Especificação do Projeto 2: Jogo de Caça-Palavras

versão 1.1

Descrição Geral

O objetivo deste projeto é implementar um jogo de caça-palavras na linguagem C, utilizando uma Árvore AVL para armazenar palavras encontradas e um Árvore Digital (Trie) para validar possíveis palavras. O projeto deverá ser modularizado. A busca pelas palavras deverá ser realizada automaticamente pelo programa, verificando todas as direções possíveis no tabuleiro.

Estrutura do Projeto

O projeto será organizado nos seguintes arquivos:

```
caça-palavras/
|-- avl.c               # Implementação da AVL
|-- avl.h               # Cabeçalho da AVL
|-- trie.c              # Implementação da Trie
|-- trie.h              # Cabeçalho da Trie
|-- jogo.c              # Lógica do jogo
|-- main.c              # Função principal
|-- palavras.txt        # Arquivo com as possíveis palavras
|-- tabuleiro.txt       # Arquivo com o tabuleiro do jogo
|-- Makefile            # Arquivo para compilar o projeto
|-- README.md           # Descrição do projeto
```

Especificações dos Arquivos

1. Arquivo AVL (`avl.c` e `avl.h`)

Objetivo: Implementar e manipular a Árvore AVL.

Funções principais

- Criar um novo nó da AVL.
- Inserir uma palavra encontrada no tabuleiro (validade na Trie) na AVL.
- Balancear a árvore após inserções, realizando rotações (simples e duplas).
- Remover uma palavra na AVL pelo usuário.
- Balancear a árvore após remoções, realizando rotações (simples e duplas).
- Imprimir as palavras armazenadas em ordem alfabética.

2. Arquivo Trie (`trie.c` e `trie.h`)

Objetivo: Implementar e manipular a Árvore Digital (Trie).

Funções principais

- Criar um novo nó da Trie.
- Inserir palavras na Trie (a partir do arquivo `palavras.txt`).
- Buscar palavras na Trie para verificar validade.
- Verificar se uma sequência de caracteres (formada no tabuleiro) corresponde a uma palavra na Trie.

3. Arquivo Jogo (`jogo.c`)

Objetivo: Implementar a lógica do jogo.

Funções principais

- `ler_tabuleiro`: Função para ler o tabuleiro do arquivo `tabuleiro.txt`.
- `ler_palavras`: Função para ler o arquivo `palavras.txt` contendo as possíveis palavras e armazená-las na Trie.
- `buscar_palavras`: Função que irá buscar automaticamente as palavras no tabuleiro, verificando em todas as direções e em ordem direta e inversa. Caso validada, inseri-la na AVL.
- `imprimir_resultados`: Função para exibir as palavras encontradas, armazenadas na AVL.

4. Função Principal (`main.c`)

Objetivo: Controlar a execução do programa.

Implementações esperadas

- Menu para interagir com o usuário.
- Chamada das funções de suporte do jogo.

Busca de Palavras no Tabuleiro

O tabuleiro será quadrado e com tamanho definido no arquivo. A busca das palavras no tabuleiro será realizada automaticamente pelo programa. A cada leitura de palavra (que poderá estar na ordem direta ou inversa), o programa verificará as possíveis combinações horizontais, verticais e diagonais no tabuleiro, comparando as sequências formadas com as palavras armazenadas na Árvore Trie. Verificando a existência da palavra no tabuleiro, a palavra encontrada, bem como as coordenadas das letras inicial e final, devem ser inseridas na Árvore AVL, tendo como chave a **palavra**.

1. Direção Horizontal

A busca horizontal será realizada em cada linha do tabuleiro. A sequência de letras será extraída de cada linha e comparada com as palavras na Trie. Caso a sequência corresponda a uma palavra válida, ela será inserida na Árvore AVL, juntamente com as coordenadas de suas letras inicial e final.

2. Direção Vertical

A busca vertical será realizada em cada coluna do tabuleiro. A sequência de letras será extraída de cada coluna e comparada com as palavras na Trie. Caso a sequência corresponda a uma palavra válida, ela será inserida na Árvore AVL, juntamente com as coordenadas de suas letras inicial e final.

2. Direção Diagonal

A busca diagonal será realizada em cada diagonal do tabuleiro. A sequência de letras será extraída de cada diagonal e comparada com as palavras na Trie. Caso a sequência corresponda a uma palavra válida, ela será inserida na Árvore AVL, juntamente com as coordenadas de suas letras inicial e final.

3. Exemplo

O programa buscará as seguintes palavras, oriundas do arquivo `palavras.txt`:

```
ganimedes calisto io metis adrasteia
ananke carme pheme europa himalia
elara pasifae sinope leda agape
```

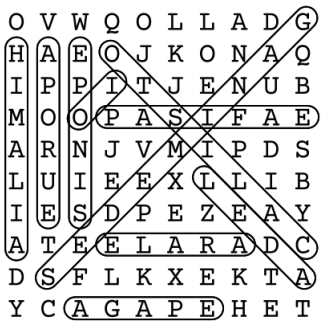
no tabuleiro, oriundo do arquivo `tabuleiro.txt`:

```
10 10
O V W Q O L L A D G
H A E O J K O N A Q
I P P I T J E N U B
M O O P A S I F A E
A R N J V M I P D S
L U I E E X L L I B
I E S D P E Z E A Y
A T E E L A R A D C
D S F L K X E K T A
Y C A G A P E H E T
```

e encontrará automaticamente as seguintes palavras:

```
ganimedes calisto io europa himalia
elara pasifae sinope leda agape
```

Palavras encontradas destacadas no tabuleiro (apenas para efeito de explicação, não precisando ser implementado):



1 Observações importantes

1. Este projeto vale 5,0 pontos.
2. O projeto poderá ser implementado em dupla.
3. Os nomes dos participantes devem ser enviados pela tarefa, no SIGAA.
4. Idem caso o projeto for implementado por apenas um aluno.
5. A data limite para envio do projeto será 24/11/25.
6. A entrega será feita via tarefa, no SIGAA, em arquivo compactado contendo todos os códigos-fonte em C.
7. O sorteio dos dias de apresentação será realizado no dia 25/11/25 e seu resultado divulgado no Sigaa.
8. Apenas um dos participantes ficará responsável por enviar o projeto compactado, via tarefa, no Sigaa.
9. A implementação do projeto deverá dar suporte a execução tanto no Windows, quanto no Linux.
10. Detecção de código gerado por IA acarretará em penalidade na avaliação deste projeto.
11. Este documento está sujeito a alterações e qualquer atualização será informada pelo Sigaa.