

## LISTA DE EXERCÍCIOS 1 (VALOR: 3,0 PONTOS)

### Descrição

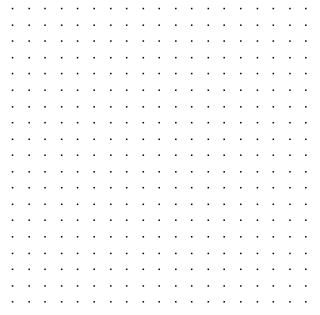
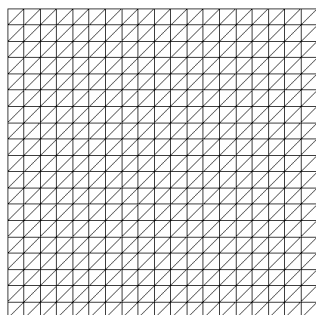
Neste trabalho deverá ser implementada a visualização de modelos como malha de triângulos e como uma nuvem de pontos. Uma imagem digital deverá ser lida para formar o modelo bidimensional.

### Imagem de entrada

A imagem de entrada poderá ser no formato PGM e lida através de funções implementadas por vocês. Caso deseje, poderá ser usado outro formato ou biblioteca de sua preferência (como OpenCV), desde que as funções para manipulação estejam implementadas em seu código-fonte e sejam compiladas juntamente com ele.

### Malha Triangular e Nuvem de Pontos

Dada uma imagem de entrada, o programa deverá ser capaz de mostrar uma malha triangular planar, como na seguinte figura à esquerda, que possui pontos relacionados com os pixels da imagem e arestas conectando pontos adjacentes (lado a lado). Também, deverá ser possível mostrar os pontos relacionados com os pixels da imagem como uma nuvem de pontos (figura abaixo à direita).



Cada ponto (da malha ou da nuvem de pontos) deve ser formado pelas coordenadas  $(x, y, 0.0)$  para cada pixel da imagem. Você deve normalizar os valores dos pontos para que a visualização do modelo, após a leitura da imagem, esteja completamente dentro da janela de visualização. A visualização inicial após iniciar o programa deve ser a da malha triangular.

### Implementação e Execução (Valor: 3,0 pontos)

As implementações devem ser feitas em linguagem C/C++ e compilar, respectivamente, usando os compiladores gcc/g++. Um arquivo Makefile deve ser disponibilizado para a compilação. A compilação deve ser feita usando o comando *make* em um terminal de comandos do Linux. Alternativamente, poderá ser usada a linguagem Python.

Em qualquer caso, o aplicativo deve executar no sistema operacional Linux usando a biblioteca “moderna” do OpenGL, com o uso de *shaders* para mostrar a visualização do modelo. *Não devem ser usadas construções como `glBegin` e `glEnd` das versões antigas do OpenGL!*

O programa deve ter o nome “modelo” e receber como argumento o caminho para a imagem que será visualizada. A execução deve iniciada em um terminal de comandos do Linux, como no exemplo abaixo.

```
$ ./modelo imagem
```

Caso seja implementado em Python, o programa deve ter o nome “modelo.py” e receber uma imagem como argumento. Sendo executado como mostrado abaixo em um terminal de comandos do Linux.

```
$ ./modelo.py imagem
```

Após abrir a janela de visualização, pelo menos a seguinte funcionalidade deve estar disponível para o usuário:

- **Visualização:** ao digitar no teclado a letra “v” o programa deve alternar entre as visualizações de malha triangular e nuvem de ponto.

## Entrega

A entrega deverá ser realizada através da Tarefa definida no SIGAA até às 15h45 do dia 14/10.

- **Código:** submeter um arquivo .zip contendo os códigos e o Makefile (códigos em C).