



### **Primeiro Trabalho**

A triangulação de polígonos planares é um dos processos mais utilizados em diversas áreas da computação. Esse processo é caracterizado pela decomposição do polígono em triângulos. O conjunto desses triângulos deve ser tal que: sua união resulta no polígono original e nenhum triângulo deve possuir interseção com os demais. A partir desse conjunto de triângulos, diversas operações, tais como classificação de pontos e desenho, podem ser feitas de forma incremental, analisando cada um dos triângulos separadamente. Na figura 1 temos um exemplo de uma triangulação de um polígono qualquer.

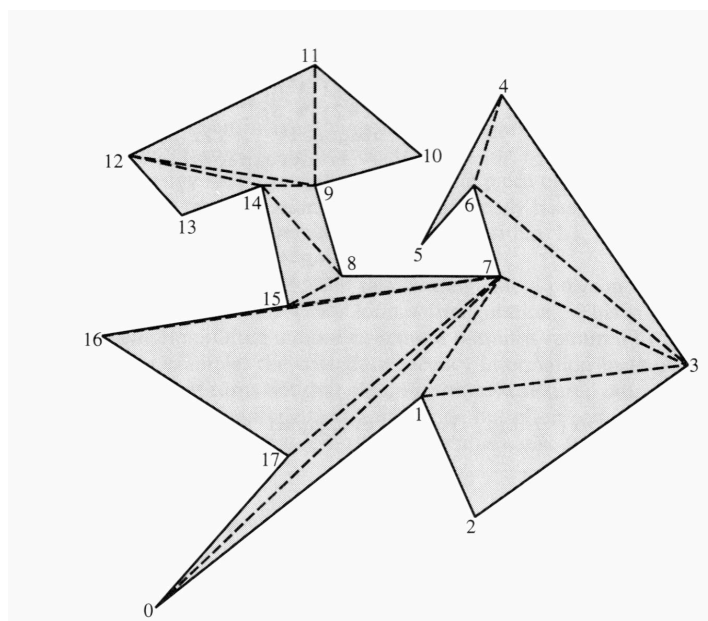


Figura 1 - Exemplo de triangulação de um polígono.

Existem vários algoritmos e critérios para a construção de uma triangulação. Um dos mais simples é denominado de *Triangulação por Otectomia*. O procedimento de otectomia em medicina está relacionado com a excisão da orelha. No contexto de uma triangulação “**orelha**” diz respeito ao vértice “central” de uma sequência de 3 vértices consecutivos do polígono, cuja aresta formada pela conexão entre os dois outros vértices é capaz de formar uma **diagonal própria e interna** ao polígono. Uma **diagonal própria** é aquela que não intercepta nenhuma outra aresta do polígono. Uma **diagonal interna** é aquela que está localizada completamente dentro do polígono. Observando a figura 1 podemos verificar exemplos de seqüências de vértices que não formam orelhas, pois não atendem a essas duas condições, como:

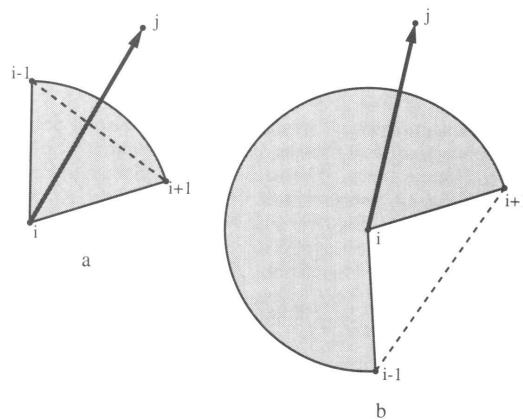
- (2,3,4), a diagonal formada pela aresta (2,4) não seria própria pois cortaria a aresta (6,7) do polígono;
- (5,6,7), a diagonal formada pela aresta (5,7) não é interna.

A partir da definição de orelha podemos escrever um algoritmo simples que constroi uma triangulação a partir da remoção sucessiva de “orelhas”, ou seja, de vértices capazes de formar um triângulo interno ao polígono e sem auto-interseção. Esse algoritmo é apresentado a seguir:

#### Algoritmo TrianguloPoligono

```
Passo 1:   Classificar todos os vértices se são “orelhas”;
Passo 2:   Enquanto for possível formar triângulos com o polígono atual faça
            Para cada vertice classificado como “orelha”;
                Remover o vértice “orelha”;
                Reclassificar os vertices afetados pela remoção;
            fim-para
        fim-enquanto.
```

O processo de classificação é o ponto crítico do algoritmo. Ele se divide em verificar se a diagonal relacionada ao possível vertice “orelha” é própria e interna. A classificação de interna pode ser feita com um teste mostrado na figura 2, que envolve a análise da diagonal  $(i,j)$ . Se a diagonal cortar o setor formado pelos vértices  $(i-1, i, i+1)$  ela é interna.



O outro teste para verificar se a diagonal intercepta alguma aresta do polígono, envolve percorrer cada uma de suas arestas que não incidem nos vértices da diagonal e verificar se há interseção.

Considerando o exposto acima, seu trabalho será, baseado no código fonte base fornecido pelo professor, implementar o algoritmo de triangulação por Otectomia. Esse código fonte está dividido em alguns módulos, a saber:

- **EstruturasDeDados.h**

Descrição das estruturas de dados a serem utilizadas nesse trabalho;

- **opGeometricas.\***

Rotinas responsáveis pelos calculos geométricos necessários ao procedimento de triangulação;

- **winGL.\***

Rotinas responsáveis pelo controle das janelas e dos desenhos;

- **Triangulacao.\***

Rotinas responsáveis pelo processo de geração da triangulação;

- **trabalho.\***

Programa principal e rotinas de tratamento de eventos de teclado, mouse e desenho.

As rotinas que você deve codificar estarão nos módulos **Triangulação, trabalho e EstruturaDeDados**.

Você deverá definir pelo menos duas estruturas de dados diferentes para o armazenamento dos dois elementos básicos trabalhados pelo programa: polígono e triangulação. Estruturas de dados adicionais podem ser criadas se você achar necessário.

O programa deve permitir que o usuário forneça o polígono iterativamente (aproveitando o código base já fornecido para captura dos cliques do mouse) e a partir de uma seleção do teclado, gerar a triangulação. Opções para reiniciar o processo de definição do polígono e para reiniciar o processo de triangulação (entrada de dados e geração da triangulação) deverão estar disponíveis a partir do teclado ou do mouse.

Após a geração da triangulação o usuário deve poder visualizar o polígono original ou a triangulação gerada.

Os trabalhos deverão ser desenvolvidos individualmente ou em duplas. O código fonte gerado deve ser comentado e legível. Acompanhando o código fonte um relatório técnico deve ser entregue, descrevendo as estruturas de dados utilizadas, uma justificativa e quais testes foram realizados para testar o funcionamento do programa (descrição, objetivo e resultado).

O trabalho deverá ser entregue até o final da aula do dia 06/06, impreterivelmente. O código fonte e a documentação deverão ser entregues gravados em um CD. Em caso de qualquer defeito de gravação, arquivos corrompidos ou qualquer outro problema similar o trabalho será considerado não entregue. Portanto, verifique bem o que for entregar!!

A cooperação entre alunos e grupos é considerada salutar. No entanto, trabalhos com alto grau de similaridade serão tratados como “plágio”, o que resultará em avaliação **zero** para todos os envolvidos.

Esse trabalho será contabilizado como 30% da nota da segunda avaliação.

Qualquer dúvida adicional procurem o professor.