PROJETO FINAL – JOGO DA MEMÓRIA

**1 – EQUIPE**

O projeto foi realizado pelos alunos do Curso de Ciências da Computação da Unipê, pertencente ao segundo período, turma B, discente da disciplina de Laboratório de Desenvolvimento de Algoritmos e Técnicas de Desenvolvimento de Algoritmos:

* Joel Adelaide Medeiros – RGM 29799384;
* Marcos Barbosa Vieira Filho - RGM: 30174503;
* Maria Helena de Medeiros Silva - RGM: 30227151.

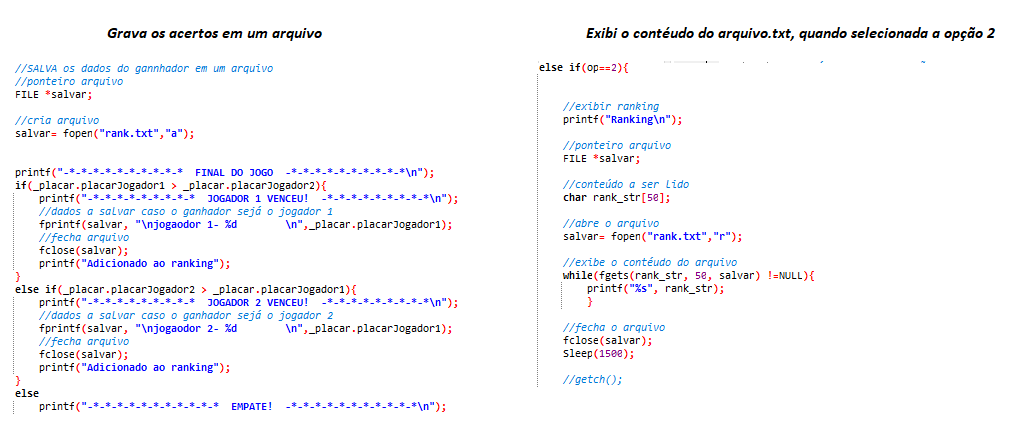
**2 – TEMA DO PROJETO**

O projeto desenvolvido foi um Jogo da Memória na linguagem C, como requisito avaliativo da nota do 2º estágio da disciplina de Desenvolvimento de Algoritmos e Técnicas de Desenvolvimento de Algoritmos.

**3 – DESCRIÇÃO DO PROJETO**

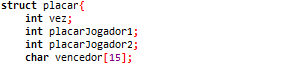
As funcionalidades contidas no projeto serão:

1. Inicio
   * Por meio do Método Void exibimos os créditos
   * Em seguida o usuário é direcionado a um Menu
2. Menu
   * São apresentadas 4 opções, entre as quais o usuário deve dar entrada em uma:
     + 1-Jogar
     + 2- Ranking
     + 3- Créditos
     + 4-Sair
   * Por meio de um While, após selecionar qualquer umas das 3 primeiras opções o usuário é redirecionado ao Menu novamente, quando selecionada a opção 4 faz a aplicação parar
   * Caso Opção 2-Ranking
     + São registrados o números de acertos do jogadores no rank.txt, por meio de um fprint, e exibidos por meio de um fgets

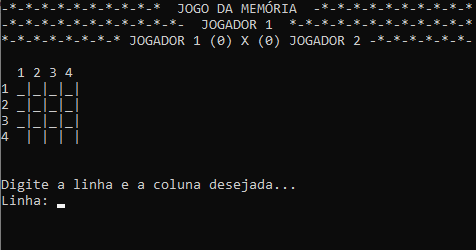
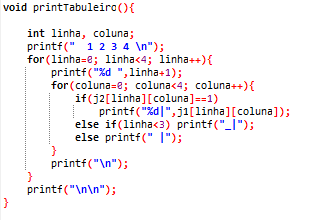


* + Caso Opção 3-Créditos
    - Exibi por meio do método Void printInicial(), os créditos do jogo
  + Caso Opção 4-Sair
    - Fecha o jogo, por meio de um exit(2);

