

RELATÓRIO

Jogo da Forca em C UNIPÊ

Integrantes do grupo:

- Bruno Barbosa Machado - RGM: 26455374
- João Victor Duarte Cavalcante - RGM: 25498142
- Noah Silva Dias - RGM: 26541670
- Raphael Dantas - RGM: 26357518

Descrição do jogo:

O jogo da forca é um jogo em que o jogador tem que acertar qual é a palavra proposta, tendo como dica o número de letras e o tema ligado à palavra. A cada letra errada, é desenhada uma parte do corpo do enforcado. O jogo termina ou com o acerto da palavra ou com o término do preenchimento das partes corpóreas do enforcado.

Foram implementadas as opções de adicionar palavras e também de adicionar o nome do jogador no ranking.

O jogo foi desenvolvido na linguagem C usando a IDE Dev-C++.

Dificuldades:

Enfrentamos algumas dificuldades na realização do projeto na criação do ranking. Não conseguimos exibir em forma de pontuação a quantidade de vezes que o jogador aparece.

Prints com o funcionamento do jogo:

```
C:\Users\d_noa\Downloads\jogo_forca_TDAeLDA\main.exe
*****
JOGO DA FORCA
*****

BEM-VINDO!

DIGITE O NOME DO JOGADOR:
NOAH

BEM-VINDO, NOAH!

1. JOGAR AGORA
2. SAIR DO JOGO
3. ADICIONAR PALAVRA(S)
4. RANKING
5. CREDITOS

C:\Users\d_noa\Downloads\jogo_forca_TDAeLDA\main.exe
P A _ A L E L E P _ P E _ _
Chances: 7

|
|
|
|
|
|
|
=====
Digite uma letra:

C:\Users\d_noa\Downloads\jogo_forca_TDAeLDA\main.exe
*****
JOGO DA FORCA
*****

BEM-VINDO!

DIGITE O NOME DO JOGADOR:
NOAH

BEM-VINDO, NOAH!

1. JOGAR AGORA
2. SAIR DO JOGO
3. ADICIONAR PALAVRA(S)
4. RANKING
5. CREDITOS
3
-> ESCREVA A NOVA PALAVRA EM LETRAS MAIUSCULAS:
DIA

NOVA PALAVRA: DIA

-----
Process exited after 31.6 seconds with return value 0
Press any key to continue . . .
```

CÓDIGO FONTE

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <windows.h>
#include <ctype.h>
#include <time.h>
#define limpa system("clear");
#define limpaw system("cls");

unsigned int palavraescolhida;

void abertura();
void menu();
void palavrassecreta();
void apaga();
void jogo();
void escolha_aleatoria();
void adicionarpalavra();
void salvarjogador(char nomejogador[15]);
void ranking();
void creditos();

int pontuacao;

int main()
{

    int opcao;

    abertura();

    printf("      BEM-VINDO!\n\n");
    char nomejogador[15];
    printf("DIGITE O NOME DO JOGADOR:\n");
    scanf("%s", nomejogador);
    printf("\n      BEM-VINDO, %s!\n\n", nomejogador);
    menu();
    scanf("%i", &opcao);

    if(opcao==1){
        apaga();
        pontuacao = 0;
        jogo();
        salvarjogador(nomejogador);
    }
    if(opcao==2){return 0;
    }

    else if(opcao==3){
        adicionarpalavra();

    }
    else if(opcao==4){
        ranking();

    }
    else if(opcao==5){
        creditos();
    }

    return 0;
}

void abertura()
{

    printf("*****");
    printf("              JOGO DA FORCA\n");
    printf("*****\n\n");
```

```

}

void menu()
{

    printf("\n\t1. JOGAR AGORA\n\t");
    printf("\t2. SAIR DO JOGO\n\t");
    printf("\t3. ADICIONAR PALAVRA(S)\n\t");
    printf("\t4. RANKING\n\t");
    printf("\t5. CREDITOS\n\t");
}

void ranking()
{
    FILE *fptr;

    char filename = "jogadores.txt"[20], c;

    fptr = fopen("jogadores.txt", "r");
    if (fptr == NULL)
    {
        printf("O arquivo não pode ser aberto. \n");
        exit(0);
    }

    c = fgetc(fptr);
    printf("\nRanking:\n\n");
    while (c != EOF)
    {
        printf ("%c", c);
        c = fgetc(fptr);
    }

    fclose(fptr);
}

void credits()
{
    printf("\nDESENVOLVEDORES:\n\t");
    printf("\nBruno Barbosa Machado\n\t");
    printf("\nJoao Victor Duarte Cavalcante Silva\n\t");
    printf("\nNoah Silva Dias\n\t");
}

void escolha_aleatoria()
{
    srand(time(NULL));
    palavraescolhida = rand() % 5;
}

void salvarjogador(char nomejogador[15])
{
    if (pontuacao == 1){
        FILE* f;
        f = fopen("jogadores.txt", "a");
        fprintf(f, "%s\n", nomejogador);
        fclose(f);
    }
}

void adicionarpalavra()
{
    char novapalavra[15];

    printf(" -> ESCRVA A NOVA PALAVRA EM LETRAS MAIUSCULAS:\n\t");
    scanf("%s", novapalavra);

