

ROTAS AÉREAS ENTRE CIDADES MARCIANAS

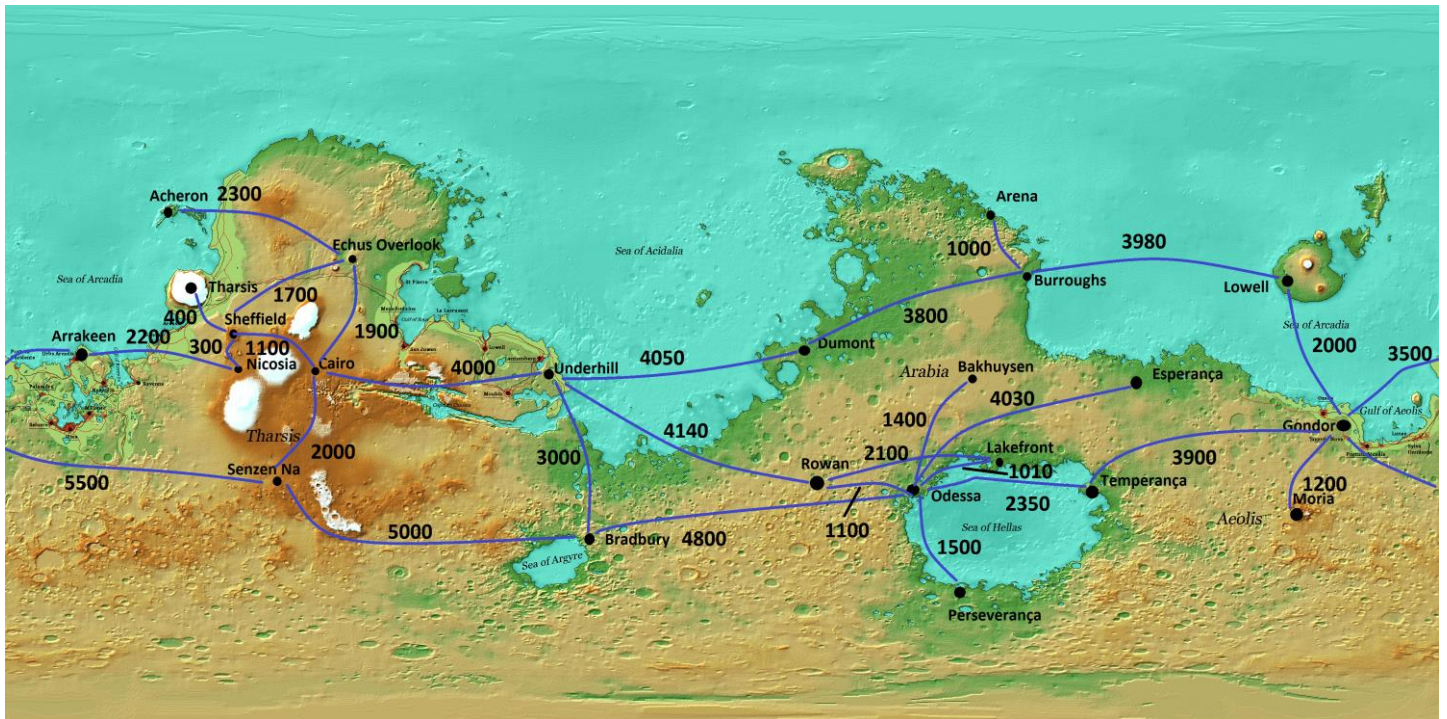
Depois da colonização e terraformação de Marte, em 2187 a Federação Marciana declarou-se independente da Terra e proporcionou aos seus habitantes uma malha aeroviária realizada por dirigíveis ligando os principais centros populacionais do planeta.

A Federação deseja fornecer aos seus usuários um aplicativo que permita verificar os caminhos entre essas principais cidades, através de viagens aéreas.

Para tanto, um arquivo texto contendo nomes de cidades, distância entre elas, tempo de percurso e preço da passagem é fornecido. Esse arquivo se chama CaminhosEntreCidadesMarte.txt.

Há também um arquivo texto chamado CidadesMarte.txt, com um número sequencial associado a cada cidade, o nome da cidade e a coordenada cartesiana (x, y) proporcional às dimensões de um mapa de Marte, onde a cidade deve ser localizada no mapa.

