

## 6ª Aula Prática – Grafos: Utilização da API de visualização de grafos

### Instruções

- Faça download do ficheiro **cal\_fp06.zip** da página da disciplina e descomprima-o (contém os ficheiros base para a realização destes exercícios em Windows)
- Abra o eclipse e importe o projeto que se encontra no ficheiro que descomprimiu.
- Compile o projeto.
- Nota: Em C/C++ Build/Settings/MinGW C++ Linker/Libraries pode ser necessário adicionar Wsock32
- ***Deverá realizar esta ficha respeitando a ordem das alíneas.***
- O código a criar pode ser colocado em FichaJUNG.cpp.

### Enunciado

#### 1. Estrutura base de um grafo

##### a) Configure o seu ambiente de desenvolvimento:

- i. Configure uma janela 600\*600

Nota: para configurar uma janela deve utilizar os seguintes comandos:

```
GraphViewer *gv = new GraphViewer(600, 600, true);  
gv->createWindow(600, 600);
```

- ii. Configure a cor dos nós para *azul*

Nota: para configurar a cor dos nós deve utilizar o seguinte comando:

```
gv->defineVertexColor("blue");
```

- iii. Configure a cor das arestas para *preto*

Nota: para configurar a cor das arestas deve utilizar o seguinte comando:

```
gv->defineEdgeColor("black");
```

##### b) Crie um nó com os seguintes atributos:

- i. Crie um nó com os seguintes atributos:

**Id: 0**

Nota: para criar um nó deve utilizar o seguinte comando:

```
gv->addNode(idNo);
```

Nota: para atribuir as novas alterações ao grafo é necessário executar o comando:

```
gv->rearrange();
```

- ii. Crie um nó com os seguintes atributos:

**Id: 1**

- iii. Crie uma aresta entre os dois nós anteriores.

Nota: para criar uma aresta deve utilizar o seguinte comando:

```
// para arestas bidireccionais  
gv->addEdge(idAresta,idNoOrigem,idNoDestino,EdgeType::UNDIRECTED);
```

```
// para arestas direccionais  
gv->addEdge(idAresta,idNoOrigem,idNoDestino, EdgeType::DIRECTED);
```

- iv. Remova o nó 1

Nota: para remover um nó deve utilizar o seguinte comando:  
`gv->removeNode(idNo);`

- v. Adicione um novo nó com os seguintes atributos:

**Id: 2**

- vi. Adicione uma aresta entre os dois nós criados anteriormente

- vii. Coloque uma legenda no nó 2 com um texto à sua escolha

Nota: para adicionar uma legenda num nó deve utilizar o seguinte comando:  
`gv->setVertexLabel(idNo, "Isto e um no");`

- viii. Coloque uma legenda na aresta com um texto à sua escolha

Nota: para adicionar uma legenda numa aresta deve utilizar o seguinte comando:  
`gv->setEdgeLabel(idAresta, "Isto e uma aresta");`

- ix. Coloque o nó 2 a Verde

Nota: para configurar a cor de um nó deve utilizar o seguinte comando:  
`gv->setVertexColor(idNo, "green");`

- x. Coloque a aresta a Amarelo

Nota: para configurar a cor de uma aresta deve utilizar o seguinte comando:  
`gv->setEdgeColor(idAresta, "yellow");`

- xi. Coloque a imagem “background.jpg” como fundo

Nota: para configurar uma imagem de fundo deve utilizar o seguinte comando:  
`//tem ser colocado antes do comando gv->createWindow(600, 600);  
gv->setBackground("background.jpg");`

## 2. Simulação de animações num grafo.

- a) Adicione nós com os seguintes atributos:

```
id: 0, x: 300, y: 50  
id: 1, x: 318, y: 58  
id: 2, x: 325, y: 75  
id: 3, x: 318, y: 93  
id: 4, x: 300, y: 100  
id: 5, x: 282, y: 93  
id: 6, x: 275, y: 75  
id: 7, x: 282, y: 58
```

id: 8, x: 150, y: 200  
id: 9, x: 300, y: 200  
id: 10, x: 450, y: 200  
id: 11, x: 300, y: 400  
id: 12, x: 200, y: 550  
id: 13, x: 400, y: 550

Nota: para definir a colocação dos nós deve iniciar o GraphViewer com o último argumento “false”:  
`GraphViewer *gv = new GraphViewer(600, 600, false);`

b) Adicione arestas com os seguintes atributos:

id: 0, idNoOrigem: 0, idNoDestino: 1  
id: 1, idNoOrigem: 1, idNoDestino: 2  
id: 2, idNoOrigem: 2, idNoDestino: 3  
id: 3, idNoOrigem: 3, idNoDestino: 4  
id: 4, idNoOrigem: 4, idNoDestino: 5  
id: 5, idNoOrigem: 5, idNoDestino: 6  
id: 6, idNoOrigem: 6, idNoDestino: 7  
id: 7, idNoOrigem: 7, idNoDestino: 0  
id: 8, idNoOrigem: 4, idNoDestino: 9  
id: 9, idNoOrigem: 9, idNoDestino: 8  
id: 10, idNoOrigem: 9, idNoDestino: 10  
id: 11, idNoOrigem: 9, idNoDestino: 11  
id: 12, idNoOrigem: 11, idNoDestino: 12  
id: 13, idNoOrigem: 11, idNoDestino: 13

c) Animação

Nota: para simular a animação dos grafos deve efectuar as alterações que pretende no grafo e em seguida voltar a dar a instrução para desenhá-lo. Para que esta animação seja perceptível pode, por exemplo, utilizar uma instrução de pausa de execução (`sleep(numSegundos)` em Linux ; `Sleep(numMiliSegundos)` em Windows)

- i. Remova os nós 12 e 13
- ii. Adicione dois nós com as seguintes atributos

id: 14, x: 250, y: 550  
id: 15, x: 350, y: 550

d) Com base nos conceitos que aprendeu faça com que a animação seja progressiva e cíclica

Nota: Devido às definições actuais da API de visualização não é possível alterar a posição de um nó. Para efectuar esta alteração deve remover o nó e adicionar um novo com um identificador diferente e com a posição pretendida.

3. Carregar um grafo do ficheiro.

a) Através da leitura dos ficheiros nodes.txt e edges.txt carrega do ficheiro o grafo associado

Os ficheiros encontram-se no seguinte formato:

Cada linha do ficheiro nodes.txt corresponde um nó:

- idNode;X;Y

Cada linha do ficheiro edges.txt corresponde uma aresta:

- idAresta;idNoOrigem;idNoDestino

