**PROJETO CONJUNTO DAS DISCIPLINAS DE TÉCNICAS E LABORATÓRIO DE DESENVOLVIMENTO DE ALGORITMOS**

**Equipe:**

Artur Freitas Plameira 25593421

Filipe Tito Calaço 26826747

José Jackson Galdino Barbosa 26423375

Pedro Augusto Lopes Gomes 26608251

Yure Almeida Diniz 26831406

**JOGO DA FORCA**

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 desenhado 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.

**REGRAS**

Para começar o jogo se desenha uma base e um risco correspondente ao lugar de cada letra.

O jogador (que está tentando adivinhar a palavra) deve escolher uma letra que ele acha que completa a palavra guardada.

Caso o jogador chute uma letra que não pertence a palavra, outra parte do corpo é desenhada.

O jogo é ganho se a palavra for adivinhada sem que o corpo do enforcado esteja completo.

**MODOS DE JOGO**

O jogador pode escolher entre jogar contra a Máquina (a palavra secreta é selecionada aleatoriamente do banco de palavras) ou jogar com outro jogador, onde um insere a palavra secreta e o outro tenta adivinhar, podendo os papéis se inverterem.

**DESCRIÇÃO GERAL**

A aplicação inicia com um menu inicial feito com switch case no qual temos as opções Jogar, Cadastrar Palavras, Ranking, Créditos e Sair, caso selecionada opção invalidade pede para digitar uma nova opção.

switch(opcao)

    {

        case '1':

            jogo();

        break;

        case '2':

            cadastra\_palavras();

        break;

        case '3':

            imprime\_ranking();

        break;

        case '4':

            creditos();

        break;

        case '5':

            system("cls");

            printf("\n\t--------------- JOGO DA FORCA ---------------\n");

            printf("\n\tSaindo do Jogo!!!\n\n");

            exit(1);

        break;

        default:

            system("cls");

            printf("\n\t--------------- JOGO DA FORCA ---------------\n");

            printf("\n\tOpcao Invalida, digite novamente!\n\n");

            system("pause");

            menu();

    }

Selecionada a opção jogar, inicia-se a função jogo que logo inicia a função modo\_jogo, pedindo ao usuário para escolher entre os modos 1-Player Vs Player ou 2-Player Vs Computador.

 modo = modo\_jogo();

    if(modo == '1')

    {

A variável modo recebe o valor do retorno da função se modo 1 ela executa o bloco de código do modo Player Vs Player, caso seja 2 executa o bloco de código do modo Player Vs Computador.

Ambos os modos quando iniciados pedem para inserir o nickname do jogador, a diferença entre eles é a forma como se obtem a palavra, no modo 1-PVP, uma pessoa informa a palavra e a dica para que outra tente adivinhar, já no modo 2-PVC a aplicação seleciona uma palavra previamente cadastrada e coloca para o jogador adivinhar.

char \*\*palavra\_dica = busca\_palavra();

        strcpy(palavra, palavra\_dica[0]);

        strcpy(dica, palavra\_dica[1]);

A variável \*\*palavra\_dica recebe o retorno da função busca\_palavra que é responsável por ir no arquivo e de modo aleatória selecionar uma palavra e junto com ela sua dica.

Depois de obter as palavras o jogo de adivinhação começa, a interface do jogo apresenta linhas que representam cada letra da palavra, um campo para apresentar as letras erradas, a imagem da forca, a dica e a quantidade de letras da palavra e o campo o usuário digita a letra que deseja.

printf("\n\t--------------- JOGO DA FORCA ---------------\n\n");

        for(i = 0; i < strlen(palavra); i++)

        {

            printf("\t%c", mascara[i]);

        }

        printf("\n");

        for(i = 0; i < strlen(palavra); i++)

        {

            if (palavra[i] == ' ')

            {

                printf("\t ");

            }

            else

            {

                printf("\t\_");

            }

        }

        printf("\n\n");

        printf("\n\tLetras Erradas: %s", strupr(erradas));

        forca(tentativas);

        printf("\n\n");

        printf("\n\tDica: %s com %d letras\n", strupr(dica), strlen(palavra));

        printf("\n\tDigite '\*' + <ENTER> para DESISTIR\n");

        printf("\n\tInforme uma letra:  ");

        letra = getchar();

        fflush(stdin);

A função forca é responsável por apresentar na tela do jogador a imagem da forca e a cada erro as partes do boneco vão sendo adicionadas. Para calcular utiliza-se a variável tentativas que começa com valor 6 e vai diminuindo 1 a cada tentativa errada.

else if(tentativas == 2)

    {

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

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

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

        printf("\n\t||               O ");

        printf("\n\t||              /|\\");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||\_\_\_");

        printf("\n\t|\_\_\_\_|");

