INSTITUTO SUPERIOR TÉCNICO

Computação Gráfica

Licenciatura em Engenharia Informática e de Computadores
Alameda / Taguspark

Exame de Recurso
29 de Junho de 2011

O exame tem a duração de **2h30**, tolerância incluída. Responda às questões no espaço reservado para o efeito, **justificando** adequadamente as respostas. O verso das páginas pode ser utilizado como folha de rascunho ou para responder a questões dessa página, quando necessário. Durante o exame apenas é permitido o uso de caneta. Não é permitido o uso de calculadoras, telemóveis ou folhas em branco para rascunho. Uma resposta errada nas perguntas de escolha múltipla desconta 1/2 da cotação da respectiva questão. Identifique **todas** as folhas do exame.

Algoritmo de *Cyrus-Beck*: tabela de cálculo para polígonos rectangulares

E_i	N_i	P_{Ei}	$P_0 - P_{Ei}$	t
$x_i=x_{min}$	$[-1\ 0]^T$	(x_{min}, y)	$(x_0 - x_{min}, y_0 - y)$	$-(x_0 - x_{min})/(x_1 - x_0)$
$x_i=x_{max}$	$[1\ 0]^T$	(x_{max}, y)	$(x_0 - x_{max}, y_0 - y)$	$-(x_0 - x_{max})/(x_1 - x_0)$
$y_i=y_{min}$	$[0\ -1]^T$	(x, y_{min})	$(x_0 - x, y_0 - y_{min})$	$-(y_0 - y_{min})/(y_1 - y_0)$
$y_i=y_{max}$	$[0\ 1]^T$	(x, y_{max})	$(x_0 - x, y_0 - y_{min})$	$-(y_0 - y_{max})/(y_1 - y_0)$

Identificação do Aluno	
Número:	Nome:

1. **[1.0v]** Considere o triângulo T={V1, V2, V3}, com V1=[0 2 3], V2=[3 3 3] e V3=[4 2 3]. Seguindo a regra da mão esquerda, indique o valor da normal ao triângulo T, n_T.

Sendo $a=[V_1\ V_2]$ e $b=[V_2\ V_3]$, temos que:

$$a \times b = \begin{bmatrix} y_a z_b - z_a y_b \\ z_a x_b - x_a z_b \\ x_a y_b - y_a x_b \end{bmatrix} = \begin{bmatrix} (3-2)(3-3) - (3-3)(2-3) \\ (3-3)(4-3) - (3-0)(3-3) \\ (3-0)(2-3) - (3-0)(4-3) \end{bmatrix}$$

$$n_T = \|a \times b\| = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$$

2. **[0.5v]** Tirou uma fotografia aos seus amigos com uma máquina fotográfica digital. Acha que o formato de imagem GIF é mais indicado para armazenar essa imagem num ficheiro que pretende enviar aos seus amigos via internet? Justifique a sua resposta.

O formato GIF não é o mais indicado para este efeito. A razão para tal prende-se com o limite do número de cores suportadas ser baixo e com o facto de o algoritmo de compressão usado neste formato não ser indicado para imagens de qualidade fotográfica.

Identificação do Aluno
Número:

3. [1.5v] Considere os seguintes comandos OpenGL:

```
gluOrtho2D(0, 2, 0, 3);
glViewport(50, 50, 400, 300);
```

a) [1.0v] Determine a matriz de mapeamento em coordenadas de viewport.

$$M = T_2 \cdot S \cdot T_1$$

Onde:

T1 é translação do canto inferior esquerdo da janela de visualização para a origem;

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & -x_{min} \\ 0 & 1 & 0 & -y_{min} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T2 é translação da origem para o canto inferior esquerdo do viewport;

$$T_2 = \begin{bmatrix} 1 & 0 & 0 & x_v \\ 0 & 1 & 0 & y_v \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 50 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

S é a escala que transforma a largura e altura da janela de visualização na largura e altura do viewport.