Continuando o projeto I já entregue, o aplicativo deverá encontrar todos os caminhos entre as cidades A e B indicadas e relacionar o mais vantajoso, de acordo com o critério de menor distância percorrida.



Como o mapa no seu tamanho original não caberá na tela, deve-se continuar permitindo que ele se ajuste ao tamanho da tela e lembre-se que isso mudará as coordenadas de exibição de cada cidade no mapa **proporcionalmente** à mudança da altura y e largura x do mapa apresentado na tela, numa proporção entre a largura e a altura da tela com a coordenada (X, Y) original da cidade.

O mapa poderá variar de tamanho caso você aumente ou diminua o tamanho do formulário. Assim, como as coordenadas X e Y são proporcionais, para determinar a o pixel (x, y) exato da cidade no mapa que está sendo exibido, a largura do mapa deve ser multiplicada pela coordenada X da cidade e a altura do mapa deve ser multiplicada pela coordenada Y da cidade. Por exemplo, se o mapa estiver sendo exibido com tamanho de 1200 x 600 pixels (largura x altura), para a cidade de Acheron do exemplo acima as coordenadas onde o círculo deve ser exibido nesse mapa será (136, 173). Você pode escrever o nome da cidade abaixo ou acima do círculo.

Os nomes dos arquivos de cidades e de caminhos devem ser solicitados ao usuário, através de caixas de diálogo. Os arquivos devem ser abertos, lidos e armazenados em listas duplamente ligadas.

A lista ligada de cidades será usada para busca de dados (por idCidade e por nomeCidade), para que se tenha acesso às coordenadas da cidade buscar, de forma que seja possível posicionar os extremos das linhas retas que determinam os caminhos entre cidades que façam parte dos percursos selecionados.

No início da execução do programa deverá também ser pedido o nome do arquivo de caminhos de ligação direta (CaminhoEntreCidadesMarte.txt), através de um OpenFileDialog; esse arquivo deverá ser aberto e lido

totalmente e armazenado em outra lista duplamente ligada (lista de ligações diretas), adequada para armazenar objetos de uma classe que representa um caminho entre duas cidades, contendo os ids das duas cidades diretamente ligadas, a distância, o tempo e o custo do percurso entre elas.

Descrição dos arquivos

CidadesMarte.txt

IdCidade – inteiro, 3 posições
 NomeCidade – string, 15 posições
 CoordenadaX – real, 7 posições
 CoordenadaY – real, 7 posições

CaminhoEntreCidadesMarte.txt

idCidadeOrigem – inteiro, 3 posições
 idCidadeDestino – inteiro, 3 posições
 distancia – inteiro, 5 posições
 tempo – inteiro, 4 posições
 custo – inteiro, 5 posições

No arquivo de ligação direta entre cidades, cada ligação é identificada pelo id da cidade de origem em conjunto com o id de uma cidade de destino, e na lista de ligações ocorrerá o mesmo. Conhecendo-se o id de uma cidade, para se descobrir o nome dessa cidade, deve-se buscar, na lista de cidades, o nó que armazena a cidade com esse id e, daí, acessar seu nome. Conhecendo-se o nome de uma cidade desejada, para se conhecer seu id deverá ser feita uma busca na lista de cidades, usando o nome como chave de busca.

Logo depois da leitura e armazenamento das cidades na lista de cidades, esta deverá ser percorrida e os nomes de cada cidade devem ser armazenados nos dois comboboxes (cbxOrigem e cbxDestino), pois o usuário deverá usá-los para selecionar as cidades cuja ligação deseja incluir, procurar, excluir ou ver os caminhos entre elas. Somente as cidades que já estão armazenadas nos comboboxes poderão ser usadas na manutenção de ligações e na busca de caminhos. Caso outra cidade seja necessária, o usuário deverá incluí-la usando a parte 1 do projeto e executar novamente a parte 2, para que a cidade seja lida e conhecida por essa segunda parte.

Os botões de navegação e manutenção de dados do topo do formulário seriam usados para disparar, eventos que façam a navegação/exibição, busca/exibição, inclusão/salvamento na lista e exclusão de uma ligação entre duas cidades mas, neste momento, você não os precisa codificar. Basta abrir, ler e armazenar os arquivos de dados fornecidos pelo professor.

