



Universidade do Porto

Faculdade de Engenharia

FEUP

Mestrado Integrado em Engenharia Informática e Computação

2º Ano – 2º Semestre

2011/2012

Relatório do Projecto – Trabalho 11

Autores:

Cristiano Carvalheiro – 100509041 – ei10041@fe.up.pt
Daniel Santos Teixeira – 100509067 – ei10067@fe.up.pt
Luís Guilherme Martins – 100509105 – ei10105@fe.up.pt

Grupo 03 Turma 02

Porto, 1 de Maio de 2012

1.Introdução

Este projecto foi realizado no âmbito da unidade curricular de Concepção e Análise de Algoritmos incluída no 2º semestre do 2º ano do Curso Mestrado Integrado em Engenharia Informática e Computação e consiste no desenvolvimento de uma aplicação em C++, que nos permita fazer a Gestão de Coleta de Lixo Numa Cidade (T11) (Trabalho 6). Neste relatório falaremos nas alterações correspondentes a Parte 2 do projecto.

Neste trabalho, pretende-se desenvolver uma aplicação em C++ que recorra à API fornecida para visualização de grafos, chamada “GraphViewer”, que seja capaz de fazer a leitura do mapa de uma cidade para um grafo representativo da rede viária da mesma. A aplicação deve receber a informação sobre o mapa da cidade a partir de três ficheiros de texto, que contêm os nós, ruas e sub-ruas.

Além disso, o programa deve permitir a definição de um nó como ponto de partida (central de recolha de lixo) para um circuito de recolha de lixo e de variados nós como pontos de recolha, por onde um ou mais camiões vão ter de passar obrigatoriamente antes de voltarem ao ponto de partida, ou seja, todos os pontos de recolha terão de ser processados pelos camiões existentes. O programa está preparado para que os circuitos definidos consigam balancear a distância mínima percorrida e o número de pontos de recolha visitados por um camião.

2.Descrição da Solução

Esta aplicação é capaz de ler mapas e analisá-los, de forma a determinar caminhos mais curtos para a recolha de lixo em pontos de recolha definidos nesses mesmos mapas. Estes mapas são lidos a partir dos ficheiros gerados pela aplicação OpenStreetMaps Parser.

O utilizador tem várias opções à sua escolha: pode ler um novo mapa, redefinir o centro de recolha, adicionar ou remover pontos de recolha, adicionar ou remover camiões e calcular os caminhos ideais de recolha para os camiões seguirem.

Ler Novo Mapa

A leitura de um novo mapa pode ser feita a partir dos ficheiros gerados pela aplicação *OpenStreetMaps Parser*. O mapa é carregado normalmente, permitindo ao utilizador seleccionar os ficheiros usados na construção desse mapa. Neste caso a informação sobre o mapa carregado é a predefinida (centro de recolha no nó 0, apenas um camião e sem nenhum ponto de recolha).

Definição do Centro de Recolha

O centro de recolha pode ser definido como qualquer nó do grafo, independentemente da posição ou estado actual. Aparece definido no grafo com a cor verde. Apenas pode existir um centro de recolha de cada vez, sendo o ponto inicial e final do circuito feito pelos camiões. Quando um novo centro de recolha é definido, todos os pontos de recolha anteriormente definidos se tornam inválidos.

Adição / Remoção de Pontos de Recolha

Pontos de recolha são pontos nos quais pelo menos um dos camiões de recolha tem de passar obrigatoriamente. Estes pontos aparecem no grafo com a cor vermelha.

Para determinar quais os nós do grafo aos quais é possível chegar a partir do centro de recolha, é necessário fazer uma pesquisa em profundidade ao grafo a partir do nó que representa o centro de recolha. Esta visita produz um vector com o código de todos os nós que são acessíveis pelo camião.

Adição / Remoção de camiões

O utilizador pode adicionar um número ilimitado de camiões. Já na remoção não é permitido ao utilizador remover todos os camiões, tendo de ficar pelo menos um camião activo para poder fazer o circuito.

Cálculo das Rotas de Recolha

Após ter definido o centro de recolha, adicionado os pontos de recolha e definido o número de camiões é possível calcular várias rotas de recolha. Estas rotas têm sempre o objectivo de passar por todos os pontos de recolha, da forma mais eficiente possível, do ponto de vista da distância total recorrida. O cálculo da menor distância entre dois pontos do grafo é feito recorrendo ao algoritmo de Floyd-Warshall. Após escolhido o método de distribuição de pontos, a rota dos vários camiões é calculada segundo esse método. Após esse cálculo estar concluído é apresentado ao utilizador uma lista com estatísticas do percurso de cada camião (distância percorrida e número de nos visitados) e a média de todos os camiões (na consola).