$$S = \begin{bmatrix} \frac{w_v}{w} & 0 & 0 & 0 \\ 0 & \frac{h_v}{h} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{400}{2} & 0 & 0 & 0 \\ 0 & \frac{300}{3} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 200 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Logo:

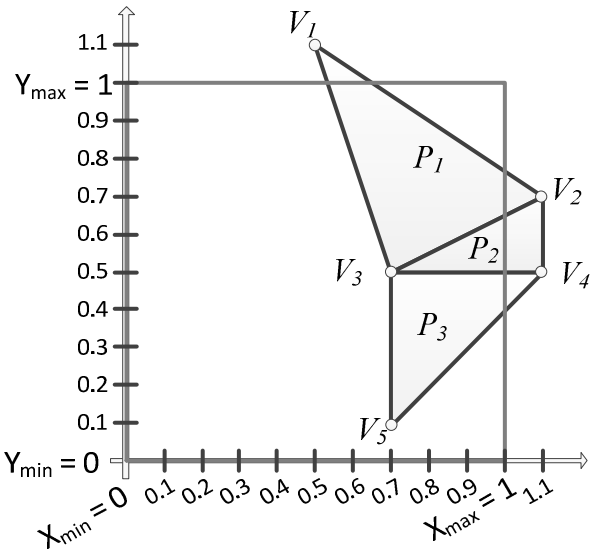
$$M = \begin{bmatrix} 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 50 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 200 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 200 & 0 & 0 & 50 \\ 0 & 100 & 0 & 50 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

b) [0.5v] Este conjunto de comandos realiza uma operação de stretching. Explique, quantificando, em que esta operação consiste.

Considerando que a relação de aspecto da janela de visualização é 2:3 e a do viewport é de 4:3. Assim sendo, a operação de stretching em causa consiste no alongamento vertical da vista, ficando os objectos com o dobro da largura que deveriam ter em relação à altura.

Identificação do Aluno
Número:

4. **[3.0v]** Considere o rectângulo de recorte e os polígonos $P_1=\{V_1, V_2, V_3\}$, $P_2=\{V_2, V_4, V_3\}$ e $P_3=\{V_3, V_4, V_5\}$ ilustrados na figura abaixo. Assuma que vai aplicar os algoritmos de recorte leccionados nas aulas teóricas.



a) **[0.5v]** Seguindo a metodologia usada no algoritmo de *Cohen-Sutherland*, indique os *outcodes* de cada um dos vértices. Explicite a convenção usada na determinação dos bits.

$OC(V_1) = 1\ 0\ 0\ 0$
$OC(V_2) = 0\ 0\ 1\ 0$
$OC(V_3) = 0\ 0\ 0\ 0$
$OC(V_4) = 0\ 0\ 1\ 0$
$OC(V_5) = 0\ 0\ 0\ 0$

bit 1 $\rightarrow Y > Y_{max}$
bit 2 $\rightarrow Y < Y_{min}$
bit 3 $\rightarrow X > X_{max}$
bit 4 $\rightarrow X < X_{min}$

b) **[0.5v]** Identifique que segmentos são trivialmente aceites, trivialmente rejeitados ou subdivididos, através da aplicação do algoritmo de *Cohen-Sutherland*.

Trivialmente aceite:
$[V_3\ V_5]$, pois $OC(V_3) = OC(V_5) = 0\ 0\ 0\ 0\ 0\ 0$

Trivialmente rejeitado:
$[V_2\ V_4]$, pois $OC(V_2)\ \&\ OC(V_4) \neq 1$

Subdivididos:
$[V_2\ V_3]$, $[V_4\ V_3]$, $[V_4\ V_5]$, $[V_1\ V_2]$ e $[V_3\ V1]$,
pois $OC(V_2)\ \&\ OC(V_3) = 0$, $OC(V_4)\ \&\ OC(V_3) = 0$, $OC(V_4)\ \&\ OC(V_5) = 0$,
$OC(V_1)\ \&\ OC(V_2) = 0$, $OC(V_3)\ \&\ OC(V_1) = 0$

Identificação do Aluno
Número:

c) **[1.5v]** Usando os algoritmos de Liang-Barsky e Cyrus-Beck, determine os valores de T_e e T_l para a aresta $[V_1\ V_2]$.

