



INSTITUTO  
SUPERIOR  
TÉCNICO

# Computação Gráfica

Licenciatura em Engenharia Informática e de Computadores  
Alameda / Taguspark

## Segundo Teste

19 de Novembro de 2014

O teste tem a duração de **1h00**, tolerância incluída. Responda às questões **unicamente** no espaço reservado para as respostas. **Identifique todas as folhas.** Durante o exame apenas é permitido o uso de caneta. Não é permitido o uso de calculadoras ou telemóveis. Uma resposta errada nas perguntas de escolha múltipla desconta 1/3 da cotação da respectiva questão. No verso desta página encontra informação de suporte à realização da prova.

**Boa sorte!**

### Identificação do Aluno

Nome:

Número:

	$30^\circ$	$45^\circ$	$60^\circ$
<i>sin</i>	0,5	0,707	0,866
<i>cos</i>	0,866	0,707	0,5
<i>tan</i>	0,578	1,0	1,732

	$30^\circ$	$45^\circ$	$60^\circ$
<i>sin</i>	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$
<i>cos</i>	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$
<i>Tan</i>	$\frac{1}{\sqrt{3}}$	1	$\sqrt{3}$

```

void glutInitWindowSize(int width, int height);

void glutInitWindowPosition(int x, int y);

void glutCreateWindow(char *name);

void glViewport(GLint x, GLint y, GLsizei width, GLsizei height);

void glOrtho( GLdouble left, GLdouble right,
              GLdouble bottom, GLdouble top,
              GLdouble nearVal, GLdouble farVal);

void gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ,
               GLdouble centerX, GLdouble centerY, GLdouble centerZ,
               GLdouble upX, GLdouble upY, GLdouble upZ);

void gluPerspective( GLdouble fovy, GLdouble aspect,
                    GLdouble zNear, GLdouble zFar);

void glTexImage2D(GLenum target, GLint level,  GLint internalFormat,
                 GLsizei width, GLsizei height,
                 GLint border, GLenum format,
                 GLenum type, const GLvoid * data)

```

1. [3.0v] Considere o seguinte trecho de código, assumindo que o cálculo de iluminação está activo e que toda a inicialização dos buffers e da janela já foi realizada.

```
void lightSource(double px, double py, double pz, double pw) {
    GLfloat ambient[] = { 0.1, 0.2, 0.2, 1.0 };
    GLfloat diffuse[] = { 0.8, 0.4, 0.4, 1.0 };
    GLfloat specular[] = { 0.4, 0.6, 0.6, 1.0 };
    GLfloat position[] = { px, py, pz, pw };
    glLightfv(GL_LIGHT0, GL_AMBIENT, ambient);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse);
    glLightfv(GL_LIGHT0, GL_SPECULAR, specular);
    glLightfv(GL_LIGHT0, GL_POSITION, position);
    glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION, 1.0);
    glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION, 1.0);
    glLightf(GL_LIGHT0, GL_QUADRATIC_ATTENUATION, 0.0);
}

void material() {
    GLfloat mat_ambient[] = { 0.5, 0.0, 1.0, 1.0 };
    GLfloat mat_diffuse[] = { 0.0, 0.5, 0.5, 1.0 };
    GLfloat mat_specular[] = { 0.0, 0.0, 0.7, 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_AND_BACK, GL_SHININESS, 2.0);
}

void clear() {
    glClearColor(1.0f, 1.0f, 1.0f, 0.0f);
    glClear(GL_COLOR_BUFFER_BIT);
}

void computeProjVisMatrices() {
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-2.0f, 2.0f, -2.0f, 2.0f, -1.0f, 1.0f);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0.0f, -1.0f, 1.0f, 0.0f, 0.0f, 0.0f, 0.0f, 0.0f, 1.0f);
}

void draw() {
    material();

    glBegin(GL_QUADS); // Superfície S
    glNormal3f(0.0f, 0.0f, 1.0f);
    glVertex3f(1.0f,0.0f,0.0f); // A
    glVertex3f(1.0f,1.0f,0.0f); // B
    glVertex3f(0.0f,1.0f,0.0f); // C
    glVertex3f(0.0f,0.0f,0.0f); // D
    glEnd();
}
```