1. Caso Opção 1-Jogar
   * Por meio de um Struct são organizados os dados de :
     + De quem é a vez de jogar
     + Número de pontos jogador 1
     + Número de pontos jogador 2



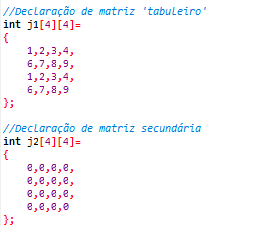
* + Um void printTopo exibe a vez de quem jogar e o número de pontos de cada jogador, apartir dos dados coletador no strict placar
  + Em seguida o void printTabuleiro, imprimi o tabulerio vazio

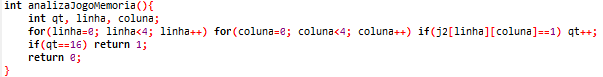
* + Em seguida o usuario insere os dados das coordenadas que deseja observar
  + Os dados são comparados de acordo com seguintes parâmetros
    - Local já foi descoberto
    - O valor inserido ultrapassa o número de linhas e colunas
    - Se o valor armazenado na primeira coordana e na segunda coordenada são iguais



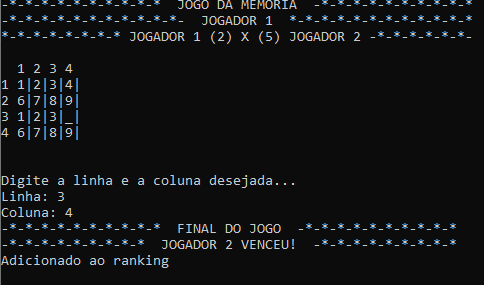
* + As matrizes j1,j2 armazenam os valores dos indices



* + O jogo acaba quando todas os indices foram descobertos, isso é analisado apartir do método a seguir



1. Fim
   * Os dois placares são comparados e o que tiver maior pontuação é armazenado no rank.txt.
   * Caso os dois placares sejam iguais, nenhum dado é armazenado no rank.txt



**APÊNDICE – Código Fonte**

****

// CENTRO UNIVERSITÁRIO DE JOÃO PESSOA - UNIPÊ

// CIÊNCIAS DA COMPUTAÇÃO

// TÉCNICAS DE DESENVOLVIMENTO DE ALGORITMOS

// LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS

// PROJETO FINAL - JOGO DA MEMÓRIA

// EQUIPE:

// Joel Adelaide Medeiros - RGM: 29799384

// Marcos Barbosa Vieira Filho - RGM: 30174503

// Maria Helena de Medeiros Silva - RGM: 30227151

//Inclusão de bibliotecas utilizadas no desenvolvimento do jogo

#include<stdio.h>

#include<windows.h>

#include<stdlib.h>

#include <locale.h>

//Declaração de matriz 'tabuleiro'

int j1[4][4]=

{

1,2,3,4,

6,7,8,9,

1,2,3,4,

6,7,8,9

};

//Declaração de matriz secundária

int j2[4][4]=

{

0,0,0,0,

0,0,0,0,

0,0,0,0,

0,0,0,0

};

//Definição de Struct para marcação do placar

struct placar{

int vez;

int placarJogador1;

int placarJogador2;

};

// Método que define alguns prints iniciais do jogo

void printInicial(){

printf("\t\t\tCENTRO UNIVERSITÁRIO DE JOÃO PESSOA - UNIPÊ\n");

Sleep(1500);

printf("\t\t\tCIÊNCIAS DA COMPUTAÇÃO\n");

Sleep(1500);

printf("\t\t\tTÉCNICAS DE DESENVOLVIMENTO DE ALGORITMOS\n");

Sleep(1500);

printf("\t\t\tLABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS\n\n");

Sleep(1500);

printf("\t\t\tPROJETO FINAL - JOGO DA MEMÓRIA\n\n");

Sleep(1500);

printf("\t\t\tEQUIPE:\n");

Sleep(1500);

printf("\t\t\tJoel Adelaide Medeiros - RGM: 29799384\n");

