Instituto de Informática
Universidade Federal do Rio Grande do Sul
Campus do Vale – Bloco IV
Av. Bento Gonçalves, 9500 – Agronomia
Caixa Postal 15.064
91501-970 Porto Alegre, RS
BRASIL

Carla Maria Dal Sasso Freitas
Professora Associada

INF01047 – Fundamentos de Computação Gráfica
Prova 1 – 30/04/2008

NOME:_____

No. MATRÍCULA: _____

Por favor, responda a TODAS as questões na prova, no verso ou nos lugares determinados.

1. (1.0 ponto) Marque V ou F, caso a afirmativa seja verdadeira ou falsa.. Acerca de vetores podemos afirmar que:

( ) O produto escalar entre dois vetores é nulo se os dois vetores tiverem a mesma direção e sentido.

( ) O produto vetorial de dois vetores tem como resultado um vetor que tem a direção da normal ao plano formado pelos dois vetores.

( ) A soma de dois pontos é uma operação válida e tem como resultado um terceiro ponto cujas coordenadas são a soma das coordenadas dos dois pontos.

( ) Um sistema de referência é uma base vetorial cujos vetores são localizados num ponto específico do espaço.

2. (1.0 ponto) Observe as matrizes de transformação abaixo. Determine o resultado da aplicação das mesmas (isoladamente) sobre um quadrado de lado 2, centrado na origem do sistema de coordenadas. Que transformações sâo essas?

(a) $\begin{bmatrix} 1 & 4 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ (b) $\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

3. (1.0 ponto) O que é o “pipeline” de visualização?

4. (1.0 ponto) Suponha que num dado momento a *window* esteja dimensionada como um retângulo cuja diagonal principal é (-10,-6) – (10,6) e a *viewport* correspondente seja uma janela de 600x 400 pixels localizada no ponto (10,10) da tela.

a) Qual a relação de aspecto entre da *window* e da *viewport*?

b) Ocorre algum tipo de distorção na exibição de objetos que estão contidos na região delimitada pela *window*?

5. (1,5 pontos) Suponha três pontos P, Q e A, no SRU. Considere P = posição de uma câmera; Q = ponto que, relativo a P, indica a direção da vertical da câmera; A= alvo da "fotografia". Descreva como obter um sistema de referência de câmera a partir destes pontos. Como essa operação é disponibilizada em OpenGL?

6. (1,0 pontos) Considere o objeto definido abaixo no Sistema de Referência do Universo:

Geometria (vértices): v1 = (0,0,0); v2 = (2,1,0); v3 = (0,2,0); v4 = (1,1,1)
Topologia (lista de triângulos): (v1,v2,v3) – (v2,v1,v4) – (v3,v2,v4) – (v1,v3,v4)

Desenhe o que vai aparecer na tela quando o universo for visualizado com projeção paralela por um observador localizado em cada uma das posições abaixo, olhando para a origem do SRU, vertical igual ao eixo y do universo:

olho = (0,0,30)	olho = (50,0,0)

Observações: ignore os limites do volume de visualização; indique claramente a posição dos eixos no desenho, assim como cada vértice do objeto; observe as proporções das distâncias nos eixos.

7. (2,0 pontos) Desenhe o que vai aparecer na tela quando o trecho de programa abaixo for executado. Suponha projeção paralela.

