Relatório Projeto – Jogo da Forca

Integrantes do grupo:

- Kauã Ravy, RGM: 34127976;

- Heitor de Oliveira Mamede, RGM: 32825501;

- Júlio César Carvalho Santos, RGM: 32904622;

- Perilo Oliveira Viana Sobrinho, RGM: 33467676;

Introdução:

Nosso grupo decidiu em conjunto fazer um Jogo da Forca através da linguagem C, onde cada um desempenho um papel importante no desenvolvimento do planejamento, algoritmo e por fim, o Jogo em si.

O Jogo da Forca é um jogo linguístico que envolve adivinhação de palavras e domínio de vocabulário. Ele consiste em: O jogador deve adivinhar a palavra sorteada, sendo esta baseada na categoria escolhida pelo próprio jogador, durante o jogo ele deve adivinhar e sugerir letras que possam formar a palavra sorteada, com intuito de salvar o homem de ser enforcado. A cada suposição errada do jogador, uma parte do corpo do homem é preenchida, e se todas as partes forem preenchidas, o jogador perde pois o homem será enforcado.

As regras desse jogo lúdico, são simples:

- A palavra será sorteada baseado na categoria selecionada pelo jogador, podendo ser animais, frutas, objetos e verbos da gramática brasileira. Essa palavra será sorteada e criará espaços a serem preenchidos pelo jogador. E o objetivo do jogador é adivinhá-las até acertar a palavra completamente. O jogo da forca é uma maneira divertida e desafiadora de testar o vocabulário e as habilidades dedutivas das pessoas, sendo muito utilizado como método educativo ou de entretenimento em espaços escolares. Nada melhor do que misturar aprendizado com brincadeiras que despertam as habilidades dedutivas e conhecimentos de vocábulos.

- Se o jogador acertar uma letra, ela será revelada na posição correta da palavra sorteada. Se errar a letra, o jogador perderá uma vida e uma parte do homem será posicionada na forca.

- O jogo apenas se encerra se o jogador acertar ou errar a palavra sorteada.

- No jogo, não haverá palavras com acentuações presentes na língua brasileira.

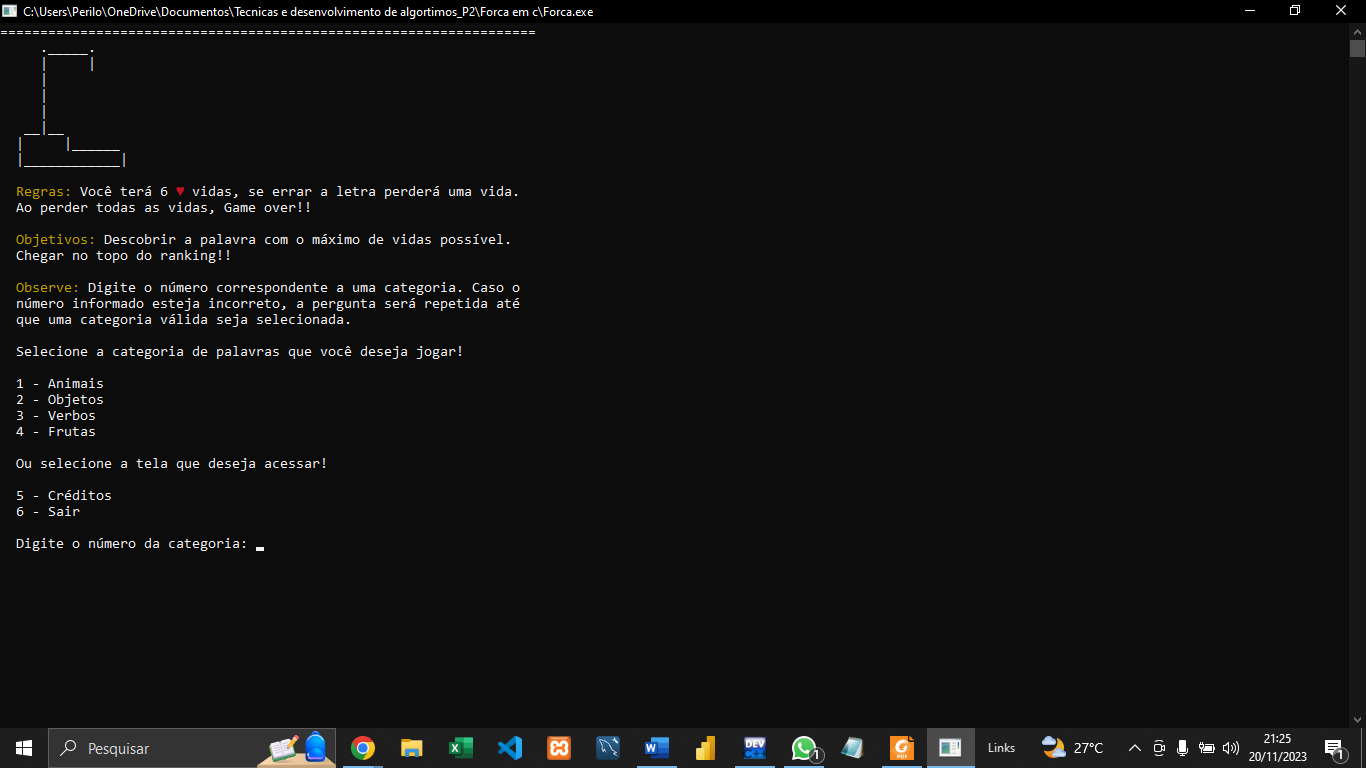
- O jogador deve digitar a letra quando for pedida, e caso ele digite algo diferente de uma letra, o sistema impedirá e pedirá novamente uma letra. Por isso, é importante a compreensão do jogador para que o jogo seja divertido e lúdico.

