



1º Teste

Computação Gráfica

LEIC-Alameda/LEIC-Tagus/LERCI

Prof. Mário Rui Gomes

Prof. João Brisson Lopes

22 de Abril de 2004

Nº _____ Nome: _____

Responda às questões seguintes justificando adequadamente todas as respostas. O Teste tem uma duração de 1h30m. Todas as páginas devem ter o nº mecanográfico.

I

1. A arquitectura lógica de uma aplicação gráfica interactiva baseia-se nos conceitos de eventos e de subscrição de serviços. Relacione estes dois conceitos com o fluxo de execução da aplicação.

A arquitectura lógica de uma aplicação gráfica interactiva baseia-se no conceito de evento, definido como uma ocorrência na vida da aplicação e no conceito de subscrição de um serviço a ser efectuado pela aplicação.

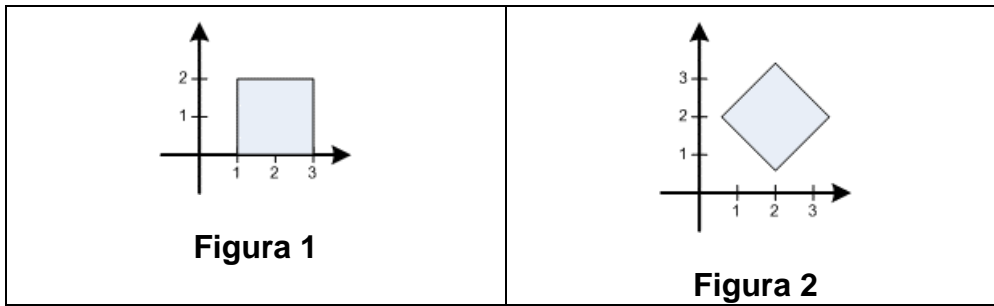
Assim, quando a aplicação é executada mostra em primeiro lugar toda a sua interface gráfica (botões, menus, ícones, etc.) assim como uma Vista inicial do Modelo.

Seguidamente, efectua-se a subscrição dos serviços disponíveis (criar círculo, alterar atributo, agrupar objectos, etc.). O mecanismo de subscrição consiste em associar um ou mais tipos de eventos (premir uma tecla, escolher uma opção de um menu, etc.) a um serviço.

O denominado ciclo principal de uma aplicação gráfica interactiva consiste numa espera activa de ocorrência de eventos. Quando ocorre um evento, é analisado o seu tipo e é executado o respectivo serviço, sendo redesenhada a Vista sobre o Modelo que poderá ter sido editado.

Normalmente existe um tipo de evento que está associado ao serviço de “sair”, o qual armazena o Modelo e termina a execução do processo.

2. Considere a transformação 2D que produz a forma representada na Figura 2 a partir do quadrado representado na Figura 1.



Indique que transformações elementares compõem essa transformação, por que ordem são aplicadas e quais as matrizes correspondentes.

Uma translação de $(-2,-1)$, seguida de uma rotação de 45° e de uma translação de $(2,2)$.

$$T_1(-2,-1) = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R(45^\circ) = \begin{bmatrix} \cos(45^\circ) & -\sin(45^\circ) & 0 \\ \sin(45^\circ) & \cos(45^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_2(2,2) = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

3. Além da modelação por instanciação de primitivas, existe outro tipo de modelação que também emprega primitivas? Em caso afirmativo, indique quais as diferenças.

Sim, existe a modelação por decomposição em células em que existem formas primitivas livremente escaláveis e posicionáveis que, adaptadas à forma dos objectos a modelar, modelam os objectos. No caso da modelação por instanciação de primitivas, estas são objectos que, embora posicionáveis, não apresentam a liberdade de escalamento livre pois as suas dimensões encontram-se restringidas pelos valores dos parâmetros das primitivas e da sua correspondência com as respectivas dimensões.

4. Que diferenças existem entre as malhas poligonais de lista explícita, as malhas de lista de vértices e as malhas de lista de arestas? Dê um exemplo de uma operação que seja fácil num destes tipos e muito difícil noutra.

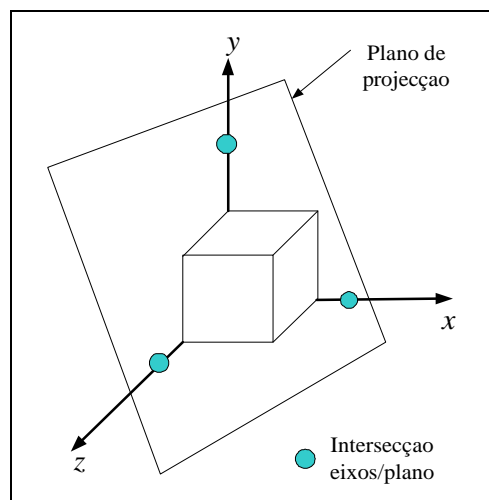
Nas malhas de lista explícita, cada polígono dispõe de uma lista de vértices própria, que não partilha com quaisquer outros polígonos, e em que cada vértice está representado

pelas suas coordenadas. Na malha por lista de vértices, existe uma lista de vértices geral e cada polígono possui uma lista de referências para os vértices que o constituem. Finalmente, na lista de arestas, existem a lista de arestas e a lista de vértices. Cada polígono possui uma lista de referências para as arestas que o constituem e cada aresta tem duas referências aos respectivos vértices e uma lista de referências para os polígonos de que faz parte.

A alteração das coordenadas de um vértice é muito difícil com a lista explícita, pois é necessário determinar quais os polígonos a que pertence para que as suas coordenadas possam ser alteradas. O mesmo não se passa com a modelação por lista de vértices dado que basta alterar um único vértice sem ter que procurar as suas cópias.

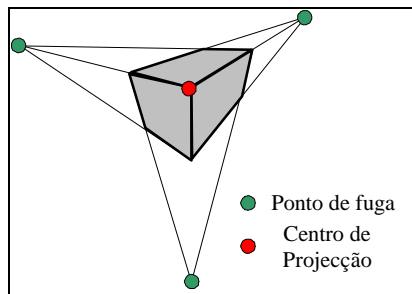
II

1. A seguinte figura apresenta um cubo assim como o plano de projecção para realizar a vista do cubo em perspectiva. Quantos pontos de fuga caracterizam esta projecção perspectiva? Justifique a resposta e apresente uma vista do cubo em que estes pontos de fuga estejam visíveis.



Na projecção perspectiva, um qualquer conjunto de linhas paralelas que não sejam paralelas ao plano de projecção converge num ponto de fuga. Neste caso, o cubo apresenta 3 conjuntos de linhas paralelas independentes que não são paralelas ao plano de projecção logo existem 3 pontos de fuga. Como se pode verificar o plano de projecção intersecta os três eixos e não é paralelo a nenhuma das faces do cubo.

A figura apresenta a localização destes pontos de fuga na imagem do cubo.



2. Apresente 2 (dois) tipos de projecções paralelas e, para cada uma delas, caracterize a relação entre a normal ao plano de projecção (NPP) e os eixos do mundo assim como a relação entre a normal ao plano de projecção (NPP) e a direcção de projecção (DP)

Projecções Ortogonais: NPP paralela a um dos eixos e DP paralela a NPP.

Projecção Axiometrica: NPP não é paralela a qualquer um dos eixos e DP é perpendicular a NPP.

Projecção Obliqua: NPP paralela a um eixo e DP nem é paralela nem perpendicular a NPP.

3. Aplique o algoritmo de Cohen-Sutherland ao recorte do segmento de recta AB com coordenadas (1, -1) e (1, 3) pelo rectângulo de recorte com coordenadas do vértice inferior esquerdo (0, 0) e do vértice superior direito (2, 1), calculando os códigos e descrevendo o critério de escolha das arestas de recorte.

Em primeiro lugar calculam-se os códigos dos vértices A e B os quais são, respectivamente, 0001 e 0010, convencionando que a sequência de bits corresponde aos limites x máximo, x mínimo, y máximo e y mínimo.

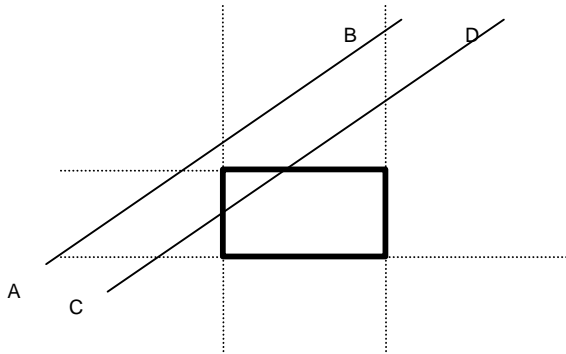
Efectuando a operação de AND bit a bit dos dois códigos, obtém-se o valor 0, pelo que será necessário subdividir o segmento de recta no ponto de intersecção com a recta sobre a qual assenta uma das arestas do rectângulo de recorte.