Determinação dos valores de T para a aresta $[V_1\ V_2]$:

$T_{left} = -(0.5 - 0.0)/(1.1 - 0.5) = -(0.5/0.6) = -5/6$

$T_{right} = -(0.5 - 1.0)/(1.1 - 0.5) = (0.5/0.6) = 5/6$

$T_{bottom} = -(1.1 - 0.0)/(0.7 - 1.1) = (1.1/0.4) = 11/4$

$T_{top} = -(1.1 - 1.0)/(0.7 - 1.1) = (0.1/0.6) = 1/4$

Ignoram-se T_{left} e T_{bottom} porque $T_{left} < 0$ e $T_{bottom} > 1$

Determinação de P_E e P_L:

$N_{right} \bullet [V_2 - V_1] > 0$, porque ângulo entre N_{top} e vector de V_1 a V_2 é menor que 90^0

$N_{top} \bullet [V_2 - V_1] < 0$, porque ângulo entre N_{top} e vector de V_1 a V_2 é maior que 90^0

$N_{right} \bullet [V_2 - V_1] > 0 \Rightarrow T_{right}$ é P_L

$N_{top} \bullet [V_2 - V_1] > 0 \Rightarrow T_{top}$ é P_E

Logo:

$T_e = 1/4$

$T_l = 5/6$

d) **[0.5v]** Descreva o recorte do polígono P_2 através da aplicação do algoritmo de Sutherland-Hodgman. Faça-o indicando as listas de vértices inicial, à entrada e saída de cada um dos passos e final. Em cada passo indique qual a aresta de recorte.

$\{ V_2, V_3, V_4 \} \rightarrow$ Left Clip (recorte pela aresta X=0)$\rightarrow \{ V_3, V_4, V_2 \} \rightarrow$

$\{ V_3, V_4, V_2 \} \rightarrow$ Top Clip (recorte pela aresta Y=1)$\rightarrow \{ V_4, V_2, V_3 \} \rightarrow$

$\{ V_4, V_2, V_3 \} \rightarrow$ Right Clip (recorte pela aresta X=1)$\rightarrow \{ V_3, I_1, I_2 \} \rightarrow$

$\{ V_3, I_1, I_2 \} \rightarrow$Bottom Clip (recorte pela aresta Y=0)$\rightarrow \{ I_1, I_2, V_3 \}$

Identificação do Aluno
Número:

5. **[2.0v]** Considere os seguintes excertos de código VRML:

```
IndexedFaceSet {
    coord Coordinate {
        point [
            1 -2 1, 1 -2 -1,
            -1 -2 -1, -1 -2 1,
            1 2 1, 1 2 -1,
            -1 2 -1, -1 2 1
        ]
    }
    coordIndex [
        0 1 2 3 -1,
            0 1 5 4 -1,
            1 2 6 5 -1,
            2 3 7 6 -1,
            3 0 4 7 -1,
            4 5 6 7
        ]
}
```

```
Extrusion {
    beginCap TRUE
    endCap TRUE
    crossSection [
            1 1,
            1 -1,
            -1 -1,
            -1 1,
            1 1
        ]
    spine [0 -2 0, 0 2 0]
}
```

a) **[1.0v]** Os dois excertos representam o mesmo objecto sólido 3D? Justifique.

Sim, ambos representam um paralelepípedo centrado na origem com 2 unidades de largura (eixo dos XX) e de profundidade (eixo dos ZZ) e 4 unidades de altura (eixo dos YY).

b) **[1.0v]** Indique, para cada um dos casos, qual a técnica de representação de malhas de polígonos utilizada.

O nó IndexedFaceSet é uma malha de polígonos representada por uma lista de vértices.

O nó Extrusion não é uma malha de polígonos. É uma extrusão ao longo de um caminho que é um tipo de representação por varrimento (sweep).

