

CENTRO UNIVERSITÁRIO DE JOÃO PESSOA

Alunos: Bruno Hebert Sousa Felix, Caio Araújo De Vasconcelos, Caio Justino De Andrade, Marllon Brenno Barbosa Serpa, Paulo Fernandes Pinheiro Neto

Jogo Da Forca em linguagem de programação C

Turma: Ciência da computação, Manhã

CENTRO UNIVERSITÁRIO DE JOÃO PESSOA

ALUNO	RGM
BRUNO HEBERT SOUSA FELIX	34895337
CAIO ARAÚJO DE VASCONCELOS	34044493
CAIO JUSTINO DE ANDRADE	34665072
MARLLON BRENNO BARBOSA SERPA	33623112
PAULO FERNANDES PINHEIRO NETO	33920095

Sumario

1. Introdução	4
1.2 objetivos do trabalho	4
1.3 Regras do jogo	4
2. Resultados encontrados	4
2.3 Descrição geral do programa.....	4
2.4 Dificuldades e Soluções	5
2.5 Demonstrativo das funcionalidades implementadas.....	5

1. Introdução

1.2 objetivos do trabalho

Executar um jogo da forca em linguagem C, utilizando os conhecimentos observados em sala de aula e os adquiridos durante a pesquisa com o objetivo de cobrir os seguintes pontos:

- Ranking em arquivo
- Banco de dados de palavras carregadas
- Menu
- Controle de jogadas

1.3 Regras do jogo

O jogo da forca é um jogo de palavras onde o programa escolhera uma palavra e os jogadores tentarão adivinhar. Cada letra incorreta resulta na perda de uma “tentativa”. O objetivo é adivinhar todas as letras antes que o número de tentativas chegue a 0

2. Resultados encontrados

2.3 Descrição geral do programa

O programa se iniciará em um menu principal dentro de um Do While, já que necessita aparecer ao menos uma vez, o menu estará oferecendo 5 opções de saídas, caso o usuário digite 1 irá ser direcionado para menu do jogo da forca, caso digite 2 irá para menu de ranking, caso digite 3 para menu de banco de palavras, caso digite 4 para os créditos dos desenvolvedores e caso escolha o número 5 menus de fechamento de jogo dentro. Dentro do menu de jogo o usuário será apresentado a duas opções, caso escolha o número 1 se dará início ao jogo da forca, caso pressione o número 2 poderá ver um tutorial de regras e o número 3 retorna ao menu principal; o jogo inicia-se puxando do arquivo “Bancodepalavras.txt” e vai alocar em uma variável chamada “palavras” com um índice randômico, alocando-a como uma palavra oculta para o usuário, tendo uma função de verificação de letras e outra verificação de palavra para validar possíveis tentativas de acerto da palavra oculta, dentro do “int jogo” teremos a declaração da função de tempo que gerente a randomização das palavras existentes, a solicitação do número de jogadores e o número da tentativas por jogados.

Iniciado o jogo o programa exibira o jogador da vez e perguntara se ele gostaria de tentar uma letra ou a palavra completa, caso o caráter selecionado pelo usuário seja não alfabético ou vazio na tela aparecerá um aviso e ele retornara a mensagem de “Próxima letra”, caso seja alfabético dará prosseguimento com o programa buscando os caractere dentro da palavra oculta alocada no Array palavra secreta, caso a tentativa seja falha o programa informara o erro e mostrara a quantidade de tentativas, caso esteja correta ele informara e

mostrara a quantidade de tentativas. Caso algum jogador vença o programa exibira a palavra e uma mensagem de vitória e perguntara seu nome alocando-o na função ranking, posteriormente modificando o arquivo ranking.txt enumerando da maior para a menor pontuação (A pontuação terá como critério a razão acerto/tentativas)

2.4 Dificuldades e Soluções

Durante o desenvolvimento do projeto enfrentamos problemas na manipulação de arquivos externos usando o C e a criação e manutenção de bibliotecas próprias, necessitando de Elucidação externa por meio de veículos de informação acadêmica e plataformas online onde foram disponibilizados materiais de estudos.

2.5 Demonstrativo das funcionalidades implementadas

Dentro do código principal existem diferentes funcionalidades como as que serão descritas abaixo, utilizaremos o critério de organização “função/descrição” explicando o nome da funcionalidade ou seu nome de função dentro do código, sua descrição e se necessário a linha de código utilizada.