O bit a 1 do primeiro vértice, A, indica que se deve realizar o recorte contra o limite inferior do rectângulo de recorte. Daqui resulta, como ponto de intersecção, o vértice A' (1, 0) que tem como código o valor 0000. Efectuando-se a operação de AND deste ponto com B, continua a ser necessário efectuar a subdivisão do segmento de recta A'B, através do cálculo de uma nova intersecção com a recta sobre a qual assenta uma das restantes arestas do rectângulo de recorte.

Como o 1º vértice, A' agora tem código 0000 (não tem qualquer bit a 1) utiliza-se o 1º bit a 1 do segundo vértice, B, para escolher a aresta com a qual será efectuado o cálculo do ponto de intersecção. Neste caso é o limite superior do rectângulo de recorte e daqui resulta o vértice B' (1, 2) o qual tem, como código 0000.

Assim obtém-se um segmento de recta recortado, $A'B'$, com ambos os vértices com código 0000, o qual é virtualmente aceite.

4. Considere o algoritmo de Cyrus-Beck aplicado aos dois segmentos de recta da figura. Classifique os tipos de pontos de cada intersecção com as rectas que suportam cada uma das arestas do rectângulo de recorte e utilize essa informação para explicar por que é que o segmento de recta AB é rejeitado e o segmento CD é aceite, após recorte.



Consideraremos que, em AB, o valor do parâmetro t cresce de 0 para 1 de A para B e, em CD, de C para D. Considerando então a recta inferior do rectângulo de recorte, como o início de ambos os segmentos se encontram no subespaço inferior, ambas as intersecções dos segmentos com esta recta são pontos classificados como PE, Potencial Entrada.

Considerando a recta superior, ambos os pontos de intersecção da recta com os segmentos são PS, Potencial Saída. Para a aresta esquerda ambos os pontos são PE, Potencial Entrada, e, para a aresta direita, ambos os pontos são PS, Potencial Saída.

Assim resulta que as intersecções do segmento de recta AB, ordenados pelo parâmetro t , serão, PE, PS, PE, PS e as do segmento CD são PE, PE, PS, PS.

Como no primeiro caso o t maior entre os de PE é maior que o menor t dos PS, teremos a situação em que o segmento de recta é rejeitado. Para o segmento CD, o maior t dos PE é inferior ao menor t dos PS, devendo o segmento ser aceite entre estes dois pontos.

III

1. A representação de segmentos de recta num dispositivo de quadrículas requer a utilização de algoritmos de rasterização. Enumere três algoritmos para realizar a rasterização, indicando, resumidamente, em que medida estes três algoritmos diferem entre si.

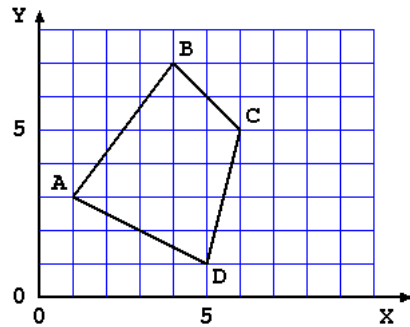
1 - Algoritmo imediato

2 - Algoritmo incremental básico

3 – Algoritmo de Bresenham

Os três algoritmos diferem principalmente na precisão dos resultados e na velocidade de execução (carga computacional). Estas diferenças devem-se ao tipo de aritmética usada. Os algoritmos imediato e o incremental usam aritmética real e arredondamentos, enquanto o algoritmo de Bresenham usa aritmética inteira.

