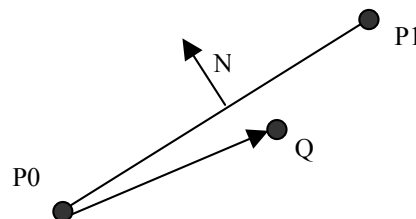


INDICATIVOS PARA RESPOSTAS DA LISTA # 1

A seguir, indicamos COM BASE EM QUE CONTEÚDO vocês podem encontrar as respostas para os exercícios. A lista é um guia de estudo ☺

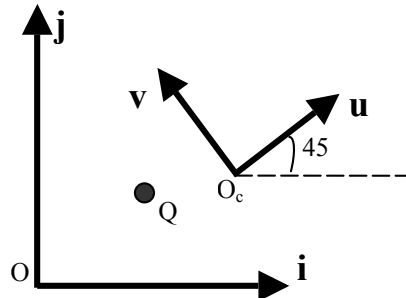
1. Para rasterização, vimos o algoritmo DDA e o de Bresenham, em sala de aula (slides rasterização, semana de 16 de março)

2. Para determinar de que lado do segmento se encontra Q, podemos usar o produto escalar entre o vetor normal ao segmento P_0 - P_1 e um vetor que vai de P_0 a Q. Veja nos slides sobre Fundamentos matemáticos!



3. Para testar se um ponto está dentro, fora ou sobre os segmentos que definem um polígono, você pode usar o que aprendeu na questão acima! Fazendo o teste acima para todos segmentos do polígono, um ponto só estará “dentro” do polígono, se estiver do mesmo lado (esquerda ou direita) para todos os segmentos. Para isso, não se esqueça, as normais aos segmentos do polígono devem sempre indicar consistentemente (todas) o lado de dentro ou o lado de fora do polígono ...

4. Essa questão é sobre mudança de base, mudança de sistema de referência. Veja como isso foi feito para a mudança do sistema de referência do universo para o sistema de referência de câmera. Você deve expressar os vetores da base (\mathbf{v}, \mathbf{u}) como combinações lineares da base (\mathbf{i}, \mathbf{j}) .



5. Para calcular o vetor normal das faces de um objeto tridimensional, você deve utilizar, para cada face, o produto vetorial entre dois segmentos do polígono que define a face. Isto também está detalhado nos slides de fundamentos matemáticos e lá também consta o uso desses vetores normais para determinar se uma face é visível ou não do ponto de vista do observador. A aula de remoção de faces ocultas do dia 27/4, repetiu essa aplicação de vetores normais das faces. Observe que deve ser garantida a consistência da informação indicada pelos vetores normais (ou todos indicam o lado de dentro do objeto ou todos indicam o lado de fora).
6. A representação de vértices e transformações em matrizes permite que a aplicação dessas transformações sobre os vértices seja feita através de operações com matrizes. Se os vértices estiverem representados em coordenadas homogêneas, podemos usar apenas a multiplicação matricial para aplicar qualquer transformação e, com isso, podemos nos valer da propriedade associativa e compor transformações facilmente.
7. Essa questão é respondida facilmente realizando as multiplicações das matrizes que representam as transformações!!

8. Considere um segmento de reta AB, com A = (1,1) e B = (5,5) e as seqüências de transformações abaixo:

- $T(-1,-1) \cdot S(2,2)$
- $S(2,2) \cdot T(-1,-1)$

8.1 Os resultados são iguais ou diferentes? Justifique sua resposta e mostre os resultados de cada seqüência.

DA MESMA FORMA QUE A QUESTÃO ANTERIOR: Faça as multiplicações e desenhe o resultado.

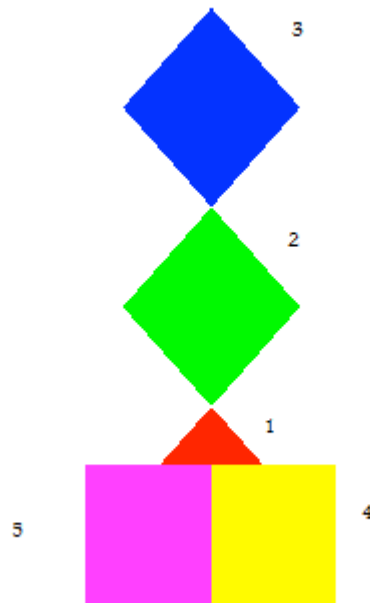
8.2 Como você faria para “animar” o segmento de reta fazendo com que sucessivamente ele realizasse uma volta completa em torno do ponto A e depois uma volta completa em torno do ponto B?

