

1. O que são coordenadas homogêneas e qual a sua utilidade?

Coordenadas homogêneas são um modo de representar pontos através da inserção de uma nova coordenada W, onde W é o elemento que permite que um mesmo ponto possua múltiplas representações. Isso permite combinar todas as transformações geométricas com multiplicações matriciais. Suas utilidades são operações com vetores que permitem operações como translação, rotação, escala e projeção perspectiva, por exemplo.

2. Qual o objetivo de se utilizar coordenadas homogêneas e como representar pontos no infinito?

O objetivo é de otimizar a aplicação de operações que envolvem transformações geométricas, como por exemplo, de rotação e escala, através da combinação de todas as transformações geométricas com multiplicações matriciais. O uso da coordenada homogênea W igual a zero permite representação de pontos no infinito (fora do espaço dimensional).

3. Explique o processo de projeção.

O processo de projeção consiste em definir como objetos tridimensionais serão visualizados, uma vez que os dispositivos de saída dos computadores, como o monitor, são bidimensionais. Na projeção as coordenadas 3D dos objetos são convertidas em coordenadas 2D de acordo com o tipo de projeção selecionada. Em OpenGL a matriz correspondente à etapa de projeção é a GL_PROJECTION.

4. Explicar glMatrixMode(...).

A função glMatrixMode define a matriz que será modificada na OpenGL. Essa matriz pode ser a GL_MODELVIEW, a GL_VIEWPORT ou a GL_PROJECTION, por exemplo. Dessa forma, ela é usada para definir a projeção, as transformações e visualização de objetos na cena, bem como a definição da janela de visualização (viewport).

5. Explicar a MODELVIEW.

A MODELVIEW contém as matrizes de visualização e modelagem. Com a MODELVIEW é possível fazer transformações geométricas nos objetos da cena e visualizá-las a partir de multiplicação de matrizes.

6. O que acontece se não for dado o comando glLoadIdentity() depois da MODELVIEW?

A MODELVIEW não está sendo reinicializada após alguma transformação geométrica. Dessa forma, como as transformações geométricas aplicadas vão sempre sendo armazenadas na MODELVIEW e ela não é reinicializada, no momento de exibir o objeto, pode não ser obtida sua posição final desejada.

7. Explicar glPushMatrix() e glPopMatrix().

Com esses comandos é possível controlar qual matriz está no topo da pilha de transformações.

A glPushMatrix() copia a matriz atual e adiciona a cópia ao topo da pilha.

A glPopMatrix() descarta a matriz do topo da pilha.

Intuitivamente, glPushMatrix() significa “lembre-se de onde você está” e a glPopMatrix significa “volte para onde estava”.

8. Comparar as projeções paralela, perspectiva e suas diferenças.

Na projeção paralela o centro de projeção é localizado no infinito, sendo todas as linhas de projeção paralelas entre si. Devido a isso, a distância entre a câmera e a cena não afeta o tamanho dos objetos.

A projeção perspectiva representa a cena vista de um ponto de observação localizado a uma distância finita. Dessa forma, a distância entre a câmera e a cena afeta o tamanho dos objetos, fazendo com que um objeto posicionado no fundo, pareça menor que um objeto de mesmo tamanho posicionado mais à frente. A projeção perspectiva é similar ao olho humano.

9. Comparação das projeções paralela e perspectiva com a câmera.

Na projeção paralela, o processo de mapeamento é feito pelos raios de projeção paralelos e assume-se que todos os raios que atingem a câmera são paralelos à direção de projeção. Na projeção perspectiva, todos os raios convergem para um ponto comum, denominado ponto de observação. Nesse caso é importante determinar o ângulo de visão da câmera.

10. Diferenciar glFrustum e gluPerspective.

A semelhança é que ambas selecionam uma projeção perspectiva.

A principal diferença entre elas é que a gluPerspective cria um volume simétrico de visualização a partir de um ângulo de visada, enquanto que a glFrustum cria um volume não necessariamente simétrico.