    }

Ao fim do jogo pode-se obter 3 resultados, venceu, perdeu ou desistiu, para cada um desses resultados tem bloco de código diferente para ser executado, dependendo do que ocorreu se a variável tentativas explicada anteriormente, chegou ao valor 0, o jogador perdeu e o jogador recebe -10 pontos, se o número de letras é igual ao número de letras da palavra o jogador venceu e ele recebe o número restante de tentativas x 5 pontos + 10 pontos de bônus por ter acertado, podendo chegar no máximo a 40 pontos e no mínimo 15 pontos, caso em algum momento o jogador optou pela opção de desistir, digitando ‘\*’ no meio do jogo, o jogador recebe -10 pontos.

else if (n\_certas == (strlen(palavra) - n\_vazio))

        {

            system("cls");

            printf("\n\t--------------- JOGO DA FORCA ---------------\n");

            forca(tentativas);

            printf("\n\n\tLetras Erradas: %s", strupr(erradas));

            printf("\n\n\tPALAVRA CORRETA: %s", strupr(palavra));

            printf("\n\n\t\*\*\*\*\*\*\*\*\*\* VOCE VENCEU! \*\*\*\*\*\*\*\*\*\*\n");

            pontuacao = ((tentativas \* 5) + 10);

            printf("\n\tJogador: %s\tRecebeu: %d Pontos\n", strupr(jogador), pontuacao);

            break;

        }

Após o fim do jogo é iniciada a função altera\_ranking para onde é enviado o nickname do jogador e sua pontuação recebida ao final do jogo. A função altera\_ranking armazena em um arquivo os nicknames e pontuações dos jogadores caso seja um jogador que já teve seu nickname armazenado sua pontuação será acumulada, caso não será adicionado uma linha com o nickname e a pontuação. Após isso a função ordena o ranking de maneira decrescente e salva um novo arquivo com as novas alterações.

for(i = 0; i < n; i++)

    {

        if (strcmp(nomes[i], jogador) == 0)

        {

            pontuacoes[i] += pontos;

            alterou = 1;

            break;

        }

    }

    if(alterou == 1)

    {

        arq\_ranking = fopen("ranking.txt", "w");

        for(i = 0; i < n; i++)

        {

            fprintf(arq\_ranking,"%s %d\n",nomes[i],pontuacoes[i]);

        }

        fclose(arq\_ranking);

    }

    if(alterou == 0)

    {

        arq\_ranking = fopen("ranking.txt", "a");

        fprintf(arq\_ranking,"%s %d\n",jogador,pontos);

        fclose(arq\_ranking);

    }

Voltando ao menu inicial, e selecionando a opção cadastro de palavras, é iniciada a função cadastra\_palavras, logo aparece as instruções de como deve ser a palavra, na qual não aceita acentuação, cedilha, caractere especial e o tamanho deve ser entre 3 e 20 caracteres, ela aceita espaço permitindo assim palavras com mais de um termo. Essas regras também valem para a dica que é pedida logo em seguida, essa verificação também ocorre na palavra e dica informadas no modo de jogo PVP, a função valida\_palavra é quem realiza esse controle retornando os valores que são usadas com parâmetro para o andamento da aplicação, 0- a palavra obedece aos requisitos, 1- usuário selecionou a opção voltar ao menu e 2- a palavra não obedeceu um dos requisitos e pede para digitar novamente.

O nickname do jogador também passar por uma validação através da função valida\_nome, no qual só permite letras maiúsculas e minúsculas sem acentuação e tamanho entre 3 e 6 caracteres, e também retorna os valores 0- caso obedeça a todos os requisitos e 1 caso não obedeça a um dos requisitos.

int valida\_palavra(char palavra[30])

{

    int i, resultado = 0;

    if (strlen(palavra)-1 == 0)

    {

        resultado = 1;

    }

    else if (strlen(palavra)-1 < 3)

    {

        printf("\n\tAVISO: A PALAVRA/DICA informada,\n\ttem menos que 3 letras, digite novamente!\n\n");

        resultado = 2;

    }

    else if (strlen(palavra)-1 >= 3 && strlen(palavra)-1 <= 20)

    {

        for (i = 0; i < strlen(palavra)-1; i++)

        {

            if (!((palavra[i] >= 'a' && palavra[i] <='z') || (palavra[i] >= 'A' && palavra[i] <= 'Z') || (palavra[i] == ' ')))

            {

                printf("\n\tAVISO: A PALAVRA/DICA informada,\n\tpossui caracteres nao permitidos, digite novamente!\n\n");

                resultado = 2;

                break;

            }

            if (palavra[i] == ' ' && palavra[i - 1] == ' ')

            {

                printf("\n\tAVISO: A PALAVRA/DICA informada,\n\tpossui espacos sequencias, digite novamente!\n\n");

                resultado = 2;

                break;

            }

        }

    }

    else if (strlen(palavra)-1 > 20)

    {

        printf("\n\tAVISO: A PALAVRA/DICA informda,\n\tultrapassou o tamanho maximo de 20 letras, digite novamente!\n\n");

        resultado = 2;

    }

    return resultado;

}

No menu ainda temos a opção ranking, que chama a função imprime\_ranking que imprime na tela do usuário o ranking ordenado de acordo com a pontuação, da maior para menor.