Identificação do Aluno
Número:

6. **[1.5v]** Considere uma projecção perspectiva cujo plano de visualização está a uma distância D = 200 da câmara. A janela de visualização tem largura W = 800 e altura H = 400.

a) **[0.5v]** Qual das seguintes utilizações da função **`gluPerspective()`** indicadas abaixo está correcta?

Nota: assinatura da função **`gluPerspective`**:

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

A: `gluPerspective(90.0f, 2.0f, 400.0f, 800.0f)`

B: `gluPerspective(90.0f, 0.5f, 200.0f, 400.0f)`

C: `gluPerspective(45.0f, 2.0f, 50.0f, 800.0f)`

D: `gluPerspective(90.0f, 2.0f, 200.0f, 400.0f)`

E: `gluPerspective(90.0f, 1.5f, 50.0f, 100.0f)`

F: Nenhuma das anteriores

A opção correcta é a D.

b) **[1.0v]** Justifique a sua escolha.

A relação de aspecto (aspect) é W/H = 800/400 = 2, o que invalida as opções B e E.

O plano de visualização deve estar entre o plano de recorte anterior (near) e o plano de recorte posterior (far), ou seja near <= D <= far. Isso invalida a opção A.

O ângulo de abertura vertical é dado por $tg(\Theta/2) = (H/2)/D \Leftrightarrow tg(\Theta/2) = 200/200 \Leftrightarrow \Theta/2 = tg^{-1}(1) \Leftrightarrow \Theta/2 = 45° \Leftrightarrow \Theta = 90°$. Isso invalida a opção C.

A opção D é válida..

Identificação do Aluno
Número:

7. **[1.0v]** Considere o seguinte código OpenGL que desenha um cubo e uma esfera vermelhas:

```
void draw() {
    glColor3f(1.0, 0.0, 0.0);
    glTranslatef(0.0, 1.0, 0.0);
    glPushMatrix();
    glTranslatef(5.0, 0.0, 0.0);
    glRotatef(45.0, 0.0, 1.0, 0.0);
    glutSolidCube(2.0);
    glPopMatrix();
    glutSolidSphere(2.0, 20, 20);
}
```

a) **[0.5v]** Qual é o produto de matrizes (no espaço homogéneo) que corresponde à transformação que é aplicada ao cubo?

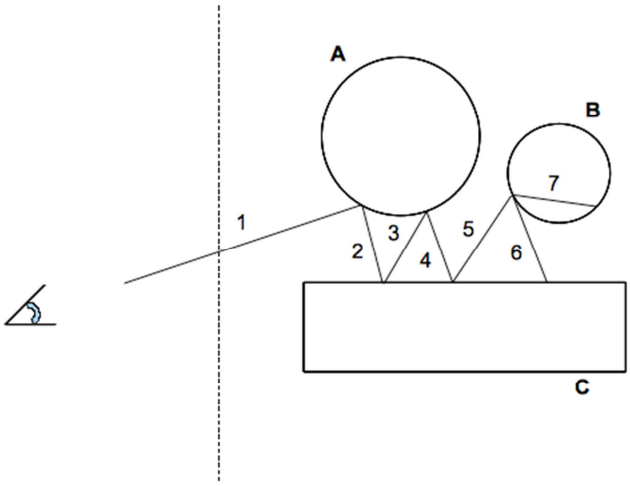
$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(45°) & \sin(45°) & \sin(45°) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(45°) & 0 & \cos(45°) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

b) **[0.5v]** A esfera é afectada pela transformação aplicada ao cubo? Justifique.

Não, só é afectada pela primeira translação. A segunda translação e a rotação estão dentro de um bloco PushMatrix/PopMatrix. Este bloco guarda o estado da matriz após a primeira translação e depois recupera-o imediatamente antes de desenhar a esfera que assim não é afectada pelas transformações posteriores à primeira translação.

Identificação do Aluno
Número:

8. **[1.5v]** Considere o seguinte diagrama que representa os raios traçados por um Ray Tracer para um determinado pixel.



a) **[0.5v]** Classifique o tipo de todos os raios identificados por números.