11. Explicar frustum de acordo com seus parâmetros (volume de projeção, foco da tela).

Ao adicionar near e far a planos que são perpendiculares ao eixo z (e paralelos ao plano de visualização), limitamos a visualização da projeção perspectiva para formar uma pirâmide truncada, ou frustum, que é um volume de visualização. O parâmetro near é a menor distância desejada para um objeto visível. Far é a maior distância desejada para um objeto visível.

12. glFrustum (-1, 1 , -1, 1, 1, 1). Desenhar e dizer onde está o observador, volume de projeção e cena.

O observador está no mesmo local da posição máxima visível de um objeto, porque os parâmetros near e far possuem o mesmo valor. Logo, nenhum objeto pode ser visualizado.

13. O que são Buffers? Qual sua utilidade para a computação gráfica?

Buffers são bancos de memória (Área de armazenamento) com o objetivo de armazenar conjuntos de valores que contém informações da imagem. Sua utilidade na CG está desde a possibilidade de se utilizar cores na imagem até em gerar imagens com realismo através da utilização de buffer de profundidade, por exemplo.

14. Qual a diferença entre o buffer de cores (color buffer) e o buffer de profundidade (depth buffer) ?

O buffer de cores armazena informações de cores para cada pixel, no modo RGBA ou Indexado. O buffer de profundidade armazena um valor de profundidade para cada pixel.

15. Explicar GLUT_SINGLE, GLUT_DOUBLE e GLUT_STEREO.

GLUT_SINGLE usa um único buffer de imagem. Ele aguarda a formação da imagem para depois exibi-la e libera a exibição da imagem com o glFlush().

O GLUT_SINGLE têm algumas desvantagens:

É preciso esperar a imagem completar para depois exibi-la, o que implica em falhas na visualização interativa. Há um descompasso entre a geração e a exibição.

GLUT_DOUBLE usa dois buffers de imagem, um para exibição e outro para receber a nova imagem em processamento. Ele aumenta a eficiência em sistemas interativos.

GLUT_STEREO usa quatro buffers para exibição. Dois buffers de exibição mais dois buffers de desenho. Possui aplicação em geração de pares de imagem. Além disso, ele necessita de hardware específico.

16. Explique Máscara de Cores (glColorMask) e dê exemplo:

A máscara de cores é utilizada em operações lógicas AND para escrita nos buffers de cor. glColorMask() permite selecionar um buffer específico para escrita, no padrão RGBA.

Exemplo: trecho de código em OpenGL

```
glColorMask(1,0,0,0); // Apenas o buffer vermelho habilitado
glutSolidTeapot(1.0);
```

```
glColorMask(0,0,1,0); // Apenas o buffer azul habilitado
glTranslated(1,0,0); // Translada o objeto de uma unidade em x.
glutSolidTeapot(1.0);
```

```
glFlush();
```

```
...
```

```
/* Desenha um teapot sobreposto no outro. A sincronização dá a cor magenta */
```

17. Para quê serve a reta normal?

Para identificar a face visível do objeto no plano que será processada e exibida, enquanto o seu interior, que não é visível, não é processado. Podemos citar como exemplo o caso de uma montanha em jogos 3D, onde apenas a parte de fora(visível) é renderizada. Isso evita o desperdício de memória e processamento. Tem importância também nos cálculos para efeitos de iluminação em superfícies.

18. Explique o z-buffer e seu funcionamento.

O z-buffer é um algoritmo utilizado para calcular a distância do observador e remover superfícies ou partes ocultas de objetos sobrepostos. Possui alto custo de memória, pois aloca memória com dimensões idênticas à da tela, ou seja, utiliza buffer de imagem mais buffer de profundidade.

Funcionamento...

19. Desenho das projeções frustum e ortogonal

- a. onde está o observador ?
 - i. Por padrão, o observador está na origem ou, se transladado ou rotacionado, na posição exata após a transformação. `glTranslate(0, 0, -30)` implica que o observador está em (0, 0, -30).
- b. onde está o centro da cena ?
 - i. Sempre na origem: (0,0,0)
- c. qual o volume da projeção ?
 - i. entre near e far

20. Explicar Viewport

A Viewport ou janela de visualização é a área retangular da janela gráfica onde é desenhada a imagem final produzida. Ela é medida em coordenadas que refletem a posição dos pixels em relação à sua margem inferior esquerda. A proporção vertical e horizontal dessa janela permite a produção de distorções na apresentação da imagem.