FILE    \*arq\_ranking;

    int     n = 0, pontuacoes[50];

    char    nomes[50][10];

    system("cls");

    printf("\n\t--------------- JOGO DA FORCA ---------------\n");

    arq\_ranking = fopen("ranking.txt", "r");

    printf("\n\t\tPOS\tJOGADOR\t\tPONTOS\n\n");

    while(fscanf(arq\_ranking,"%s %d", nomes[n], &pontuacoes[n]) != EOF)

    {

        printf("\t\t%do.\t%s\t\t%d\n", n + 1, nomes[n], pontuacoes[n]);

        n++;

    }

    fclose(arq\_ranking);

Ainda temos no menu a opção créditos, imprime na tela o nomes dos participantes do projeto.

void creditos()

{

    system("cls");

    printf("\n\t--------------- JOGO DA FORCA ---------------\n");

    printf("\n\tCriado por...:\n");

    printf("\n\t\tARTHUR FREITAS PALMEIRA");

    printf("\n\t\tFILIPE TITO CALACO");

    printf("\n\t\tJOSE JACKSON GALDINO BARBOSA");

    printf("\n\t\tPEDRO AUGUSTO LOPES GOMES");

    printf("\n\t\tYURE ALMEIDA DINIZ\n\n");

    system("pause");

    menu();

}

**DIFICULDADES**

1- Retornar da Função busca\_palavra um vetor de strings(matriz de char) com a palavra e a dica

Para resolver esse problema foi utilizado a criação de uma variável char \*\*selecionadas, que é um ponteiro de ponteiro com alocação de memória no tamanho de uma matriz 2 por 30 atravez da função malloc.

seleciondas = (char\*\*)malloc(2 \* sizeof(char));

    for(i = 0; i < 2; i++)

    {

        seleciondas[i] = (char\*)malloc(30 \* sizeof(char));

    }

    strcpy(seleciondas[0], palavras[r]);

    r++;

    strcpy(seleciondas[1], palavras[r]);

    return seleciondas;

Para receber os valores do retorno da função busca\_palavra na função jogo e utilizar eles, teve que ser declarado a variável ponteiro de ponteiro, char \*\*palavra\_dica e copiado os valores do retorno par ela.

char \*\*palavra\_dica = busca\_palavra();

        strcpy(palavra, palavra\_dica[0]);

        strcpy(dica, palavra\_dica[1]);

2- Selecionar palavra aleatória

Como palavra e dica estão em um mesmo arquivo, palavra numa linha e dica na linha abaixo, teve que ser feiro um loop na função que gera números aleatórios onde o loop só seria encerrado caso o número sorteado fosse um número par, pois as palavras estão localizadas nas linhas pares e para pegar a sua dica é só seleciona o número da linha da palavra + 1.

do

    {

        srand((unsigned)time(NULL));

        r = rand() % n;

    } while (r %2 != 0);

3- Alteração e Ordenação do ranking

A função altera\_ranking, quando iniciada com o nickname e pontuação obtida pelo jogador, ela verifica se aquele nickname já existe se sim é alterado apenas a pontuação e mantém o nickname, para fazer isso antes de tudo foi criado dois vetores um de string com os nomes e um de int com a pontuação, através do modo de abertura de arquivo “r”, um loop while com paramento fscanf != EOF, ele percorre todo o arquivo e armazenando os dasdos nos dois vetores. Um for percorre os vetores e através da função strcmp com retorno 0, é realizada a comparação do nickname do jogador que terminou a partida com os que já estão no arquivo. Se encontrar um nickname igual é acumulado a pontuação obtida com a já existente na mesma posição de vetor do nickname. Para que essa alteração seja salva no arquivo é utilizada a variável alterou como verificadora para saber quando o arquivo deve ser reescrito ou não, caso um nickname igual seja encontrado alterou recebe o valor 1, com isso um for percorre os vetores agora com o valor da pontuação altera e rescreve o arquivo através do modo de abertura arquivo “w”, caso um nickname igual não seja encontrado o arquivo é aberto no modo “a” e adiciono o novo nickname e sua pontuação ao final do arquivo.

Depois de inserir um novo jogador no ranking ou alterado a pontuação de um já existente vem a ordenação de forma decrescente em relação a pontuação dos jogadores. Para isso foi utilizado a técnica de Bubble Sort que funciona assim a posição atual é comparada com a próxima posição e, se a posição atual for menor que a posição posterior, é realizada a troca dos valores nessa posição, jogando o maior valor pra cima.

