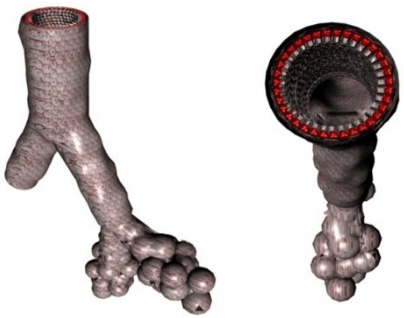
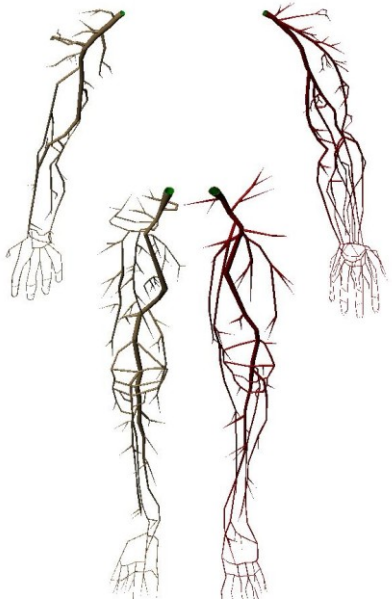
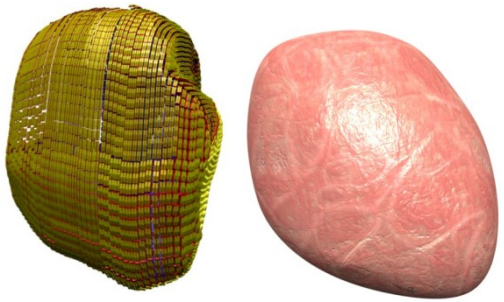
OpenGl 02



Computação Gráfica

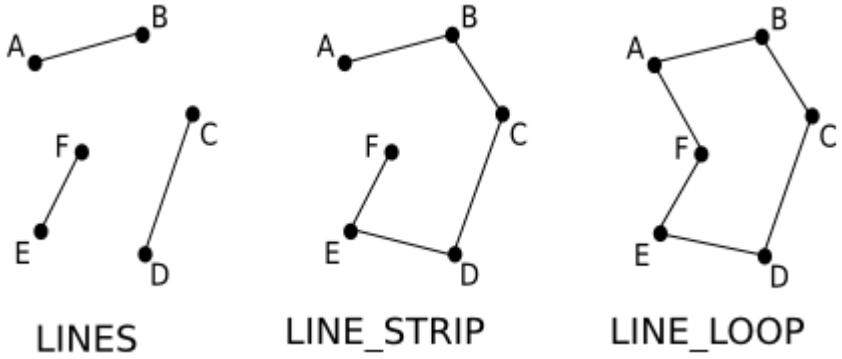
Prof. Dr. Leandro Alves Neves





2. Rasterização: OpenGL

- **GL_POINTS** ⇨ define cada vértice com um ponto.
- **GL_LINES** ⇨ cada par de vértices gera um segmento de linha independente. Vértices n-1 e n definem o segmento de linha.
- **GL_LINE_LOOP** ⇨ desenha um grupo de segmentos de linhas conectados do primeiro ao último vértice, voltando ao primeiro.
- **GL_LINE_STRIP** ⇨ desenha um grupo de segmentos de linhas conectados do primeiro ao último vértice.



2. Rasterização: OpenGL

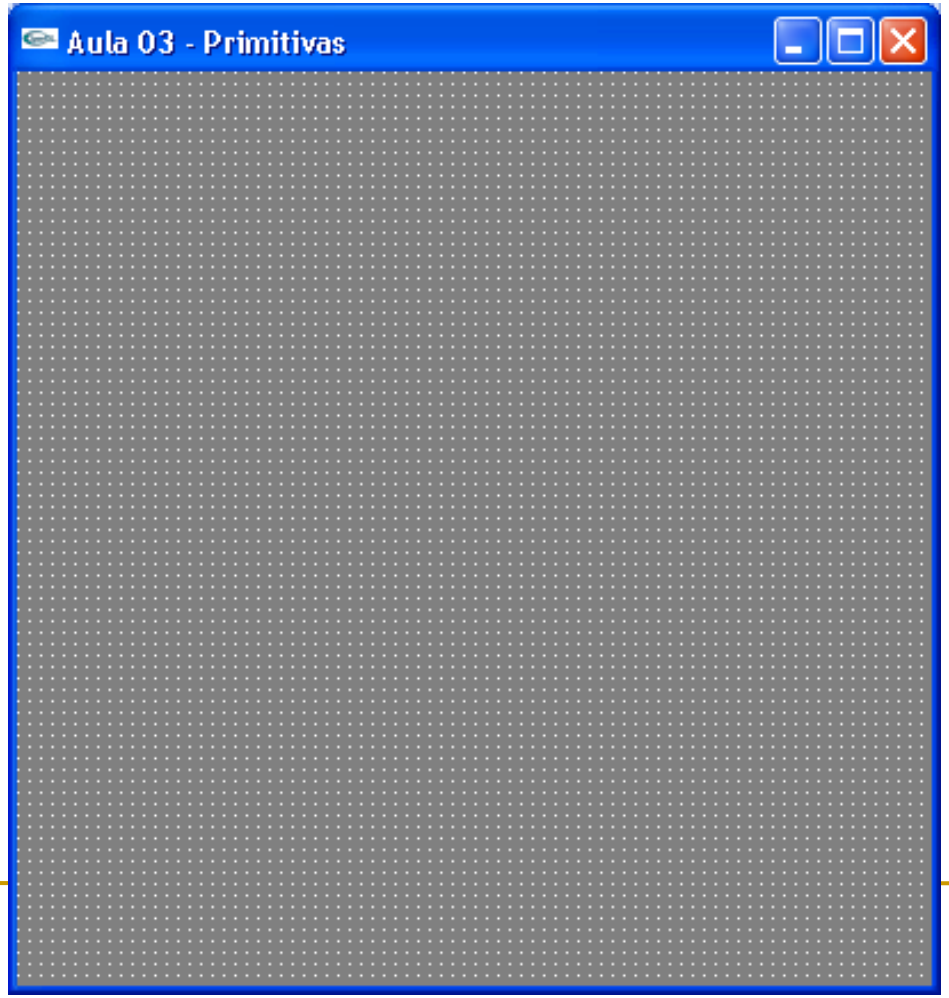
```
// Inicializa parâmetros de rendering
void DISPLAY ()
{// Define a cor de fundo da janela de visualização como preta
   glClearColor(0.5, 0.5, 0.5, 0);
   glMatrixMode(GL_PROJECTION); //Ativa matriz de projeção
   glLoadIdentity();//"Limpa" ou "transforma" a matriz em identidade, reduzindo possíveis erros.
   gluOrtho2D(-200,200,-200,200);//Define tipo de projeção (2D) e o tamanho
   glMatrixMode(GL_MODELVIEW);//Ativa matriz de visualização
      glLoadIdentity();//"Limpa" ou "transforma" a matriz em identidade, reduzindo possíveis erros.
      //Limpa a janela de visualização com a cor de fundo especificada
      glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

glColor3ub(200,100,50);//Define uma cor para a primitiva
      glBegin(GL_POINTS);
        glVertex3f(-10,10,0);
        glVertex3f(-10,-10,0);
        glVertex3f(10,-10,0);
        glVertex3f(10,10,0);
      glEnd();
glutSwapBuffers(); }
```