1 – Raio Primário
2, 3, 4, 5, 6 – Raios Reflectidos
7 – Raios Refractados

b) **[0.5v]** O que pode concluir sobre o limite máximo de intersecções? Justifique.

O limite máximo de intersecções é 6. Se fosse maior o algoritmo não teria sido interrompido: o objecto C teria reflectido o raio 6 e o objecto B teria reflectido e refractado o raio 7.

c) **[0.5v]** Assumindo que esta cena contém duas fontes de luz, quantos raios do tipo *shadow feeler* seriam calculados na determinação da cor do pixel?

Seriam calculados 14 raios shadow feeler. Em cada intersecção de um raio com um objecto, é emitido um raio shadow feeler por cada fonte de luz. Neste caso, como são detectadas 7 intersecções, são calculados 7*2=14 shadow feelers.

Identificação do Aluno
Número:

9. **[0.5v]** Identifique e descreva os três blocos fundamentais de uma arquitectura raster.

1. Processador Gráfico – responsável pela geração de imagem
2. Memória de imagem baseada em pixels
3. Controlador de refrescamento de video: responsável por ler a memória de imagem e converter o contéudo em sinal de video para que este seja enviado para o monitor

10. **[1.0v]** Explique detalhadamente a razão pela qual as APIs gráficas utilizam a Transformação Perspectiva.

A transformação perspectiva transforma o frustum normalizado num volume ortogonal canónico. O objectivo é simplificar a operação de visibilidade: ao fazer com que os raios projectores se tornem paralelos entre si determinação a oclusão de objectos é realizada por simples comparações das duas primeiras coordenadas.

Resulta ainda que a operação de recorte também é simplificada (um único algoritmo para ambas as situações de projecção).

Identificação do Aluno
Número:

11. [1.0v] Suponha que escreveu a seguinte linha de código:

```
gluLookAt(0.0, 0.0, 7.0, 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) [0.5v] Indique a matriz de translação em coordenadas homogéneas referente à Transformação de Visualização realizada internamente pelo pipeline OpenGL de modo a colocar os objectos da cena no referencial da câmara

b) [0.5v] Calcule a matriz de rotação em coordenadas homogéneas 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 7) e o observador está olhar para (0, 0, 0) (2º argumento) logo VPN [0 0 -7] => 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 x v)

Internamente efectuam-se uma translação seguida de uma rotação:

$$\text{Translacção} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Rotação} = \begin{bmatrix} +1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Identificação do Aluno
Número:

12. [1.0v] 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:

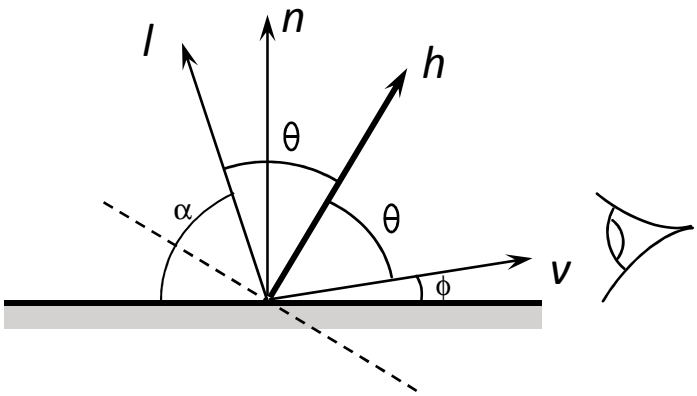
```
GLfloat ambient[] = { 0.3, 0.0, 0.0, 1.0 };
GLfloat diffuse[] = { 1.0, 0.5, 0.5, 1.0 };
GLfloat specular[] = { 1.0, 1.0, 1.0, 1.0 };

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 α de 70º e que o observador olha para a superfície segundo um ângulo ϕ de 20º.