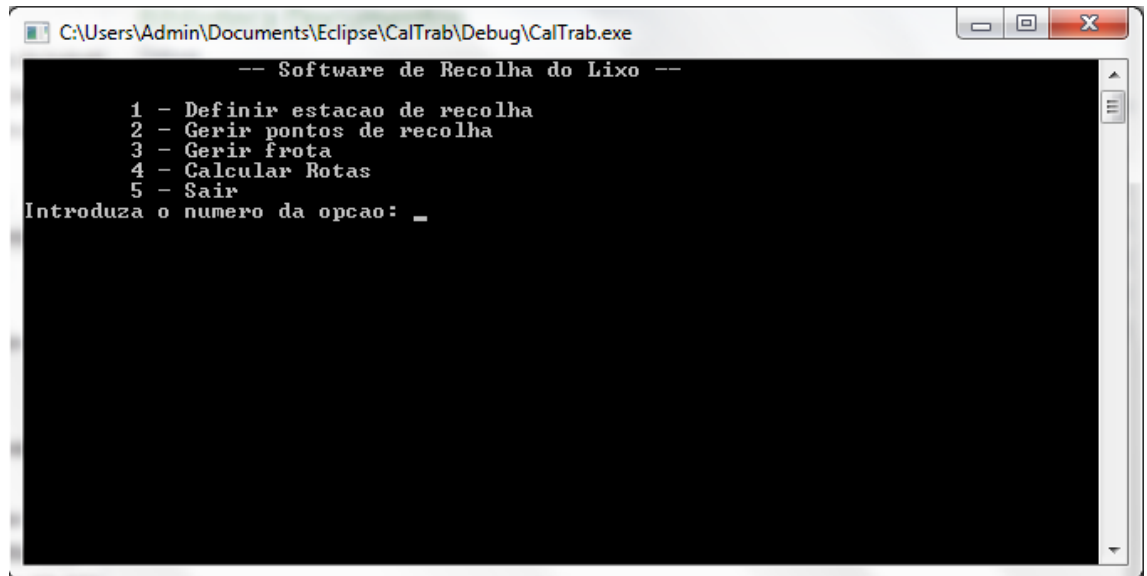
3. Diagrama de classes



4.Casos de Utilização

O programa vai carregar automaticamente os três ficheiros que se encontram com o executável de nome “nodes.txt”, “roads.txt” e “subroads.txt”.

O utilizador tem cinco opções no menu principal:



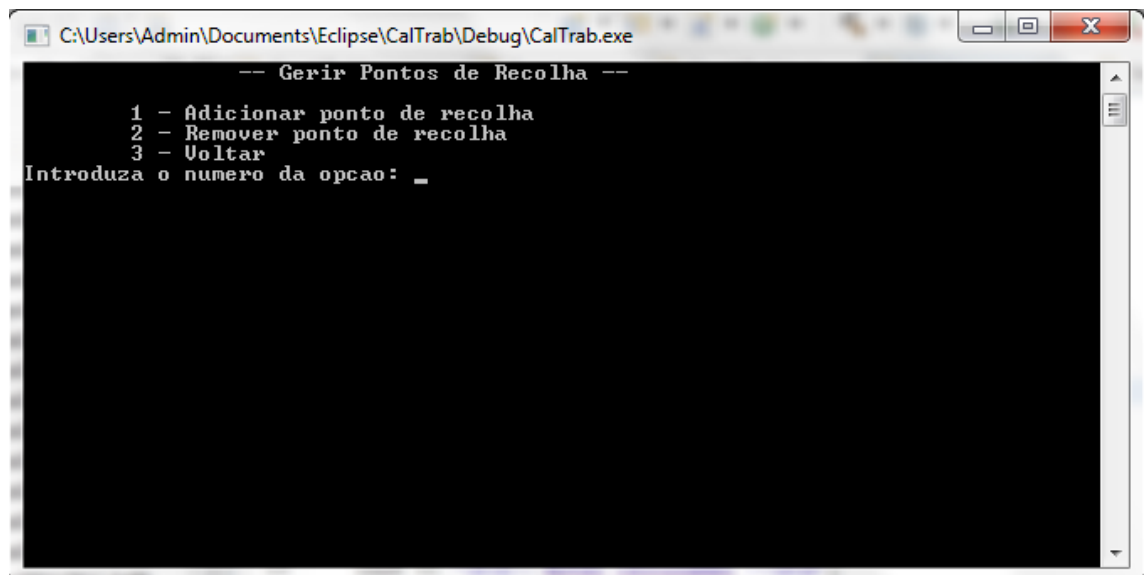
```
C:\Users\Admin\Documents\Eclipse\CalTrab\Debug\CalTrab.exe

-- Software de Recolha do Lixo --

1 - Definir estacao de recolha
2 - Gerir pontos de recolha
3 - Gerir frota
4 - Calcular Rotas
5 - Sair
Introduza o numero da opcao: _
```