Exercícios

1. Construa um programa C, com OpenGl, para obter o desenho apresentado abaixo, através do comando GL_POINTS:

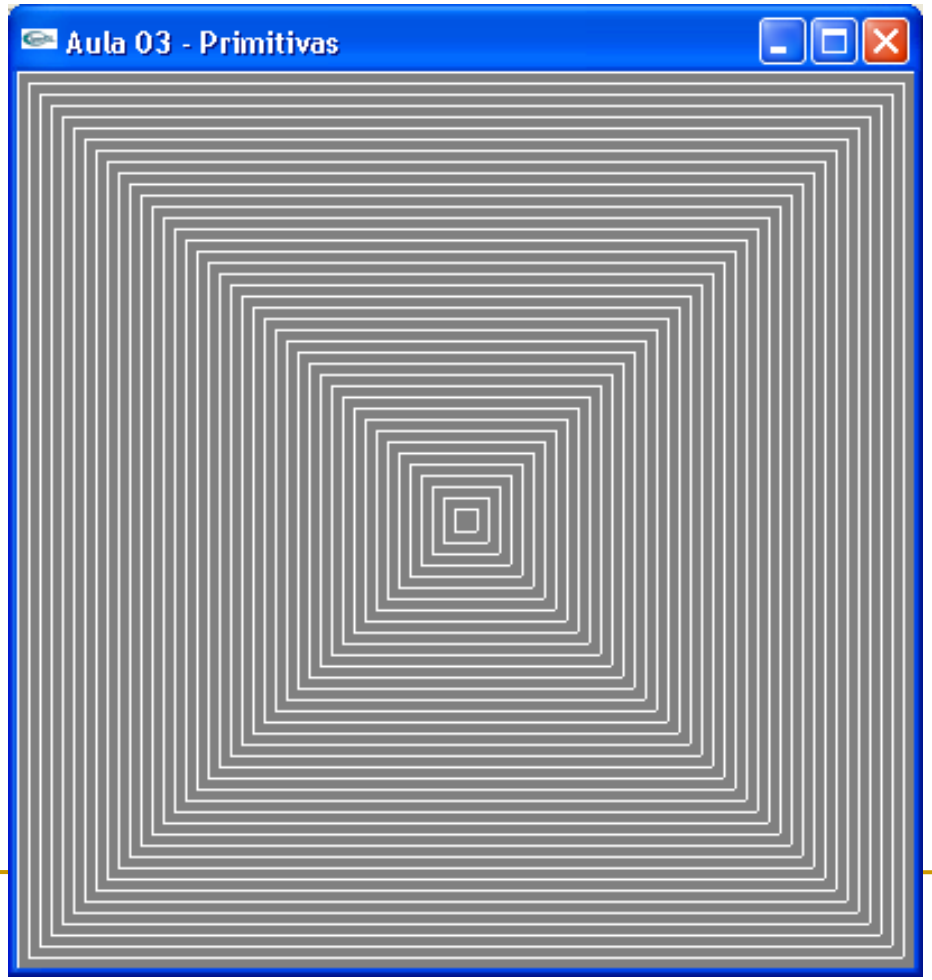
Exercícios

2. Construa um programa C, com OpenGl, para obter um desenho que represente uma matriz de pontos, desenhada através do comando GL_POINTS.



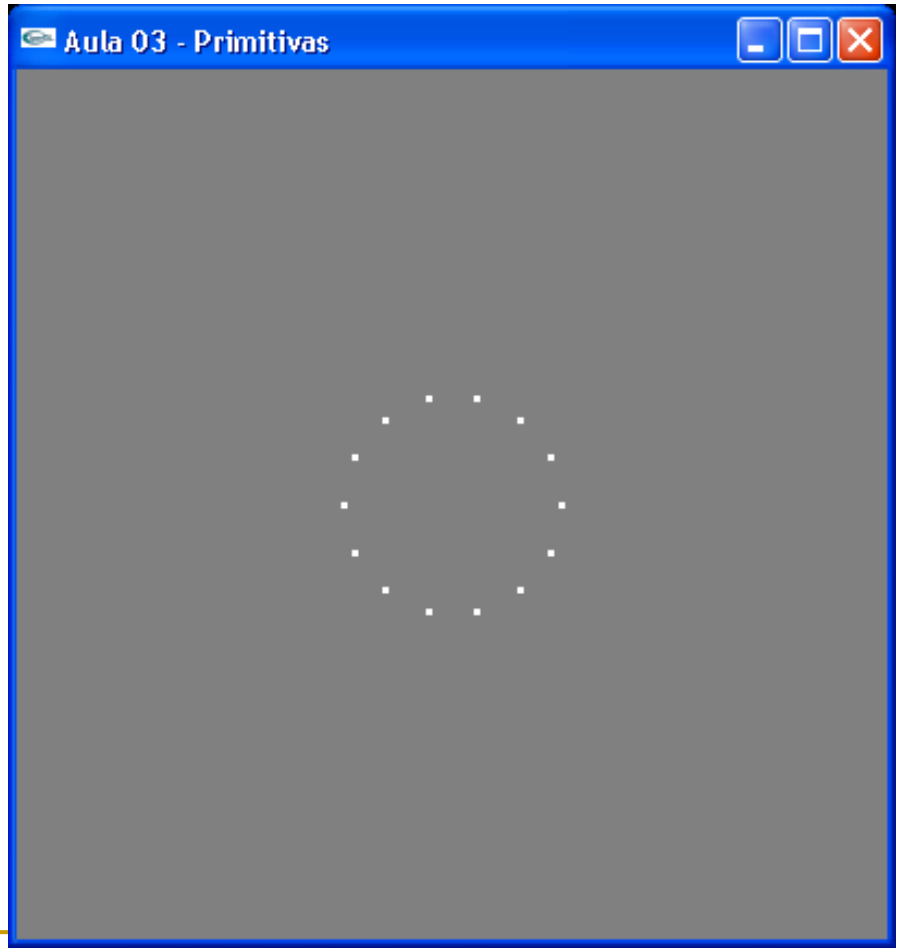
Exercícios

3. Construa um programa C, com OpenGl, para obter um desenho que represente uma matriz de linhas, desenhada através do comando GL_LINES.



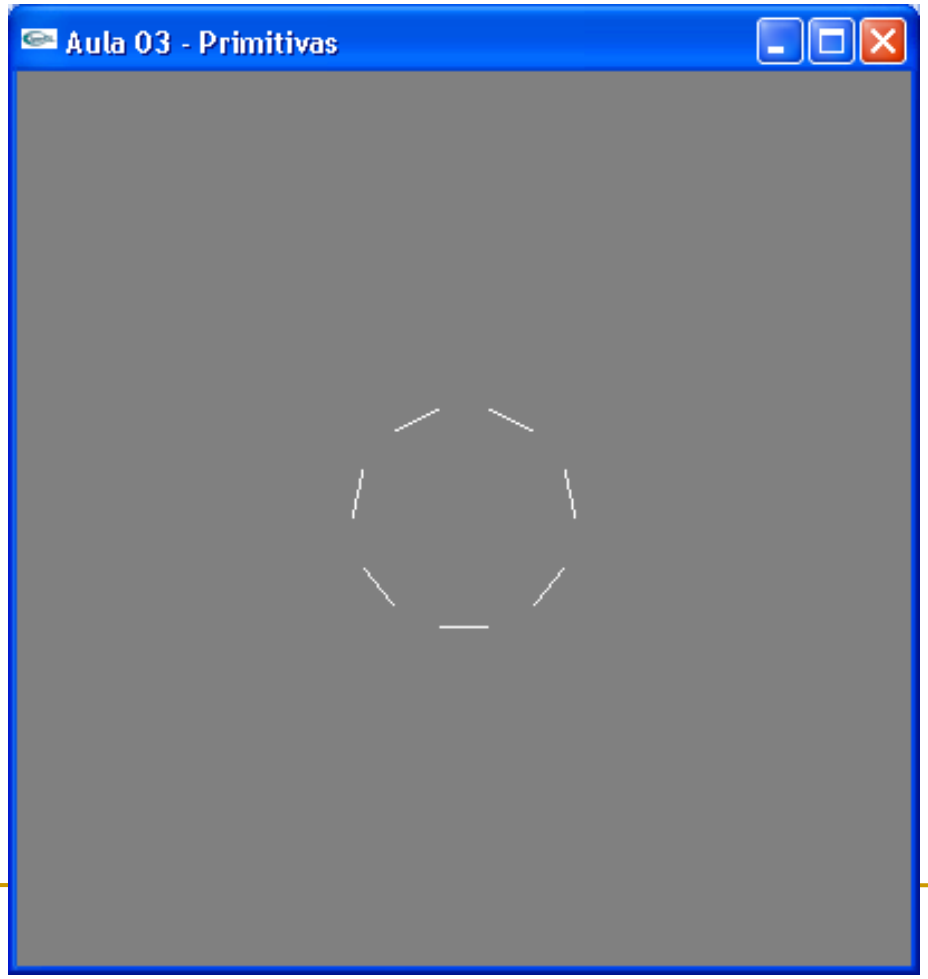
Exercícios

4. Construa um programa C, com OpenGl, para obter um desenho que represente uma circunferência, desenhada através do comando GL_POINTS.



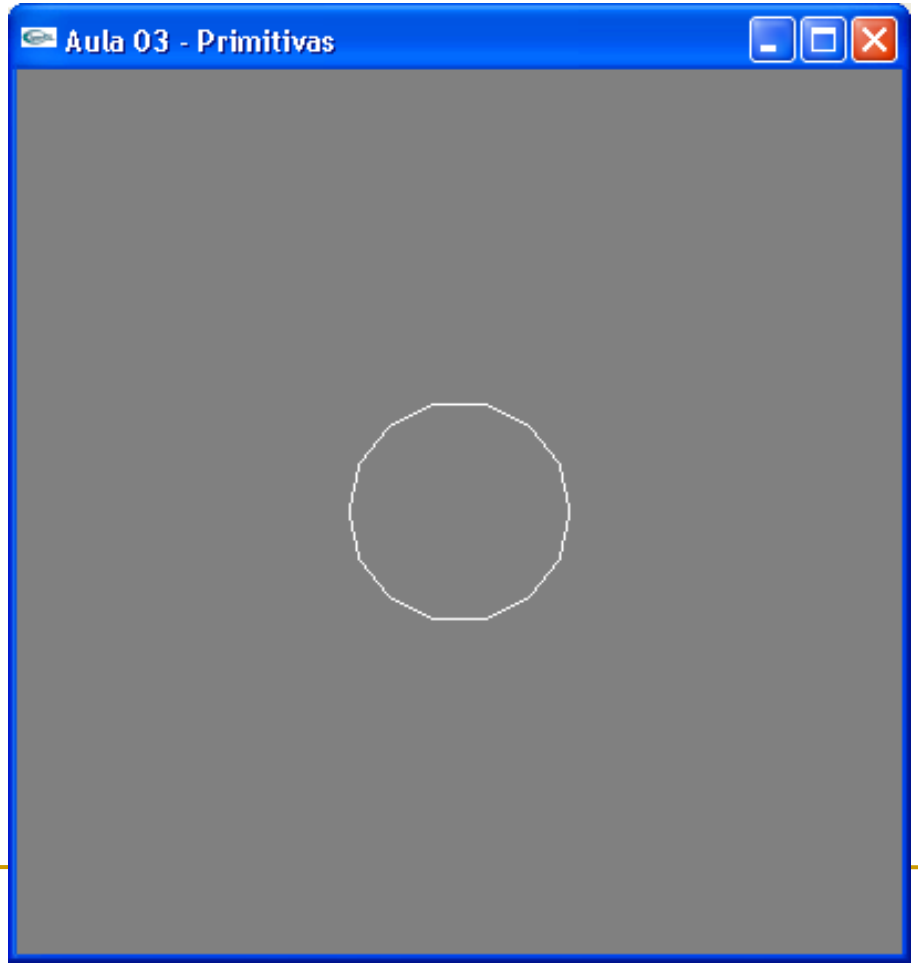
Exercícios

5. Construa um programa C, com OpenGl, para obter um desenho que represente uma circunferência, desenhada através do comando GL_LINES.