Sleep(1500);

printf("\t\t\tMarcos Barbosa Vieira Filho - RGM: 30174503\n");

Sleep(1500);

printf("\t\t\tMaria Helena de Medeiros Silva - RGM: 30227151\n\n\n");

Sleep(1500);

system("cls");

}

// Método que exibi, o placar do jogo

void printTopo(struct placar \_placar){

printf("-\*-\*-\*-\*-\*-\*-\*-\*-\*-\* JOGO DA MEMÓRIA -\*-\*-\*-\*-\*-\*-\*-\*-\*-\*\n");

if(\_placar.vez == 1)

printf("-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*- JOGADOR 1 \*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*\n");

else if(\_placar.vez == 2)

printf("-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*- JOGADOR 2 \*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*\n");

printf("\*-\*-\*-\*-\*-\*-\*-\* JOGADOR 1 (%d) X (%d) JOGADOR 2 -\*-\*-\*-\*-\*-\*-\n\n", \_placar.placarJogador1, \_placar.placarJogador2);

}

// Método que printa o tabuleiro do jogo

void printTabuleiro(){

int linha, coluna;

printf(" 1 2 3 4 \n");

for(linha=0; linha<4; linha++){

printf("%d ",linha+1);

for(coluna=0; coluna<4; coluna++){

if(j2[linha][coluna]==1)

printf("%d|",j1[linha][coluna]);

else if(linha<3) printf("\_|");

else printf(" |");

}

printf("\n");

}

printf("\n\n");

}

// Método de análise do jogo da memoria

int analizaJogoMemoria(){

int qt, linha, coluna;

for(linha=0; linha<4; linha++) for(coluna=0; coluna<4; coluna++) if(j2[linha][coluna]==1) qt++;

if(qt==16) return 1;

return 0;

}

// Método marca pontos

struct placar PontuaPlacar(struct placar \_placar){

if(\_placar.vez == 1){

\_placar.placarJogador1 += 1;

\_placar.vez = 2;

}

else if(\_placar.vez == 2){

\_placar.placarJogador2 += 1;

\_placar.vez = 1;

}

return \_placar;

}

// Método não marca ponto

struct placar PontuaPlacarErro(struct placar \_placar){

if(\_placar.vez == 1){

\_placar.placarJogador2 -= 1;

}

else if(\_placar.vez == 2){

\_placar.placarJogador1 -= 1;

}

return \_placar;

}

// Método que printa o resultado final

void printFinal(struct placar \_placar){

//SALVA os dados do gannhador em um arquivo

//ponteiro arquivo

FILE \*salvar;

//cria arquivo

salvar= fopen("rank.txt","a");

printf("-\*-\*-\*-\*-\*-\*-\*-\*-\*-\* FINAL DO JOGO -\*-\*-\*-\*-\*-\*-\*-\*-\*-\*\n");

if(\_placar.placarJogador1 > \_placar.placarJogador2){

printf("-\*-\*-\*-\*-\*-\*-\*-\*-\* JOGADOR 1 VENCEU! -\*-\*-\*-\*-\*-\*-\*-\*-\*\n");

//dados a salvar caso o ganhador sejá o jogador 1

fprintf(salvar, "\njogaodor 1- %d \n",\_placar.placarJogador1);

//fecha arquivo

fclose(salvar);

printf("Adicionado ao ranking");

}

else if(\_placar.placarJogador2 > \_placar.placarJogador1){

printf("-\*-\*-\*-\*-\*-\*-\*-\*-\* JOGADOR 2 VENCEU! -\*-\*-\*-\*-\*-\*-\*-\*-\*\n");

//dados a salvar caso o ganhador sejá o jogador 2

fprintf(salvar, "\njogaodor 2- %d \n",\_placar.placarJogador1);

//fecha arquivo

fclose(salvar);

printf("Adicionado ao ranking");

}

else

printf("-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\* EMPATE! -\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*\n");

}

// Método de inicialização MAIN()

int main(){

// Configuração de linguagem utilizada (Português)

setlocale(LC\_ALL, "Portuguese");

