**CENTRO UNIVERSITÁRIO DE JOÃO PESSOA**

**CIÊNCIAS DA COMPUTAÇÃO**

JOÃO VICTOR MAIA BRANCO (29100159),

LARA LOPES MARQUES (30044804),

LUCAS KAIKY PESSOA PINHO DE CARVALHO (30081084),

RODRIGO PEREIRA DE ALMEIDA (30173591)

JOGO DA VELHA EM C - RELATÓRIO

JOÃO PESSOA

2022

JOÃO VICTOR MAIA BRANCO

LARA LOPES MARQUES

LUCAS KAIKY PESSOA PINHO DE CARVALHO

RODRIGO PEREIRA DE ALMEIDA

T TÉCNICAS E LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS

Relatório apresentado na matéria de

técnicas e laboratório de desenvolvimento

de algoritmos, ciências da computação,

a ser utilizado para finalizar o semestre 2022.2

Professor Dr.Leonardo Ângelo Virginio de Souto

e Professor Walace Sartori Bonfim.

JOÃO PESSOA

2022

**SUMÁRIO**

**1. INTRODUÇÃO----------------------------------------------------------------------------------------1**

**2. RESULTADOS--------------------------------------------------------------------------------------2/3**

**3. APÊNDICE-----------------------------------------------------------------------------------------4/10**

**INTRODUÇÃO**

O Tic-Tac-Toe, mais conhecido no Brasil por Jogo da Velha, é um jogo extremamente tradicional que foi passado por gerações em diversas culturas diferentes. A sua origem é incerta, pois muitos dizem que o jogo foi criado na Inglaterra por mulheres que se reuniam para um chá da tarde, que a partir do tempo foram envelhecendo e já não enxergavam tão bem quanto antes para bordar, criando-se assim uma opção diferente de entretenimento. Entretanto, há pessoas que acreditam que o jogo é mais antigo, pois foram encontrados tabuleiros do jogo escavados em rochas de templos antigos no Antigo Egito, provavelmente feitos por escravos há mais de 3 mil anos atrás.

O jogo é realizado por duas pessoas, sendo cada um dos participantes representados por um símbolo (X ou O). Para o jogador ganhar é necessário conseguir formar uma linha com os símbolos iguais,independente da posição, seja ela, diagonal, horizontal e vertical.

As regras do Jogo da velha são simples:

· É necessário ter um tabuleiro de matriz 3 por 3 (3 linhas e 3 colunas)

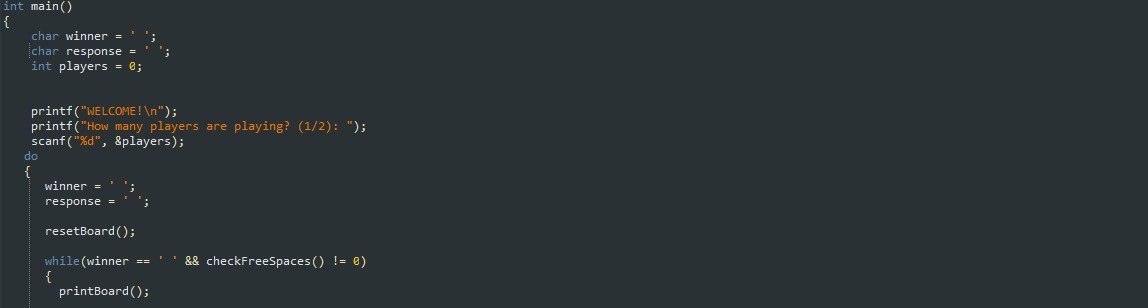
· Dois jogadores escolhem a sua marcação, geralmente sendo um ‘O’ e um ‘X’

· O objetivo é conseguir uma sequência da sua marcação em uma mesma linha, coluna ou diagonal, além de tentar impedir o adversário de realizar este feito.

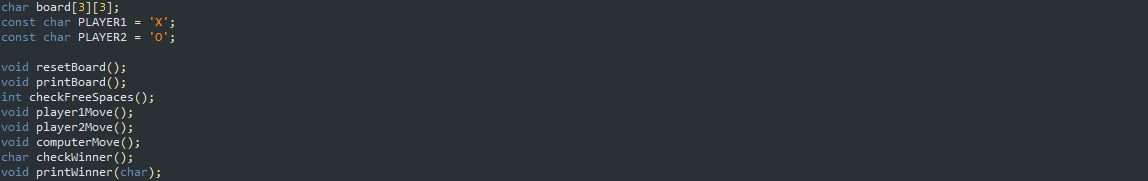
1

**RESULTADOS**

Diversas funções foram necessárias para efetuar cada passo do jogo, como o print do tabuleiro ‘printBoard()’, o movimento dos jogadores ‘player1Move()’ e ‘player2Move()’, no qual o jogo pede pro jogador selecionar a linha e a coluna desejada e caso haja apenas um jogador, o ‘computerMove()’ aparece como adversário, selecionando aleatoriamente uma linha e uma coluna. A cada jogada é realizada a função ‘checkFreeSpaces()’, no qual checa as casas disponíveis para efetuar jogadas, assim como a função ‘checkWinner()’ é ativada para conferir se a última jogada realizou a regra de vitória do jogo, analisando cada casa da matriz e vendo as marcações em cada uma. No final da partida, a ‘printWinner()’ mostra o vencedor –se houver- ou o empate, e após isso é perguntado se haverá outra partida, além de perguntar quantos jogadores irão participar. Se houver outro jogo, a ‘resetBoard()’ entra em ação limpando todas as casas da matriz, e o loop do jogo da velha é realizado