Bubble sort da aplicação:

for (x = 0; x < n ; x++)

    {

        for(y = x; y < n ; y++)

        {

            if(pontuacoes[y] > pontuacoes[x])

            {

                aux = pontuacoes[x];

                strcpy(nAux, nomes[x]);

                pontuacoes[x] = pontuacoes[y];

                strcpy(nomes[x], nomes[y]);

                pontuacoes[y] = aux;

                strcpy(nomes[y], nAux);

            }

        }

    }

Demonstração do Ranking

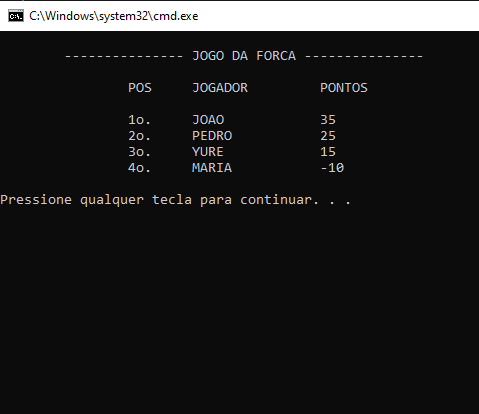


Fig1. Ranking anterior.



Fig. 2 Pedro obteve 35 pontos.

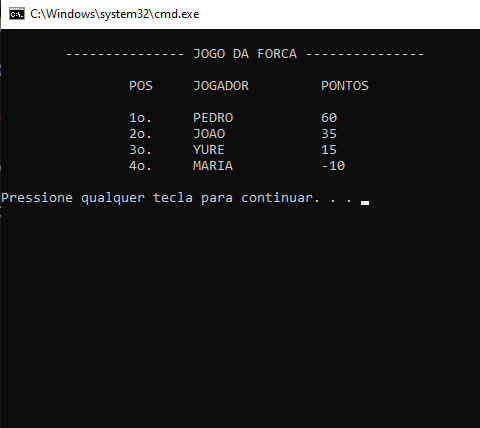


Fig.3 Novo rankeamento ordenado com nova pontuação de Pedro.



Fig. 4 Jeanne obteve 35 pontos.

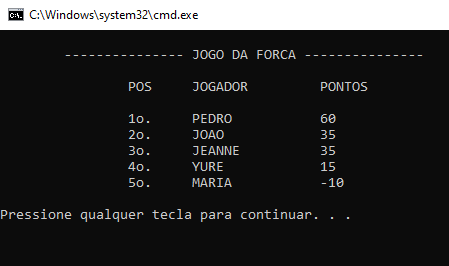


Fig.5 Novo rankeamento agora como adição de Jeanne.

**APÊNDICE**

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <ctype.h>

#include <time.h>

//Declaracao das funcoes

void    menu();

char    modo\_jogo();

void    forca(int);

void    jogo();

void    cadastra\_palavras();

int     valida\_palavra(char[]);

int     valida\_nome(char[]);

char    \*\*busca\_palavra();

void    altera\_ranking(char[], int);

void    imprime\_ranking();

void    creditos();

// Mostra o menu principal para o jogador

void menu()

{

    int opcao;

    system("cls");

    printf("\n\t--------------- JOGO DA FORCA ---------------\n");

    printf("\n\tMENU PRINCIPAL\n");

    printf("\n\t 1 - Jogar");

    printf("\n\t 2 - Cadastro de Palavras");

    printf("\n\t 3 - Ranking");

    printf("\n\t 4 - Creditos");

    printf("\n\t 5 - Sair");

    printf("\n\n\tEscolha uma das opcoes...: ");

    opcao = getchar();

    fflush(stdin);

    switch(opcao)

    {

        case '1':

            jogo();

        break;

        case '2':

            cadastra\_palavras();

        break;

        case '3':

            imprime\_ranking();

        break;

        case '4':

            creditos();

        break;

        case '5':

            system("cls");

            printf("\n\t--------------- JOGO DA FORCA ---------------\n");

            printf("\n\tSaindo do Jogo!!!\n\n");

            exit(1);

        break;

        default:

            system("cls");

            printf("\n\t--------------- JOGO DA FORCA ---------------\n");

            printf("\n\tOpcao Invalida, digite novamente!\n\n");

            system("pause");

            menu();

    }

}

// Seleciona o modo de jogo PVC ou PVP

char modo\_jogo()

{

    int opcao;

    system("cls");

    printf("\n\t--------------- JOGO DA FORCA ---------------\n");

    printf("\n\tMODOS DE JOGO\n");

    printf("\n\t 1 - Modo Player Vs Player");

    printf("\n\t 2 - Modo Player Vs Computador");

    printf("\n\n\tEscolha uma das opcoes ou\n\t<ENTER> para SAIR......: ");

    opcao = getchar();

    fflush(stdin);

    if(opcao == '\n')

