

INSTITUTO
SUPERIOR
TÉCNICO

Computação Gráfica

Licenciatura em Engenharia Informática e de Computadores
Alameda / Taguspark

Primeiro Teste

26 de Outubro de 2013

O teste tem a duração de **1h00**, 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. **Identifique ambas as folhas.** As restantes folhas podem ser utilizadas como folhas de rascunho. 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.

Respostas:

1. [1.0v] _____

2.a) [1.0v] _____

2.b) [1.0v] A= _____ B= _____

2.c) [1.0v] Cor: _____

3.a) [1.0v]

3.b) [0.5v]

3.c) [0.5v]

4.a) [1.0v]

4.b) [1.0v]

4.c) [1.0v]

Identificação do Aluno

Nome:

Número:

Identificação do Aluno	
Nome:	Número:

5.a) [1.0v] _____ 5.b) [1.0v] _____

6.a) [1.0v] _____

6.b) [1.0v]



6.c) [1.0v]



7.a) [1.0v] $I =$ _____

7.b) [1.0v]

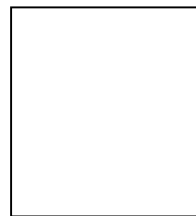
$\vec{n} =$



$\vec{l} =$



$\vec{h} =$



7.c) [1.0v]

$I_a =$

$I_d =$

$I_s =$

8.a) [1.0v]

8.b) [1.0v]

$R =$

$G =$

$B =$

8.c) [1.0v]

$R =$

$G =$

$B =$

Identificação do Aluno

Nome:

Número:

	30°	45°	60°
\sin	0,5	0,707	0,866
\cos	0,866	0,707	0,5

	30°	45°	60°
\sin	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$
\cos	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$

```

void glutInitWindowSize(int width, int height);
void glutInitWindowPosition(int x, int y);
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);

```

Identificação do Aluno

Nome:

Número:

1. [3.0v] Considere o seguinte programa em OpenGL.

```
01:  int main(int argc, char *argv[]) {
02:      glutInit(&argc, argv);
03:      glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
04:      glutInitWindowSize (400, 400);
05:      glutInitWindowPosition (-1, -1);
06:      glutCreateWindow("Teste");
07:      glutDisplayFunc(myDisplay);
08:      glutReshapeFunc(myReshape);
09:      glutMainLoop();
10:      return 0;
11:  }
12:  void myReshape(GLsizei w, GLsizei h) {
13:      glViewport(0, 0, w, h);
14:      glMatrixMode(GL_PROJECTION);
15:      glLoadIdentity();
16:      glOrtho(-2.0f, 2.0f, -2.0f, 2.0f, -2.0f, 2.0f);
17:      glMatrixMode(GL_MODELVIEW);
18:      glLoadIdentity();
19:  }
20:  void myDisplay(void) {
21:      glColor3f(0.0f, 0.0f, 0.0f);
22:      glClearColor(1.0f, 1.0f, 1.0f, 0.0f);
23:      glClear(GL_COLOR_BUFFER_BIT);
24:      glBegin(GL_POLYGON);
25:          glVertex3f(-1.0, A, 0.5f);
26:          glVertex3f(1.0f, -1.0f, 0.5f);
27:          glVertex3f(1.0f, 1.0f, 0.5f);
28:          glVertex3f(-1.0f, 1.0f, B );
29:      glEnd();
30:      glFlush();
31:  }
```

a) [1.0v] O que faz a linha de código número 4?

(escolha múltipla: indique a opção correcta na página de respostas)

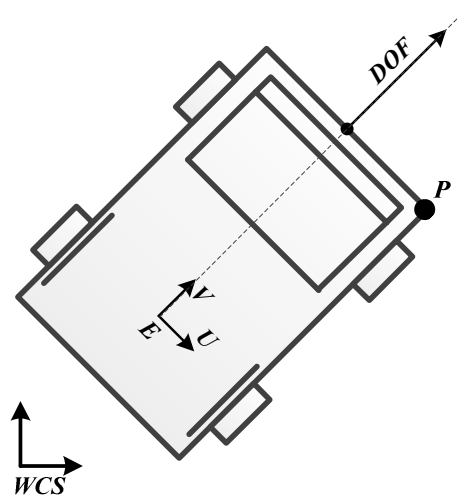
- A:** Preenche o viewport com a cor vermelha;
- B:** Invoca a função myDisplay;
- C:** Regista a callback myDisplay;
- D:** Preenche o viewport com a cor azul;
- E:** Cria uma janela de 400 por 400;
- F:** Nenhuma das anteriores.

b) [1.0v] Indique os valores de A e B de forma a que o código acima desenhe um quadrado.**c) [1.0v]** Com que cor será desenhado o quadrado?**Identificação do Aluno**

Nome:

Número:

2. [1.0v] O pipeline de geometria do OpenGL transforma:
(escolha múltipla: indique a opção correcta na página de respostas)
- A: Polígonos côncavos em triângulos;
B: Coordenadas WCS dos vértices em coordenadas da janela;
C: Triângulos em polígonos côncavos;
D: Coordenadas WCS dos vértices em coordenadas VRC;
E: Nenhuma das anteriores.
3. [2.0v] Considere que é aplicada a um objecto uma transformação em 3D composta por uma rotação de 45° sobre o eixo ZZ, seguida de uma escala uniforme de 2.
- a) [1.0] Calcule as matrizes de transformação referentes à rotação e à escala.
- b) [0.5] Calcule a matriz da transformação composta.
- c) [0.5] Sabendo que o centro do objecto for dado pelo ponto $P=[0 \ 1 \ 2]^T$, diga quais as coordenadas do centro do objecto após sofrer a transformação indicada acima.
4. [3.0v] Considere a cena 2D abaixo representada, onde a posição e direcção de um veículo são dadas pelo ponto E e pelo vector DOF, respectivamente.