- O usuário terá 6 chances de adivinhar a palavra.

- Utilize se possível apenas letras minúsculas e uma de cada vez.

Resultados:

Imagem 1: Inicialmente, foram feitas as categorias para que o usuário selecione quais categorias de palavras, ele deseja adivinhar, sendo elas: Animais, Objetos, Verbos e Frutas.



Legendas: Aqui está a interface do nosso projeto da forca, no qual o user pode selecionar a categoria para jogar, ver os créditos ou sair do jogo.

Imagem 2:

Tela de computador

Descrição gerada automaticamente

Legenda: Acertou uma letra.

Imagem 3: Caso você perca o jogo

Tela de computador

Descrição gerada automaticamente

Legendas: Informará qual a palavra, um contador informará seu número de derrotas bem como perguntará se você deseja jogar novamente ou se deseja sair do jogo.

Imagem 4: Créditos acerca dos desenvolvedores!

Tela de computador com fundo preto

Descrição gerada automaticamente

Legenda: Tela de créditos, com o nome dos desenvolvedores do jogo.

Repositório com projeto + vídeo:

<https://github.com/Perilocc/Projeto---Jogo-na-Linguagem-C> /disponível no GitHub.

Apêndice com Código Fonte:

#include <stdio.h>

#include <locale.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

#include <ctype.h>

#include <iostream>

#include <windows.h>

void limpar\_console() {

#ifdef \_WIN32

system("cls");

#else

system("clear");

#endif

}

void desenha\_boneco(int vida) {

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

int coracao = 3;

switch (vida) {

case 0:

//Morto

SetConsoleTextAttribute(hConsole, 15);

printf("===================================================================\n");

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

printf(" | |\n");

printf(" | ");

SetConsoleTextAttribute(hConsole, 4);

printf("O\n");

SetConsoleTextAttribute(hConsole, 15);

printf(" | ");

SetConsoleTextAttribute(hConsole, 4);

printf ("/");

printf("|");

printf("\\ \n");

SetConsoleTextAttribute(hConsole, 15);

printf(" | ");

SetConsoleTextAttribute(hConsole, 4);

printf("/");

printf(" \\ \n");

SetConsoleTextAttribute(hConsole, 15);

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

printf(" %c %c %c %c %c %c\n", coracao, coracao, coracao, coracao, coracao, coracao);

break;

case 1:

//Perna Esquerda

SetConsoleTextAttribute(hConsole, 15);

printf("===================================================================\n");

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

printf(" | |\n");

printf(" | ");

printf("O\n");

printf(" | ");

printf ("/");

printf("|");

printf("\\ \n");

printf(" | ");

printf("/\n");

