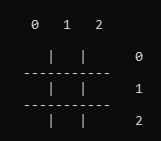
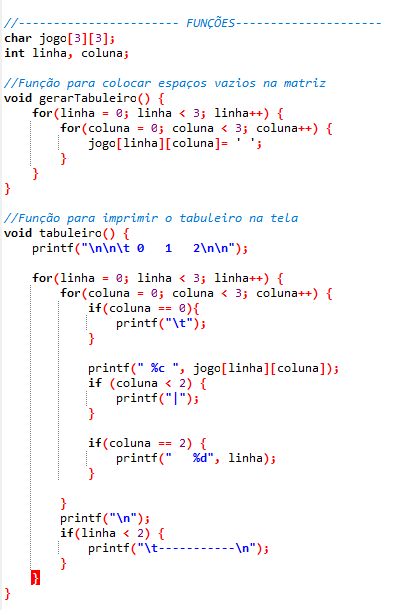
  
  
Técnicas e Desenvolvimento de Algoritmo  
  
Professor Wallace Sartori Bonfim  
  
Paulo Mariano Fernandes 32951507  
Daniel da Costa Bezerril 32555148  
Caio Kleivson Serafim Galdino Alves 34885862  
Pedro Henrique de Brito 33400504  
Filipe Marinho de Melo Klomfass 33115877  
  
  
Jogo da Velha 3x3  
  
Introdução:  
  
No jogo, dois jogadores participam. Os jogadores preenchem a área 3x3 com símbolos (O para um jogador e X para o outro), em cada rodada cada um seleciona apenas um espaço para marcar, com o objetivo sendo conseguir marcar uma linha de 3 símbolos idênticos, podendo ser uma linha horizontal, vertical ou diagonal. Caso isto não aconteça é considerado um empate.

Funcionamento:  
  
De início, são criadas variáveis com o objetivo de criar o tabuleiro do jogo, uma do tipo char e duas do tipo int, seguido logo por duas funções: uma para colocar espaços vazios na matriz a partir de comandos for (para que possam ser preenchidos durante o jogo) num tamanho 3x3, e outra para imprimir o tabuleiro na tela, também a partir de comandos for, como por exemplo o comando   
  
for(linha = 0; linha < 3; linha++) {  
for(coluna = 0; coluna < 3; coluna++) {.  
  
  
  
  
  
A próxima seção do código é a detecção da condição de vitória. A partir de funções, são realizadas verificações em todos os tipos de combinação de símbolos, verificando se houve uma vitória em uma linha, coluna ou diagonal. Após realizar essa verificação, o código tem duas respostas: caso tenha acontecido uma jogada vencedora, o código retorna 1 e acaba o jogo, mas caso o código retorne 0, o jogo continua. O código de verificar uma vitória em uma diagonal necessitou ser dividido em dois.  
  
Continuando, está próxima seção do código  
  
Apêndice:   
#include <stdio.h>

#include <string.h>

#include <locale.h>

//----------------------- FUNÇÕES---------------------

char jogo[3][3];

int linha, coluna;

//Função para colocar espaços vazios na matriz

void gerarTabuleiro() {

for(linha = 0; linha < 3; linha++) {

for(coluna = 0; coluna < 3; coluna++) {

jogo[linha][coluna]= ' ';

}

}

}

//Função para imprimir o tabuleiro na tela

void tabuleiro() {

printf("\n\n\t 0 1 2\n\n");

for(linha = 0; linha < 3; linha++) {

for(coluna = 0; coluna < 3; coluna++) {

if(coluna == 0){

printf("\t");

}

printf(" %c ", jogo[linha][coluna]);

if (coluna < 2) {

printf("|");

}

if(coluna == 2) {

printf(" %d", linha);

}

}

printf("\n");

if(linha < 2) {

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

}

}

}

/\*Função que ira verificar se alguem ganhou em todas as linhas

(se retornar 1 é considerado vitória, se retornar 0 o jogo continua)\*/

int verificarLinhas(char coluna) {

int ganhouPorLinha = 0;

int ganhou = 0;

int linha;

for(linha = 0; linha < 3; linha++) {

if(jogo[linha][0] == coluna && jogo[linha][1] == coluna && jogo[linha][2] == coluna) {

ganhouPorLinha = 1;

}

else {

ganhouPorLinha = 0;

}

if(ganhouPorLinha == 1){

ganhou = 1;

}

}

return ganhou;

}

/\*Função que ira verificar se alguem ganhou em todas as colunas

(se retornar 1 é considerado vitória, se retornar 0 o jogo continua) \*/

int verificarColunas(char j) {

int ganhou = 0;

int ganhouPorColuna = 0;

for(coluna = 0; coluna < 3; coluna++) {

if(jogo[0][coluna] == j && jogo[1][coluna] == j && jogo[2][coluna] == j) {

ganhouPorColuna = 1;

}

else {

ganhouPorColuna = 0;

}

if(ganhouPorColuna == 1){

ganhou = 1;

}

}

return ganhou;

}

/\*Função que verifica se alguem ganhou em uma das diagonais

(se retornar 1 é considerado vitória, se retornar 0 o jogo continua) \*/

int verificarDiag1(char coluna) {

if(jogo[0][0] == coluna && jogo[1][1] == coluna && jogo[2][2] == coluna) {

return 1;

}

else {

return 0;