ESSE É REALIZADO COM DOIS COMANDOS “FOR”, um para rotacionar o ponto B em torno de A, 360 graus (com um passo angular fixo), e outro para rotacionar A em torno de B de forma análoga.

9 Nesta questão, você deve observar os limites da window em relação aos vértices dos objetos. Na opção (a), ambos os objetos estão no interior da window. Já na opção (b) ocorre recorte porque dois vértices tem coordenadas fora do limite superior em X. Você pode “desenhar” os objetos e depois a window para perceber o que vai ser recortado.

10 Adaptando o programa para colorir os polígonos, tem-se a seguinte saída.

```
glBegin(GL_POLYGON);
    glVertex2f(0.0f,0.0f);
    glVertex2f(10.0f,0.0f);
    glVertex2f(10.0f,10.0f);
    glVertex2f(0.0f,10.0f);
glEnd();
}
void FazDesenho()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-40, 40, -10, 60);
    glMatrixMode(GL_MODELVIEW);
    glRotatef(45,0,0,1);
    glColor3f(1.0f, 0.0f, 0.0f);
    Desenha(); // 1
    glTranslatef(10,10,0);
    glColor3f(0.0f, 1.0f, 0.0f);
    Desenha(); // 2
    glTranslatef(10,10,0);
    glColor3f(0.0f, 0.0f, 1.0f);
    Desenha(); // 3
    glTranslatef(-20,-20,0);
    glRotatef(-45,0,0,1);
    glColor3f(1.0f, 1.0f, 0.0f);
    Desenha(); // 4
    glTranslatef(-10,0,0);
    glColor3f(1.0f, 0.0f, 1.0f);
    Desenha(); // 5
    glutSwapBuffers();
}
```



11 Considere duas matrizes homogêneas de transformação T1 e T2. T1T2 é igual a T2T1 quando (marque V se a frase for verdadeira e F se for falsa):

- (V) Ambas representam translações puras.
- (V) Ambas representam escalas puras com fatores iguais ou diferentes.
- (F) Uma representa uma rotação e a outra uma translação.
- (V) Uma representa uma rotação e a outra uma escala com fatores iguais em todas as dimensões.

12 Observando as matrizes de rotação no material, identifica-se:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0,707 & -0,707 & 0 \\ 0 & 0,707 & 0,707 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotação

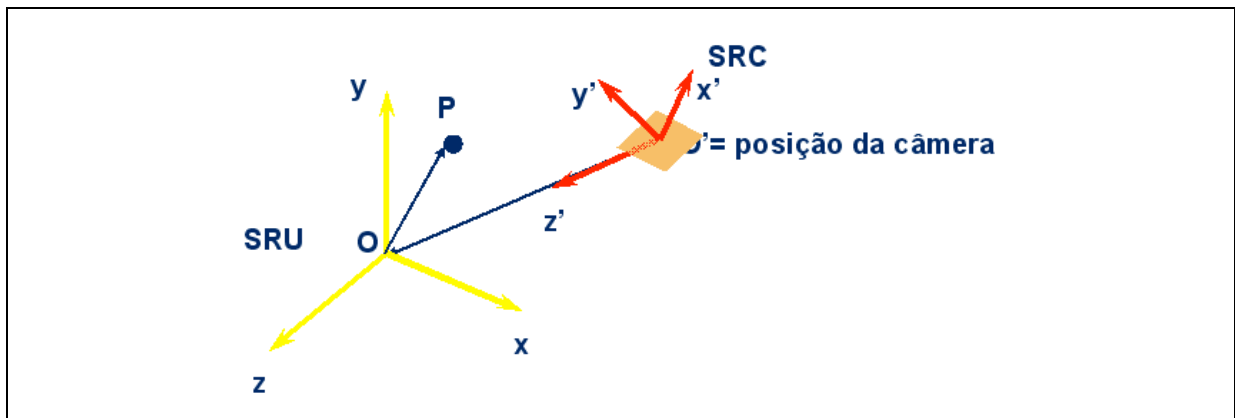
$$\begin{bmatrix} 1 & 0 & 0 & 8 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Translação

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Mudança de escala

13 Observe a figura a seguir que mostra a relação entre o SRU e o SRC. Marque V ou F nas afirmações a seguir, identificando as afirmações verdadeiras ou falsas, respectivamente.



