

CIÊNCIAS DA COMPUTAÇÃO
LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS
TÉCNICAS DE DESENVOLVIMENTO DE ALGORITMOS

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.
- Sidnei Marcos- RGM: 29818591

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. Início

- 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

CIÊNCIAS DA COMPUTAÇÃO

LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS

TÉCNICAS DE DESENVOLVIMENTO DE ALGORITMOS

aplicação parar

- Caso Opção 2-Ranking
 - São registrados o números de acertos do jogadores no rank.txt, por meio de um fprintf, e exibidos por meio de um fopen

Grava acertos em um arquivo

```
// Método que grava o resultado final
void printFinal(struct placar _placar){
    //SALVA os dados do ganhador em um arquivo
    //abre o 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 seja o jogador 1
        fprintf(salvar, "%d %s %d\n", _placar.placarJogador1, _placar.nomeJogador1, _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 seja o jogador 2
        fprintf(salvar, "%d %s %d\n", _placar.placarJogador2, _placar.nomeJogador2, _placar.placarJogador2);
        //fecha arquivo
        fclose(salvar);
        printf("Adicionado ao ranking");
    }
    else
        printf("***** EMPATE! *****\n");
}
```

Exibi o conteúdo do arquivo

```
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");
    //le o conteúdo do arquivo
    while(fgets(rank_str, 50, salvar) != NULL){
        printf("%s", rank_str);
    }
    //fecha o arquivo
    fclose(salvar);
    Sleep(1500);
    getch();
}
```

- 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);

3. 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
 - Nome do jogador 1
 - Nome do jogador 2

```
struct placar{
    int vez;
    int placarJogador1;
    int placarJogador2;
    char nomeJogador1[20];
    char nomeJogador2[20];
};
```

- Ambos o jogadores inserem seus nomes
- Um void printTopo exibe a vez de quem jogar e o número de pontos de cada jogador, apartir dos dados coletador no struct placar
- Em seguida o void printTabuleiro, imprime o tabuleiro vazio

```
***** JOGO DA MEMÓRIA *****
***** JOGADOR 1 *****
***** JOGADOR 2 *****
JOGADOR 1 - (lula) (0) X (0) - (lulu) JOGADOR 2

 1 2 3 4
1 -|-|-|
2 -|-|-|
3 -|-|-|
4 -|-|-|

Digite a linha e a coluna desejada...
Linha:
```

```
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");
}
```

- 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

CIÊNCIAS DA COMPUTAÇÃO

LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS

TÉCNICAS DE DESENVOLVIMENTO DE ALGORITMOS

- O valor inserido ultrapassa o número de linhas e colunas
- Se o valor armazenado na primeira coordenada e na segunda coordenada são iguais

```

verifica os dados da primeira coordenada
//verifica se o local já foi descoberto
if(j2[linha][coluna]==1){
    printf("\nTente novamente! Local já descoberto! \n");
    Sleep(1500);
    getch();
    goto jgd1;
}

//se os valores ultrapassarem o limite
if(linha<0 || linha>3 || coluna<0 || coluna>4){
    printf("\nTente novamente! Insira valores entre 1 e 4!\n");
    Sleep(1500);
    getch();
    goto jgd1;
}

j2[linha][coluna]=1;
jgd2: system("cls");

printtopo(_placar);
printtabuleiro();

verifica os dados da segunda coordenada
//verifica se o local já foi descoberto
if(j2[linha2][coluna2]==1){
    printf("\nTente novamente! Local já descoberto! \n");
    Sleep(1500);
    getch();
    goto jgd2;
}

//se os valores ultrapassarem o limite
if(linha2<0 || linha2>3 || coluna2<0 || coluna2>4){
    printf("\nTente novamente! Insira valores entre 1 e 4!\n");
    Sleep(1500);
    getch();
    goto jgd2;
}

j2[linha2][coluna2] = 1;
_placar = PontuaPlacar(_placar);

//verifica se o conteúdo da primeira coordenada e igual a 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();
    
```

- As matrizes j1,j2 armazenam os seguintes valores dos índices

```

//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
};
    
```

- O jogo acaba quando todas os índices foram descobertos, isso é analisado a partir do método a seguir

```

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;
}
    
```

4. 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

```

***** JOGO DA MEMÓRIA *****
***** JOGADOR 2 *****
JOGADOR 1 - (lolo) (7) X (0) - (lulu) JOGADOR 2

 1 2 3 4
1 9|1|4|3|
2 7|7|1|8|
3 8|4|2|3|
4 6|2|9| |

Digite a linha e a coluna desejada...
Linha: 4
Coluna: 4
***** FINAL DO JOGO *****
***** JOGADOR 1 VENCEU! *****
Adicionado ao ranking
    
```

```

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 seja o jogador 1
    fprintf(salvar, "\n %s- %d\n", _placar.nomeJogador1, _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 seja o jogador 2
    fprintf(salvar, "\n %s- %d\n", _placar.nomeJogador2, _placar.placarJogador2);
    //fecha arquivo
    fclose(salvar);
    printf("Adicionado ao ranking");
}
else
    printf("***** EMPATE! *****\n");
    
```

CIÊNCIAS DA COMPUTAÇÃO
LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS
TÉCNICAS DE DESENVOLVIMENTO DE ALGORITMOS
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
// Sidnei Marcos- RGM: 29818591
```

```
//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;
    char nomeJogador1[20];
    char nomeJogador2[20];
};
```

```
// Método que define alguns prints iniciais do jogo
```

```
void printInicial(){
```

```
    printf("\t\tCENTRO UNIVERSITÁRIO DE JOÃO PESSOA - UNIPÊ\n");
    Sleep(1500);
    printf("\t\tCIÊNCIAS DA COMPUTAÇÃO\n");
```

PROJETO FINAL – JOGO DA MEMÓRIA
Laboratório de Desenvolvimento de Algoritmos
Técnicas de Desenvolvimento de Algoritmos

CIÊNCIAS DA COMPUTAÇÃO
LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS
TÉCNICAS DE DESENVOLVIMENTO DE ALGORITMOS

```
Sleep(1500);
printf("\t\tTÉCNICAS DE DESENVOLVIMENTO DE ALGORITMOS\n");
Sleep(1500);
printf("\t\tLABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS\n\n");
Sleep(1500);
printf("\t\tPROJETO FINAL - JOGO DA MEMÓRIA\n\n");
Sleep(1500);
printf("\t\tEQUIPE:\n");
Sleep(1500);
printf("\t\tJoel Adelaide Medeiros - RGM: 29799384\n");
Sleep(1500);
printf("\t\tMarcos Barbosa Vieira Filho - RGM: 30174503\n");
Sleep(1500);
printf("\t\tMaria Helena de Medeiros Silva - RGM: 30227151\n");
Sleep(1500);
printf("\t\tSidnei Marcos- RGM: 29818591\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 - (%s) (%d) X (%d) - (%s) JOGADOR 2 \n\n", _placar.nomeJogador1,
    _placar.placarJogador1, _placar.placarJogador2, _placar.nomeJogador2);

}

// 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 analisaJogoMemoria(){
    int qt, linha, coluna;
    PROJETO FINAL – JOGO DA MEMÓRIA
    Laboratório de Desenvolvimento de Algoritmos
    Técnicas de Deesnvolvimento de Algoritmos
```

CIÊNCIAS DA COMPUTAÇÃO
LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS
TÉCNICAS DE DESENVOLVIMENTO DE ALGORITMOS

```
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 ganhador 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 seja o jogador 1
        fprintf(salvar, "\n %s- %d \n", _placar.nomeJogador1, _placar.placarJogador1);
        //fecha arquivo
        fclose(salvar);
        printf("Adicionado ao ranking");
    }
    else if(_placar.placarJogador2 > _placar.placarJogador1){
PROJETO FINAL – JOGO DA MEMÓRIA  
Laboratório de Desenvolvimento de Algoritmos  
Técnicas de Desenvolvimento de Algoritmos
```

CIÊNCIAS DA COMPUTAÇÃO

LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS

TÉCNICAS DE DESENVOLVIMENTO DE ALGORITMOS

```
printf("_*_*_*_*_*_*_*_* JOGADOR 2 VENCEU! *_*_*_*_*_*_*_**\n");
//dados a salvar caso o ganhador seja o jogador 2
fprintf(salvar, "\n %s- %d      \n", _placar.nomeJogador2, _placar.placarJogador2);
//fecha arquivo
fclose(salvar);
printf("Adicionado ao ranking");
}
else
    printf("_*_*_*_*_*_*_*_* EMPATE! *_*_*_*_*_*_*_**\n");
}

void embaralharMatriz(){
    int i,j,r,y,temp;

    for ( i = 0; i < 4; i++ ) {
        for ( j = 0; j < 4; j++ ) {
            r = rand () % 4;
            y = rand () % 4;
            temp = j1 [ i ] [ j ];
            j1 [ i ] [ j ] = j1 [ r ] [ y ];
            j1 [ r ] [ y ] = temp;
        }
    }

    for ( i = 0; i < 4; i++ ) {
        for ( j = 0; j < 4; j++ ) {

            j2[i][j] = 0;}}
}
```

```
// 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");
```

CIÊNCIAS DA COMPUTAÇÃO
LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS
TÉCNICAS DE DESENVOLVIMENTO DE ALGORITMOS

```
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;

    embaralharMatriz();

if(op==1){
//PROCESSAMENTO

gameover=0;

system("cls");

printf("Digite o nome do Jogador 1: ");
scanf("%s", _placar.nomeJogador1);
printf("Digite o nome do Jogador 2: ");
scanf("%s", _placar.nomeJogador2);

system("cls");

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

CIÊNCIAS DA COMPUTAÇÃO
LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS
TÉCNICAS DE DESENVOLVIMENTO DE ALGORITMOS

```
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 = analisaJogoMemoria();
}
```

CIÊNCIAS DA COMPUTAÇÃO
LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS
TÉCNICAS DE DESENVOLVIMENTO DE ALGORITMOS

//SAÍDA POS FINALIZAÇÃO DO JOGO DA MEMÓRIA

```
        printFinal(_placar);
    Sleep(10000);
    system("cls");
    getch();
}
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 conteúdo 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
    system("cls");
    printInicial();
}

}

//acaba o jogo
getchar();
exit(2);
}
```