2. Considere o polígono da figura seguinte.



- Assumindo que está a usar o algoritmo de Preenchimento por Varrimento (scan-line), preencha a Tabela de Lados inicial.
- Apresente agora a Tabela de Lados Activos em duas situações diferentes, uma assumindo que a linha de varrimento está na posição $y=3$ e outra em $y=6$.

a) A tabela de lados tem as seguintes entradas não nulas:

$$TL[1] = [(AD, Y_{max}=3, X(Y_{min})=5, 1/m = -2), (CD, Y_{max}=5, X(Y_{min})=5, 1/m = 1/4)]$$

$$TL[3] = [(AB, Y_{max}=7, X(Y_{min})=1, 1/m = 3/4)]$$

$$TL[5] = [(BC, Y_{max}=7, X(Y_{min})=6, 1/m = -1)]$$

$$b) TLA[3] = [(AB, Y_{max}=7, X=1, 1/m = 3/4), (CD, Y_{max}=5, X=5, 1/m = 1/4)]$$

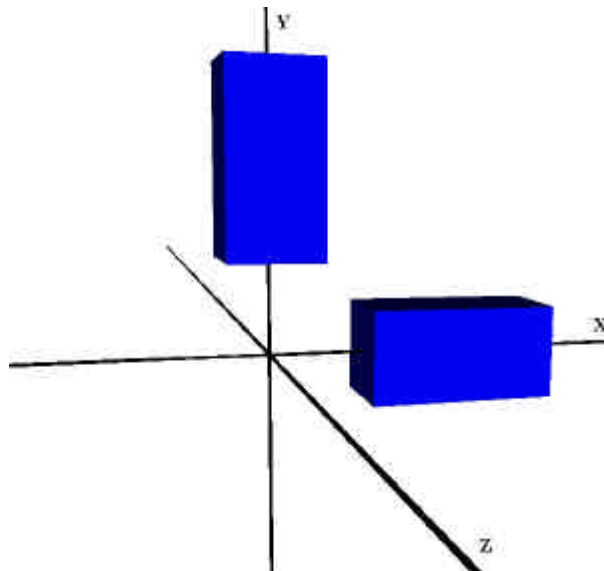
$$TLA[6] = [(AB, Y_{max}=7, X=13/4, 1/m = 3/4), (BC, Y_{max}=7, X=5, 1/m = -1)]$$

3. Defina o nó VRML PointLight e descreva 4 dos seus campos.

O nó VRML PointLight é uma luz pontual que irradia em todas as direcções a partir de um ponto, que é a sua localização no espaço. Os campos do nó PointLight são:

- *attenuation*: atenuação da luz com a distância;
- *ambientIntensity*: a contribuição para a luz ambiente geral da cena;
- *color*: a cor da luz emitida;
- *intensity*: o brilho da luz (intensidade da luz emitida);
- *location*: localização da luz no mundo;
- *on*: liga e desliga a luz;
- *radius*: distância máxima afectada pela luz.

4. Considere o nó VRML Transform e a imagem seguinte:



Quais das seguintes alternativas produz a imagem apresentada? Justifique.

Alternativa A	Alternativa B
<pre> DEF TransfA Transform { translation 2 0 0 children [DEF Caixa Shape { geometry Box { size 2 1 1 } appearance Appearance { material Material { diffuseColor 0 0 1 } } }] } Transform { rotation 0 0 1 1.57 children USE TransfA } </pre>	<pre> DEF TransfB Transform { translation 2 0 0 rotation 0 0 1 1.57 children [DEF Caixa Shape { geometry Box { size 2 1 1 } appearance Appearance { material Material { diffuseColor 0 0 1 } } }] } Transform { translation 0 2 0 children USE Caixa } </pre>

A alternativa A define um nó (TransfA) em que, a um paralelepípedo centrado na origem e orientado segundo o eixo dos XX, é aplicada uma translação segundo X de 2 unidades. A seguir, uma cópia deste nó é rodada de $+90^\circ$ em torno do eixo dos ZZ, o que corresponde à figura.

A alternativa B define um nó (Caixa) contendo um paralelepípedo centrado na origem e com o maior eixo segundo X. A este nó é primeiro aplicada uma rotação de $+90^\circ$ e só depois uma translação de 2 unidades segundo X, porque a ordem das transformações de um único nó Transform é esta. O paralelepípedo fica na posição pretendida encontra-se

na posição pretendida mas com o eixo principal segundo Z, o que não corresponde à figura.

Assim, a alternativa correcta é a alternativa A.

Cotação

	I	II	III
1.	1,5	2	1
2.	1,5	1,5	2
3.	1,5	2	1,5
4.	2	1,5	2
Total	6,5	7	6,5