2



3

Apêndice:

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

#include <locale.h>

//Entrada: Inicialização das variáveis, do tabuleiro e das funções

char board[3][3];

const char PLAYER1 = 'X';

const char PLAYER2 = 'O';

void resetBoard();

void printBoard();

int checkFreeSpaces();

void player1Move();

void player2Move();

void computerMove();

char checkWinner();

void printWinner(char);

int main()

{

//Processamento: O tabuleiro é resetado, os participantes jogam até houver um vencedor, e ai o ganhador aparece na tela

setlocale(LC\_ALL, "Portuguese");

char winner = ' ';

char response = ' ';

int players = 0;

printf("BEM-VINDO!\n");

printf("QUANTOS JOGADORES IRÃO PARTICIPAR?? (1/2): ");

scanf("%d", &players);

do

{

winner = ' ';

response = ' ';

resetBoard();

while(winner == ' ' && checkFreeSpaces() != 0)

{

printBoard();

player1Move();

winner = checkWinner();

if(winner != ' ' || checkFreeSpaces() == 0)

{

break;

}

if(players == 2){

printBoard();

player2Move();

winner = checkWinner();

if(winner != ' ' || checkFreeSpaces() == 0)

{

break;

}

}

else{

computerMove();

winner = checkWinner();

if(winner != ' ' || checkFreeSpaces() == 0)

{

break;

}

}

}

printBoard();

printWinner(winner);

printf("\nQUER JOGAR NOVAMENTE? (Y/N): ");

scanf("%c");

scanf("%c", &response);

response = toupper(response);

if(response == 'Y'){

printf("QUANTOS JOGADORES IRÃO PARTICIPAR? (1/2): ");

scanf("%d", &players);

}

} while (response == 'Y');

printf("OBRIGADO POR JOGAR!");

return 0;

}

//Função que resetará o tabuleiro assim que chamado.

void resetBoard()

{

for(int i = 0; i < 3; i++)

{

for(int j = 0; j < 3; j++)

{

board[i][j] = ' ';

}

}

}

//Função para estilizado o tabuleiro do jogo

void printBoard()

{

printf(" %c | %c | %c ", board[0][0], board[0][1], board[0][2]);

printf("\n---|---|---\n");

printf(" %c | %c | %c ", board[1][0], board[1][1], board[1][2]);

printf("\n---|---|---\n");

printf(" %c | %c | %c ", board[2][0], board[2][1], board[2][2]);

printf("\n");

}

//função para chegar se há espaços vazios.

int checkFreeSpaces()

{

int freeSpaces = 9;

for(int i = 0; i < 3; i++)

{

for(int j = 0; j < 3; j++)

{

if(board[i][j] != ' ')

{

freeSpaces--;

}

}

}

return freeSpaces;

}

//Função com os movimentos do jogador 1.

void player1Move()

{

int x;

int y;

do

{

printf("INSERIR LINHA #(1-3): ");

scanf("%d", &x);

x--;

printf("INSIRA COLUNA #(1-3): ");

scanf("%d", &y);

y--;

if(board[x][y] != ' ')

{

printf("Movimento inválido!\n");

}

else

{

board[x][y] = PLAYER1;

break;

}

} while (board[x][y] != ' ');

}

//Função que torna possível o multijogador e os movimentos do player 2.

void player2Move()

{

int x;

int y;

do

{

printf("INSERIR LINHA #(1-3): ");

scanf("%d", &x);

x--;

printf("INSERIR COLUNA#(1-3): ");

scanf("%d", &y);

y--;

if(board[x][y] != ' ')

{

printf("Movimento inválido!\n");

}

else

{

board[x][y] = PLAYER2;

break;

}

} while (board[x][y] != ' ');

}

//Função que realiza os movimentos do adversário (o computador)

void computerMove()

{

srand(time(0));

int x;

int y;

if(checkFreeSpaces() > 0)

{

do

{

x = rand() % 3;

y = rand() % 3;

} while (board[x][y] != ' ');

board[x][y] = PLAYER2;

}

else

{

printWinner(' ');

}

}

//Função que irá conferir o vencedor.

char checkWinner()

{

//verificar linhas

for(int i = 0; i < 3; i++)

{

if(board[i][0] == board[i][1] && board[i][0] == board[i][2])

{

return board[i][0];

}

}

//verificar colunas

for(int i = 0; i < 3; i++)

{

if(board[0][i] == board[1][i] && board[0][i] == board[2][i])

{

return board[0][i];

}

}

//verificar diagonals

if(board[0][0] == board[1][1] && board[0][0] == board[2][2])

{

return board[0][0];

}

if(board[0][2] == board[1][1] && board[0][2] == board[2][0])

{

return board[0][2];

}

return ' ';

}

//saída: mostrar quem ganhou a partida

void printWinner(char winner)

{

if(winner == PLAYER1)

{

printf("JOGADOR 1 GANHOU!");

}

else if(winner == PLAYER2)

{

printf("JOGADOR 2 GANHOU!");

}

else{

printf("EMPATE!");

}

}