1º - Definir estação de recolha – ao escolher a opção numero 1 o utilizador poderá alterar a localização actual da central de recolha escolhendo o vértice ao qual esta deve corresponder.

2º - Gerir pontos de recolha – permite ao utilizador adicionar/remover um ponto de recolha num vértice à escolha:

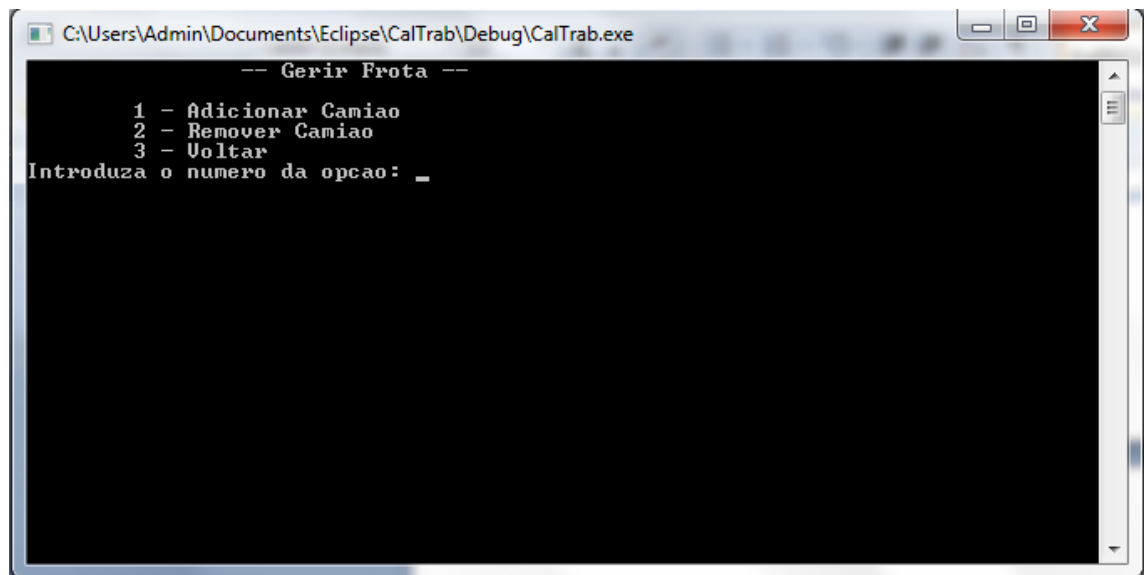


```
C:\Users\Admin\Documents\Eclipse\CalTrab\Debug\CalTrab.exe

-- Gerir Pontos de Recolha --

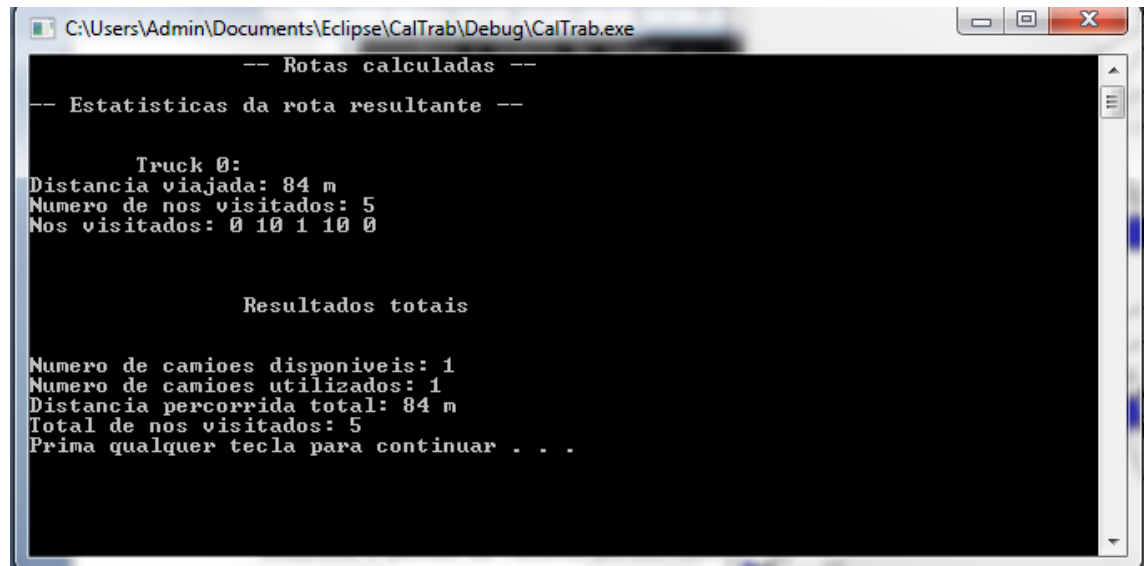
1 - Adicionar ponto de recolha
2 - Remover ponto de recolha
3 - Voltar
Introduza o numero da opcao: _
```