**Identificação do Aluno**

Nome:

Número:

- a) [1.5] Indique qual dos trechos de código abaixo vai produzir uma cena iluminada por uma luz pontual à distância  $d=1.0$  da superfície 'S'.  
(Escolha múltipla: indique apenas uma opção. Resposta errada desconta 1/3 da cotação)

**OPÇÃO A**

```
void myDisplay() {
    clear();
    computeProjVisMatrices();
    lightSource(0, 1, 1, 0);
    material();
    draw();
    glFlush();
}
```

**OPÇÃO B**

```
void myDisplay() {
    clear();
    computeProjVisMatrices();
    lightSource(0, 1, 1, 1);
    material();
    draw();
    glFlush();
}
```

**OPÇÃO C**

```
void myDisplay() {
    lightSource(0, 1, 1, 0);
    clear();
    computeProjVisMatrices();
    draw();
    glFlush();
}
```

**OPÇÃO D**

```
void myDisplay() {
    lightSource(0, 1, 1, 1);
    clear();
    computeProjVisMatrices();
    draw();
    glFlush();
}
```

**OPÇÃO E**

```
void myDisplay() {
    clear();
    lightSource(0, 1, 1, 0);
    computeProjVisMatrices();
    draw();
    glFlush();
}
```

**OPÇÃO F**

```
void myDisplay() {
    lightSource(0, 1, 1, 1);
    draw();
    clear();
    computeProjVisMatrices();
    glFlush();
}
```

**Resposta correcta: B**

- b) [1.5] Indique o valor dos vectores  $\mathbf{v}$ ,  $\mathbf{n}$ ,  $\mathbf{l}$  e  $\mathbf{h}$  usados no modelo de iluminação de Blinn-Phong para o cálculo da cor do vértice  $\mathbf{D}$ .

$$\mathbf{v} = \|[0 \ -1 \ 1] - [0 \ 0 \ 0]\| = \|[0 \ -1 \ 1]\| = \begin{bmatrix} 0 & -\sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}$$

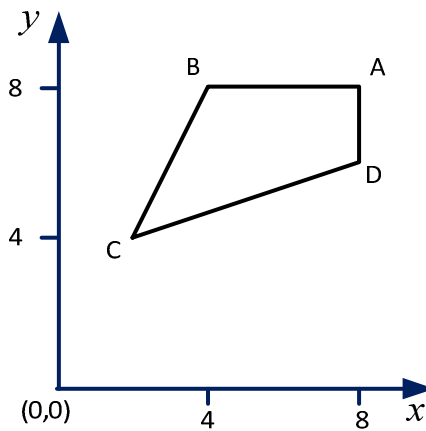
$$\mathbf{n} = [0 \ 0 \ 1]$$

$$\mathbf{l} = [0 \ 1 \ 1] - [0 \ 0 \ 0] = \|[0 \ 1 \ 1]\| = \begin{bmatrix} 0 & \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}$$

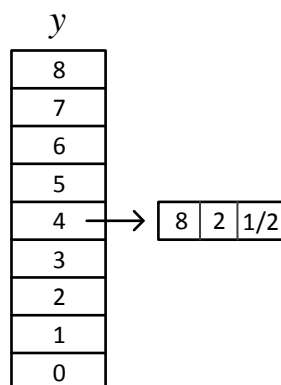
$$\mathbf{h} = \frac{\begin{bmatrix} 0 & -\sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} + \begin{bmatrix} 0 & \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}}{\left\| \begin{bmatrix} 0 & -\sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} + \begin{bmatrix} 0 & \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} \right\|} = [0 \ 0 \ 1]$$

