



1º Exame

Computação Gráfica

LEIC-A/T
Prof. Mário Rui Gomes
Prof. João Pereira
Prof. Daniel Gonçalves
5 de Julho 2008

Nº _____ Nome: _____ B

O exame tem a duração de 2 horas, tolerância incluída. Responda às questões unicamente nestas duas primeiras folhas, justificando adequadamente as respostas de desenvolvimento. Só estas duas folhas deverão ser entregues, e como tal, serão as únicas avaliadas. As restantes folhas podem ser utilizadas como folhas de rascunho. Uma resposta errada nas perguntas de escolha múltipla desconta 1/3 da cotação.

Respostas:

1. [1,5]

 2. [1]

3. [1]

--

 4. [0,5]

--	--

5. [1]

 6. [1,5]

7.a) [0,5]

 7.b) [1]

8. [1]

--

Nº

Nome: _____ B

9. [1,5]

10.a) [0,5]

A	B	C	D	E	F

10.b) [0,5]

10.c) [0,5]

10.d) [0,5]

11.a) [0,5]

11.b) [0,5]

11.c) [0,5]

11.d) [0,5]

Xmin	Xmax

12. [1,5]

13.a) [0,5]

13.b) [0,5]

13.c) [0,5]

14.a) [0,5]

14.b) [0,5]

14.c) [0,5]

15. [1]

--



1. Considere a representação *raster* de uma imagem:

- A. Não se observa o efeito de *aliasing*.
- B. A quantidade de memória necessária para armazenar a imagem é independente da complexidade da cena.
- C. É uma representação contínua.
- D. É particularmente adequada para representar redes de estradas.

B

2. Calcule as dimensões da janela de visualização do *frustum* (janela localizada no plano *near*), sabendo que foi introduzida a linha de comando:

```
gluPerspective(90.0f, 1.33f, 15.0, 120.0)
```

Nota: assinatura da função `gluPerspective`:

```
void gluPerspective(Gldouble fovy, Gldouble aspect, Gldouble near, Gldouble far)
```

```
tg (30°) = 0.577      tg (45°) = 1.0      tg (60°) = 1.732
```

O ângulo fornecido como primeiro argumento respeita a abertura vertical em graus.

*A relação existente entre a semi-altura e a distância à janela de visualização (definida pelo argumento *near*) é a tangente da semi-abertura vertical ou seja:*

*top = tg (90/2) * 15 e como bottom = -top significa que a altura da janela vale 30*tg(45°);*

*Dado que a relação de aspecto é 1,33 temos que right = 1,33*top e left = -right. Assim a largura da janela vale 1,33*30*tg(45)*

tg (45°) = 1, 0

Resposta:

Altura = 30

*Largura = 1,33*30=39,9*

3. Uma unidade lógica de entrada de dados a funcionar em modo de funcionamento por Acontecimento

- E. São usados dois processos, de Disparo e de Medida.
- F. O processo de Disparo pede ao processo de Medida o que originou o disparo.
- G. A comunicação entre os dois processos não é efectuada através de uma Fila de Espera.
- H. Não é necessário usar o processo de Disparo

A

4. Considere o vector v que passa pelos pontos $(1, 2, 2)$ e $(-2, 3, -2)$.

Calcule as componentes do vector v em coordenadas homogêneas.

Componentes cartesianas de v $[-3 \ 1 \ -4]$ logo em coord. homogêneas $[-3 \ 1 \ -4 \ 0]$

5. Considere a seguinte transformação de translação descrita no espaço cartesiano:

$$\begin{aligned} X' &= x + 3 \\ Y' &= y - 5 \\ Z' &= z + 8 \end{aligned}$$

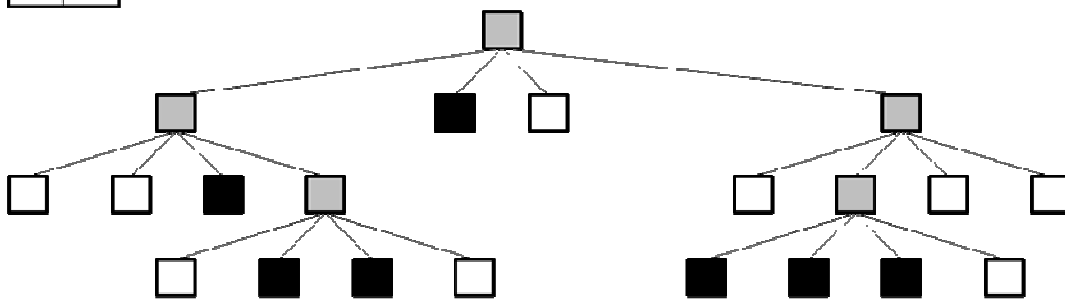
Determine a matriz de translação em coordenadas homogêneas.

RESPOSTA:

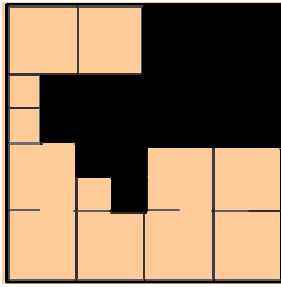
$$\begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & -5 \\ 0 & 0 & 1 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

6. Assumindo que os quadrantes aparecem representados na árvore, da esquerda para a direita, na ordem apresentada abaixo, desenhe a área representada pela *quadtree* seguinte:

1	2
4	3



Resposta:



7. Suponha que escreveu a seguinte linha de código:

```
gluLookAt(0.0, 0.0, -5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
```

Nota: assinatura da função `gluLookAt`:

```
gluLookAt(eyex, eyey, eyez,
           centerx, centery, centerz,
           upx, upy, upz)
```

a) Indique a matriz de translação em coordenadas homogêneas referente à Transformação de Visualização realizada internamente pelo pipeline OpenGL de modo a colocar os objectos da cena no referencial da câmara.

b) Calcule a matriz de rotação em coordenadas homogêneas realizada internamente pelo pipeline OpenGL na Transformação de Visualização.

A transformação de visualização implica uma mudança de referencial do Mundo para o referencial da Câmara. Assim, em termos de mudança de referencial, esta transformação equivale a fazer coincidir o referencial da câmara com o referencial do Mundo. Isso implica efectuar uma translação seguida de uma rotação. A matriz de translação é definida pelo vector de translacção $[-VRPx \ -VRPy \ -VRPz]$. A matriz de rotação é dada em termos de linhas, respectivamente, pelas componentes dos versores u, v e $-n$.

VRP (1º argumento) é o (0 0 -5) e o observador está olhar para (0, 0, 0) (2º argumento) logo VPN $[0 \ 0 \ 5]$ => $n \ [0 \ 0 \ +1]$

View-up $[0 \ 1 \ 0]$ (3º argumento) => $v \ [0 \ 1 \ 0]$ e portanto $u \ [-1 \ 0 \ 0]$ (referencial mão esquerda: $u = n \times v$)

Internamente efectuam-se uma translação seguida de uma rotação:

RESPOSTA:

$$\text{Translacção} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & +5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Rotação} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

8. Indique, justificando, quais as transformações necessárias para transformar o *frustum* genérico no volume canónico ortogonal.

Duas transformações de escala para converter o volume de visualização genérico no volume canónico perspectivo. Primeiro a escala que faz com que a inclinação dos planos laterais do frustum genérico tenham um declive unitário, e depois uma outra escala que faça com que o plano posterior $z=F$ passe para $z=1$.

Poder-se-á ainda especificar a Transformação Perspectiva de modo a converter o volume canónico perspectivo num volume canónico ortogonal com o objectivo de simplificar as operações de recorte e de visibilidade.

9. Considere uma cena muito simples constituída por uma fonte de luz e uma superfície plana. As características de iluminação e de reflexão desta cena são descritas pelas seguintes funções OpenGL:

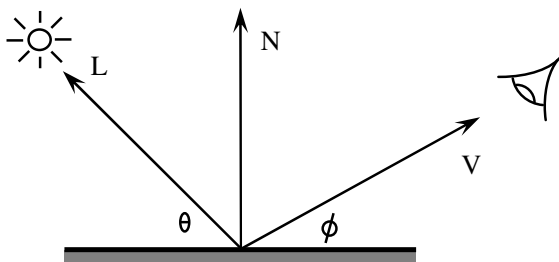
```
GLfloat ambient[] = { 0.3, 0.0, 0.0, 1.0 };
GLfloat diffuse[] = { 1.0, 0.5, 0.5, 1.0 };
GLfloat specular[] = { 1.0, 1.0, 1.0, 1.0 };
```

```
glLightfv(GL_LIGHT0, GL_AMBIENT, ambient);
glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse);
glLightfv(GL_LIGHT0, GL_SPECULAR, specular);
```

```
GLfloat mat_ambient[] = { 1.0, 1.0, 1.0, 1.0 };
GLfloat mat_diffuse[] = { 0.6, 0.1, 1.0, 1.0 };
GLfloat mat_specular[] = { 1.0, 1.0, 1.0, 1.0 };
```

```
glMaterialfv (GL_FRONT_AND_BACK, GL_AMBIENT, mat_ambient);
glMaterialfv (GL_FRONT_AND_BACK, GL_DIFFUSE, mat_diffuse);
glMaterialfv (GL_FRONT_AND_BACK, GL_SPECULAR, mat_specular);
glMaterialf (GL_FRONT, GL_SHININESS, 10.0);
```