    {

        menu();

    }

    else if (opcao != '1' && opcao != '2')

    {

        printf("\n\tOpcao invalida, digite novamente!\n\n");

        system("pause");

        jogo();

    }

    return opcao;

}

// Imprime a forca na tela

void forca(int tentativas)

{

    if(tentativas == 6)

    {

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

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

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

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||\_\_\_");

        printf("\n\t|\_\_\_\_|");

    }

    else if(tentativas == 5)

    {

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

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

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

        printf("\n\t||               O ");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||\_\_\_");

        printf("\n\t|\_\_\_\_|");

    }

    else if(tentativas == 4)

    {

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

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

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

        printf("\n\t||               O ");

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

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||\_\_\_");

        printf("\n\t|\_\_\_\_|");

    }

    else if(tentativas == 3)

    {

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

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

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

        printf("\n\t||               O ");

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

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||\_\_\_");

        printf("\n\t|\_\_\_\_|");

    }

    else if(tentativas == 2)

    {

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

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

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

        printf("\n\t||               O ");

        printf("\n\t||              /|\\");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||\_\_\_");

        printf("\n\t|\_\_\_\_|");

    }

    else if(tentativas == 1)

    {

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

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

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

        printf("\n\t||               O ");

        printf("\n\t||              /|\\");

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

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t||\_\_\_");

        printf("\n\t|\_\_\_\_|");

    }

    else if(tentativas == 0)

    {

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

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

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

        printf("\n\t||               0 ");

        printf("\n\t||              /|\\");

        printf("\n\t||              / \\");

        printf("\n\t||");

        printf("\n\t||");

        printf("\n\t|| : Mais sorte da proxima vez...");

        printf("\n\t||\_\_\_");

        printf("\n\t|\_\_\_\_|");

    }

}

//Jogo no modo PVP

void jogo()

{

    char    palavra[30], dica[30], mascara[30], erradas[20], modo, jogador[10];

    int     i, acertou, letra, e, tentativas = 6, n\_certas = 0, n\_vazio = 0, erro = 0, pontuacao = 0;

    modo = modo\_jogo();

    if(modo == '1')

    {

        do

        {

            system("cls");

            printf("\n\t--------------- JOGO DA FORCA ---------------\n");

            printf("\n\tJOGO PVP\n");

            printf("\n\tInstrucoes:\n\tO Nickname NAO podem conter:\n\tACENTO, CEDILHA, CARACTERE ESPECIAL ou NUMERO,\n\tno MINIMO 3 letras e no MAXIMO 6 letras.\n");

            printf("\n\tInforme seu Nickname ou\n\t<ENTER> para SAIR...:  ");

            fgets(jogador, 10, stdin);

            fflush(stdin);

            erro = valida\_nome(jogador);

            jogador[strlen(jogador)-1] = '\0';

        } while (erro != 0);

        while(1)

        {

            while(1)

            {

                system("cls");

                printf("\n\t--------------- JOGO DA FORCA ---------------\n");

                printf("\n\tJOGO PVP\n");

                printf("\n\tInstrucoes:\n\tA PALAVRA e a DICA NAO podem conter:\n\tACENTO, CEDILHA, CARACTERE ESPECIAL ou NUMERO,\n\tno MINIMO 3 letras e no MAXIMO 20 letras.");

                printf("\n\n\tInforme a PALAVRA que deseja JOGAR\n\tou <ENTER> para SAIR...:  ");

                fgets(palavra, 30, stdin);

                fflush(stdin);

                erro = 0;

                erro = valida\_palavra(palavra);

                if (erro == 1)

                {

                    menu();

                    break;

                }

                else if (erro == 2)

                {

                    system("pause");

                }

                else

                {

                    break;

                }

            }

            printf("\n\tInforme a DICA da palavra\n\tou <ENTER> para SAIR...:  ");

            fgets(dica, 30, stdin);

            fflush(stdin);

            erro = 0;

            erro = valida\_palavra(dica);

            if (erro == 1)

            {

                menu();

                break;

            }

            else if (erro == 2)

            {

                system("pause");

            }

            else

            {

                break;

            }

        }

        palavra[strlen(palavra) - 1] = '\0';

        dica[strlen(dica) - 1] = '\0';

    }

    else if (modo == '2')

    {

        do

        {

            system("cls");

            printf("\n\t--------------- JOGO DA FORCA ---------------\n");

            printf("\n\tJOGO PVC\n");

            printf("\n\tInstrucoes:\n\tO Nickname NAO podem conter:\n\tACENTO, CEDILHA, CARACTERE ESPECIAL ou NUMERO,\n\tno MINIMO 3 letras e no MAXIMO 6 letras.\n");

            printf("\n\tInforme seu Nickname ou\n\t<ENTER> para SAIR...:  ");

            fgets(jogador, 10, stdin);

            fflush(stdin);

            erro = valida\_nome(jogador);

            jogador[strlen(jogador)-1] = '\0';