Função	Descrição
*escolherPalavraAleatoria(*nomeArquivo)	Escolhe uma palavra aleatória do arquivo BancoPalavras2.txt
Parâmetro *nomeArquivo	

```
// Função para escolher uma palavra aleatória do arquivo
char *escolherPalavraAleatoria(const char *nomeArquivo) {
    FILE *arquivo = fopen("BancoPalavras2.txt", "r");
    if (arquivo == NULL) {
        perror("Erro ao abrir o arquivo");
        exit(EXIT_FAILURE);
    }
}
```

Função	Descrição
InicializarPalavraOculta(*palavraEscolhida, *palavraOculta)	Dá início a palavra aleatória escolhida no banco de palavras
Parâmetro *palavraEscolhida	Copia e recebe uma palavra oculta que será usada no jogo
Parâmetro *palavraOculta	Recebe a palavra aleatória e através da quantidade caractere determina o tamanho de “_” na força

Função	Descrição
verificarPalavraCompleta(*palavraEscolhida, *tentativa)	Verifica se o jogador conseguiu acertar o chute da palavra oculta escolhida
Parâmetro *palavraEscolhida	É comparada através do parâmetro ‘strcmp’ com uma string escrita pelo jogador

Função	Descrição
verificarPalavraCompleta(*palavraEscolhida, *tentativa)	Verifica se o jogador conseguiu acertar o chute da palavra oculta escolhida
Parâmetro *palavraEscolhida	É comparada através do parâmetro ‘strcmp’ com uma string escrita pelo jogador

Para a implementação dos sistemas de rodadas e multiplayer fizemos utilização de um sistema de modificações simples no código a variar de acordo com o número de jogadores inseridos, isso de maneira bem empregada garantiu a possibilidade de dez ou mais jogadores.

Função	Descrição
Multiplayer	Verifica o número de jogadores e define os rounds baseado na vez de cada um deles

```
// Solicitar ao usuário o número de jogadores
printf("Capacidade de Jogadores: \n");
printf("[ 1 ] Solo\n");
printf("[ 2 ] Dois\n");
printf("[ 3 ] Três\n");
printf("[ 4 ] Quatro\n");
printf("[ 5++ ] 5 ou mais\n");
printf("Digite o numero de jogadores: ");
scanf("%d", &numJogadores);

if (numJogadores <= 0) {
    printf("Numero de jogadores invalido. Saindo...\n");
    return 1;
}

char *palavra = escolherPalavraAleatoria("Bancopalavras2.txt");
char palavraOculta[MAX_PALAVRA];
int tentativas = 0;
int jogadorAtual = 0;

if (palavra == NULL) {
    printf("Erro ao escolher a palavra. Saindo...\n");
    return 1;
}
```

Função	Descrição
verificarPalavraCompleta(*palavraEscolhida, *tentativa)	Verifica se o jogador conseguiu acertar o chute da palavra oculta escolhida
Parâmetro *palavraEscolhida	É comparada através do parâmetro 'strcmp' com uma string escrita pelo jogador

Função	Descrição
mostrarRank(FILE *rfile1)	Exibe a funcionalidade Ranking de vencedores
Parâmetro *FILE *rfile1	Endereço do arquivo onde se encontra o nome dos vencedores

Função	Descrição
armazenaRank(FILE *rfile2)	Exibe a funcionalidade de armazenar o nome dos jogadores no ranking após sua vitória
Parâmetro *FILE *rfile2	Endereço do arquivo onde se encontra o espaço para gravar o nome dos vencedores

```

void armazenaRank(FILE *rfile2){
    if (rfile2 == NULL) {
        printf("Arquivo não encontrado.\n");
        system("pause");
        exit(1);
    }

    char nomeWin[5];
    printf("Informe seu nome para o Ranking");
    scanf("%s", nomeWin);
    fprintf(rfile2, "\n");
    fputs(nomeWin, rfile2);
}

void mostrarRank(FILE *rfile1){
    if (rfile1 == NULL) {
        printf("Arquivo não encontrado.\n");
        system("pause");
        exit(1);
    }

    char mRank[100];
    while(fgets(mRank, 100, rfile1) != NULL){
        printf("%s", mRank);
    }
    printf("\n");
    system("pause");
}