Considere ainda que a fonte de luz está apontada à superfície segundo o ângulo θ de 60° e que o observador olha para a superfície segundo um ângulo ϕ de 15° .



Calcule as componentes de cor no ponto da superfície para onde a câmara aponta segundo o modelo de reflexão de Phong.

$$\cos 30^\circ = 0,866$$

$$\cos 45^\circ = 0,707$$

$$\cos 60^\circ = 0,5$$

$VertexColor = ambient\ light * ambient\ material + (\max\{L \cdot N, 0\}) * diffuse\ light * diffuse\ material + (\max\{R \cdot V, 0\}) * exp\ shininess * specular\ light * specular\ material$

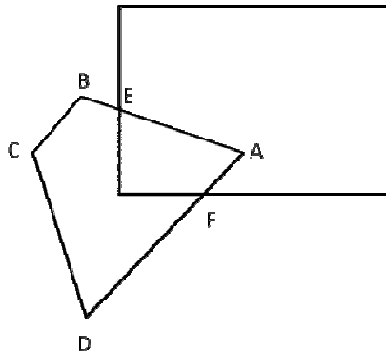
Versão A:

$$R: 0,3 * 1,0 + 1,0 * 0,6 * \cos 30^\circ + 1,0 * 1,0 * (\cos 45^\circ) * exp 10$$

$$G: 0,0 * 1,0 + 0,5 * 0,1 * \cos 30^\circ + 1,0 * 1,0 * (\cos 45^\circ) * exp 10$$

$$B: 0,0 * 1,0 + 0,5 * 1,0 * \cos 30^\circ + 1,0 * 1,0 * (\cos 45^\circ) * exp 10$$

10. Vai efectuar o recorte do polígono trapezoidal abaixo face ao rectângulo, usando o algoritmo de Cohen-Sutherland:



a) Indique quais os outcodes dos vários pontos indicados na figura?

$$OC(A) = 0000$$

$$OC(B) = 0001$$

$$OC(C) = 0001$$

$$OC(D) = 0101$$

$$OC(E) = 0000$$

$$OC(F) = 0000$$

b) Quais os segmentos trivialmente aceites? Que condição teve que verificar para o saber?

$$\text{Nenhum, } OC1 = OC2 = 0$$

Também aceitámos [AE] [AF]

c) Quais os segmentos trivialmente rejeitados? Que condição teve que verificar para o saber?

$[CB], [CD], OC1 \text{ \& } OC2 \neq 0$

Também aceitámos $[CB], [CD], [BE], [DF]$

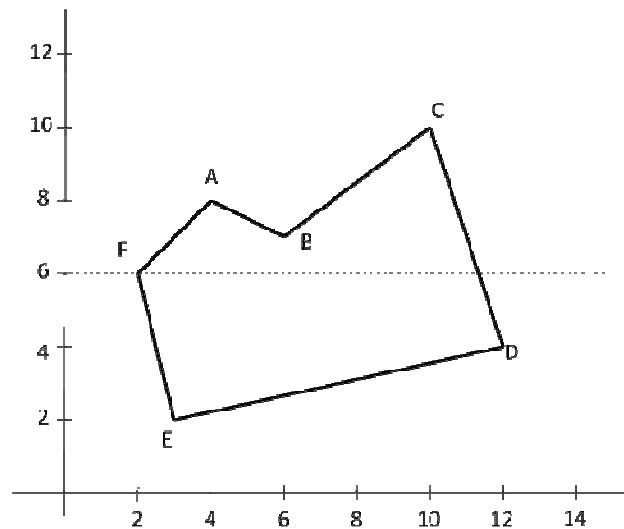
d) Como efectua o recorte da aresta AB? Escreva, um por linha, os passos seguidos.