        } while (erro != 0);

        char \*\*palavra\_dica = busca\_palavra();

        strcpy(palavra, palavra\_dica[0]);

        strcpy(dica, palavra\_dica[1]);

    }

    for(i = 0; i < strlen(palavra); i++)

    {

        mascara[i] = ' ';

        if (palavra[i] == ' ')

        {

            n\_vazio++;

        }

    }

    while (1) {

        system("cls");

        printf("\n\t--------------- JOGO DA FORCA ---------------\n\n");

        for(i = 0; i < strlen(palavra); i++)

        {

            printf("\t%c", mascara[i]);

        }

        printf("\n");

        for(i = 0; i < strlen(palavra); i++)

        {

            if (palavra[i] == ' ')

            {

                printf("\t ");

            }

            else

            {

                printf("\t\_");

            }

        }

        printf("\n\n");

        printf("\n\tLetras Erradas: %s", strupr(erradas));

        forca(tentativas);

        printf("\n\n");

        printf("\n\tDica: %s com %d letras\n", strupr(dica), strlen(palavra));

        printf("\n\tDigite '\*' + <ENTER> para DESISTIR\n");

        printf("\n\tInforme uma letra:  ");

        letra = getchar();

        fflush(stdin);

        acertou = 0;

        if ((letra >= 'a' && letra <='z') || (letra >= 'A' && letra <= 'Z') || letra == ' ' || letra == '\n' || letra == '\*')

        {

            for (i = 0; i < strlen(palavra); i++)

            {

                if (mascara[i] == toupper(letra) || letra == '\n' || letra == '\*')

                {

                    acertou = 1;

                }

                else if (toupper(palavra[i]) == toupper(letra))

                {

                    mascara[i] = toupper(letra);

                    n\_certas++;

                    acertou = 1;

                }

            }

        }

        else

        {

            acertou = 1;

        }

        e = 0;

        for (i = 0; i < strlen(erradas); i++)

        {

            if(erradas[i] == toupper(letra))

            {

                e = 1;

            }

        }

        if (acertou == 0 && e == 0)

        {

                erradas[strlen(erradas)] = toupper(letra);

                erradas[strlen(erradas)] = ' ';

                erradas[strlen(erradas)] = '\0';

                tentativas--;

        }

        if (tentativas == 0)

        {

            system("cls");

            printf("\n\t--------------- JOGO DA FORCA ---------------\n");

            forca(tentativas);

            printf("\n\n\tLetras Erradas: %s", strupr(erradas));

            printf("\n\n\tPALAVRA CORRETA: %s", strupr(palavra));

            printf("\n\n\t---------- VOCE PERDEU! ----------\n");

            pontuacao = -10;

            printf("\n\tJogador: %s\tRecebeu: %d Pontos\n", strupr(jogador), pontuacao);

            break;

        }

        else if (n\_certas == (strlen(palavra) - n\_vazio))

        {

            system("cls");

            printf("\n\t--------------- JOGO DA FORCA ---------------\n");

            forca(tentativas);

            printf("\n\n\tLetras Erradas: %s", strupr(erradas));

            printf("\n\n\tPALAVRA CORRETA: %s", strupr(palavra));

            printf("\n\n\t\*\*\*\*\*\*\*\*\*\* VOCE VENCEU! \*\*\*\*\*\*\*\*\*\*\n");

            pontuacao = ((tentativas \* 5) + 10);

            printf("\n\tJogador: %s\tRecebeu: %d Pontos\n", strupr(jogador), pontuacao);

            break;

        }

        else if (letra == '\*')

        {

            system("cls");

            printf("\n\t--------------- JOGO DA FORCA ---------------\n");

            forca(tentativas);

            printf("\n\n\tLetras Erradas: %s", strupr(erradas));

            printf("\n\n\tPALAVRA CORRETA: %s", strupr(palavra));

            printf("\n\n\t\_\_\_\_\_\_\_\_\_\_ VOCE DESISTIU! \_\_\_\_\_\_\_\_\_\_\n");

            pontuacao = -10;

            printf("\n\tJogador: %s\tRecebeu: %d Pontos\n", strupr(jogador), pontuacao);

            break;

        }

    }

    altera\_ranking(jogador, pontuacao);

    while (1)

    {

        printf("\n\tDeseja jogar novamente(S/N)...:  ");

        letra = getchar();

        fflush(stdin);

        if (letra == 's' || letra == 'S')

        {

            jogo();

            break;

        }

        else if (letra == 'n' || letra == 'N')

        {

            menu();

            break;

        }

        else

        {

            printf("\n\tOpcao Invalida!\n\tDigite S para Sim ou N para Nao.\n");

        }

    }

}

//Cadastra as Palavras e as Dicas no arquivo palavras.txt

void cadastra\_palavras()

