Descrição da solução

## Estruturas de dados

### Representação do grafo do problema

A base do programa é a abstração Graph (definida no ficheiro *Graph.h*), baseada naquela que é usada nos exercícios da aula prática 5 – mas é evidentemente estendida e modificada, suportando funcionalidades específicas do tema. O suporte para a funcionalidade GraphViewer API fornecida pela framework JUNG, e recomendada no moodle para visualização de grafos, é encapsulada completamente nesta abstração, usando o módulo GraphViewer.cpp/.h fornecido e inalterado.

**Graph** A abstração de *G* é a base do programa. Corresponde na realidade ao sistema de ruas e interseções. É estritamente orientado e cíclico.

**Vertex** A abstração de um vértice *vi*. Corresponde na realidade a uma interseção de ruas, ou a uma parte de uma rua para delinear o seu trajeto curvilíneo. Cada entrada do ficheiro *nodes* é um vértice *vi*.

**Edge** A abstração de uma aresta *ej* . Corresponde a uma porção maioritariamente retilínea de uma rua no mapa real, sem interseções ou bifurcações. Cada entrada do ficheiro *subroads* é uma aresta *ej* .

**Road** A abstração de uma rua *r*. Corresponde a uma rua no mapa real, e é constituído por uma sequência (caminho) de arestas *ej* , *j* = 1*, . . . , k*. Cada entrada do ficheiro *roads* é uma rua *r*.

**Subroad** A abstração de uma porção de uma estrada *s*. À semelhança da abstração Edge, corresponde a uma porção maioritariamente retilínea de uma rua no mapa real, sem interseções ou bifurcações, embora esta seja destinada a gerir e armazenar informação usada pela Edge. Cada entrada do ficheiro *subroads* é uma porção *s*.

Graph encapsula V, R, a API de visualização do grafo e ainda todos os algoritmos implementados, além das estruturas de dados auxiliares à programação dinâmica Graph.

A classe Graph contém os atributos:

* width, height: dimensões do mapa do grafo.
* vertexSet: vetor de apontador para os vértices que constituem o grafo.
* accidentedVertexSet: vetor de apontadores para os vértices que constituem o grafo e que se encontram intransitáveis.
* gv: apontador para o objeto da classe GraphViewer responsável pela representação visual do grafo.
* scale: fator de escala do mapa visualizado no ecrã.

Cada Vertex detém a sua posição no mapa do grafo (coordenadas cartesianas (xi, yi)

inteiras) mas não detém informação acerca da sua posição geográfica, já que as coordenadas

geográficas são descartadas assim que forem mapeadas para as coordenadas (xi, yi) do grafo. Detém ainda, como dados, as arestas que dele saem, i.e. todas as arestas eij, efetivamente encapsulando E.

A classe Vertex contém os seguintes atributos:

* id: inteiro positivo que identifica, de forma única, um vértice.
* x, y: coordenadas cartesianas.
* accidented: booleano que indica se o vértice se encontra acidentado ou não. Os vértices acidentados estão cortados ao transito impossibilitando a passagem de veículos.
* adj: vetor de arestas com origem neste vértice e que o ligam a outros distintos.
* accidentedAdj: vetor de arestas com origem neste vértice que se estão atualmente intransitáveis.
* graph: apontador para o objeto da classe Graph que contém o vértice e que o representa.

Cada Edge encapsula, essencialmente, a informação da subroad que é representada por uma aresta e os vértices que a delimitam.

A classe Edge contém os seguintes atributos:

* id: inteiro positivo que identifica uma aresta de forma única.
* source: vértice de início da aresta.
* dest: vértice de destino de aresta.
* accidented: booleano que indica se a aresta se encontra intransitável ou não.
* graph: apontador para o objeto da classe Graph que contém a aresta e que a representa.
* subroad: apontador para o objeto da classe Subroad que esta aresta representa.

Cada Road encapsula a informação que descreve a rua real em *M*: o seu nome e o comprimento.

A classe Road contém os atributos seguintes:

* id: inteiro positivo que identifica uma estrada de forma única.
* name: nome da estrada.
* bothways: booleano que indica se a estrada é de dois sentidos ou não.
* totalDistance: distância total da estrada.
* maxSpeed: limite de velocidade da estrada calculado a partir da distância.

Cada SubRoad contém a informação do mundo real sobre uma Edge.

A classe Subroad contem os seguintes atributos:

* distance: comprimento da sub estrada.
* road: apontador para a rua à qual pertence.
* actualCapacity: quantidade de carros presentes na sub estrada.
* maxCapacity: quantidade máxima de carros que podem existir na sub estrada.

Para além dos atributos referidos, foram também usados outros adicionais, nas classes Vertex e Edge, específicos para certos atributos, que serão abordados mais á frente.