21. A ordem de transformações altera na GL_MODELVIEW? Onde o objeto vai ficar depois? Explique.

Sim, pois a cada transformação a OpenGL cria uma matriz de transformação específica para ela e a seguir multiplica essa matriz pela matriz MODELVIEW atual. A posição final do objeto é obtida pela multiplicação dessa matriz pelos vértices do objeto.

22. Desenhar glViewport (100, 50, 200, 100)

23. Explicar um trecho de código

Erros:

1. Tentativa de desenhar na matriz de projeção
2. loadIdentity não chamada antes de desenhar
3. ddd

**24. Qual a diferença entre os modelos de iluminação local e global?
Quais as vantagens e desvantagens de cada método?**

Os modelos de iluminação locais tratam da contribuição direta de uma fonte de luz (considera o objeto individualmente). Esses modelos levam em conta um ponto individual em uma superfície e as fontes de luz que o iluminam diretamente. Produzem bom resultado e têm cálculos mais rápidos, porém consideram os objetos individualmente, o que implica em menor realismo.

Modelos de iluminação globais tratam da contribuição indireta de fontes de luz, bem como reflexões de superfícies. Levam em conta a luz que é refletida a partir de outros objetos. Nesses modelos toda a cena é considerada, o que implica em maior realismo, porém com um custo computacional elevado.

25. Explique as diferenças dos modelos de Phong e de Gouraud.

Ambos buscam obter mais suavidade na exibição de objetos formados por faces. (Se quiser apenas a diferença, considere o vermelho).

A iluminação de Gouraud elimina a descontinuidade entre os polígonos de uma superfície por interpolação das intensidades. Possui efeitos luminosos muito localizados, o que implica em pontos de brilho especular atenuados. Além disso, os efeitos luminosos no meio do polígono podem ser complicados.

Utiliza a normal de cada polígono para encontrar a normal dos vértices da superfícies. Calcula a intensidade de cada vértice e aplica a iluminação de cada polígono através da interpolação das intensidades dos vértices para cada borda.

A iluminação de Phong elimina a descontinuidade entre os polígonos de uma superfície por interpolação dos vetores normais além das intensidades (Gouraud).

Resolve o problema do meio polígono, porém é mais custoso. No modelo de Phong, a luz em qualquer ponto é composta por três componentes: luz difusa, luz especular e luz ambiente. Essas três componentes são aditivas e determinam o aspecto final da iluminação e da cor de um determinado ponto na cena ou da superfície de um determinado polígono plano contido nela.

26. Gouraud. Vantagem e desvantagem.

Eficiência no meio local e a desvantagem é que não funciona globalmente. Falar sobre local e global. [PRECISA SER ELABORADA MELHOR]
a iluminação em superfícies pode ser exibida com formas não esperadas e a intensidade de interpolação linear pode causar listra brilhante ou escura na superfície

27. Explique iluminação global e o método de ray-tracing.

28. Explicar gluLookAt() de acordo com seus parâmetros

29. Explicar gluPerspective() de acordo com seus parâmetros

`gluPerspective (alfa, aspecto, near, far)`

A função `gluPerspective` cria um volume simétrico de visualização a partir de um ângulo de visada.

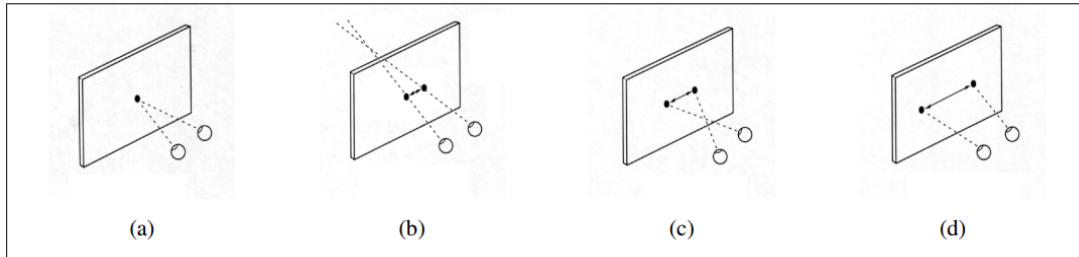
Alfa é o ângulo em y para determinar a altura do volume;
aspecto é a razão entre a largura e a altura do volume;
near é a menor distância desejada para um objeto visível;
far é a maior distância desejada para um objeto visível.