```


Função	Descrição
alocarPalavra(FILE *file2)	Capaz de alocar palavras novas no banco de dados da força, para que o próprio usuário possa personalizar sua experiencia
Parâmetro *FILE *file2	Endereço do arquivo onde se encontra o espaço para gravar as novas palavras

```

void alocaPalavra(FILE *file2) {
    if (file2 == NULL) {
        printf("Arquivo não encontrado.\n");
        system("pause");
        exit(1);
    }

    char palavraNova[50];
    printf("\n--- Alocador de Palavras \n---\n");
    printf("\nInforme a palavra: ");
    scanf("%s", palavraNova);
    fprintf(file2, "\n");
    fputs(palavraNova, file2);
}

```

APÊNDICE - I

Demonstrativo do código fonte

```
//  
//Menu Jogo da Forca  
//  
//  
//Bibliotecas Externas  
#include <stdio.h>  
#include <string.h>  
#include <stdlib.h>  
#include <locale.h>  
  
//Bibliotecas Internas  
#include "Jogos.h"  
#include "Rank.h"  
#include "Banco.h"  
  
//Funções dos Menus  
//->Menu Jogo  
void menuJogo(int gameOp2){  
    setlocale(0, "portuguese");  
    int op_menu_jogo;  
    do{  
        printf("---\n Bem Vindo ao Jogo da Forca \n---\n");  
        printf("[ 1 ] Iniciar Jogo\n");  
        printf("[ 2 ] Tutorial/Regras\n");  
        printf("[ 3 ] Menu Principal\n");  
        scanf("%d", &op_menu_jogo);  
        system("cls");  
  
        switch(op_menu_jogo){
```

```

        case 1:
            Jogo();
            break;

        case 2:
            printf("Objetivos do jogo\nRegras:\n");
            printf("- Cada jogador tem 8 tentativas em caso de falha\n");
            printf("- Os jogadores irão revezar os chutes a cada tentativa\n");
            printf("[ 1 ] Voltar");
            scanf("%d",&gameOp2);
            system("cls");
            break;

        case 3:
            main();
            break;

        default:
            break;
    }
}while(gameOp2 != 1);
}

//->Menu Creditos
void menuCreditos(int gameOp4){
    printf("--- Desenvolvedores ---\n");
    printf("Lider do projeto e Programador:\n - Caio Justino\n");
    printf("Programadores:\n - Paulo Fernandes\n - Caio Araújo\n - Bruno Hebert\n - Marllon Brenno\n");
    printf("Obrigado pelo Suporte\n");

    printf("[ 1 ] Voltar\n");
    scanf("%d", &gameOp4);
    if(gameOp4 == 1){

    }
    system("cls");
}

```

```

//Menu Principal
int main(void){
    int gameOp;

    int gameOp2;

    int gameOp3;

    int gameOp4;

    int gameOp5;

    int gameOp6;

    do{

        //opções do jogador(MENU)
        printf("--- JOGO DA FORCA ---\n");

        printf("[ 1 ] JOGAR\n");
        printf("[ 2 ] RANKING\n");
        printf("[ 3 ] ADICIONAR PALAVRAS\n");
        printf("[ 4 ] CREDITOS\n");
        printf("[ 5 ] SAIR\n");


        //Escolha do usuário
        printf(" ");
        scanf("%d", &gameOp);
        system("cls");


        //Switch de Cenários
        switch(gameOp){

            case 1:
                menuJogo(gameOp2);

                system("cls");

                break;

            case 2:
                Rank();

                system("cls");

                break;

            case 3:

```

```

        Banco();

        system("cls");

        break;

case 4:

        menuCreditos(gameOp5);

        system("cls");

        break;

case 5:

        printf("Certeza de que deseja fechar o Jogo ?\n");
        printf("    [ 1 ]Fechar    [2]Voltar\n");
        scanf("%d", &gameOp6);
        system("cls");

        if(gameOp6 == 1){

                printf("Certo,fechando jogo...\n");

                system("pause");

                exit(1);

                break;

        }

        else if(gameOp6 == 2){

                printf("Certo,voltando ao jogo...");

                break;

        }

        system("cls");

        break;

default:

        break;

}

}while(gameOp6 != 1);

}

```