

# Lista 01 - INF 2604 - Geometria Computacional

**Professor:** Helio Lopes

**Aluno:** Carlos Raoni de Alencar Mendes

## Biblioteca geometry2d

Os algoritmos solicitados na lista foram desenvolvidos através da implementação de uma biblioteca de classes chamada *geometry2d*, utilizando a linguagem de programação C++. Segue abaixo a descrição dos arquivos que contemplam a biblioteca *geometry2d*, os arquivos abaixo se encontram no diretório *src/* do projeto:

Arquivo	Descrição
<i>geometry2d.h</i>	Arquivo cabeçalho para uso da biblioteca <i>geometry2d</i> .
<i>point2d.h</i>	Arquivo cabeçalho da classe que encapsula a representação de um ponto em duas dimensões.
<i>point2d.cpp</i>	Arquivo de implementação da classe que encapsula a representação de um ponto em duas dimensões.
<i>primitives.h</i>	Arquivo cabeçalho com a definição do protótipo das funções que implementam as primitivas geométricas utilizadas na biblioteca <i>geometry2d</i> .
<i>primitives.cpp</i>	Arquivo com a implementação das primitivas geométricas utilizadas na biblioteca <i>geometry2d</i> .
<i>point_in_polygon.h</i>	Arquivo de cabeçalho com as definições de classes e estruturas utilizadas na implementação dos algoritmos de ponto no polígono.
<i>point_in_polygon.cpp</i>	Arquivo de implementação dos algoritmos de ponto no polígono.
<i>point_in_polygon_gl.cpp</i>	Arquivo contendo a implementação de uma interface gráfica desenvolvida utilizando a biblioteca <i>OpenGL</i> para teste dos algoritmos de ponto no polígono.
<i>convex_hull.h</i>	Arquivo de cabeçalho com as definições de classes e estruturas utilizadas na implementação dos algoritmos de fecho convexo.
<i>convex_hull.cpp</i>	Arquivo de implementação dos algoritmos de fecho convexo.
<i>convex_hull_gl.cpp</i>	Arquivo contendo a implementação de uma interface gráfica desenvolvida utilizando a biblioteca <i>OpenGL</i> para teste dos algoritmos de fecho convexo.
<i>triangulation.h</i>	Arquivo de cabeçalho com as definições de classes e estruturas utilizadas na implementação do algoritmo de triangulação de polígonos.
<i>triangulation.cpp</i>	Arquivo de implementação do algoritmo de triangulação de polígonos.
<i>triangulation_gl.cpp</i>	Arquivo contendo a implementação de uma interface gráfica desenvolvida utilizando a biblioteca <i>OpenGL</i> para teste do algoritmo de triangulação de polígonos.

- 1) Os arquivos que contemplam a implementação do algoritmo estão descritos acima. Para teste do algoritmo você deve executar o arquivo batch próprio do seu sistema operacional que está contido na raiz do projeto:

- Windows: `runPoinInPolygonGUI.bat`
- Linux: `runPoinInPolygonGUI.sh`

Após a execução do batch escolha a opção “0 – Crossing Mumber”. O executável também exibe informações no console de como utilizar a interface gráfica, além de quê é necessário observar o console onde a saída do algoritmo de ponto do polígono será exibida, ou seja, a informação se o ponto está dentro ou fora do polígono será exibida apenas no console.

- 2) Siga as instruções da questão anterior apenas alterando a opção de algoritmo a ser escolhida ao executar a interface gráfica de teste, escolha a opção “1 – Winding Mumber”.
- 3) O algoritmo de índice de rotação tende a ser mais instável que o algoritmo de número de arestas cruzadas, já que o mesmo utiliza aritmética de ponto flutuante tanto na determinação do índice de rotação quanto no momento de comparação do índice de rotação com os valores 0, 1 e -1. Comparação esta que é utilizada justamente para determinar se o ponto está dentro ou fora do polígono. Para minimizar este problema podemos utilizar uma comparação mais robusta como por exemplo a que foi implementado no trabalho, ao invés de comparar diretamente o índice de rotação com 0 o teste realizado foi se a o valor absoluto do índice de rotação era inferior a um determinado épsilon (constante de precisão de ponto flutuante). Outra desvantagem do algoritmo de índice de rotação é a necessidade de uso da função de determinação do arco-cosseno no momento do cálculo do ângulo orientado, esta função é sabidamente cara computacionalmente. Uma maneira de tentar minimizar esta outra desvantagem é procurar uma primitiva pseudo-ângulo que se encaixe para o cálculo do índice de rotação.
- 4) Os arquivos que contemplam a implementação do algoritmo estão descritos acima. Para teste do algoritmo você deve executar o arquivo batch próprio do seu sistema operacional que está contido na raiz do projeto:

- Windows: `runTriangulationGUI.bat`
- Linux: `runTriangulationGUI.sh`

O executável irá exibir no console informações de como utilizar a interface gráfica.

- 5) Os arquivos que contemplam a implementação dos algoritmos estão descritos acima. Para teste dos algoritmos você deve executar o arquivo batch próprio do seu sistema operacional que está contido na raiz do projeto:

- Windows: `runConvexHullGUI.bat`
- Linux: `runConvexHullGUI.sh`

Após a execução do batch escolha a opção “0 – Graham-Scan” para teste do algoritmo de Graham para determinação do fecho convexo e “1 – Divide and Conquer” para teste do algoritmo de dividir para conquistar para determinação do fecho convexo. O executável também exibe informações no console de como utilizar a interface gráfica.