{

    FILE    \*arq\_palavras;

    char    palavra[30], dica[30];

    int     erro = 0;

    arq\_palavras = fopen("palavras.txt","a");

    if(arq\_palavras == NULL)

    {

        printf("\n\tErro na abertura do arquivo, palavras.txt\n\n");

        fclose(arq\_palavras);

        system("pause");

        menu();

    }

    system("cls");

    printf("\n\t--------------- JOGO DA FORCA ---------------\n");

    printf("\n\tCADASTRO DE PALAVRAS\n");

    printf("\n\tInstrucoes:\n\tA PALAVRA e a DICA NAO podem conter:\n\tACENTO, CEDILHA, CARACTERE ESPECIAL ou NUMERO,\n\tno MINIMO 3 letras e no MAXIMO 20 letras.");

    printf("\n\n\tInforme a PALAVRA que deseja cadastrar\n\tou <ENTER> para SAIR...:  ");

    fflush(stdin);

    fgets(palavra, 30, stdin);

    erro = valida\_palavra(palavra);

    if (erro == 1)

    {

        fclose(arq\_palavras);

        menu();

    }

    else if (erro == 2)

    {

        fclose(arq\_palavras);

        system("pause");

        cadastra\_palavras();

    }

    printf("\n\tInforme a DICA da palavra\n\tou <ENTER> para SAIR...:  ");

    fflush(stdin);

    fgets(dica, 30, stdin);

    erro = 0;

    erro = valida\_palavra(dica);

    if (erro == 1)

    {

        fclose(arq\_palavras);

        system("pause");

        menu();

    }

    else if (erro == 2)

    {

        fclose(arq\_palavras);

        system("pause");

        cadastra\_palavras();

    }

    fputs(palavra, arq\_palavras);

    if(ferror(arq\_palavras))

    {

        perror("\n\tErro no cadastro da palavra\n\n");

        fclose(arq\_palavras);

        system("pause");

        menu();

    }

    fputs(dica, arq\_palavras);

    if(ferror(arq\_palavras))

    {

        perror("\n\tErro no cadastro da dica\n\n");

        fclose(arq\_palavras);

        system("pause");

        menu();

    }

    fclose(arq\_palavras);

    cadastra\_palavras();

}

// Valida palavra a ser cadastrada ou no Jogo PVP

int valida\_palavra(char palavra[30])

{

    int i, resultado = 0;

    if (strlen(palavra)-1 == 0)

    {

        resultado = 1;

    }

    else if (strlen(palavra)-1 < 3)

    {

        printf("\n\tAVISO: A PALAVRA/DICA informada,\n\ttem menos que 3 letras, digite novamente!\n\n");

        resultado = 2;

    }

    else if (strlen(palavra)-1 >= 3 && strlen(palavra)-1 <= 20)

    {

        for (i = 0; i < strlen(palavra)-1; i++)

        {

            if (!((palavra[i] >= 'a' && palavra[i] <='z') || (palavra[i] >= 'A' && palavra[i] <= 'Z') || (palavra[i] == ' ')))

            {

                printf("\n\tAVISO: A PALAVRA/DICA informada,\n\tpossui caracteres nao permitidos, digite novamente!\n\n");

                resultado = 2;

                break;

            }

            if (palavra[i] == ' ' && palavra[i - 1] == ' ')

            {

                printf("\n\tAVISO: A PALAVRA/DICA informada,\n\tpossui espacos sequencias, digite novamente!\n\n");

                resultado = 2;

                break;

            }

        }

    }

    else if (strlen(palavra)-1 > 20)

    {

        printf("\n\tAVISO: A PALAVRA/DICA informda,\n\tultrapassou o tamanho maximo de 20 letras, digite novamente!\n\n");

        resultado = 2;

    }

    return resultado;

}

//Valida o Nickname do jogador

int valida\_nome(char nome[10])

{

    int i, resultado = 0;

    if (strlen(nome)-1 == 0)

    {

        menu();

    }

    else if (strlen(nome)-1 < 3)

    {

        printf("\n\tAVISO: O NICKNAME informado,\n\ttem menos que 3 letras, digite novamente!\n\n");

        system("pause");

        resultado = 1;

    }

    else if (strlen(nome)-1 >= 3 && strlen(nome)-1 <= 6)

    {

        for (i = 0; i < strlen(nome)-1; i++)

        {

            if (!((nome[i] >= 'a' && nome[i] <='z') || (nome[i] >= 'A' && nome[i] <= 'Z')))

            {

                printf("\n\tAVISO: O NICKNAME informado,\n\tpossui caracteres nao permitidos, digite novamente!\n\n");

                system("pause");

                resultado = 1;

                break;

            }

        }

    }

    else if (strlen(nome)-1 > 6)