2. [2.0v] Pretende-se proceder ao sombreado do quadrilátero  $Q=\{A,B,C,D\}$ , ilustrado na figura abaixo recorrendo ao algoritmo da scan-line. Os vértices do

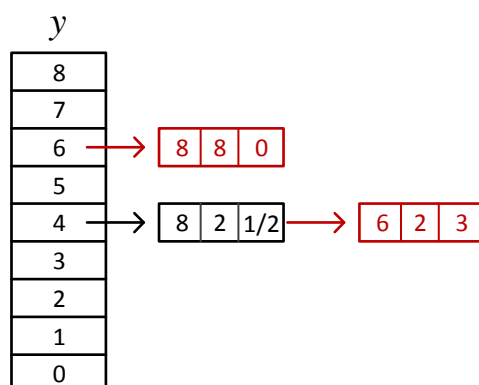
quadrilátero têm coordenadas  $A=[8 \ 8 \ 1]$ ,  $B=[4 \ 8 \ 1]$ ,  $C=[2 \ 4 \ 1]$ ,  $D=[8 \ 6 \ 1]$ . Através da aplicação prévia do modelo de iluminação de Blinn-Phong, determinou-se que os vértices têm as cores  $I_A = [1/5 \ 0 \ 0]$ ,  $I_B = [2/5 \ 0 \ 0]$ ,  $I_C = [4/5 \ 0 \ 0]$ ,  $I_D = [3/5 \ 0 \ 0]$ .



- a) **[0.5v]** Complete a lista de arestas ilustrada abaixo com os valores com que esta seria inicializada pelo algoritmo da *scan-line*.



Resposta:



**Identificação do Aluno**

Nome:

Número:

- b) [0.9v] Diga qual a cor atribuída à quadrícula na posição (4,6) pelo algoritmo de sombreamento de Gouraud.

$$R(4, 6) = 0.6; G(4, 6) = 0.0; B(4, 6) = 0.0$$

- c) [0.6v] Indique a cor com que ficam as quadrículas (8,7) e (7,8) após terminar a rasterização, assumindo que este é o único polígono da cena e que o buffer de cor foi inicializado a branco.

$$R(8, 7) = 0.4; G(8, 7) = 0.0; B(8, 7) = 0.0$$

$$R(7, 8) = 1.0; G(7, 8) = 1.0; B(7, 8) = 1.0$$

3. [3.0v] Considere o seguinte trecho de código OpenGL:

```
(...)  
glMatrixMode(GL_PROJECTION);  
glLoadIdentity();  
glOrtho(-2.0f, 0.5f, -2.0f, 2.0f, -1.0f, 1.0f);  
glMatrixMode(GL_MODELVIEW);  
glLoadIdentity();  
  
glBegin(GL_QUADS);  
glNormal3f(0.0f, 0.0f, 1.0f);  
glVertex3f(1.0f, 0.0f, 0.0f); // A  
glVertex3f(1.0f, 1.0f, 0.0f); // B  
glVertex3f(0.0f, 1.0f, 0.0f); // C  
glVertex3f(0.0f, 0.0f, 0.0f); // D  
glEnd();  
(...)
```

- a) [0.5] No pipeline OpenGL o recorte é efectuado em coordenadas de recorte (*clipping coordinates*). Indique, nessas coordenadas, os limites contra os quais é realizado o recorte.

$$-1 \leq x, y, z \leq 1$$

- b) [1.0] Indique os *outcodes* dos quatro vértices do quadrilátero desenhado pelo código apresentado.

$$b_1: z < -1; b_2: z > 1; b_3: y > 1; b_4: y < -1; b_5: x > 1; b_6: x < -1 \text{ (clip)}$$

$$b_1: z < -1; b_2: z > 1; b_3: y > 2; b_4: y < -2; b_5: x > 1/2; b_6: x < -2 \text{ (world)}$$

$$OC_A = 000010$$

$$OC_B = 000010$$

$$OC_C = 000000$$

$$OC_D = 000000$$

- c) [1.5] Qual das seguintes opções completa a frase “Na aplicação da primeira iteração do algoritmo de Cohen-Sutherland...”?