30. O que é estereoscopia? Qual a importância da paralaxe?

A estereoscopia é a ciência e arte que trabalha com imagens para produzir um modelo visual tridimensional com características análogas às características da mesma imagem quando vista através da visão binocular real.

A paralaxe (distância horizontal entre as imagens esquerda e direita) é importante porque o seu valor no par estéreo determinará a distância ou intervalo horizontal entre quaisquer dois pontos nas imagens.

31. Associe e explique cada paralaxe.



(a) Paralaxe zero.

Quando os pares estéreos possuem paralaxe zero, não há qualquer intervalo entre as imagens.

(b) Paralaxe positiva.

É possível notar profundidade na imagem de fusão.

(c) Paralaxe negativa.

Ocorre quando as linhas de visão estão cruzadas, ou seja, o olho esquerdo visualiza a imagem direita e o olho direito visualiza a imagem esquerda.

(d) Paralaxe divergente

Ocorre quando seu valor é maior que o espaçamento interocular. Esta situação, no entanto, não ocorre no mundo real e deve ser evitada nos pares estéreos devido ao grande desconforto gerado ao observador.

32. Explicar as diferenças entre os métodos off-axis, on-axis e rotação.

Na projeção off-axis assume-se a existência de dois centros de projeção, sendo que a visão esquerda é produzida baseada no centro de projeção esquerdo e a visão direita é produzida baseada no centro de projeção direito. A assimetria dos dois volumes de visualização está relacionada ao efeito de imagem “saindo da tela”. Para que isso ocorra APENAS a função `glFrustum` pode ser utilizada, visto que ela permite a definição do formato assimétrico do volume de visualização. Este é o método que mais se aproxima do modo como o ser humano enxerga o mundo real.

Na projeção on-axis é utilizado apenas um centro de projeção em conjunto com translações horizontais dos dados. Neste caso, a obtenção de cada imagem do par estereoscópico é feita através de três passos: translação da imagem para a direita ou esquerda, projeção perspectiva e translação da imagem para o sentido contrário da primeira translação.

No método de rotação para a obtenção do par estereoscópico, observa-se que esta técnica é bastante rápida computacionalmente, uma vez que as imagens são obtidas através da simples rotação vertical do centro de projeção. Neste método deve ser utilizada projeção paralela para evitar o efeito de paralaxe vertical.

33. Explicar o que está ocorrendo no trecho de código abaixo:

```

glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluPerspective(60, 4/3, dnear, dfar);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
// imagem esquerda
gluLookAt(x0,y0,z0, xref,yref,zref, Vx,Vy,Vz);
glutWireTeapot(30);
// imagem direita
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
gluLookAt(x0,y0,z0, xref,yref,zref, Vx,Vy,Vz);
glRotated(4,0,1,0);
glutWireTeapot(30);

```

O método de rotação está sendo implementado com projeção perspectiva, fazendo com que a imagem 3D resultante apresente paralaxe vertical e distorções que afetam a qualidade da imagem.

34. Como o buffer de profundidade afeta a máscara de cores no anaglifo ?

O anaglifo trabalha com lentes vermelha e azul, supondo que seria habilitado apenas o buffer vermelho para a escrita da imagem esquerda do par estereoscópico e o buffer azul seria habilitado apenas para a escrita da imagem direita. Desse modo, durante a apresentação no monitor será observada uma grande imagem magenta com deslocamentos à esquerda e à direita em vermelho ou azul. A cor magenta é o resultado da combinação do que foi desenhado no buffer azul com o que foi desenhado no buffer vermelho. O resultado disso é que apenas um dos olhos percebe cada uma das cores, mesmo elas se referindo ao mesmo objeto. Na tentativa de juntar as duas imagens, o cérebro sobrepõe uma à outra, dando a sensação de profundidade.