}

}

/\*Função que verifica se alguem ganhou em uma das diagonais

(se retornar 1 é considerado vitória, se retornar 0 o jogo continua) \*/

int verificarDiag2(char coluna) {

if(jogo[0][2] == coluna && jogo[1][1] == coluna && jogo[2][0] == coluna) {

return 1;

}

else {

return 0;

}

}

//Função que valida as coordenadas digitadas

int validar(int linha, int coluna) {

if(linha >= 0 && linha < 3 && coluna >= 0 && coluna < 3 && jogo[linha][coluna] == ' '){

return 1;

}

else {

return 0;

}

}

//Função para ler as coordeanadas enviadas e armazenar

void Coordenadas(char j){

int l, c;

printf("\nDigite a linha que deseja marcar: ");

scanf("%d", &l);

printf("\nDigite a coluna que deseja marcar: ");

scanf("%d", &c);

while(validar(l,c) == 0) {

printf("\nEscolha uma posição válida. Digite a linha que deseja marcar: ");

scanf("%d", &l);

printf("\nDigite a coluna que deseja marcar: ");

scanf("%d", &c);

}

jogo[l][c] = j;

}

//Função que calcula a quantidade de casas vazias dentro do tabuleiro

int quantVazias() {

int quantidade = 0;

for(linha = 0; linha < 3; linha++) {

for(coluna = 0; coluna < 3; coluna++) {

if(jogo[linha][coluna] == ' ') {

quantidade++;

}

}

}

return quantidade;

}

/\*Função que ira fazer a verificação de ganhador e que ira trocar o simbolo marcado no tabuleiro(X,O)

a cada rodada\*/

int jogada() {

int jogador = 1, vitoriaX = 0, vitoriaO = 0;

char jogador1 = 'X', jogador2 = 'O';

do{

tabuleiro();

//verificações para ver se o jogador 1 ganhou

if(jogador == 1){

Coordenadas(jogador1);

jogador++;

vitoriaX += verificarLinhas(jogador1);

vitoriaX += verificarColunas(jogador1);

vitoriaX += verificarDiag1(jogador1);

vitoriaX += verificarDiag2(jogador1);

}

//verificações para ver se o jogador 2 ganhou

else {

Coordenadas(jogador2);

jogador = 1;

vitoriaO += verificarLinhas(jogador2);

vitoriaO += verificarColunas(jogador2);

vitoriaO += verificarDiag1(jogador2);

vitoriaO += verificarDiag2(jogador2);

}

}while(vitoriaX == 0 && vitoriaO == 0 && quantVazias() > 0);

tabuleiro();

if(vitoriaO == 1) {

printf("\nParabéns Jogador 2 você ganhou!\n");

return 2;

}

else if(vitoriaX == 1) {

printf("\nParabéns Jogador 1 você ganhou!\n");

return 1;

}

else {

printf("\nQue pena, deu empate!\n");

}

}

// Criando a função de imprimir o placar com os parâmetros de jogador como ponteiro

void placar(int resultado, int \*jogador1, int \*jogador2){

if(resultado == 1){

(\*jogador1)++;

}

if(resultado == 2){

(\*jogador2)++;

}

printf("Placar: \n %d x %d \n", \*jogador1, \*jogador2);

}

void limparPlacar(int \*jogador1, int \*jogador2) {

\*jogador1 = 0;

\*jogador2 = 0;

}

//----------------------------------- FIM DAS FUNÇÕES --------------------------------------------

int main () {

setlocale(0,"portuguese");

int opcaoMenu; // Vai gerenciar as opções do Menu

int opcao; // Gerencia se o jogador quer continuar jogando

//Parâmetros do jogo

int resultado = 0;

int jogador1 = 0;

int jogador2 = 0;

do{

printf("\n===============MENU====================\n\n");

printf("1. Jogar\n");

printf("\n2. Créditos\n");

printf("\n3. Regras\n");

printf("\n0. Sair do jogo\n");

printf("\nEscolha uma opção: ");

scanf("%d", &opcaoMenu);

switch (opcaoMenu) {

case 1:

// Código para jogar contra outro jogador

printf("\nHora de jogar!\n");

do{

gerarTabuleiro();

resultado = jogada();

placar(resultado, &jogador1, &jogador2);

printf("\n1. Jogar novamente \n");

printf("0. Voltar ao menu \n");

scanf("%d", &opcao);

} while(opcao == 1);

limparPlacar(&jogador1, &jogador2);

break;

case 2:

// Código para jogar contra a máquina (a ser implementado)

printf("\n\tCREDITOS DO JOGO\n\n");

printf("-Pedro Henrique de Brito\n");

printf("-Daniel da Costa Bezerril\n");

printf("-Paulo Mariano Fernandes\n");

printf("-Filipe Marinho de Melo Klomfass\n");

printf("-Caio Kleivson Serafim Galdino\n");

break;

case 3:

// Código para jogar contra a máquina (a ser implementado)

printf("\n\tREGRAS:\n");

printf("\nDois jogares irão fazer jogadas alternadas por rodadas no tabuleiro\nO primeiro será o X e outro o O. Ganha o jogo aquele que primeiro\nfizer uma sequência de 3 na horizontal, na vertical ou na diagnoal\nBoa sorte e bom jogo!\n");

break;

case 0:

printf("Saindo do jogo...\n");

return 0; // Termina o programa

default:

printf("\nOpção inválida! Por favor, escolha uma opção válida.\n");

break;

}

} while (opcaoMenu != 0);

}