(Escolha múltipla: indique apenas uma opção. Resposta errada desconta 1/3 da cotação)

**A:** "...os segmentos AB e BC são trivialmente rejeitados e os CD e DA são trivialmente aceites."

**B:** "...os segmentos AB e CD são trivialmente aceites, o segmento BC é subdividido e o segmento DA é trivialmente rejeitado."

**C:** "...o segmento CD é trivialmente aceite, o segmento BC é trivialmente rejeitado e os segmentos AB e DA são subdivididos."

**D:** "...o segmento DA é trivialmente aceite, o segmento BC é trivialmente rejeitado e os segmentos AB e CD são subdivididos."

**E:** Nenhuma das anteriores

**Resposta: E**

**Identificação do Aluno**

Nome:

Número:

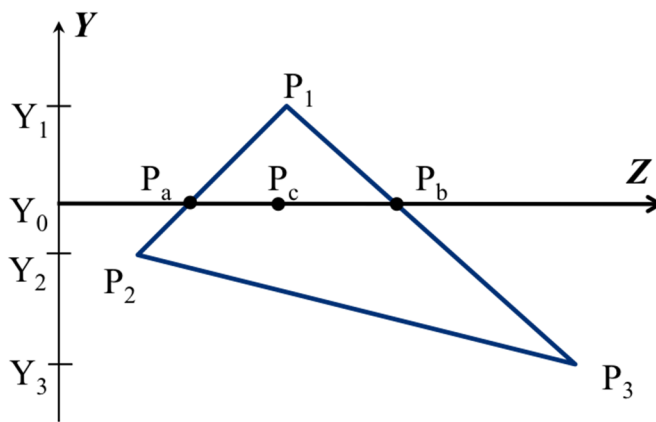
4. [1.0v] O algoritmo de Bresenham serve para desenhar...

(Escolha múltipla: indique apenas uma opção. Resposta errada desconta 1/3 da cotação)

- A: ... linhas rectas num terminal vectorial.
- B: ... segmentos de recta num dispositivo de quadrícula.
- C: ... sequências esparsas de pontos num ecrã caligráfico.
- D: ... triângulos não convexos.
- E: Nenhuma das opções acima

Resposta: B

5. [2.0v] Considere a imagem abaixo e a informação seguinte:



- 
- $Y_1=20, Y_0=0, Y_2=-5, Y_3=-20$
- Os pontos  $P_1, P_2$  e  $P_3$  tem  $Z=5, 10$  e  $30$  respectivamente
- $d(P_c, P_a) = d(P_b, P_c)/2$
- Os pontos  $P_1, P_2$  e  $P_3$  estão no plano  $X=10$

a) [1.0v] Usando interpolação bilinear calcule o valor de  $Z$  em  $P_c$ .

$Z_c =$  \_\_\_\_\_

Resposta: calcular pela fórmula  $Z_c = (2Z_a + Z_b) / 3$

Para  $P_1, P_2$  e  $P_3$  com  $Z=10, 5$  e  $30$  resp

$Z_a=6$  (via cálculo da recta que passa por  $P_1$  e  $P_2$ )

$Z_b=20$  (via cálculo da recta que passa por  $P_1$  e  $P_3$ )

$Z_c = (2Z_a + Z_b)/3 = 32/3$

Para  $P_1, P_2$  e  $P_3$  com  $Z=5, 10$  e  $30$  resp

$Z_a=9$  (via cálculo da recta que passa por  $P_1$  e  $P_2$ )

$Z_b=5+25/2$  (via cálculo da recta que passa por  $P_1$  e  $P_3$ )



$$Z_c = (2Z_a + Z_b)/3 = 71/6$$

- a) [1.0v] Suponha que para  $X=10$  e  $Y$  entre  $-30$  e  $+30$ , os valores de  $Z$  armazenados no  $Z$ -buffer são de  $Z=6$ . Qual a visibilidade dos pontos  $P_1$ ,  $P_2$  e  $P_3$  ?

