Relatório Técnico

Aluna: Maria Mariana Varela Cavalcanti Souto

Matrícula: 20250033894

Período: 2025.2

Nome do projeto: Jogo de Palavras Cruzadas

1. Introdução

1.1. Contexto do Projeto

Este projeto foi desenvolvido para a disciplina de Introdução a Técnicas de Programação (ITP). O objetivo foi usar os conceitos aprendidos anteriormente e adicionar novos tópicos, como matrizes, strings e, principalmente, ponteiros e gerenciamento de memória.

1.2. Objetivos do Projeto

O objetivo principal era criar um jogo que usasse de forma prática os seguintes conceitos:

- Matrizes (para o tabuleiro).
- Strings (para as palavras e dicas).
- Estruturas de Repetição Aninhadas (loops dentro de loops).
- Ponteiros e Alocação Dinâmica (malloc e free).

1.3. Descrição do projeto

O projeto é um jogo de palavras cruzadas. Nele mostra um tabuleiro vazio e uma lista de dicas e, então, o jogador escolhe o número da dica e digita a palavra. Se o jogador acertar, a palavra aparece no tabuleiro. O jogo acaba quando o jogador acerta tudo.

2. Como o Projeto Foi Feito

2.1. Ferramentas Utilizadas

- **Linguagem:** C.
- **Compilador:** GCC, que eu rodava no terminal para transformar o código em programa.
- **Editor:** Google colab

2.2. Organização do Código

Para o código não virar uma bagunça, eu quebrei o jogo em várias funções. Criei uma função para inicializarTabuleiro, outra para exibirTabuleiro, uma para exibirDicas e assim por diante.

A parte principal foi criar uma "ficha" (uma struct) para cada palavra do jogo. Nessa ficha, eu guardei as posições das palavras no tabuleiro (linha e coluna), as direções (Horizontal ou Vertical), as dicas e as respostas certas.

3. Análise do Código e aplicações do conteúdo

Os conceitos usados no projeto foram:

- **Matrizes (Arrays Bidimensionais):**
  O jogo usa dois tabuleiros que são, na verdade, matrizes char[10][10]. Um deles era o gabarito (com as respostas certas) e o outro era o tabuleiro do jogador, que começava vazio e ia sendo preenchido.
- **Estruturas de Repetição Aninhadas:** Usei o for dentro de outro for o tempo todo. Principalmente para "varrer" a matriz, seja para imprimir o tabuleiro na tela, ou para limpar ele no começo e checar se o jogador já tinha ganhado.
- **Strings:**
  Utilizei strings nas dicas, no gabarito das respostas e o palpite que o usuário digitava. Usei strcmp (da string.h) para checar se o palpite do usuário era igual à resposta certa.
- **Ponteiros e Alocação Dinâmica:**
  Esse foi o maior desafio, mas deixou o projeto bem melhor. Em vez de criar um vetor com tamanho fixo para as palavras, eu usei malloc para pedir memória só quando o programa precisava.
  Cada dica e cada palavra-resposta foram criadas com malloc. Isso significa que se uma dica é longa, ela usa mais memória; se é curta, usa menos. Isso evitou o desperdício.
  No final do programa, eu criei uma função liberarMemoria que usa free para devolver toda a memória que eu pedi, garantindo que não tem vazamento de memória.

4. Dificuldades e Soluções

4.1. Dificuldades Encontradas

Minha maior dificuldade foi com os ponteiros. No começo, o programa dava muito erro e fechava sozinho (os segmentation faults). Eu me confundia na hora de alocar e liberar a memória das structs e das strings que estavam dentro delas.

Outro problema foi com o scanf, que às vezes deixava um "Enter" sobrando e atrapalhava a próxima leitura.

4.2. O que eu fiz para resolver

Para o problema do scanf, eu usei um comando (while (getchar() != '\n');) para limpar o buffer (a sujeira) depois de cada leitura. Já no problema da memória, criei uma função só para pedir memória (criarPalavra) e outra só para devolver (liberarMemoria). Fazer esse caminho de ida e volta me ajudou a organizar a lógica e evitar os erros.

5. Conclusão

5.1. O que eu aprendi

Acho que aprendi a deixar um pouco mais complexo meus projetos, mesmo com as dificuldades, já que lidar com ponteiros pra mim e algumas manipulações de strings foram bem desafiadoras, além disso aprendi a usar a memória do jeito certo. Entender como malloc e free funcionam foi o mais importante

5.2. Evolução do Projeto

Senti que com o jogo da velha que eu fiz na unidade 1 era como se tudo tivesse um tamanho fixo. Então, esse projeto das palavras cruzadas é muito mais flexível e abriu mais espaço para eu explorar os aprendizados da disciplina. Se eu quiser adicionar mais 20 palavras, eu não preciso mudar o programa todo, só preciso chamar mais vezes a função criarPalavra.

5.3. Próximos Passos

Para melhorar, eu poderia fazer o programa ler as palavras e dicas de um arquivo .txt. Assim, qualquer pessoa poderia criar seu próprio jogo de palavras cruzadas sem precisar mexer no código.