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

SetConsoleTextAttribute(hConsole, 4);

printf(" %c ", coracao, coracao);

SetConsoleTextAttribute(hConsole, 15);

printf("%c %c %c %c %c\n", coracao, coracao, coracao, coracao, coracao);

break;

case 2:

//Braco Direito

SetConsoleTextAttribute(hConsole, 15);

printf("===================================================================\n");

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

printf(" | |\n");

printf(" | ");

printf("O\n");

printf(" | ");

printf ("/");

printf("|");

printf("\\ \n");

printf(" | \n");

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

SetConsoleTextAttribute(hConsole, 4);

printf(" %c %c ", coracao, coracao);

SetConsoleTextAttribute(hConsole, 15);

printf("%c %c %c %c\n", coracao, coracao, coracao, coracao);

break;

case 3:

//Braco Esquerdo

SetConsoleTextAttribute(hConsole, 15);

printf("===================================================================\n");

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

printf(" | |\n");

printf(" | ");

printf("O\n");

printf(" | ");

printf ("/");

printf("|\n");

printf(" | \n");

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

SetConsoleTextAttribute(hConsole, 4);

printf(" %c %c %c ", coracao, coracao, coracao);

SetConsoleTextAttribute(hConsole, 15);

printf("%c %c %c\n", coracao, coracao, coracao);

break;

case 4:

//Tronco

SetConsoleTextAttribute(hConsole, 15);

printf("===================================================================\n");

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

printf(" | |\n");

printf(" | ");

printf("O\n");

printf(" | ");

printf(" |\n");;

printf(" | \n");

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

SetConsoleTextAttribute(hConsole, 4);

printf(" %c %c %c %c ", coracao, coracao, coracao, coracao);

SetConsoleTextAttribute(hConsole, 15);

printf("%c %c\n", coracao, coracao);

break;

case 5:

//Cabeça

SetConsoleTextAttribute(hConsole, 15);

printf("===================================================================\n");

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

printf(" | |\n");

printf(" | ");

printf("O\n");

printf(" | \n");

printf(" | \n");

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

SetConsoleTextAttribute(hConsole, 4);

printf(" %c %c %c %c %c ", coracao, coracao, coracao, coracao, coracao);

SetConsoleTextAttribute(hConsole, 15);

printf("%c\n", coracao, coracao);

break;

case 6:

//Vivo

SetConsoleTextAttribute(hConsole, 15);

printf("===================================================================\n");

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

printf(" | |\n");

printf(" | \n");

printf(" | \n");

printf(" | \n");

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

SetConsoleTextAttribute(hConsole, 4);

printf(" %c %c %c %c %c %c\n", coracao, coracao, coracao, coracao, coracao, coracao);

SetConsoleTextAttribute(hConsole, 15);

break;

case 7:

//Menu

SetConsoleTextAttribute(hConsole, 15);

printf("===================================================================\n");

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

printf(" | |\n");

printf(" | \n");

printf(" | \n");

printf(" | \n");

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

}

}

void tela\_saida() {

printf("=============================================\n");

printf(" Obrigado por Jogar!\n");

printf("=============================================\n");

}

int main(){

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, 15);

setlocale(LC\_ALL, "Portuguese");

// Listas de palavras da forca!! e Lista de variáveis presentes no programa!!

char animais[][20] = {"tartaruga", "zebra", "coala", "camelo", "macaco", "cachorro", "cobra", "gato", "galo", "galinha", "cavalo", "vaca", "ganso", "pato", "asno", "avestruz", "rato", "capivara", "carpa", "marmota"};

char objetos[][20] = {"anzol", "balde", "bola", "borracha", "faca", "canivete", "enxada", "flauta", "funil", "dado", "lanterna", "luva", "martelo", "mouse", "prato", "garfo", "colher", "copo", "tijolo", "mala"};

char verbos[][20] = {"comer", "beber", "chutar", "bater", "lutar", "chicotear", "passear", "dirigir", "compartilhar", "pesquisar", "obedecer", "ouvir", "elevar", "quebrar", "praticar", "errar", "trair", "invadir", "planejar", "falsificar"};

char frutas[][20] = {"laranja", "banana", "oliveira", "jaca", "jabuticaba", "abacaxi", "amora", "goiaba", "abacate", "acerola", "graviola", "cacau", "caqui", "carambola", "figo", "kiwi", "jambo", "manga", "morango", "pitaya"};

// Lista de contadores

int letras\_certas, chances, contagem\_erros, i, underline;

//Outras variáveis importantes

int indice, tamanho;

char categoria;

int acertou = 1;

int coracao = 3;

char letra, \*dica;

char letras\_usadas[26] = "";

char x, y, z, a, b, c, d, e, f, g;

// Variáveis envolvidas no do{} while()

int vitorias, derrotas;

char resposta;

vitorias = 0;

derrotas = 0;

do{

memset(letras\_usadas, 0, sizeof(letras\_usadas)); // reseta a lista de letras usadas ao começar outra tentativa!!

resposta = '\0';

categoria = '\0';

srand(time(NULL));/\* Esse comando serve para resetar o rand. Pois se não houver ele, por algum motivo o rand pega a mesma palavra sempre!

Provavelmente ele salva na memória algum endereço!!\*/

chances = 6;

char palavra[20];

int categoriaErrada = 0, multiCaractere = 0;

do {

// Início!!

//Aqui fica as intruções iniciais

if (categoria != '5' && categoria != '6' || categoria == '9') {

limpar\_console();

printf("===================================================================\n");

printf("-------------------------");

SetConsoleTextAttribute(hConsole, 6);

printf("Vamos Jogar Forca");

SetConsoleTextAttribute(hConsole, 15);

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

desenha\_boneco(7);

SetConsoleTextAttribute(hConsole, 6);

printf("\n Regras:");

SetConsoleTextAttribute(hConsole, 15);

printf(" Você terá 6");

SetConsoleTextAttribute(hConsole, 4);

printf(" %c ", coracao);

SetConsoleTextAttribute(hConsole, 15);

printf("vidas, se errar a letra perderá uma vida.\n Ao perder todas as vidas, Game over!!\n");

SetConsoleTextAttribute(hConsole, 6);

printf("\n Objetivos:");

SetConsoleTextAttribute(hConsole, 15);

printf(" Descobrir a palavra com o máximo de vidas possível.\n Chegar no topo do ranking!!\n");

SetConsoleTextAttribute(hConsole, 6);

printf("\n Observe:");

SetConsoleTextAttribute(hConsole, 15);

printf(" Digite o número correspondente a uma categoria. Caso o \n número informado esteja incorreto, a pergunta será repetida até \n que uma categoria válida seja selecionada.\n");

printf("\n Selecione a categoria de palavras que você deseja jogar!\n\n 1 - Animais\n 2 - Objetos\n 3 - Verbos\n 4 - Frutas\n\n");

printf(" Ou selecione a tela que deseja acessar!\n\n 5 - Créditos\n 6 - Sair\n\n");

if ( multiCaractere == 1 || categoriaErrada == 1) {

SetConsoleTextAttribute(hConsole, 4);

printf(" Essa categoria ou tela é inválida ou não existe, digite outro número!!\n\n");

SetConsoleTextAttribute(hConsole, 15);

} // Interação do usuário com escolha de categoria

printf(" Digite o número da categoria: ");

scanf("%c", &categoria);

categoriaErrada = 0;

if (categoria != '\n' && (getchar()) != '\n') {

multiCaractere = 1;

} else {

multiCaractere = 0;

}

fflush(stdin);

}

//Seleção da categoria e sorteio da palavra!!

switch (categoria) {

case '1':

indice = rand() % (sizeof(animais) / sizeof(animais[0]));

strcpy(palavra, animais[indice]);

strcpy(dica, "Animal");

break;

case '2':

indice = rand() % (sizeof(objetos) / sizeof(objetos[0]));

strcpy(palavra, objetos[indice]);

strcpy(dica, "Objeto");

break;

case '3':

indice = rand() % (sizeof(verbos) / sizeof(verbos[0]));

strcpy(palavra, verbos[indice]);

strcpy(dica, "Verbo");

break;

case '4':

indice = rand() % (sizeof(frutas) / sizeof(frutas[0]));

strcpy(palavra, frutas[indice]);

strcpy(dica, "Fruta");

break;