(Escolha múltipla: indique apenas uma opção. Resposta errada desconta 1/3 da cotação)

- A:  $P_1$  oculto,  $P_2$  oculto,  $P_3$  oculto
- B:  $P_1$  oculto,  $P_2$  oculto,  $P_3$  visível
- C:  $P_1$  oculto,  $P_2$  visível,  $P_3$  oculto
- D:  $P_1$  oculto,  $P_2$  visível,  $P_3$  visível
- E:  $P_1$  visível,  $P_2$  oculto,  $P_3$  oculto
- F:  $P_1$  visível,  $P_2$  oculto,  $P_3$  visível
- G:  $P_1$  visível,  $P_2$  visível,  $P_3$  oculto
- H:  $P_1$  visível,  $P_2$  visível,  $P_3$  visível

**Resposta: C ( $P_1$ : oculto,  $P_2$ : visível,  $P_3$ : oculto)**

6. [4.0v] Responda às seguintes perguntas sobre mapeamento de texturas

- a) [2.0] Considere o seguinte excerto de código que pretende carregar uma textura com  $64^2$  texels.

```
1      Glubyte my_texture_2d[32][32][4]
...    ....
6      glEnable(GL_TEXTURE_2D);
7      glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, 64, 64, 0,
                  GL_RGBA, GL_UNSIGNED_BYTE, my_texture_2d);
```

Sabendo que só pode alterar uma, qual a linha teria de alterar para que o código execute correctamente ? Re-escreva-a.

Linha: \_\_\_\_\_

**Resposta 1 : `Glubyte my_texels[64][64][4]`**

#### Identificação do Aluno

Nome:

Número:

**b) [1.0v]** O método de "bump mapping" estudado nas aulas teóricas usa:  
(Escolha múltipla: indique apenas uma opção. Resposta errada desconta 1/3 da cotação)

- A:** Texturas em ficheiros especiais BUMP
- B:** Interpolação trilinear das componentes RGB em cada pixel
- C:** Perturbação das coordenadas Z em cada pixel
- D:** Perturbação das normais em cada pixel
- E:** Nenhuma das abordagens acima

**Resposta: D**

**c) [1.0v]** Os "mipmaps" permitem...  
(Escolha múltipla: indique apenas uma opção. Resposta errada desconta 1/3 da cotação)

- A:** ... utilizar uma textura em várias escalas.
- B:** ... interpolação trilinear das componentes da textura.
- C:** ... efectuar point sampling.
- D:** ... aumentar o efeito de aliasing.
- E:** Nenhuma das opções anteriores

**Resposta: A**

7. [1.5v] Tem uma cor no formato RGB. Qual o cálculo a efectuar para determinar essa mesma cor no formato CMY?

C=\_\_\_\_\_ M=\_\_\_\_\_ Y=\_\_\_\_\_

8. [1.5v] Considere a figura abaixo. Qual o formato de ficheiro que escolheria para armazenar esta imagem?



O formato JFIF (JPEG)

9. [2.0v] Considere o funcionamento do algoritmo do Ray-Tracing numa janela com 256x256 quadrículas. Considere ainda que existem 3 fontes de luz, a cena é composta por dois objectos opacos convexos e são efectuados 1000 cálculos de intersecção. Qual o número mínimo de raios de luz primários, de sombra, de reflexão e de transmissão?

Primários: 256x256

Sombra: 1000x3

Reflexão: 1000

Transmissão: 0

**Identificação do Aluno**

Nome:

Número: