
Computação Gráfica

Trabalho sobre Inclusão de Pontos em Diagramas de Voronoi

2023/2

Introdução

Este trabalho, que deverá ser feito em dupla ou individualmente, consiste em desenvolver programa que avalie algoritmos de inclusão de pontos em polígonos de um diagrama de Voronoi, usando OpenGL.

O programa deverá ler de um arquivo um conjunto de polígonos que formam um diagrama de Voronoi e exibir os polígonos na tela. Esta parte do programa já está pronta no código disponibilizado no Moodle.

A partir disto o programa deverá gerar um ponto que se move sobre o diagrama e, à cada movimento, deve ser informado em qual dos polígonos do conjunto está o ponto.

Não serão aceitos trabalhos que não sejam genéricos quanto ao número de objetos gráficos manipulados.

Complementação do Diagrama de Voronoi

Após a carga dos polígonos do Diagrama de Voronoi, a estrutura de ver a ser atualizada de forma a **armazenar para cada aresta do polígono, qual o outro polígono compartilha aquela aresta**. Isto deve ser feito somente uma vez. A sugestão é fazer isto dentro da função **init()**, como segue.

```
Voro.LePoligonos("ListaDePoligonos-V2.txt");  
Voro.obtemLimites(Min,Max);  
Voro.obtemVizinhosDasArestas();  
CoresDosPoligonos = new int[Voro.getNPoligonos()];
```

Além disto, para cada polígono deve ser gerado um **envelope**. Sugere-se utilizar a estrutura da classe **Envelope** que está no código disponibilizado no Moodle.

Métodos de Inclusão de Pontos em Polígonos

Inicialmente o teste deve ser feito para verificar se o ponto ainda está no mesmo polígono que estava antes do último movimento. Se estiver, nenhum outro teste deve ser feito. Como os polígonos são todos convexos, o teste deve ser realizado com o algoritmo de inclusão de pontos em polígonos convexos. Como saída, deve ser informado quantas vezes a função **ProdVetorial** foi chamada.

Quando se detectar que o ponto saiu do polígono, o programa deve determinar qual o novo polígono contém o ponto através de três algoritmos:

- **Inclusão de pontos em polígonos côncavos:** O teste de inclusão deve ser realizado somente com os polígonos cujos envelopes cruzarem a linha horizontal usada para o teste. Neste caso, deve ser contabilizado o número de vezes que a função **HaIntersecao** foi chamada.

- **Inclusão de pontos em polígonos convexos:** O teste de inclusão deve ser realizado somente com os polígonos cujos envelopes contiverem o ponto. Neste caso, devem ser contabilizados o número de vezes que a função **ProdVetorial** for chamada;
 - **Inclusão de pontos em polígonos convexos utilizando a informação de vizinhança disponível no Diagrama de Voronoi:** o novo polígono deve ser determinado testando qual aresta foi a “cruzada” quando o ponto saiu do polígono. Deve ser contabilizado o número de vezes que a função **ProdVetorial** foi chamada.
-

Entrega

- Data de entrega no *Moodle* e apresentação: 30/09/2023 até às 19:00.
- Os trabalhos podem ser desenvolvidos em duplas.
- Para a entrega do trabalho deverá ser gerado um relatório que demonstre o comportamento dos algoritmos com diagramas de 20, 100 e 500 polígonos.
- Os arquivos, contendo os fontes do programa, devem ser compactados e submetidos pelo *Moodle* até a data e hora especificadas. **ENVIE APENAS ARQUIVOS .ZIP, ou seja, não envie 7z, rar, tar.gz, tgz, tar.bz2, etc.**
- **Deve ser**
- Para a entrega deverá ser criado **um vídeo**, de até 2 minutos, mostrando o funcionamento do programa e um relatório demonstrando que o trabalho atende aos requisitos da especificação e contendo o link para o vídeo.
- **A cópia parcial ou completa do trabalho terá como consequência a atribuição de nota ZERO ao trabalho dos alunos envolvidos.**

FIM.