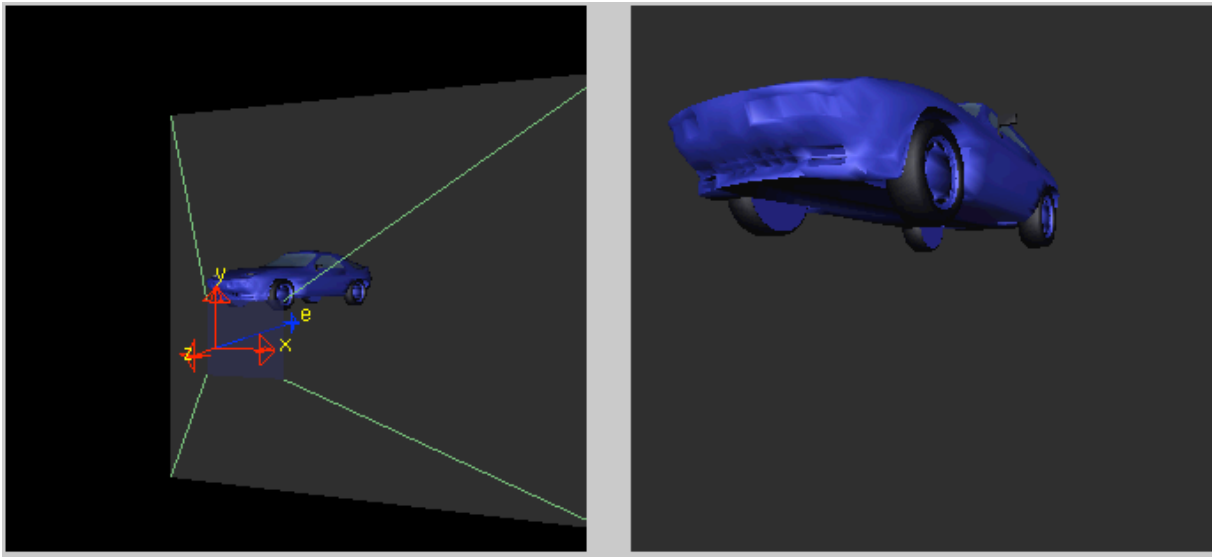
```
glClear(GL_COLOR_BUFFER_BIT);

glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
gluLookAt (4,4,-10,0,0,0,0,1,0);

// Desenha eixos
glColor3f(1.0f, 0.0f, 0.0f); // eixos Y vermelho
glBegin(GL_LINES);
       glVertex3f (0.0, -10.0, 0.0);
       glVertex3f (0.0, 10.0, 0.0);
       glColor3f(0.0f, 0.0f, 0.0f); // eixo X preto
       glVertex3f (-10.0, 0.0, 0.0);
       glVertex3f (10.0, 0.0, 0.0);
       glColor3f(1.0f, 1.0f, 0.0f); // eixo Z amarelo
       glVertex3f (0.0, 0.0, -10.0);
       glVertex3f (0.0, 0.0, 10.0);
glEnd();
```

```
glColor3f(1.0,0.0,0.0);
glPushMatrix();
glutWireCube(4);  // cubo 1
glTranslatef(-1.5,-1.5,-1.5);
glScalef(0.25,0.25,0.25);
glColor3f(0.0,1.0,0.0);
glutWireCube(4);    // cubo 2
glTranslatef(12,6,12);
glColor3f(0.0,0.0,1.0);
glutWireCube(4);   // cubo 3
glPopMatrix();
glTranslatef(-6,0,0);
glScalef (0.5, 0.5, 0.5);
glColor3f(0.0,0.0,0.0);
glutWireCube(4);   // cubo 4
```

8. (1,5 pontos) Indique, dentre as alternativas, qual a seqüência de transformações usada para obter a imagem da direita. A imagem da esquerda mostra o objeto transformado no SRU. Originalmente o objeto estava alinhado com o eixo Z do SRU.




```
glTranslatef( 0.00  , 0.45  , 0.00  );
  glRotatef( -30.0 , 0.00  , 1.00  , 0.00  );
   glScalef( 1.00  , 1.00  , 1.00  );
    glBegin( . . . );
```

(a)

```
glTranslatef( 0.00  , 0.45  , 0.00  );
  glRotatef( 30.0  , 0.00  , 1.00  , 0.00  );
   glScalef( 1.00  , 1.00  , 1.00  );
    glBegin( . . . );
```

(b)

```
glTranslatef( 0.00  , -0.45 , 0.00  );
  glRotatef( 30.0  , 0.00  , 1.00  , 0.00  );
   glScalef( 1.00  , 1.00  , 1.00  );
    glBegin( . . . );
```

(c)

```
  glRotatef( 30.0  , 0.00  , 1.00  , 0.00  );
glTranslatef( 0.00  , 0.45  , 0.00  );
   glScalef( 1.00  , 1.00  , 1.00  );
    glBegin( . . . );
```

(d)