Sabendo que:

- Posição do veículo: $E_{WCS} = [2 \ 2]^T$
- Direcção de veículo (não normalizada): $DOF_{WCS} = [1 \ 1]^T$
- Pisca direito (no referencial do veículo): $P_{UV} = [2 \ 4]^T$

Identificação do Aluno

Nome:

Número:

- a) [1.0] Apresente a matriz de transformação composta $M_{UV \rightarrow WCS}$, que realiza a mudança do referencial UV para o sistema de coordenadas do mundo.
- b) [1.0] Indique as coordenadas do pisca direito do veículo, P_{WCS} , em coordenadas do mundo.
- c) [1.0] Caso o veículo se deslocasse duas unidades no sentido DOF, qual seria a nova posição do veículo, E'_{WCS} , em coordenadas do mundo?

5. [2.0v] Considere o seguinte excerto de código de OpenGL

```
void displayer(); {
    glPushMatrix();
    glScalef( 0.5, 1.0, 2.0);           // Transformação S1
    glPushMatrix();
    glTranslatef( -10.0, 10.0, 100.); // Transformação T1
    showObject( obj1);
    glPopMatrix();
    glRotatef( -90.0, 0.0, 1.0, 0.0); // Transformação R1
    glScalef( 2.0, 1.0, 0.5);         // Transformação S2
    showObject( obj2);
    glPopMatrix();
    showObject( obj3);
    glRotatef( 180.0, 1.0, 0.0, 0.0); // Transformação R2
    showObject( obj4);
    glPopMatrix();
    glFlush();
}
```

- a) [1.0v] Quais são as transformações que afectam o objecto obj2 quando a função displayer () é executada.
- b) [1.0v] Após a execução da função Displayer qual o conteúdo da Matriz ModelView?

6. [3.0v] Considere o seguinte comando OpenGL

```
glOrtho (-10, 10, -20, 20, 100, 200);
```

O OpenGL efectua uma transformação de normalização para o volume de visualização ortogonal canónico.

- a) [1.0v] Quais são os limites desse volume?
- b) [1.0v] Considerando que a transformação de normalização consiste no produto de duas matrizes geométricas $T_2 * T_1$ Calcule a matriz correspondente à transformação T_2 .
- c) [1.0v] Qual a transformação que é efectuada pela matriz T_1 ?

Identificação do Aluno

Nome:

Número:

7. [3.0v] Considere o seguinte extrato de código OpenGL:

```
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[] = { 0.0, 10.0, 0.0, 1.0 };
glLightfv(GL_LIGHT0, GL_AMBIENT, ambient);
glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse);
glLightfv(GL_LIGHT0, GL_SPECULAR, specular);
glLightfv(GL_LIGHT0, GL_POSITION, position );

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, GL_AMBIENT, mat_ambient);
glMaterialfv (GL_FRONT, GL_DIFFUSE, mat_diffuse);
glMaterialfv (GL_FRONT, GL_SPECULAR, mat_specular);
glMaterialf (GL_FRONT, GL_SHININESS,50.0);

glLoadIdentity();
gluLookAt(0.0, 2.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0 );
glBegin(GL_TRIANGLE);
glNormal3f(1.0, 0.0, 0.0);
glVertex3f(1.0, 0.0, 0.0); // Ponto A
glNormal3f(0.0, 1.0, 0.0);
glVertex3f(0.0, 0.0, 0.0); // Ponto B
glVertex3f(0.0, 0.0, 1.0); // Ponto C
glEnd();
```

- a) [1.0v] Escreva a fórmula de cálculo da intensidade num ponto segunda o Modelo de Iluminação de Blinn-Phong para uma única fonte de luz pontual e sem considerar a atenuação atmosférica .
- b) [1.0v] Calcule o valor dos versores \vec{n} , \vec{l} , \vec{h} usados no cálculo da cor do vértice B.
- c) [1.0v] Calcule o valor das intensidades ambiente, difusa e especular da componente Azul do ponto B:

8. [4.0v] Considere o rectângulo com os seguintes vértices:

A = (10, 0) B = (20, 0) C=(20, 10) D=(10, 10)

As cores desse vértices são:

A=(100, 100, 0), B=(200, 200, 0), C=(200, 200, 0), D=(100, 100, 0)

- a) [1.0v] Indique qual o algoritmo que o OpenGL usa para efectuar o cálculo da iluminação.
- b) [1.0v] Considere que o OpenGL está a usar o algoritmo de sombreado constante (flat shading). Nesse caso calcule a cor do ponto D=(15, 5).
- c) [1.0v] Considere que na função de inicialização do programa é usado o seguinte comandoOpenGL: glShadeModel(GL_SMOOTH). Calcule a cor na mesma quadrícula D=(15, 5).

Identificação do Aluno

Nome:

Número: