

Universidade de Évora

Estruturas de Dados e Algoritmos I

Jogo do Boggle



Autores:
José Santos nº 43017
Filipe Alfaiate nº 43315

Professora:
Lígia Ferreira



Índice

Introdução	3
Classes Implementadas	4
Conclusão	6

Introdução

Boggle é um jogo de palavras inventado por Allan Turoff e originalmente distribuído pela Parker Brothers. O jogo é jogado com letras posicionadas numa matriz, nas quais os jogadores tentam encontrar palavras em sequências de letras adjacentes.

O trabalho consiste em escrever o código do jogo, de forma a que ele funcione conforme as regras, e que todas as palavras encontradas sejam guardadas num ficheiro de texto. Para a realização do mesmo criamos as seguintes classes: Letra, Position e Boogle. Juntamente, foram agregadas as implementações das Linked List e das HashTables, realizadas durante o período de lecionado.

Classes Implementadas

Position:

Esta classe tem como objetivo juntar a ideia de linha e coluna. Foram inicializadas duas variáveis, *int PosX* e *int PosY*. O construtor atribui um valor as mesmas. Criamos dois métodos: o método *equals()*, que comparar duas posições, caso *PosX* e *PosY* forem iguais devolve *true*, e o método *toString()*, para o print ser do tipo (PosX,PosY).

Letra:

A classe *Letra()* tem como objetivo juntar a Letra do jogo, com a sua posição no tabuleiro. O construtor atribui os valores às variáveis *String w* e *Position p*. Concebemos o método *equals()*, que compara a *Position* e o *toString*, para o print ser do tipo (W:(PosX,PosY)).

Boogle:

São criadas duas HashTables, uma para as palavras(*wordHash*) e outra para os prefixos(*preHash*), uma *LinkedList ll*, e uma matriz de letras, *Letra[][] matriz*.

No construtor é realizada a chamada do método *criarHash()*, que tem como objetivo colocar o dicionário dentro da *wordHash* e os prefixos na *preHash*, começando por abrir o ficheiro de texto, e ler cada palavra uma a uma, coloca a palavra na *wordHash*, e com um ciclo *for()* para criar todos os prefixos e coloca-los na *preHash*.

O método *ler_boogle()*, é chamado na *Main*, e tem como argumento o ficheiro de texto com 16 caracteres, fornecido pela professora, que é usado para colocar cada caracter na sua posição, ordenadamente, dentro da matriz.

O método *print()* faz um print do tabuleiro de jogo.

O método *solve()*, é chamado na *Main*, gera um ficheiro de texto onde são colocadas as respostas.

Depois, usando dois ciclos *for()*, é chamado o método *respostas()*, que tem como objetivo enviar cada Letra da matriz para *respostas()*. O método *respostas()*, tem como argumentos, *int x* e *int y* que são usados para percorrer o matriz recursivamente.

LinkedList ll, é o local onde são guardadas as Letras da *String s*, usado para comparar com as palavras dentro das HashTables. O *BufferedWriter writer*, é usado para escrever no ficheiro de texto a *String s* e a *LinkedList ll*.

Existem 4 condições para ser verdade:

- Verificar que os valores de *x* e *y* tem de ser maiores ou iguais a 0 e menores que 4;
- Verificar se a Letra associada à posição (*x*, *y*) da matriz não está na Linked List, caso não esteja na matriz, a *String s* é atualizada, com (*s+=(matriz[x][y].w).toLowerCase()*) e é adicionada a Letra à Linked List.
- Verificar se *String s* está na *wordHash*, se se verificar, esta é adicionada ao ficheiro de texto;
- Verificar se *String s* está na *preHash* para continuar a recursividade, no final é removida a última letra da Linked List, no caso de não ser nem prefixo nem palavra.

Main:

Por fim, criamos a classe *Main*, onde aplicamos a classe *Boogle* anteriormente descrita. É usado um *BufferedWriter* igual ao método usado *solve()*, serve para atribuir à *String tabuleiro* a linha de 16 caracteres que servirá para criar o tabuleiro de letras do jogo.

Depois criamos o objeto *game* da classe *Boogle* e chamamos os métodos, *game.ler_boogle(tabuleiro)*, *game.print()*, *game.solve()*,

Conclusão

Durante a execução do trabalho, não houve grandes dúvidas ou dificuldades, apenas houve um obstáculo maior de ultrapassar, que foi de como colocar as palavras encontradas num array. Mas, mesmo assim, este foi ultrapassado com a orientação da professora, por sugestão de colocar as palavras encontradas num ficheiro de texto.

Com este trabalho absorvemos mais conhecimento e prática na escrita de código e até mesmo diferentes formas de como solucionar os problemas que vão surgindo no decorrer do trabalho.

Por fim, consideramos que este trabalho se encontra bem sintetizado, organizado e simples, demonstrando objetivamente os conceitos e conteúdos interiorizados no decorrer das aulas.