Calcule as componentes de cor no ponto da superfície para onde a câmara aponta segundo o modelo de reflexão de **Phong modificado**, usando a **aproximação de Blinn**.
Nota: cos(30º)=0,866; cos(45º)=0,707; cos(60º)=0,5

VertexColor = ambient light * ambient material + (max {L . N, 0}) * diffuse light * diffuse material + (max {N . H, 0}) exp shininess * specular light * specular material

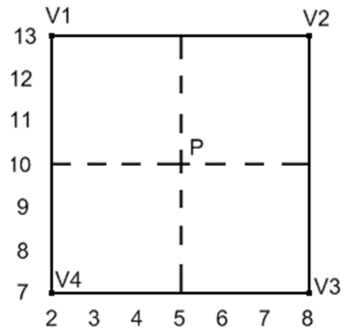
R: $0,3*1,0 + 1,0*0,6*\cos 20^o + 1,0*1,0*(\cos 25^o)^{10}$

G: $0,0*1,0 + 0,5*0,1*\cos 20^o + 1,0*1,0*(\cos 25^o)^{10}$

B: $0,0*1,0 + 0,5*1,0*\cos 20^o + 1,0*1,0*(\cos 25^o)^{10}$

Identificação do Aluno
Número:

13. [0.5v] .Considere o seguinte quadrado, em coordenadas de ecrã, prestes a ser sombreado pelo método de Gouraud. Determine a cor RGB no ponto médio P do quadrado, conhecendo as seguintes cores para os vértices: V_1=(160, 120, 128), V_2=(120, 140, 128), V_3=(80, 140, 128), V_4=(100, 120, 128).



O método de sombreamento de Gouraud consiste em interpolar linearmente, componente a componente, as cores dos vértices para os pontos interiores dos polígonos. Pela figura, verifica-se que a cor do ponto P pode ser determinada calculando primeiro a cor dos pontos (2,10) e (8,10) para os quais, por se encontrarem no ponto médio dos lados V_1V_4 e V_2V_3, bastará calcular a média das componentes de cor. Também o ponto P é o ponto médio do segmento que une aqueles pontos e, assim, basta calcular a média das cores desses pontos. Logo (notando que a componente azul é igual em todos os vértices)

P = [((160+100)/2 + (120+80)/2) / 2; (120+120)/2 + (140+140)/2) /2; 128]

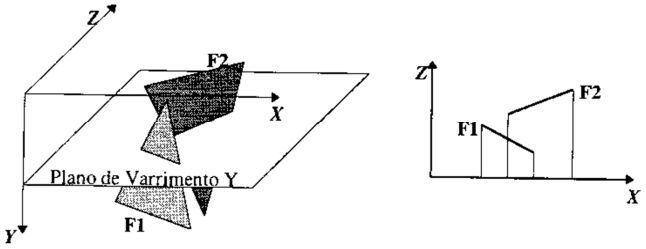
P = [(130+100)/2; (120+140)/2; 128] = [115, 130, 128]

14. [1.0v] Apresente as regras que se devem aplicar para discretizar um polígono com preenchimento e a ordem da sua aplicação.

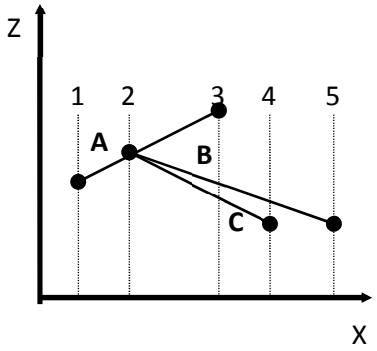
O algoritmo de linha de varrimento empregue na discretização de polígonos com preenchimento segue as seguintes regras: 1) os lados horizontais são não são considerados; 2) Um lado de um polígono passa a estar activo (para a determinação de intersecções com a linha de varrimento) quando o valor da abcissa da linha de varrimento é igual ao menor valor da coordenada y desse lado; 3) Um lado de um polígono não se encontra activo quando a abcissa da linha de varrimento for igual ao maior valor de y desse segmento; 4) Quando ordenadas todas as intersecções dos lados activos de um polígono com a linha de varrimento segundo x crescente, as intersecções de ordem par correspondem a pontos de entrada no interior do polígono e a sua coordenada x deve ser arredondada para um valor inteiro por excesso, enquanto as intersecções de ordem par correspondem a pontos de saída cujas coordenadas x devem ser arredondadas por defeito para valores inteiros.