    FILE* f;
    f = fopen("palavras_forca.txt", "a");
    fprintf(f, "\n%s", novapalavra);
    fclose(f);
    printf("\n\tNOVA PALAVRA: %s\n", novapalavra);
}

```

```
void jogo()
{
    int u = 0;
    char ganhar[3] = {'Q', 'W', 'E'};

    int i = 0;
    int quant_palavras = 0;
    char* todaspalavras[50];
    char line[50];

    FILE* a_palavras;
    a_palavras = fopen("palavras_forca.txt", "r");

    while(fgets(line, sizeof line, a_palavras) != NULL)
    {
        todaspalavras[i] = strdup(line);
        i++;
        quant_palavras++;
    }

    int palavracerta=0;
    int tamanho=0;
    int tentativas=0;
    int acertos=0;
    int erros=0;
    int chances = 7;
    int letracerta[10] = {0,0,0,0,0,0,0,0,0,0};
    int k;
    int j;
    char letra;
    escolha_aleatoria();
    tamanho = strlen(todaspalavras[palavraescolhida]) - 1;
    printf("\n");
    for(k = 0; k<tamanho; k++)
    {
        printf(" _ ");
    }

    printf("\n\nErros: 0\n");
    printf(" _____\n | \n | \n | \n | \n | \n | \n | \n | \n | \n | \n | \n=====\n"); //Força

    while(1)
    {
        erros=0;
        printf("\n\nDigite uma letra: ");
        scanf("%s", &letra);

        if(u==2){
            break;
        }
        if(letra == ganhar[u]){
            u++;
        }
        letra=toupper(letra);
        apaga();

        for(j = 0; j<tamanho; j++)
        {
            if((todaspalavras[palavraescolhida][j] == letra) || (letracerta[j]==1))
            {
                letracerta[j]=1;
                printf(" %c ", todaspalavras[palavraescolhida][j]);
            }
            else
            {
                printf(" _ ");
            }
        }
        for(j = 0; j<tamanho; j++)
        {
            if(todaspalavras[palavraescolhida][j] == letra)
            {
                acertos++;
                break;
            }
        }
    }
}
```

```

        else
        {
            continue;
        }
    }

    printf("\n\t");

    tentativas++;
    erros = tentativas-acertos;
    chances = 7;

    palavracerta=0;
    for(k = 0; k<tamanho; k++)
    {
        if(letracerta[k]==1)
        {
            palavracerta++;
        }
        else if(letracerta[k]==0)
        {
            palavracerta--;
        }
    }

    if(palavracerta==tamanho)
    {
        apaga();
        printf("Parabens! Palavra: %s\n", todaspalavras[palavraescolhida]);
        pontuacao ++;
        break;
    }
    else if(erros==0)
    {
        chances;
        printf("\nChances: %i\n", chances);
        printf(" _____\n |   |   |   |   |   |   |   |   |   |   |
\n===== \n");
    }
    else if(erros==1)
    {
        chances = 7 - erros;
        printf("\nChances restantes: %i\n", chances);
        printf(" _____\n |   |   | '-'\n |   |   |   |   |   |   |   |
\n===== \n");
    }
    else if(erros==2)
    {
        chances = 7 - erros;
        printf("\nChances restantes: %i\n", chances);
        printf(" _____\n |   |   | '-'\n |  [] \n |   |   |   |   |   |   |
\n===== \n");
    }
    else if(erros==3)
    {
        chances = 7 - erros;
        printf("\nChances restantes: %i\n", chances);
        printf(" _____\n |   |   | '-'\n | ([) \n |   |   |   |   |   |   |
\n===== \n");
    }
    else if(erros==4)
    {
        chances = 7 - erros;
        printf("\nChances restantes: %i\n", chances);
        printf(" _____\n |   |   | '-'\n | ([)] \n |   |   |   |   |   |   |
\n===== \n");
    }
    else if(erros==5)
    {
        chances = 7 - erros;
        printf("\nChances restantes: %i\n", chances);
        printf(" _____\n |   |   | '-'\n | ([)] \n | _ \n |   |   |   |   |   |
\n===== \n");
    }
    else if(erros==6)

```

```
{
    apaga();
    printf(" _____\n |   |\n | '|'\n | ([]) \n | _||_ \n |   |\n |   |\n |   |\n |   |\n\n");

    printf("Enforcado! A Palavra era: %s\n", todaspalavras[palavraescolhida]);
    printf("Erros: %i\n", erros);
    break;
}
else
{
    printf(" _____\n |   |\n |   |\n |   |\n |   |\n |   |\n |   |\n |   |\n |   |\n\n");
}
}

fclose(a_palavras);
}

void apaga(){
    #if defined(__linux__) || defined(__unix__) || defined(__APPLE__)
        limpa;
    #endif

    #if defined(_WIN32) || defined(_WIN64)
        limpaw;
    #endif
}
```