	<p>LABORATÓRIO DE PROGRAMAÇÃO</p> <p>Projeto II 2020/21</p> <p>Entrada de texto preditiva</p>
<p>Nota importante: A fraude denota uma grave falta de ética e constitui um comportamento não admissível num estudante do ensino superior e futuro profissional. Qualquer tentativa de fraude pode levar à reprovação na disciplina tanto do facilitador como do prevaricador, para além de outras consequências previstas na lei.</p>	

Objetivo

Pretende-se com este projeto desenvolver um sistema de entrada de texto preditivo, tal como usado em telemóveis antigos com teclados de 12 teclas.

Competências

1. Escrita de código modular, corretamente formatado e indentado.
2. Definição de novos tipos de dados.
3. Domínio de estruturas de dados dinâmicas.
4. Utilização combinada de estruturas de dados, como tabelas de hash, dicionários e listas ligadas.
5. Implementação eficiente de algoritmos de procura e de modificação de estruturas de dados.
6. Armazenamento não-volátil de estruturas de memória em ficheiros.

Descrição do Problema

Antes da invenção dos smartphones com ecrãs tácteis que conhecemos hoje, os telemóveis baseavam a entrada de informação em pequenos teclados, tipicamente de 3 colunas x 4 linhas de teclas, como os ilustrados na Figura 1. Através destes teclados era possível a introdução de textos com um número de letras (e outros símbolos) que excediam largamente as 12 teclas diferentes de tais teclados. Associada a cada tecla, exibiam-se as diferentes letras que tal tecla permitia escrever. Assim, por exemplo para tecla com o número 2 na Figura 1, escreviam-se as

letras “a”, “b” e “c”, indicando que esta tecla permitia, para além da introdução do número “2”, a introdução destas 3 letras. Sem um algoritmo inteligente e preditivo de introdução de texto (que será o objectivo deste trabalho), a forma de aceder às diferentes letras e símbolos de cada tecla era pelo número de vezes sucessivas que se carregava na tecla (existe um intervalo de tempo pré-definido para considerar um toque sucessivo na tecla como um toque de indexação ao símbolo pretendido). Por exemplo, sem um algoritmo inteligente e preditivo de introdução de texto, a sequência para escrever a palavra “aula” seria:

- Um toque na tecla “2”;
- Dois toques na tecla “8” (indexando a letra “u”);
- Três toques na tecla 5;
- E um toque na “2”.

Ou seja, a introdução de uma palavra de 4 letras necessitava de 7 toques nas teclas.



Figura 1. Representação visual do teclado para entrada de texto

A ideia de um algoritmo inteligente e preditivo para acelerar a introdução de texto através destes teclados é usar um dicionário de palavras para determinar com apenas um toque por tecla qual a palavra que o utilizador pretende escrever, atribuindo automaticamente a letra (ou símbolo) respectivo das diferentes possibilidades associadas a cada tecla. Haverá casos em que existe mais do que uma palavra possível no dicionário para a mesma sequência de teclas, mas dado que serão poucas alternativas é possível mostrá-las ao utilizador e permitir-lhe que escolha a que pretende. Melhor ainda, nestes casos em que as mesmas sequências de teclas indexam várias palavras diferentes, a ordem de escolha de palavras que é apresentada ao utilizador usa a frequência com que o utilizador escreve cada uma destas palavras para seleccionar primeiro a mais frequente, depois a de segunda maior frequência, etc, tornando quase unívoco o processo de introdução de palavras para muitos dos textos introduzidos por cada utilizador específico.

Com um algoritmo inteligente e preditivo um utilizador que quisesse introduzir a palavra “aula” daria apenas os seguintes toques nas teclas:

- Um toque na tecla “2”;
- Um toque na tecla “8”;
- Um toque na tecla “5”;
- E um toque na tecla “2”.

Finda esta sequência, e se a palavra mais frequente no dicionário específico para este utilizador fosse “aula”, nenhum toque adicional seria necessário. Caso o utilizador pretendesse a palavra “bula” (note-se que para um médico ou farmacêutico o respectivo dicionário ponderado pelo histórico de palavras introduzidas poderia mostrar esta palavra como primeira alternativa), que tem a mesma sequência de teclas, o utilizador poderia usar toques na tecla “*” para navegar nas alternativas exibidas e escolher a pretendida.

Por vezes o utilizador pode querer introduzir novas palavras que não estão no dicionário (por exemplo, um nome de um amigo), sendo que as novas palavras passarão a fazer parte do dicionário. A introdução de novas palavras faz-se desligando o modo de escrita inteligente e introduzindo a nova palavra pelo método convencional (toques sucessivos para indexar a letra pretendida). Para introduzir uma nova palavra o utilizador deve navegar com a tecla “*” até ao final das alternativas de palavras sugeridas.