Identificação do Aluno
Número:

15. [1.0v] O algoritmo de RSO à Linha de Varrimento (Watkins) usa o ciclo vertical em que cada iteração, correspondente a uma linha de varrimento, implica uma disposição espacial dos polígonos no plano ZoX , como pode observar pela ilustração abaixo.



Considere que para uma determinada cena, a disposição espacial dos polígonos no plano ZoX, definido por uma determinada linha de varrimento Y_{curr}, encontra-se ilustrada na figura abaixo. Estabeleça o conteúdo da Lista de Arestas Activas e efectue, justificando detalhadamente, os ciclos do algoritmo na direcção horizontal (eixo X) e na profundidade (eixo Z), mencionando a Lista de Polígonos activos.



Pela figura conclui-se que se encontram 3 polígonos activos (A, B e C), logo devem estar activas 6 arestas (esquerda e direita para cada um deles) e o conteúdo da Lista de Arestas activas será

{ Aesq, Besq, Cesq, Adir, Cdir, Bdir}

Podemos então aplicar o algoritmo de Watkins como segue

X	Cor do pixel	Porquê	Obs.
1	A	Único polígono	Entrada em A
2	A	Z igual em todos, continua A	Entrada em B e C
3	C	C -> Menor Z	Saída de A
4	C	C -> Menor Z	Saída de C
5	B	Único polígono	Saída de B

Identificação do Aluno
Número:

16. [1.0v] Quais são as características fundamentais do modelo de cor RBG? Mencione o respectivo domínio no espaço RGB e compare-o com o espaço da especificação da cor de uma fonte de luz que empregue o modelo RGB em OpenGL, explicando a razão desta diferença.

O modelo RGB é um modelo aditivo que tem como cores as cores primárias vermelho (R), verde (G) e azul(B), ocupa um espaço cúbico com a cor negra na origem (0,0,0) e a cor branca no vértice oposto (1,1,1). A diferença da especificação de uma cor neste modelo para OpenGL consiste em que, ao especificar a cor de fontes de luz, os valores das componentes de cor do modelo RGB deixam de estar obrigatoriamente compreendidas no intervalo [0,1] (ou outro equivalente) e passam a poder assumir valores entre 0 e infinito, não estando limitados àquele intervalo, permitindo assim fontes de luz de diferentes intensidades (que serão posteriormente normalizadas depois de somadas as intensidades das fontes de luz).

17. [1.0v] Uma fonte de luz emite luz com cor amarela pura incide num dado ponto de uma superfície segundo um ângulo de 45º com a normal à superfície.

Nota: sen (45º) = cos (45º) = $\sqrt{2}/2$

a) [0.5v] Se a superfície só reflectir especularmente, que características de reflexão deverá apresentar a superfície para que um observador olhando segundo um ângulo de 30º com a normal à superfície possa ver a luz reflectida?

Neste caso, ainda que o observador se encontre no plano formado pela normal e pela linha unindo a fonte de luz ao ponto de reflexão, o observador não está a ver o ponto segundo uma direcção (30º) que não é a direcção de reflexão especular perfeita (45º). Logo, para que possa observar luz reflectida nas condições enunciadas, é necessário que a reflexão especular não seja perfeita e, portanto, que exista dispersão da luz reflectida em torno da direcção de reflexão especular perfeita.

b) [0.5v] Se a superfície for reflectora difusa sem nenhum brilho com os coeficientes de reflexão (0,5; 0,75; 0,25), calcule a intensidade da cor da luz reflectida (não existe componente ambiente).

A cor amarela pura tem [1;1;0] como componentes RGB (ou [0;0;1] no espaço CMY). Neste caso, a cor da luz reflectida será

RGB -→ (0,5*1 $\sqrt{2}$ /2; 0,75*1* $\sqrt{2}$ /2; 0,25*0*)

Identificação do Aluno
Número: