

# Relatório do Trabalho de Computação Gráfica: Algoritmo de Graham Scan

Alexandre Luis Batista da Silva<sup>[115058603]</sup>, Daniel Atkinson Oliveira<sup>[114054054]</sup>,  
Pedro Moya Pinto<sup>[113083474]</sup>, Ricardo Denilson<sup>[106077357]</sup>, and Thairon  
Silva<sup>[111474718]</sup>

Universidade Federal do Rio de Janeiro, Rio de Janeiro RJ, Brasil  
alexandre.lubasi@gmail.com, danatkho@gmail.com,  
pedropintomoya@gmail.com, r.denilson@gmail.com, thairon.ssilva@gmail.com

## Introdução

O trabalho descrito neste relatório foi realizado como projeto final da disciplina "Computação Gráfica" da Universidade Federal do Rio de Janeiro. A restrição imposta sobre o projeto foi que ele fosse relacionado a uma das seguintes áreas de computação gráfica:

- Geometria computacional;
- Modelagem geométrica;
- Processamento de imagens;
- Visualização.

A proposta do grupo é reproduzir o algoritmo Graham scan, que se enquadra na área de geometria computacional.

Mais sofisticado que o Jarvis March, e muito mais eficiente, o algoritmo Graham Scan foi publicado por Ronald Graham em 1972. Tem complexidade  $O(n \log n)$  e se os pontos já estão ordenados por uma das coordenadas ou por um ângulo de um vetor fixo, então o algoritmo tem complexidade  $O(n)$ .

## Funcionamento do Algoritmo

O primeiro passo do algoritmo é achar o vértice de menor coordenada  $y$ . O quesito de desempate caso hajam mais de um ponto com o menor  $y$  é o menor  $x$ . Esse será o ponto  $p_0$ .

Em seguida, os pontos são ordenados em ordem crescente quanto ao ângulo que cada ponto forma com o ponto  $p_0$  e o eixo  $x$ .

O algoritmo processa analisando cada vértice ordenado. Para cada ponto é primeiro determinado se, baseado nos dois pontos anteriores, o ponto constitui uma curva à direita, à esquerda ou se os três pontos são colineares. Se for uma curva à direita ou os três pontos forem colineares, o penúltimo ponto não faz parte do fecho e sim está dentro dele. O mesmo se aplica ao próximo ponto e aos dois que precediam o ponto que foi julgado estar dentro do fecho, o que é repetido até que haja uma curva à esquerda. A partir disso o algoritmo passa para o próximo ponto de forma ordenada, menos os pontos que já foram considerados fazerem parte de dentro do fecho, que não serão revisitados.

## Implementação

O nosso algoritmo apresenta uma interface feita em C++ usando o OpenGL onde podem ser colocados quaisquer número de pontos de forma não ordenada. Esse vértices são armazenados em um arquivo .txt onde em cada linha estarão as coordenadas x e y, respectivamente, de cada vértice. Uma aplicação em C lê o arquivo e calcula os vértices que fazem parte do fecho convexo e os coloca, de forma ordenada, em um outro arquivo .txt que então será lido pela interface e então é desenhado o fecho.

## Referências

1. Página da Wikipedia sobre o algoritmo  
[https://en.wikipedia.org/wiki/Graham\\_scan](https://en.wikipedia.org/wiki/Graham_scan)
2. Portal GeeksforGeeks  
<https://www.geeksforgeeks.org/convex-hull-set-2-graham-scan/>
3. Implementação do código  
<https://github.com/danielatk/graham-scan>