// Créditos

case '5':

do {

limpar\_console();

printf("=============================================\n");

printf(" Agradecimento Especial!\n");

printf("=============================================\n\n");

printf(" Agradecemos a contribuição dos integrantes:\n\n");

printf(" - Júlio César Carvalho Santos\n");

printf(" - Heitor de Oliveira Mamede\n");

printf(" - Perilo\n");

printf(" - Kauã Ravy Alves Pontes\n\n");

printf(" Pelo empenho e dedicação ao projeto!\n\n");

printf("=============================================\n");

if (multiCaractere == 1 || categoriaErrada == 1) {

SetConsoleTextAttribute(hConsole, 4);

printf("\n Essa categoria ou tela é inválida ou não existe, digite outro número!!\n\n");

SetConsoleTextAttribute(hConsole, 15);

}

printf(" Selecione a tela que deseja acessar!\n\n 1 - Ínicio\n 2 - Sair\n\n");

printf(" Digite o número da categoria: ");

scanf(" %c", &categoria);

categoriaErrada = 0;

if (categoria != '\n' && (getchar()) != '\n') {

multiCaractere = 1;

} else {

multiCaractere = 0;

}

if (categoria != '1' && categoria != '2') {

categoriaErrada = 1;

}

fflush(stdin);

} while (categoria < '1' || categoria > '2' || categoriaErrada == 1 || multiCaractere == 1);

if (categoria == '1')

categoria = '9';

if (categoria == '2')

categoria = '6';

break;

case '6':

limpar\_console();

tela\_saida();

exit(0);

default:

categoriaErrada = 1;

}

} while (categoria < '1' || categoria > '4' || categoriaErrada == 1 || multiCaractere == 1);

limpar\_console();

printf("===================================================================\n");

printf(" Vamos Começar, adivinhe a palavra!!!\n");

printf(" As palavras que serão adivinhadas, não possuem acentos!!\n Não digite letras repetidas!!\n Você possui 6");

SetConsoleTextAttribute(hConsole, 4);

printf(" %c");

SetConsoleTextAttribute(hConsole, 15);

printf(" vidas.\n", coracao);

desenha\_boneco(chances);

// Tamanho da palavra sorteada

tamanho = strlen(palavra);

// Adapta o sistema para palavras com até 10 letras, no qual se a palavra possui 5 letras, haverão 5 underlines para o usuário substituir!

char variaveis[10] = {x, y, z, a, b, c, d, e, f, g};

char variaveis\_com\_underline[10] = {};

for (underline = 0; underline < tamanho; underline++) { // Obs: a variável underline é um contador

variaveis[underline] = '\_';

if (variaveis[underline] == '\_') {

variaveis\_com\_underline[underline] = variaveis[underline];

}

}

//Funcionamento básico: Vai mostrar as letras utilizadas, mostrar o progresso e pedir ao usuário a letra. tudo isso em loop para ser realizado até ganhar ou perder.

for(letras\_certas = 0; letras\_certas < tamanho;){

printf(" Dica: %s com %d letras.\n", dica, tamanho);

printf("\n Letras Utilizadas: %s\n\n",letras\_usadas);

printf(" %s\n\n", variaveis\_com\_underline);

printf(" Digite uma letra: ");

scanf(" %[^\n]", &letra);

limpar\_console();

acertou = 1;

/\*Verificação se realmente é letra e se a letra já foi utilizada anteriormente.

Pensei em colocar repetição de letra como punição, mas aí seria muito malvado!

Mesmo já tendo colocado uma lista das letras usadas para a pessoa ver, mas como wallace disse, é usuário\*/

//Filtro 1 - Transformar todas as letras em suas versões minúsculas!

if (isupper(letra)) {

letra = tolower(letra);

}

//Filtro 2 - Verificar se o usuário digitou mais de um caractere e impedir a continuidade se isso ocorrer!

if (strlen(&letra) > 1) {

desenha\_boneco(chances);

SetConsoleTextAttribute(hConsole, 6);

printf("\n Você digitou mais de um caractere e isso não é permitido!! Tente novamente!\n\n");

SetConsoleTextAttribute(hConsole, 15);

printf("===================================================================\n\n");

continue;

