



UNIVERSIDADE FEDERAL DO PIAUÍ  
CENTRO DE CIÊNCIAS DA NATUREZA  
DEPARTAMENTO DE COMPUTAÇÃO  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO  
DC/CCN032 - COMPUTAÇÃO GRÁFICA - 60h - 2020.1  
Prof. Dr. Laurindo de Sousa Britto Neto  
laurindoneto@ufpi.edu.br

**TRABALHO #04**

**Valor do Trabalho: 10,0 pontos**

- 1) Desenvolva um Editor Gráfico em C/C++, estilo o Paint da Microsoft, utilizando as bibliotecas GLUT/OpenGL. O foco deste trabalho consiste em implementar várias técnicas de Rasterização e Transformações Geométricas. Para isso, o programa deve atender, no mínimo, os requisitos listados abaixo:
  - a) Leia os comentários do programa “linha.cpp”. Utilizando o programa “linha.cpp” como referência, implemente o algoritmo de Bresenham para traçado de linhas, utilizando apenas *GL\_POINTS* como parâmetro da função *glBegin()*. Esse parâmetro indica que cada vértice deve ser tratado como um ponto simples;
  - b) O Algoritmo de Bresenham implementado só funciona no primeiro octante. Desenvolva funções com os algoritmos para redução ao primeiro octante e sua transformação inversa para que possam ser desenhadas retas em todos os octantes;
  - c) Utilizando o Algoritmo de Bresenham implementado, desenvolva uma função para desenhar quadriláteros, dadas as coordenadas do topo esquerdo e da base direita desse quadrilátero;
  - d) Utilizando o Algoritmo de Bresenham implementado, desenvolva uma função para desenhar triângulos a cada três coordenadas.
  - e) Utilizando o Algoritmo de Bresenham implementado, desenvolva uma função para desenhar polígonos, dadas quatro ou mais coordenadas.
  - f) Utilizando o programa “linha.cpp” como referência, implemente funções para realizar transformações geométricas como a translação, escala, cisalhamento, reflexão e rotação das formas geométricas rasterizadas. Obs.: as funções de transformações geométricas da API do OpenGL ou de qualquer outra biblioteca não podem ser utilizadas.
  - g) Utilizando o programa “linha.cpp” como referência, implemente uma função com o Algoritmo de Bresenham para rasterização de circunferências. Utilize apenas *GL\_POINTS* como parâmetro da função *glBegin()*. A função de rasterização receberá como parâmetro as coordenadas do centro da circunferência e o tamanho do raio.
  - h) Utilizando o programa “linha.cpp” como referência, implemente a função de rasterização de polígonos. Nesse caso, a função preencherá as formas geométricas rasterizadas com exceção da circunferência.
  - i) Utilizando o programa “linha.cpp” como referência, implemente o algoritmo *Flood Fill* com vizinhança 4. Pode ser usada a função *glReadPixel()* do *OpenGL* para ler a cor do pixel na janela *GLUT*, ou uma matriz auxiliar para a implementação. Nesse caso, a função preencherá as formas geométricas rasterizadas.