Subdividir $[AB]$ em $[AE]$ e $[EB]$

Rejeitar $[EB]$

Aceitar $[AE]$ ($OC(E)=OC(A)=0$)

11. Considere o seguinte polígono, que quer discretizar e preencher usando o algoritmo de *scan-line*:



a) Qual o conteúdo da Tabela de Arestas Activas na linha 6 (basta indicar os nomes das arestas)

$([FA], [DC])$

b) Quais os valores guardados na Tabela de Arestas para caracterizar a aresta $[EF]$?

$Y_{max}=6$

$X = 3$

$1/m = 1/(4/-1) = -1/4$

- c) Ao passar de uma *scanline* para a próxima, qual o incremento para x ao longo da aresta [AB]?

$$dx=1/m=1/(2/-1) = -2$$

- d) Quais o menor e maior x preenchidos na linha 6?

Será entre a aresta [FA] e a aresta [DC]. Sabemos que o x de [FA] na linha 6 é 2. Para saber o x de [DC] na linha 6, temos que fazer algumas contas. Há várias formas de fazer, mas usando o algoritmo incremental, sabemos que para [DC], $1/m = -2/6 = -1/3$. Como a aresta começa na linha 4, na linha 6 já subimos duas, pelo que temos que somar $2/m = -2/3$, tornando o x de [DC] na linha 6 = $12-2/3 \approx 11.33$. Como os valores são sempre arredondados para dentro dos polígonos, a resposta final é:

$$X_{min}=2, X_{max}= 11.$$

12. O algoritmo Z-Buffer permite:

- A. Calcular qual o objecto que se encontra mais próximo da câmara e que ocupa uma dada quadrícula.
- B. Calcular qual o objecto mais próximo da câmara.
- C. Calcular as distâncias de cada objecto a uma fonte de luz.
- D. Calcula os objectos invisíveis de uma cena.

A

13. Uma imagem tem *pixeis* da cor (32, 21, 128) (RGB).

- a) Dispondo apenas de 21 bits para representar cada pixel, qual das seguintes situações é preferível:

- A. Usamos um número maior de bits para a componente azul
- B. Usamos um igual número de bits para cada componente
- C. Usamos um número maior de bits para as componentes vermelha e azul
- D. Usamos um número maior de bits para as componentes vermelha e verde

D ou B

- b) Qual a representação dessa cor no espaço de cor CMY?

$$(223,234,127)$$

c) Se souber que, na imagem, não tem mais do que 240 cores diferentes, qual é outra possibilidade acertada para poupar espaço de armazenamento?

- A. Usar 3 bits por componente de cor em cada pixel
- B. Reduzir as dimensões da imagem
- C. Usar uma tabela de cores com 8 bits por cor.
- D. Converter a imagem para níveis de cinzento

C ou A

14. Considere uma cena com 400 polígonos a ser desenhada pelo método de *Ray-tracing* num *viewport* com uma resolução de 1024 por 1024 quadriculas.

a) Se não for empregue nenhuma técnica de optimização, determine o número máximo de testes de intersecção entre os objectos da cena e os raios primários que o algoritmo de *Ray-tracing* em questão poderia gerar.

$$1024 \times 1024 \times 400 =$$

b) Considere que a cena possui 400 objectos opacos e em que apenas um deles é reflector. Calcule o número máximo de testes de intersecção entre os objectos da cena e os raios primários e secundários que o algoritmo de ray-tracing em questão poderia gerar, tendo em atenção que o nível máximo de profundidade para a árvore de raios é de 2.

$$1024 \times 1024 \times 400 + 1024 \times 1024 \times 399 =$$

c) Considere agora que se utilizou uma estratégia de esferas envolventes em que cada uma dessas esferas contém 10 objectos. Determine o número total de possíveis intersecções de raios primários com os volumes envolventes e os objectos.

Usaram-se 40 esferas envolventes . Logo

$$1024 * 1024 * (40 + 10) =$$

15. O que se entende por Factor de Forma? Qual o seu significado geométrico? Qual a aplicabilidade do cálculo de Factores de Forma quando ocorre emissão e reflexão de energia especular?

O denominada factor de forma, representa a fracção da radiosidade total que abandona um superfície j e atinge uma superfície i. Como estamos a tratar de

superfícies com reflexão e emissão difusas, a radiosidade proveniente de uma tal superfície é uniforme. Os factores de forma assumem um significado geométrico preciso correspondente à forma como cada superfície “vê” outra superfície. Isto corresponde ainda ao ângulo sólido do campo de visão da superfície que “vê” ocupado pela superfície que é “vista”. Os factores de forma dependem unicamente da geometria das superfícies e da sua disposição espacial relativa.