Note-se que os símbolos de pontuação como “,”, “.”, “?””, etc, são indexados pela tecla “1”, sendo usual que esta tecla não funcione no modo de escrita preditivo, mas sim no modo convencional (toque sucessivos vão indexando os diferentes símbolos de pontuação).

Em acréscimo à escrita inteligente e preditiva de texto, os projectos devem permitir a leitura de ficheiros de texto como forma de carregar e ponderar os dicionários. Ou seja, a frequência de uma determinada palavra num ficheiro irá ponderar a ordenação dessa palavra pelo mecanismo de escrita inteligente e preditiva. Um dos ficheiros a carregar será o ficheiro lusiadas.txt, disponível no Moodle, visto nas aulas práticas.

Os dicionários ponderados na estrutura desenhada por cada projecto são mantidos de forma não-volátil num ficheiro, sendo carregados no início de cada programa. Os textos escritos interactivamente pelo utilizador durante uma execução do programa alteram em memória a estrutura que guarda o dicionário de palavras e a respectiva ponderação e ordenação. Ao terminar a execução do programa, esta estrutura é guardada no ficheiro.

A interação com o utilizador pode ser realizada através de uma consola em modo de texto para permitir a entrada de dados e a apresentação de resultados, podendo ser mostrada em modo textual uma simulação das 12 teclas, tal como ilustrado na Figura 1, com as letras e símbolos indexados por cada tecla. Será no entanto mais legível e facilitará a simulação de uma sessão de escrita de texto através de um telemóvel de 12 teclas se usarem uma GUI, usando o GTK Toolkit, tal como visto nas aulas, e produzindo uma interface semelhante à apresentada na Figura 2 (note que nesta figura as palavras vêm de um dicionário de língua inglesa). Os projetos que não recorrerem a uma interface gráfica terão uma classificação

máxima limitada a 80%.

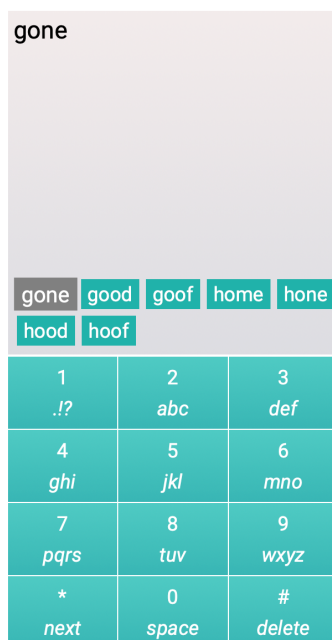


Figura 2. Possível interface gráfica para o projecto

Implementação

A aplicação deve ser implementada na linguagem C, podendo usar qualquer estrutura de dados que achem adequada para este problema. A eficiência da representação em memória que facilite a procura da palavra a introduzir é um dos pontos principais do trabalho, assim como a otimização do espaço em memória ocupado por esta estrutura (a memória dos telemóveis nos anos 90 era escassa).

Utilize ficheiros para armazenar os dicionários ponderados sempre que sair da aplicação, para os recuperar, sempre que reentrar, e para os atualizar, sempre que se justifique, durante a execução do programa.

Composição dos grupos

O trabalho deverá ser realizado pelos mesmo grupos do projecto I.

Data de entrega e material a entregar

O projecto 2 tem como data limite de entrega as **23h59 do dia 4 de junho**, que corresponde ao fim do segundo semestre.

Cada grupo deve entregar obrigatoriamente:

1. Upload de zipFile no Moodle com:
 - Todo o código fonte (.c e .h).
 - Ficheiros de dados para teste.
 - Relatório (em formato pdf).
2. O relatório deve incluir os seguintes aspetos:
 - Esquema com as estruturas de dados utilizadas.
 - Estrutura geral do programa.
 - Estrutura dos ficheiros de armazenamento dos dicionários.
 - Breve explicação de como o programa se executa.

Entregas com atraso serão penalizadas de acordo com a seguinte fórmula:

$$5\% * 2^{MD-1}$$

Em que MD significa cada meio dia de atraso. Como exemplo, um trabalho entregue na tarde seguinte ao dia de entrega terá **1 dia de atraso**. Assim, se fosse avaliado com **16**, teria **14.4** (10% de penalização).

Defesa final do trabalho

O trabalho deverá ser defendido através de uma prova oral com todos os elementos do grupo. Os estudantes que não comparecerem à defesa do trabalho terão a classificação de **zero** valores no projeto.

A nota do projeto será atribuída em função da avaliação do trabalho entregue e da sua defesa.

Apesar do trabalho ser essencialmente um trabalho de grupo as notas são individuais, podendo ser atribuídas notas diferentes a cada elemento do grupo, sempre que tal se justifique.

As defesas decorrerão após a entrega dos trabalhos, em data a anunciar. Será disponibilizado um mapa de defesas onde os grupos se deverão inscrever.