Pessoa

Técnicas e Desenvolvimento de Algoritmos

Jogo da Velha

Aramis Lins Barreto Neto - 32830062

Arthur Xavier Cavalcante - 34448403

Augusto Vieira Medeiros - 33688648

Pedro Lucas Souto - 33372241

Rafael Cirne Medeiros - 32632312

Introdução

O Jogo é baseado nas regras padrões do jogo da velha tradicional, onde duas pessoas jogam alternadamente, preenchendo cada um dos espaços vazios. Tradicionalmente se faz com X e O como elementos para a marcação. Cada participante deve inserir a posição a qual deseja jogar. Vence o jogador que conseguir formar primeiro uma linha com três símbolos iguais, seja ela na horizontal, vertical ou diagonal.

Resultados

Nossa primeira dificuldade foi como iriamos resolver caso desse empate nas jogadas, foi quando pensamos em criar uma função para verificar se o tabuleiro já estava totalmente preenchido, pois se já estivesse as 9 posições preenchidas e não houvesse nenhum vencedor, era porque nenhum jogador completou as fileiras com os símbolos iguais.

```
37  // Função para verificar se o tabuleiro está cheio (empate)
38  int tabuleiroCheio(char* tabuleiro) {
39      for (int i = 0; i < size; i++) {
40          if (*(tabuleiro + i) == ' ')
41              return 0; // Não está cheio
42      }
43      return 1; // Está cheio
44  }
45
```

Implementamos uma função para imprimir nosso tabuleiro, para quando quisermos chamar ele novamente, não termos que aumentar as linhas dentro da ***main***.

```
6
7   // Função para exibir o tabuleiro
8   void exibirTabuleiro(char* tabuleiro) {
9       printf("  0 1 2\n");
10      for (int i = 0; i < 3; i++) {
11          printf("%d ", i);
12          for (int j = 0; j < 3; j++) {
13              printf("%c ", *(tabuleiro + i * 3 + j));
14          }
15          printf("\n");
16      }
17      printf("\n");
18  }
```

Para iniciar nosso jogo teríamos que fazer uma alocação da memória deixando mais fácil a utilização dos Bytes para o jogo.

```
    char* tabuleiro = (char*)malloc(size * sizeof(char));
    for (int i = 0; i < size ; i++) {
        *(tabuleiro + i) = ' ';
    }
```

Criamos uma função para verificar as jogadas e saber qual foi o jogador que venceu, fazendo com que dentro da ***main*** toda vez que o jogador, fizesse a jogada realizasse essa verificação.

```
// Função para verificar se alguém ganhou
int verificarGanhador(char* tabuleiro, char jogador) {
    for (int i = 0; i < 3; i++) {
        if ((*(tabuleiro + i * 3) == jogador && *(tabuleiro + i * 3 + 1) == jogador && *(tabuleiro + i * 3 + 2) == jogador) ||
            (*(tabuleiro + i) == jogador && *(tabuleiro + i + 3) == jogador && *(tabuleiro + i + 6) == jogador)) {
            return 1; // Ganhou
        }
    }

    if ((*(tabuleiro) == jogador && *(tabuleiro + 4) == jogador && *(tabuleiro + 8) == jogador) ||
        (*(tabuleiro + 2) == jogador && *(tabuleiro + 4) == jogador && *(tabuleiro + 6) == jogador)) {
        return 1; // Ganhou
    }

    return 0; // Não ganhou
}
```

APÊNDICE

- Aplicação Integral
  Utilização do DEV C++

JogoDaVelha.cpp

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

#define size 9

// Função para exibir o tabuleiro
void exibirTabuleiro(char* tabuleiro) {
    printf("  0 1 2\n");
    for (int i = 0; i < 3; i++) {
        printf("%d ", i);
        for (int j = 0; j < 3; j++) {
            printf("%c ", *(tabuleiro + i * 3 + j));
        }
        printf("\n");
    }
    printf("\n");
}

// Função para verificar se alguém ganhou
int verificarGanhador(char* tabuleiro, char jogador) {
    for (int i = 0; i < 3; i++) {
        if ((*(tabuleiro + i * 3) == jogador && *(tabuleiro + i * 3 + 1) == jogador && *(tabuleiro + i * 3 + 2) == jogador) ||
            (*(tabuleiro + i) == jogador && *(tabuleiro + i + 3) == jogador && *(tabuleiro + i + 6) == jogador)) {
            return 1; // Ganhou
        }
    }

    if ((*(tabuleiro) == jogador && *(tabuleiro + 4) == jogador && *(tabuleiro + 8) == jogador) ||
        (*(tabuleiro + 2) == jogador && *(tabuleiro + 4) == jogador && *(tabuleiro + 6) == jogador)) {
        return 1; // Ganhou
    }

    return 0; // Não ganhou
}

// Função para verificar se o tabuleiro está cheio (empate)
int tabuleiroCheio(char* tabuleiro) {
    for (int i = 0; i < size; i++) {
        if (*(tabuleiro + i) == ' ')
            return 0; // Não está cheio
    }
    return 1; // Está cheio
}

```

```c
int main() {
    setlocale(0, "Portuguese");

    char* tabuleiro = (char*)malloc(size * sizeof(char));
    for (int i = 0; i < size ; i++) {
        *(tabuleiro + i) = ' ';
    }

    char jogadorAtual = 'X';

    while (1) {
        // Exibir tabuleiro
        exibirTabuleiro(tabuleiro);

        // Obter a jogada do jogador
        int linha, coluna;
        printf("Jogador %c, faça sua jogada (linha e coluna): ", jogadorAtual);
        scanf("%d %d", &linha, &coluna);

        // Verificar se a posição é válida
        if (linha < 0 || linha >= 3 || coluna < 0 || coluna >= 3 || *(tabuleiro + linha * 3 + coluna) != ' ') {
            printf("Jogada inválida. Tente novamente.\n");
            continue;
        }

        // Realizar a jogada
        *(tabuleiro + linha * 3 + coluna) = jogadorAtual;
```

```c
        // Verificar se o jogador venceu
        if (verificarGanhador(tabuleiro, jogadorAtual)) {
            exibirTabuleiro(tabuleiro);
            printf("Parabéns, jogador %c! Você venceu!\n", jogadorAtual);
            break;
        }

        // Verificar empate
        if (tabuleiroCheio(tabuleiro)) {
            exibirTabuleiro(tabuleiro);
            printf("Empate! O jogo acabou.\n");
            break;
        }

        // Trocar o jogador
        jogadorAtual = (jogadorAtual == 'X') ? 'O' : 'X';
    }

    free(tabuleiro); // Liberar a memória alocada dinamicamente

    return 0;
}
```