



1º Teste de Computação Gráfica 2º Ano

Licenciatura em Eng. Informática e de Computadores

Prof. responsável – Mário Rui Gomes

2 de Maio de 2000

Nº _____ Nome: _____ Sala: _____

Responda às questões seguintes justificando adequadamente todas as respostas. Se necessário utilize o verso da respectiva folha.

I

1. Descreva a diferença fundamental entre a norma **GKS** e a norma **PHIGS**.

A norma ISO, Graphic Kernel System especifica a realização de um núcleo gráfico 2D enquanto que a norma ISO PHIGS, Programmer's Hierarchical Graphics Systems especifica a realização de um modelador e visualizador 3D. Ambas definem Interfaces Aplicacionais de Programação, API a serem usadas no contexto de várias linguagens de programação.

2. Considere que está a utilizar uma unidade de visualização gráfica com resolução de 1200 x 1200. O que acontece se substituir essa unidade por um ecrã com uma resolução de 960 x 960. Como se trata este problema no pipeline de visualização?

Ao substituir o ecrã vai só conseguir ver os 960x960 dos 1200x1200 quadrículas da imagem. Para resolver este problema existe um andar no pipeline de visualização que produz uma representação da cena em *normalized device coordinates*.

3. Descreva os fundamentos tecnológicos que estão na base do sucesso dos primeiros terminais Gráficos Tektronix.

Nos primeiros terminais gráficos Tektronix designados por DVST's (Direct View Storage Tubes) a imagem é armazenada no fósforo do ecrã o qual tem um tempo de decaimento muito lento, continuando a emitir luz ao longo de períodos de tempo prolongados.

II

1. Determine as coordenadas homogéneas do ponto $[-1 \ 2]$ do espaço 2D quando $w=3$.

Ao ponto do espaço 2D de coordenadas $[-1 \ 2]$ corresponde o ponto $[-1 \ 2 \ 1]$ do espaço homogéneo. Quando $w=3$ ter-se-á então $[-3 \ 6 \ 3]$.

2. Uma transformação composta consiste numa translação seguida de uma transformação de escala uniforme. Será que a ordem destas duas transformações é comutativa? Demonstre a sua afirmação.

Em geral, as transformações lineares não são comutativas excepto nos casos de translação+translação, escala+escala e rotação+rotação (este última só é comutativa em 2D). Como a transformação composta proposta não pertence a nenhum destes tipos, então as transformações elementares em causa (escala e translação) não são comutativas.

Demonstração alternativa que pode ser feita em 2D ou 3D, mas sempre no espaço homogêneo dado que uma das transformações elementares (translação) não é uma transformação linear.

Em 2D

Translação seguida de escala
$$\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f & 0 & af \\ 0 & f & bf \\ 0 & 0 & 1 \end{bmatrix}$$

Escala seguida de translação
$$\begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f & 0 & a \\ 0 & f & b \\ 0 & 0 & 1 \end{bmatrix}$$

III

1. Explique a função desempenhada e a utilidade dos parâmetros da câmara virtual designados por "planos de recorte".

Os planos de recorte da câmara virtual destinam-se a delimitar em profundidade (Z) que parte da cena é visível. É assim possível retirar objectos entre o centro de projecção e o plano anterior de recorte e todos os objectos existentes para lá do plano posterior de recorte.

2. A transformação de uma cena do sistema de coordenadas do mundo para o sistema de coordenadas da câmara virtual é realizada de acordo com determinados parâmetros da câmara virtual. Que parâmetros são estes e qual é a sua função?

Posição e orientação da câmara virtual. À posição corresponde uma translação da origem do espaço do mundo para a posição da câmara virtual (centro de projecção) enquanto a orientação da câmara corresponde a uma rotação.

IV

1. As transformações de projecção perspectiva mantêm ou não a amplitude dos ângulos? Justifique.

As transformações projectivas não preservam necessariamente o paralelismo das linhas, logo não podem necessariamente preservar os ângulos entre elas. A excepção são os ângulos entre linhas existentes em planos paralelos ao plano de projecção.

2. Determine as coordenadas x, y dos pontos projectados correspondentes aos extremos do segmento de recta que une os pontos [1 2 3] e [4 -5 2] (num referencial com origem no centro de projecção) quando o plano de projecção dista de 6 (seis) unidades do centro de projecção.

A matriz de projecção genérica é

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}, \text{ que no caso presente com } d=6 \text{ é } M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/6 & 0 \end{bmatrix}$$

aplicando-a então aos pontos A e B

$$A' = M \times A = M \times \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1/2 \end{bmatrix} \quad e \quad B' = M \times b = M \times \begin{bmatrix} 4 \\ -5 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ -5 \\ 0 \\ 1/3 \end{bmatrix}$$

cuja passagem do espaço homogêneo para coordenadas do plano de projecção leva a

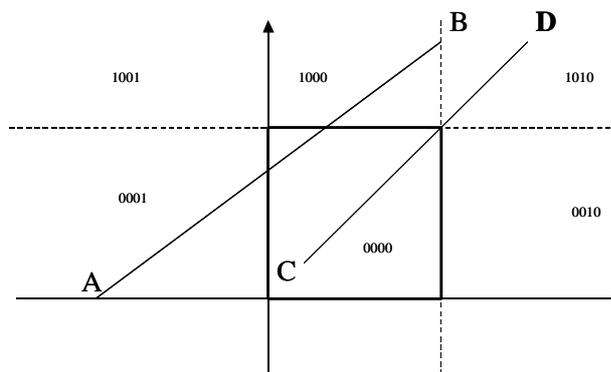
$$A'' = \begin{bmatrix} 2 \\ 4 \end{bmatrix} \text{ e } B'' = \begin{bmatrix} 12 \\ -15 \end{bmatrix}$$

V

1. O que se entende por Base de Dados de Objectos Activos? Qual a importância desse conceito no contexto de um pipeline de visualização.