//Definição de Struct a ser utilizada

struct placar \_placar;

//Declaração de Variáveis

int linha, coluna, linha2, coluna2;

int op=1, gameover=0;

//Chama Método de Prints Iniciais

printInicial();

//Loop Inicial do Menu

while(op!=4){

system("cls");

//ENTRADA do Menu

printf("-\*-\*-\*-\*-\*-\*-\*-\*-\*-\* JOGO DA MEMÓRIA -\*-\*-\*-\*-\*-\*-\*-\*-\*-\*\n\n");

printf("Escolha uma das opções: \n");

printf("\n(1) - Jogar\n");

printf("(2) - Ranking\n");

printf("(3) - Créditos\n");

printf("(4) - Sair\n");

printf("\nOpção Desejada: ");

scanf("%d", &op);

//placar inicial

\_placar.placarJogador1 = 0;

\_placar.placarJogador2 = 0;

\_placar.vez = 1;

if(op==1){

//PROCESSAMENTO

while(!gameover){

jgd1: system("cls");

printTopo(\_placar);

printTabuleiro();

//ENTRADA do Usuário - Linha e Coluna desejada da primeira coordenada

printf("Digite a linha e a coluna desejada...\n");

printf("Linha: ");

scanf("%d", &linha);

printf("Coluna: ");

scanf("%d", &coluna);

linha--;

coluna--;

//verifica se o local já foi descoberto

if(j2[linha][coluna]==1){

printf("\nTente novamente! Local já descoberto! \n");

Sleep(1500);

getchar();

goto jgd1;

}

//se os valores ultrapassam o limite

if(linha<0 || linha>3 || coluna<0 || coluna>4){

printf("\nTente novamente! Insira valores entre 1 e 4!\n");

Sleep(1500);

getchar();

goto jgd1;

}

j2[linha][coluna]=1;

jgd2: system("cls");

printTopo(\_placar);

printTabuleiro();

//ENTRADA do Usuário - Linha e Coluna desejada da segunda coordenada

printf("Digite a linha e a coluna desejada...\n");

printf("Linha: ");

scanf("%d", &linha2);

printf("Coluna: ");

scanf("%d", &coluna2);

linha2--;

coluna2--;

//verifica se o local já foi descoberto

if(j2[linha2][coluna2]==1){

printf("\nTente novamente! Local já descoberto! \n");

Sleep(1500);

getchar();

goto jgd2;

}

//se os valores ultrapassam o limite

if(linha2<0 || linha2>3 || coluna2<0 || coluna2>4){

printf("\nTente novamente! Insira valores entre 1 e 4!\n");

Sleep(1500);

getchar();

goto jgd2;

}

j2[linha2][coluna2] = 1;

\_placar = PontuaPlacar(\_placar);

//verifica se o conteúdo da primeira coordenada e igual o da segunda

if(j1[linha][coluna] != j1[linha2][coluna2]){

\_placar = PontuaPlacarErro(\_placar);

system("cls");

printTopo(\_placar);

printTabuleiro();

printf("\nERROU!!!\n");

Sleep(1500);

j2[linha][coluna]=0;

j2[linha2][coluna2]=0;

}

gameover = analizaJogoMemoria();

}

//SAÍDA POS FINALIZAÇÃO DO JOGO DA MEMÓRIA

printFinal(\_placar);

Sleep(10000);

system("cls");

getchar();

}

else if(op==2){

//exibir ranking

printf("Ranking\n");

//ponteiro arquivo

FILE \*salvar;

//conteúdo a ser lido

char rank\_str[50];

//abre o arquivo

salvar= fopen("rank.txt","r");

//exibe o contéudo do arquivo

while(fgets(rank\_str, 50, salvar) !=NULL){

printf("%s", rank\_str);

}

//fecha o arquivo

fclose(salvar);

Sleep(1500);

//getch();

}

else if(op==3){

//mostra os créditos

printInicial();

}

}

//acaba o jogo

getchar();

exit(2);

}