}

//Filtro 3 - Verifica se o caractere digitado é uma letra, !isalpha = se diferente de letra, entra nos comandos dentro do if

if (!isalpha(letra)){

desenha\_boneco(chances);

SetConsoleTextAttribute(hConsole, 6);

printf("\n É necessário digitar uma letra e esse caractere não é uma letra!! Tente Novamente, digitando uma letra!\n\n");

SetConsoleTextAttribute(hConsole, 15);

printf("===================================================================\n\n");

continue;

}

//Filtro 4 - Verificar se a letra já foi utilizada anteriormente, e impedir isso!!

if(strchr(letras\_usadas, letra) != NULL) {

desenha\_boneco(chances);

SetConsoleTextAttribute(hConsole, 6);

printf("\n Você já utilizou essa letra e isso não é permitido!! Tente novamente!\n\n");

SetConsoleTextAttribute(hConsole, 15);

printf("===================================================================\n\n");

continue;

}

//sistema de concatenação de letras em uma string, para não precisar fazer um loop para printar cada letra usada!

letras\_usadas[strlen(letras\_usadas)] = letra;

letras\_usadas[strlen(letras\_usadas) + 1] = '\0';

contagem\_erros = 0;

// Abaixo há a verificação se as letras estão presentes ou não da palavra selecionada aleatoriamente

for(i = 0; i <= tamanho;i++){

if (letra == palavra[i]){

letras\_certas++;

variaveis\_com\_underline[i] = letra;

if (acertou == 1) {

desenha\_boneco(chances);

SetConsoleTextAttribute(hConsole, 10);

printf("\n Você acertou a letra '%c' está presente na palavra misteriosa.\n\n", letra);

SetConsoleTextAttribute(hConsole, 15);

acertou = 0;

}

}

else{

contagem\_erros++;

if (contagem\_erros > tamanho){//Afirma se o usuário errou alguma letra e qual letra ele errou

chances--;

desenha\_boneco(chances);

SetConsoleTextAttribute(hConsole, 12);

printf("\n Você errou, a letra '%c' não está presente na palavra misteriosa.\n\n", letra);

SetConsoleTextAttribute(hConsole, 15);

if (chances == 0){ //Verifica se o usuário perdeu todas as vidas, ou seja, se deu game over

derrotas++;

SetConsoleTextAttribute(hConsole, 12);

printf(" Você perdeu todas as suas vidas.\n\n");

SetConsoleTextAttribute(hConsole, 15);

printf(" A palavra correta era: ");

SetConsoleTextAttribute(hConsole, 6);

printf("%s\n\n", palavra);

SetConsoleTextAttribute(hConsole, 15);

if (letras\_certas > 0){

printf(" Você acertou %d letra(s)!!\n\n", letras\_certas);

}else{

printf(" Infelizmente você não acertou nenhuma letra!!\n\n");

}

printf("===================================================================\n\n");

goto out; // go to serve para ir para uma parte específica do código, e é bom para fugir dos loops

}else{//Afirma quantas vidas remanescentes o usuário possui

printf(" Só lhe restam %d vida(s).\n\n", chances);

}

}

else{

continue;

}

}

}

printf("===================================================================\n\n");

}

//Verificação se venceu -- Transformar em função

if (letras\_certas == tamanho) {

limpar\_console();

printf(" Você acertou a palavra %s!!\n\n", palavra);

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

vitorias++;

}

out: // checkpoint do goto out

printf("\n Você venceu %d e perdeu %d\n", vitorias, derrotas);

do {

printf(" Deseja jogar novamente?\n 1 - Sim\n 2 - Não\n");

printf("\n Resposta: ");

scanf(" %c", &resposta);

if (resposta != '\n' && (getchar()) != '\n') {

multiCaractere = 1;

} else {

multiCaractere = 0;

}

// Filtro de resposta 2

if ((resposta != '1' && resposta != '2') || multiCaractere == 1) {

limpar\_console();

SetConsoleTextAttribute(hConsole, 4);

printf(" Essa resposta é inválida ou não existe, digite outro número!!\n\n");

SetConsoleTextAttribute(hConsole, 15);

}

} while (resposta != '1' && resposta != '2');

} while(resposta == '1');

limpar\_console();

tela\_saida();

};