(V) A transformação de câmera pode ser representada como uma seqüência de transformações geométricas aplicadas às primitivas geométricas.

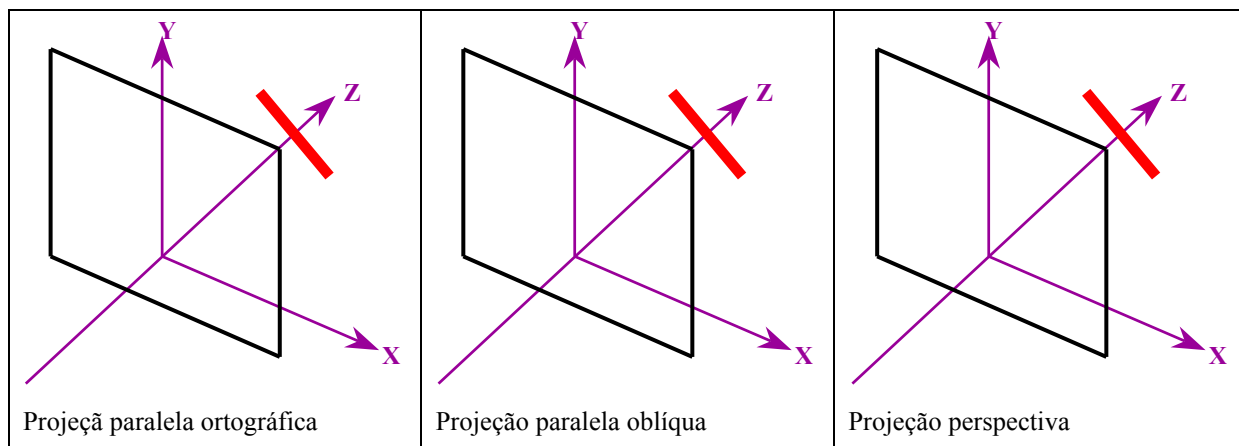
(F) Considerando w o vetor da base que determina a direção do eixo z' pode-se afirmar que w é sempre obtido a partir da posição da câmera e da origem do SRU.

(F) A transformação de câmera corresponde à etapa final do pipeline de visualização 3D.

(V) As primitivas geométricas, após a transformação de câmera, tem coordenadas relativas ao ponto indicado como posição do observador.

(V) Na API OpenGL, a transformação de câmera faz parte das transformações compostas na matriz model view.

14 A questão requer que vocês observem a diferença das projetantes nos diferentes tipos de projeção!



15. O programa utilizado está disponível no Moodle. Executem o programa modificando os parâmetros para identificar as visões fornecidas.



1



2



3



4

```

fovy aspect zNear zFar
gluPerspective( 60.0 , 1.00 , 1.0 , 10.0 );
gluLookAt( 0.00 , 2.00 , 2.00 , <- eye
          0.00 , 0.00 , 0.00 , <- center
          0.00 , 1.00 , 0.00 ); <- up

```

()

```

fovy aspect zNear zFar
gluPerspective( 60.0 , 1.00 , 1.0 , 10.0 );
gluLookAt( 2.00 , 0.00 , 2.00 , <- eye
          0.00 , 0.00 , 0.00 , <- center
          0.00 , 1.00 , 0.00 ); <- up

```

()

```

fovy aspect zNear zFar
gluPerspective( 60.0 , 1.00 , 1.0 , 10.0 );
gluLookAt( -2.00 , 0.00 , 2.00 , <- eye
          0.00 , 0.00 , 0.00 , <- center
          0.00 , 1.00 , 0.00 ); <- up

```

()

```

fovy aspect zNear zFar
gluPerspective( 60.0 , 1.00 , 1.0 , 10.0 );
gluLookAt( -2.00 , 0.00 , 2.00 , <- eye
          0.00 , 0.00 , 0.00 , <- center
          0.00 , 1.00 , 0.00 ); <- up

```

()