3º - Gerir frota – permite ao utilizador adicionar/remover camiões da frota de veículos:

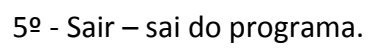


```
-- Gerir Frota --  
1 - Adicionar Camiao  
2 - Remover Camiao  
3 - Voltar  
Introduza o numero da opcao: _
```

4º - Calcular Rotas – calcula a rota “ideal”, ou seja, com base no número total de camiões, calcula a rota mais eficiente passando obrigatoriamente por todos os pontos de recolha existentes. Apresenta ao utilizador um ecrã com a distância total, a distância e pontos de recolha percorridos por cada um dos camiões. No fim apresenta uma animação do gráfico que mostra o caminho percorrido pelo(s) cami(ões):



```
-- Rotas calculadas --  
-- Estatisticas da rota resultante --  
  
Truck 0:  
Distancia viajada: 84 m  
Numero de nos visitados: 5  
Nos visitados: 0 10 1 10 0  
  
Resultados totais  
  
Numero de camioes disponiveis: 1  
Numero de camioes utilizados: 1  
Distancia percorrida total: 84 m  
Total de nos visitados: 5  
Prima qualquer tecla para continuar . . .
```



5.Dificuldades encontradas no desenvolvimento do trabalho

Uma das principais dificuldades deste projecto, foi a decisão de escolher o algoritmo a utilizar. Tendo como objectivo encontrar o caminho mais curto entre dois pontos do grafo, haveria vários algoritmos que poderiam ser utilizados cada um com o seu específico grau de dificuldade. Acabamos por nos decidir pelo algoritmo de Floyd-Warshall pelo seu grau de dificuldade mais reduzido e sua fácil aplicação.

6.Conclusão

Uma vez concluído o projecto, podemos constatar que os objectivos foram cumpridos na sua maioria. O grupo está satisfeito com a sua prestação uma vez que, dada a complexidade do problema que não possui (ainda) uma solução óptima, conseguiu contornar essa dificuldade com algum engenho para produzir resultados bastante conclusivos e fiáveis dentro do possível.

O algoritmo utilizado foi o de Floyd-Warshall que se baseia em programação dinâmica ao contrário do que tinha sido inicialmente pensado, o algoritmo do carteiro chinês. O grupo optou pela primeira solução pois era menos complexa que a outra, para além de que “fugia” um pouco ao conceito a que nos dispusemos que seria escolher os pontos de recolha dentro de um mapa (grafo) que seria disponibilizado.

Podemos então considerar que facilitamos um pouco na complexidade do algoritmo mas por outro lado utilizamos um algoritmo mais rápido e mais “económico” em termos de recursos, existindo então um equilíbrio.

7.Bibliografia

FORTUNA, José Vinicius. Implementações # Algoritmo de Floyd-Warshall, 2010.

<http://lampiao.ic.unicamp.br/maratona/?name=implementa#toc15> - última visita em 01/05/12

WikiMedia. Depth-first search, 2012.

http://en.wikipedia.org/wiki/Depth-first_search - última visita em 01/05/12

WikiMedia.Algoritmo de Floyd-Warshall, 2012.

http://en.wikipedia.org/wiki/Floyd-Warshall_algorithm - última visita em 01/05/12