    {

        printf("\n\tAVISO: O NICKNAME informado,\n\tultrapassou o tamanho maximo de 6 letras,\n\tdigite novamente!\n\n");

        system("pause");

        resultado = 1;

    }

    return resultado;

}

// Pega palavras do arquivo e carrega para o jogo PVC

char \*\*busca\_palavra()

{

    int n = 0, i, r = 0;

    char palavras[400][30], linhas[30];

    char \*\*seleciondas;

    FILE \*arq\_palavras;

    arq\_palavras = fopen("palavras.txt", "r");

    if (arq\_palavras == NULL)

    {

        printf("\n\tErro na leitura do arquivo, palavras.txt\n\n");

        system("pause");

        menu();

    }

    while(fgets(linhas, 30, arq\_palavras)!= NULL)

    {

        linhas[strlen(linhas) - 1] = '\0';

        strcpy(palavras[n], linhas);

        n++;

    }

    fclose(arq\_palavras);

    do

    {

        srand((unsigned)time(NULL));

        r = rand() % n;

    } while (r %2 != 0);

    seleciondas = (char\*\*)malloc(2 \* sizeof(char));

    for(i = 0; i < 2; i++)

    {

        seleciondas[i] = (char\*)malloc(30 \* sizeof(char));

    }

    strcpy(seleciondas[0], palavras[r]);

    r++;

    strcpy(seleciondas[1], palavras[r]);

    return seleciondas;

}

// Criar e ordena o ranking

void altera\_ranking(char jogador[10], int pontos)

{

    FILE \*arq\_ranking;

    char nomes[50][10], nAux[10];

    int pontuacoes[50], n = 0, alterou = 0, i, x = 0, y = 0, aux = 0;

    arq\_ranking = fopen("ranking.txt", "r");

    while(fscanf(arq\_ranking,"%s %d", nomes[n], &pontuacoes[n]) != EOF)

    {

        n++;

    }

    fclose(arq\_ranking);

    for(i = 0; i < n; i++)

    {

        if (strcmp(nomes[i], jogador) == 0)

        {

            pontuacoes[i] += pontos;

            alterou = 1;

            break;

        }

    }

    if(alterou == 1)

    {

        arq\_ranking = fopen("ranking.txt", "w");

        for(i = 0; i < n; i++)

        {

            fprintf(arq\_ranking,"%s %d\n",nomes[i],pontuacoes[i]);

        }

        fclose(arq\_ranking);

    }

    if(alterou == 0)

    {

        arq\_ranking = fopen("ranking.txt", "a");

        fprintf(arq\_ranking,"%s %d\n",jogador,pontos);

        fclose(arq\_ranking);

    }

    //Ordenando o Ranking

    arq\_ranking = fopen("ranking.txt", "r");

    n = 0;

    while(fscanf(arq\_ranking,"%s %d", nomes[n], &pontuacoes[n]) != EOF)

    {

        n++;

    }

    fclose(arq\_ranking);

    for (x = 0; x < n ; x++)

    {

        for(y = x; y < n ; y++)

        {

            if(pontuacoes[y] > pontuacoes[x])

            {

                aux = pontuacoes[x];

                strcpy(nAux, nomes[x]);

                pontuacoes[x] = pontuacoes[y];

                strcpy(nomes[x], nomes[y]);

                pontuacoes[y] = aux;

                strcpy(nomes[y], nAux);

            }

        }

    }

    arq\_ranking = fopen("ranking.txt", "w");

    for(i = 0; i < n; i++)

    {

        fprintf(arq\_ranking,"%s %d\n",nomes[i],pontuacoes[i]);

    }

    fclose(arq\_ranking);

}

void imprime\_ranking()

{

    FILE    \*arq\_ranking;

    int     n = 0, pontuacoes[50];

    char    nomes[50][10];

    system("cls");

    printf("\n\t--------------- JOGO DA FORCA ---------------\n");

    arq\_ranking = fopen("ranking.txt", "r");

    printf("\n\t\tPOS\tJOGADOR\t\tPONTOS\n\n");

    while(fscanf(arq\_ranking,"%s %d", nomes[n], &pontuacoes[n]) != EOF)

    {

        printf("\t\t%do.\t%s\t\t%d\n", n + 1, nomes[n], pontuacoes[n]);

        n++;

    }

    fclose(arq\_ranking);

}

// Creditos

void creditos()

{

    system("cls");

    printf("\n\t--------------- JOGO DA FORCA ---------------\n");

    printf("\n\tCriado por...:\n");

    printf("\n\t\tARTHUR FREITAS PALMEIRA");

    printf("\n\t\tFILIPE TITO CALACO");

    printf("\n\t\tJOSE JACKSON GALDINO BARBOSA");

    printf("\n\t\tPEDRO AUGUSTO LOPES GOMES");

    printf("\n\t\tYURE ALMEIDA DINIZ\n\n");

    system("pause");

    menu();

}