

DISCIPLINA: Computação Gráfica

SIGLA: CGR0001

IMPORTANTE: *utilize a apostila da disciplina assim como bibliografia e outros materiais para sua resposta. Indique a bibliografia utilizada para sua resposta.*

1) São características de um modelo:

- a) Capacidade de criar, representar, simular
- b) Capacidade de processar, visualizar, modificar
- c) Capacidade de modificar, excluir, analisar

2) Onda eletromagnética caracterizada por uma frequência:

- a) Luz
- b) Água
- c) Cor

3) Datiloscopia trabalha com quais sub áreas da CG?

- a) Processamento de Imagem + Visão Computacional
- b) Processamento de Imagem + Síntese de Imagens
- c) Visão Computacional + Síntese de Imagens
- d) Síntese de Imagens + Modelagem Geométrica

4) Jogos eletrônicos tradicionais (ou "games") trabalham com quais sub áreas da CG?

- a) Processamento de Imagem + Visão Computacional
- b) Processamento de Imagem + Síntese de Imagens
- c) Visão Computacional + Síntese de Imagens
- d) Síntese de Imagens + Modelagem Geométrica

5) Active games (ou "games/jogos ativos"), como D2R e Moveltrando criados aqui no DCC, trabalham principalmente com quais sub áreas da CG?

- a) Processamento de Imagem + Visão Computacional + Síntese de Imagem
- b) Processamento de Imagem + Síntese de Imagens + Modelagem Geométrica
- c) Visão Computacional + Modelagem Geométrica + Processamento de Imagens
- d) Síntese de Imagens + Modelagem Geométrica + Visão Computacional

6) O modelo de cor RGB é representado geometricamente por um cubo. Cada cor é representada por 3 números que representam as quantidades de primárias Vermelho (*Red*), Verde (*Green*) e Azul (*Blue*). Faça um desenho deste cubo com os nomes usuais das cores representadas nos vértices do cubo. Porque foram desenvolvidos outros modelos de cores como HSV? Explique os modelos HSV e HLS (incluindo sua representação geométrica) e cite alguma vantagem e desvantagem (ou dificuldade) no seu uso.

7) Cite pelo menos três dispositivos utilizados para Computação Gráfica ou Realidade Virtual e explique o funcionamento dos mesmos.

8) Quais os TIPOS de OBJETOS que podem ser modelados por computador? Cite um tipo de objeto que (ainda) não pode ser modelado pelo computador e explique o motivo da impossibilidade.

9) Das diversas formas de representação de objetos, indique quais delas são por Superfície e quais podem ser consideradas como Sólidos.

10) Qual a melhor forma de Representação de Sólidos em sua opinião? Justifique a resposta.

11) Dado um objeto definido por um arquivo gerado por um determinado programa de modelagem, como é possível validar se o objeto é um poliedro válido ou não? Somente a fórmula de Euler é suficiente?

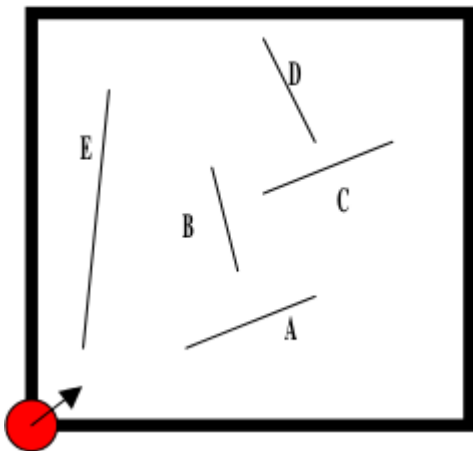
12) Cite e descreva um método para criação (modelagem) de vegetações como árvores, flores e similares.

13) Como são modelados objetos naturais como fogos, fumaça, explosão, nuvens, cachoeiras, chuva e outras superfícies deformáveis?

14) Programas de modelagem geométrica como Maya e 3DS utilizam diversos métodos de modelagem (Sweep, CSG,...) mas utilizam outra nomenclatura e algumas vezes com parâmetros distintos. Em sua opinião, qual o motivo desta diferença?

15) Cite as principais diferenças entre as Octrees e BSP-Trees, descrevendo também suas vantagens e desvantagens (uma em relação à outra).

16) Crie uma BSP-tree a partir dos segmentos de linha definidos na figura abaixo. Apresente a estrutura desta árvore. Usando a posição do observador no canto inferior esquerdo mostrado na figura e a BSP-tree criada, aplique o algoritmo de ordenação de visibilidade de BSP-trees, enumere os nodos da árvore (incluindo os nodos folhas) visitados seguindo a estratégia back-to-front (de trás para frente) e apresente a ordenação dos segmentos produzidos.