O conceito de Bases de Dados de Objectos Activos designa os objectos que vão sendo produzidos ou seleccionados por cada andar do pipeline de visualização. A importância deste conceito é muito importante uma vez que após a execução de cada andar o número de objectos activos deve ir diminuindo. Idealmente na fase de *rasterização* só devem existir nessa base de dados os objectos que irão ser visualizados.

2. Aplique o algoritmo de Cohen-Suntherland de recorte sobre uma janela com vértices cujas coordenadas são [0, 0], [1, 0], [1, 1], [0, 1] aos seguintes segmentos de recta: [-1, 0] ->



[1, 1.5] e [0.2, 0.2] -> [1.5, 1.5].

A janela tem a forma de um quadrado com o vértice inferior esquerdo na origem do referências. Começando pelo 2º segmento de recta (CD na figura) codificando C e D e efectuando o & dos códigos obtemos o valor 0000 mas o código de D é diferente de 0000 pelo que temos que calcular o ponto de intersecção com a aresta superior da janela. Obtemos um valor do parâmetro tal que o ponto de intersecção, com coordenadas [1, 1], tem o código 0000 aplicando a operação & entre esse ponto e C obtemos 0000 e ambos com código 0000 pelo que todo o novo segmento inferior se encontra dentro da janela. Aplicando a mesma operação entre o ponto de intersecção e o ponto D obtemos um resultado 0010 pelo que todo o novo segmento inferior é rejeitado.

Aplicando o mesmo algoritmo a AB calculamos primeiro a intersecção com a aresta superior do quadrado e o segmento que resulta calculamos a intersecção com a aresta da esquerda do quadrado, resultando o pequeno segmento do interior da janela. A escolha de qual a aresta a com a qual se deverá efectuar o cálculo de intersecção depende da análise dos códigos dos vértices do segmento de recta. Assim na primeira intersecção é usado o código de B (1000) pelo que a intersecção deve ser efectuada com a aresta superior da janela. No segundo caso é usado o código de A (0001), uma vez que o código do segundo novo vértice já é 0000, pelo que é usada a aresta da esquerda da janela de recorte.

VI

1. O que se entende por *rasterização*? Considerando o exemplo do algoritmo de Bresenham como efectuaria a *rasterização* de uma circunferência?

Por *rasterização* entende-se o processo que transforma uma representação vectorial de um objecto gráfico num conjunto de quadrículas.

Induzindo o modo como adaptaria o algoritmo de Bresenham para rasterizar circunferências, em primeiro lugar identificaria a simetria dessa forma geométrica e concluiria que, tal como acontece com o segmento de recta bastaria aplicar o algoritmo a um octante. Seguidamente estudaria o

tipo de declives presentes no octante mais favorável (com declive entre 0 e 1) da circunferência e aplicaria o algoritmo de Bresenham mas utilizando a fórmula da circunferência.

2. Descreva o Algoritmo Incremental Básico e justifique a principal razão para ele ter deixado de ser adoptado.

O algoritmo Incremental Básico ou *Digital Differential Analyzer (DDA)* baseia-se no cálculo de cada quadrícula fazendo variar X de 1 ($X_{i+1} = X_i + 1$) e calculando $Y_{i+1} = \text{Round}(Y_i + m)$ o que provoca acumulação de erros pelo que deixou de ser adoptada. A este facto deve adicionar-se a melhor eficácia do algoritmo que o substituiu, o algoritmo de Bresenham.

VII

1. Explique a diferença entre os campos "emissiveColor" e "diffuseColor". A que tipo de nó VRML pertencem estes campos?

O campo "emissiveColor" especifica a cor da luz emitida por uma superfície (objecto) enquanto o campo "diffuseColor" especifica a cor da luz reflectida por uma superfície (objecto).

2. Escreva o código VRML para definir uma pirâmide quadrangular de lado unitário e com a altura de 2 (duas) unidades em que a base seja amarela e as faces laterais sejam de cor azul, empregando um único nó.

```
Shape { geometry IndexedFaceSet
  { coord Coordinate { point[ .5 .5 0, .5 -.5 0,
    -.5 -.5 0, -.5 .5 0,
    0 0 2 ]
  }
  coordIndex[ 0 1 2 3 -1,
    0 1 4 -1, 1 2 4 -1, 2 3 4 -1, 3 0 4 -1]
  color Color{ color[ 1 1 0, 0 0 1 ] }
  colorIndex [ 0 0 0 0 -1,
    1 1 1 -1, 1 1 1 -1, 1 1 1 -1, 1 1 1 -1]
  }
}
```

Cotação

	I	II	III	IV	V	VI	VII
1.	1	1	1	1	1	2	1
2.	1	2	1	2	2	1	2
3.	1						
Total	3	3	2	3	3	3	3