Código-fonte

Projeto 1 - Algoritmos e Programação II

Aluno: Mihael Rommel Barbosa Xavier / RA: 10239617

Projeto I:

Programa que simula jogos como wordle, termo, entre outros do estilo adivinhe a palavra. Chamei esse jogo de "MACK-Advinha", ele atende as especificações do projeto, porém fiz algumas mudanças. O dicionário (arquivo disponível no GitHub da matéria) continha palavras de diversos números de caracteres, algumas de apenas 3 caracteres, ou mais que 5. Tendo em vista que no modelo de jogo desses citados (worldle, termo, letreco, etc), a palavra a ser adivinhada sempre é uma palavra de 5 letras, criei uma forma de filtrar as palavras do arquivo dicionário para apenas sortear as palavras de 5 caracteres. Além disso, como o arquivo dicionário contém palvras bem desconhecidas (o que pode ser interessante para o jogo, no caso de ser a palavra a ser adivinhada) vi que muitas das minhas tentativas não eram aceitas por não estarem no dicionário, assim, achei mais fácil filtrar a entrada da tentativa apenas pelo tamanho do caractere. O programa utiliza de buffers para ler o arquivo .txt e ponteiros para manipulá-lo, strlen para verifcar se palavra contém 5 caracteres.

Código-Fonte

//Projeto 1 - Alg e Prog 2 - Aluno: Mhael Rommel Barbosa Xavier - RA: 10239617

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

#include <ctype.h>

#define MAX\_PALAVRAS 6621

#define COMPRIMENTO\_PALAVRA 5

#define MAX\_TENTATIVAS 6

// como decidi imprimir a tentativa em maiusculo, criei essa funcao para padronizar a entrada, evitando problemas

void converter\_para\_minuscula(char \*palavra) {

for (int i = 0; palavra[i]; i++) {

palavra[i] = tolower(palavra[i]);

}

}

int carregar\_dicionario(const char \*nome\_arquivo, char palavras[MAX\_PALAVRAS][COMPRIMENTO\_PALAVRA + 1]) {

FILE \*arquivo = fopen(nome\_arquivo, "r");

if (!arquivo) {

perror("Erro ao abrir dicionário");

return -1;

}

char buffer[100];

int contador = 0;

while (fgets(buffer, sizeof(buffer), arquivo)) {

size\_t tamanho = strlen(buffer);

if (buffer[tamanho - 1] == '\n') {

buffer[tamanho - 1] = '\0';

}

if (strlen(buffer) == COMPRIMENTO\_PALAVRA) {

converter\_para\_minuscula(buffer);

strcpy(palavras[contador++], buffer);

if (contador >= MAX\_PALAVRAS) {

break;

}

}

}

fclose(arquivo);

return contador;

}

void imprimir\_feedback(const char \*tentativa, const char \*palavra\_alvo) {

char tentativa\_min[COMPRIMENTO\_PALAVRA + 1];

strcpy(tentativa\_min, tentativa);

converter\_para\_minuscula(tentativa\_min);

// imprime a tentativa com os carecteres separados por espaco.

for (int i = 0; i < COMPRIMENTO\_PALAVRA; i++) {

printf("%c ", toupper(tentativa[i]));

}

printf("\n");

for (int i = 0; i < COMPRIMENTO\_PALAVRA; i++) {

if (tentativa\_min[i] == palavra\_alvo[i]) {

printf("^ "); // letra correta na posicao certa.

} else {

int letraEncontrada = 0;

for (int j = 0; j < COMPRIMENTO\_PALAVRA; j++) {

if (tentativa\_min[i] == palavra\_alvo[j]) {

letraEncontrada = 1;

break; // letra correta, posicao errada

}

}

printf("%c ", letraEncontrada ? '!' : 'X'); // letra incorreta

}

}

printf("\n");

}

int main() {

char dicionario[MAX\_PALAVRAS][COMPRIMENTO\_PALAVRA + 1];

char palavra\_alvo[COMPRIMENTO\_PALAVRA + 1];

char tentativa[COMPRIMENTO\_PALAVRA + 1];

int qtd\_palavras = carregar\_dicionario("sem\_acentos.txt", dicionario);

time\_t inicio = time(NULL), fim;

double tempo\_gasto;

int tentativas = 0;

if (qtd\_palavras == -1) {

fprintf(stderr, "Falha ao carregar o dicionário.\n");

return 1;

}

srand(time(NULL));

strcpy(palavra\_alvo, dicionario[rand() % qtd\_palavras]);

printf("Bem-vindo ao jogo MACK-Advinha!\n");

printf("Adivinhe a palavra de 5 letras.\n");

while (tentativas < MAX\_TENTATIVAS) {

printf("Tentativa %d de %d. Digite sua tentativa: ", tentativas + 1, MAX\_TENTATIVAS);

scanf("%s", tentativa);

if (strlen(tentativa) != COMPRIMENTO\_PALAVRA) {

printf("Por favor, digite uma palavra com exatamente %d letras.\n", COMPRIMENTO\_PALAVRA);

continue;

}

imprimir\_feedback(tentativa, palavra\_alvo);

if (strcasecmp(tentativa, palavra\_alvo) == 0) {

fim = time(NULL);

tempo\_gasto = difftime(fim, inicio); // calcula o tempo gasto em segundos

char nome\_jogador[50]; // aqui podemos aumentar o numero conforme a necessidade

printf("Parabéns! Você adivinhou a palavra corretamente!\n");

printf("Por favor, digite seu nome: ");

scanf("%s", nome\_jogador);

FILE \*arquivo\_scores = fopen("scores.txt", "a");

if (arquivo\_scores == NULL) {

perror("Erro ao abrir o arquivo de scores");

return 1;

}

fprintf(arquivo\_scores, "%s %s %d %.2f\n", nome\_jogador, palavra\_alvo, tentativas + 1, tempo\_gasto);

fclose(arquivo\_scores);

printf("Sua pontuação foi salva!\n");

break;

}

tentativas++;

}

if (tentativas == MAX\_TENTATIVAS) {

printf("Você não conseguiu adivinhar. A palavra correta era '%s'.\n", palavra\_alvo);

}

return 0;

}