17) Qual a vantagem e desvantagem de se utilizar estrutura de dados como a Quad-Edge para armazenar e manipular (modelar) objetos 3D?

18) Determine quais são as etapas para síntese de imagens 2D e descreva brevemente cada uma. Que tipo de operações devemos fazer nessas etapas?

19) Em relação aos algoritmos de varredura, qual a diferença entre o método incremental (DDA) e o do Ponto Médio (Bresenham)? Cite vantagens e desvantagens.

20) Usando seu conhecimento de álgebra vetorial, proponha um método que determine se um ponto está dentro ou não de um polígono T qualquer, especificado pelos seus vértices.

21) O algoritmo de Cohen-Sutherland estabelece código de 4 bits para os pontos extremos das linhas a serem recortadas. Qual o número máximo de bits diferentes para um ponto qualquer?

22) Em relação às transformações geométricas que podemos aplicar aos modelos, existe alguma relação entre as operações de escala e reflexão? Discurse sobre sua resposta.

23) Algumas sequências de transformações geométricas são comutativas, ou seja, tanto faz a ordem em que elas são aplicadas ($M_1M_2 = M_2M_1$). Decida e prove se as seguintes transformações são comutativas ou não:

- a. duas escalas em sequência
- b. duas rotações em sequência
- c. duas translações em sequência
- d. uma escala seguida de uma rotação
- e. uma translação seguida de uma escala
- f. uma rotação seguida de uma translação

24) Qual a principal vantagem do uso de coordenadas homogêneas em computação gráfica?

25) Considere a matriz de rotação 4x4 abaixo

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

Qual o ângulo e o eixo de rotação?

26) O mapeamento de um ponto P em coordenadas de janela (*window*) para um ponto Q em coordenadas de janela de exibição (*viewport*) é feito por uma translação, seguida de uma escala e de outra translação:

$$(q_x, q_y) = [(p_x - t1_x) s_x, (p_y - t1_y) s_y] + (t2_x, t2_y)$$

Expresse a transformação acima como multiplicação de 3 matrizes utilizando coordenadas homogêneas e obtenha a matriz resultante desta multiplicação.

27) Ao realizar as operações de multiplicação entre um vetor e uma matriz, devemos saber se o vetor é em linha ou em coluna. Explique qual a diferença em fazer as operações entre vetor linha e vetor coluna.

28) Qual o motivo de definirmos o volume de visualização (*View Volume*)? Isto é realmente necessário? Justifique sua resposta.

29) Defina o que é *z-buffer*. Explique porque você acredita ou não que esta técnica é na média independente do número de primitivas numa cena (você pode pensar em polígonos como primitivas).

30) Explique a filtragem de faces traseiras (*back face culling*). Em média, quantas faces são eliminadas utilizando-se esta filtragem?

31) Qual a condição necessária que o objeto deve satisfazer para que o teste utilizado na filtragem de faces traseiras (*back face culling*) funcione?

32) Descreva como o vetor normal pode ser utilizado para auxiliar no processo de remoção de superfícies ocultas.

33) Com o observador na posição (2,5,3) e dados os pontos a seguir com seus respectivos vetores normais, quais pontos podem ser eliminados pelo teste de *back face culling*?

P0: (5, 2, -2); N0: (-1, 0, 0)

P1: (-1, 7, -5); N1: (0, 1, 0)

P2: (4, 6, 1); N2: (0, 0, -1)

- (a) Nenhum deles
- (b) Somente P0 e P2
- (c) Somente P1 e P2
- (d) Todos eles

34) Sobre o método *z-buffer* (*depth buffer*), responda as questões abaixo.

a) Suponha que a componente z do *z-buffer* esteja sendo armazenada em 1 byte e que o intervalo possível de valores de z é (0.3,1) inclusive. Qual a distância mínima entre 2 objetos para que possa ser determinado que um se encontra em frente ao outro? Em outras palavras, qual a precisão da representação em 1 byte apenas.

b) Caso o intervalo possível de valores fosse (0.5,1) mudaria sua resposta em relação ao item anterior? Como?

c) Caso a precisão do *z-buffer* fosse de 2 bytes, mudaria sua resposta em relação ao item 'a)' acima? Como?

d) Qual a diferença entre *frame-buffer* e *z-buffer*?

e) Uma das vantagens do *z-buffer* é que as primitivas podem ser enviadas em qualquer ordem para processamento. Isto implica que duas imagens obtidas pela aplicação de *z-buffer* onde as mesmas primitivas foram enviadas em ordem diferente serão exatamente iguais? Justifique.

35) Qual a diferença principal entre as bibliotecas GL e GLU de OpenGL?