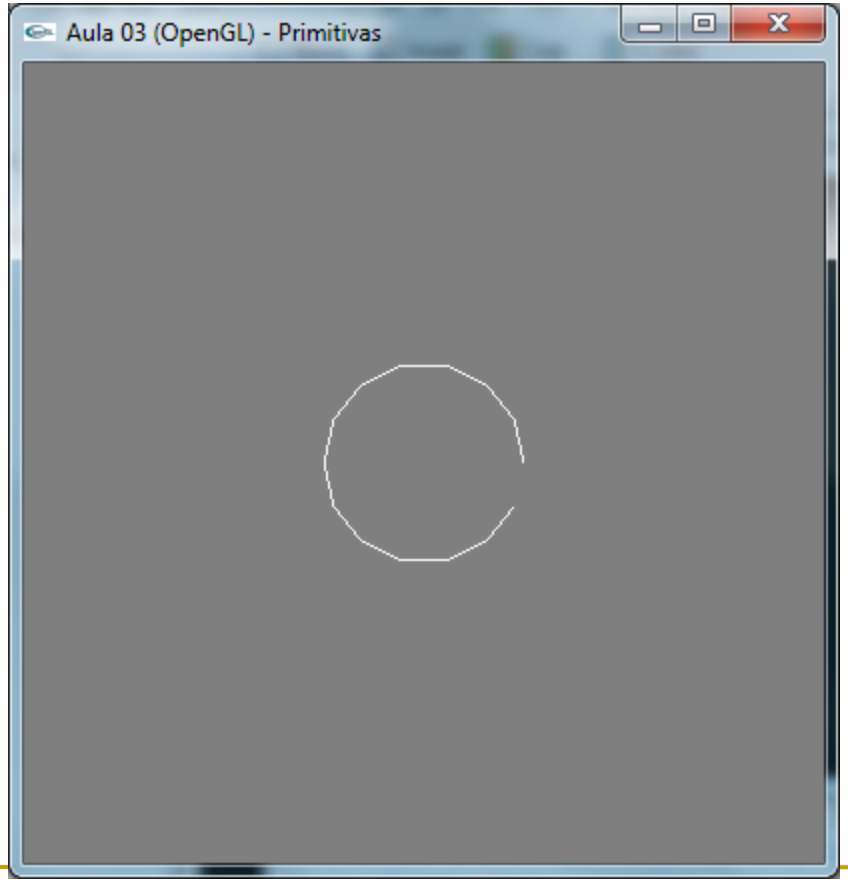
Exercícios

6. Construa um programa C, com OpenGl, para obter um desenho que represente uma circunferência, desenhada através do comando GL_LINE_LOOP.



Exercícios

7. Construa um programa C, com OpenGl, para obter um desenho que represente uma circunferência, desenhada através do comando GL_LINE_STRIP.



Exercícios

8. Implemente o algoritmo de Bresenham para circunferências, com todos os octantes, em uma biblioteca (CIRCLE.h). A função para desenhar circunferências deve receber os parâmetros: raio, posição x, posição y. Esses parâmetros são fornecidos pelo usuário. O desenho da circunferência deve ocorrer na biblioteca.

Exemplo: Circulo(raio, x, y);

9. Considere o exercício 8 e defina também a função “quadrado.h”. O desenho deve ocorrer na biblioteca.

Exemplo: Quadrado(float r, float g, float b, float x, float y, float z, float tam, float transp);

Bibliografia

1. Azevedo, E., Conci, A. Computação Gráfica: Teoria e Prática. Rio de Janeiro: Elsevier, 2003.González, R. C., Woods, R. E.

2. Traina, A. J. M., Oliveira, M. C. F. Apostila de Computação Gráfica. Disponível em: www.icmc.sc.usp.br\gbdi, 2004.

3. Takahashi, R. et al. Apostila: Curso Básico de OpenGl: Programa de Aprimoramento Discente em Modelagem Geométrica.Computacional, UFMG, 2003

4. Cohen, M., Manssour, I. H. OpenGl: uma abordagem prática. São Paulo, Novatec, 2006.