O usuário indicará as cidades envolvidas em uma ligação (um caminho direto) através dos comboboxes cbxOrigem e cbxDestino. Isso ocorrerá para as operações de inclusão, exclusão e busca de caminhos.

Nesses comboboxes devemos colocar os **nomes** de todas as cidades armazenadas na lista duplamente ligada de cidades. Quando o usuário selecionar o nome de uma cidade num dos comboboxes, deveremos **procurar tal nome na lista de cidades**, para sabermos o seu idCidade.

Ao final da execução do programa deve-se salvar a lista duplamente ligada de ligações diretas.

Quando o usuário pressionar o botão [Achar caminhos], deve-se gerar uma matriz de adjacências, com número de linhas e de colunas igual ao número de cidades da lista de cidades. Em seguida, deve-se percorrer a lista ligada de ligações diretas para gerar o conteúdo de uma matriz de adjacências, onde cada linha e cada coluna da matriz corresponde às posições em que as cidades de origem e de destino estão armazenadas na lista ligada de cidades. O conteúdo de cada posição dessa matriz armazenará a distância que representa a ligação entre a cidade de origem (índice de linha) e a cidade de destino (índice de coluna).

O id da cidade não necessariamente corresponde a um índice, por isso é que precisamos buscar a cidade na lista para saber em que posição sequencial ela se encontra na lista, e essa posição é que será o índice da cidade na matriz.

Em seguida à geração da matriz de adjacências, deveremos procurar todos os caminhos entre as cidades de origem e destino, respectivamente selecionadas nos comboboxes cbxOrigem e cbxDestino.

Melhor caminho (www km)

Achar caminhos

Passando por

- Acheron
- EchusOverlook
- Cairo
- Underhill
- Bradbury

Km do caminho selecionado: (xxxx km)

Caminhos encontrados

	1a	2a	3a	4a	5a	6a	7a	8a	9a
	Acheron	EchusOver...	Sheffield	Nicosia	Arrakeen	Gondor	Temperança	Odessa	Bradbury
	Acheron	EchusOver...	Cairo	Senzen Na	Bradbury				
	Acheron	EchusOver...	Cairo	Underhill	Bradbury				
	Acheron	EchusOver...	Cairo	Senzen Na	Gondor	Lowell	Burroughs	Underhill	Bradbury

Cada caminho encontrado deverá ser exibido em uma linha do DataGridView dgvCaminhosEncontrados.

O menor desses caminhos deverá ser exibido no DataGridView dgvMelhorCaminho.

Quando se clicar numa das linhas do dgvCaminhosEncontrados ou no dgvMelhorCaminho, o trajeto correspondente a essa linha deverá ser desenhado no mapa, com linhas ligando as cidades de cada ligação do trajeto.

IMPORTANTE

- Trabalho feito **pela mesma dupla que fez a parte 1 do projeto**;
- Desenvolver em C# no Visual Studio;
- Comentar adequadamente o programa e o código programado;
- Nomear os identificadores de forma adequada;
- No início dos arquivos fonte, digitar comentário com os RAs e nomes dos alunos;

- Aproveitem as aulas da disciplina para o desenvolvimento e redação do relatório;
- **Relatório de desenvolvimento** deve ser feito num arquivo cujo nome é: RA1_RA2_RelatorioProjeto1ED.PDF (exemplo: 21101_21192_RelatorioProjeto1ED.pdf). Deve conter imagens da execução da busca de caminhos;
- O relatório **deve** ser entregue em formato **PDF**;
- Entrega: **30/06/2019, até às 12h15 (28 dias)**, pelo Google Classroom
- Material a ser entregue: pasta **do projeto, arquivos de dados e PDF compactados em um único arquivo, cujo nome será** RA1_RA2_Projeto3ED.rar (21101_21192_Projeto1ED.rar, por exemplo).

Fontes de informação:

<https://www.deviantart.com/axiaterraartunion/art/Mars-Political-Map-294675891>

<http://www.teses.usp.br/teses/disponiveis/8/8147/tde-21102009-170528/pt-br.php>

https://en.wikipedia.org/wiki/Mars_trilogy