36) Considere a seguinte chamada da função `gluLookat`:

```
gluLookAt(1.0, 2.0, 3.0, 2.0, 4.0, 5.0, 0.0, 1.0, 0.0);
```

a) Qual é o VRP (*view reference point*)?

b) Qual é o VPN (*view plane normal*)?

c) Qual é o VUP (*view-up vector*)?

37) Considere a seguinte sequência de chamadas OpenGL:

```
glColor3f(1.0,1.0,1.0);
glVertex3f(1.0,1.0,1.0);
glVertex3f(2.0,2.0,3.0);
glColor3f(0.0,1.0,1.0);
```

(a) Qual a cor do vértice (1,1,1)?

(b) Qual a cor do vértice (2,2,3)?

38) Dado o código OpenGL abaixo, escreva a matriz resultante após a chamada `glRotate` (você não precisa fazer a multiplicação propriamente dita, apenas deixar indicado).

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glTranslatef( 5, 2, 1 );
glRotatef( 1.0f, 0.0, 0.0, 1.0f );
```

39) Qual a função da etapa de projeção no *pipeline* de visualização 3D?

40) No modelo de iluminação proposto por Phong, explique:

- A. como a lei de cosseno de Lambert é utilizada e qual sua finalidade;
- B. como é calculada a componente especular;
- C. para que serve a componente ambiente;
- D. de que forma é aplicada uma atenuação atmosférica.

41) Para que servem os algoritmos de *shading* (tonalização/sombreamento)?

42) Descreva como são realizados e explique a diferença entre os seguintes modelos de *shading*: *Flat*, Gouraud e Phong. Como utilizar cada um deles em OpenGL?

43) Quais as semelhanças e diferenças entre o tipo de luz pontual e a direcional/paralela. Como elas são diferenciadas pela biblioteca OpenGL?

44) Na imagem abaixo, informe qual o tipo de luz foi utilizada e informe o que falta para um maior realismo (dica: sol e sombras).



45) Como podemos definir uma luz do tipo *Spot*? Cite onde é possível aplicar este tipo de luz, ou seja, quais objetos podemos “modelar” com este tipo de luz?

46) Para que serve uma luz emitente (*glowing object*)? Como ela é implementada em OpenGL?

47) Cite e comente as principais diferenças entre modelos de iluminação locais e globais. Cite exemplo de sistemas que utilizam iluminação local. Cite exemplo de algoritmos ou métodos de iluminação global.

48) Em sistemas como *Ray Casting*, *Ray Tracing* e *Path Tracing*, precisamos traçar uma linha e encontrar o objeto mais próximo interceptado por essa reta. Mostre como é possível realizar a intersecção dessa reta com um triângulo.

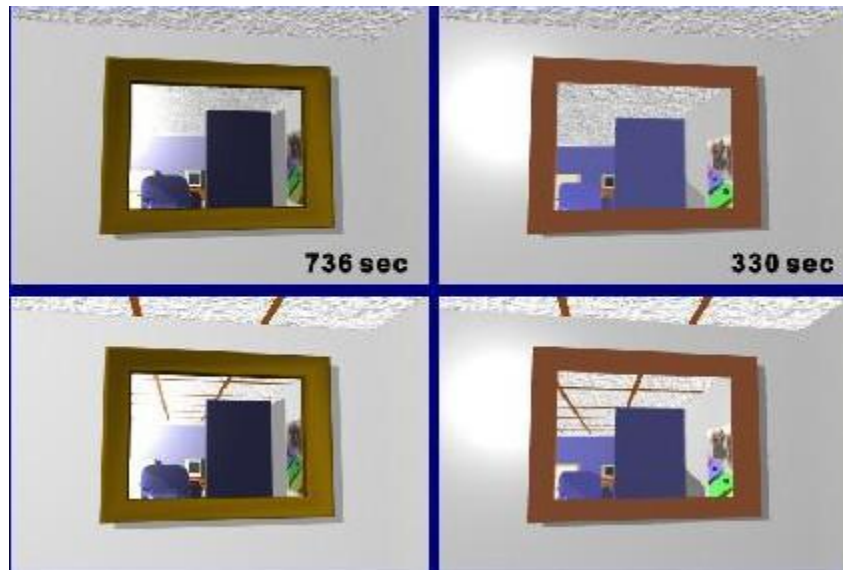
49) Defina como pode ser realizada a intersecção de uma linha com uma esfera? Mostre um exemplo numérico.

50) Seria possível não utilizar texturas em sistemas de computação gráfica? Como devemos proceder utilizando apenas cores e materiais para criar objetos realistas (pessoas, animais,...)?

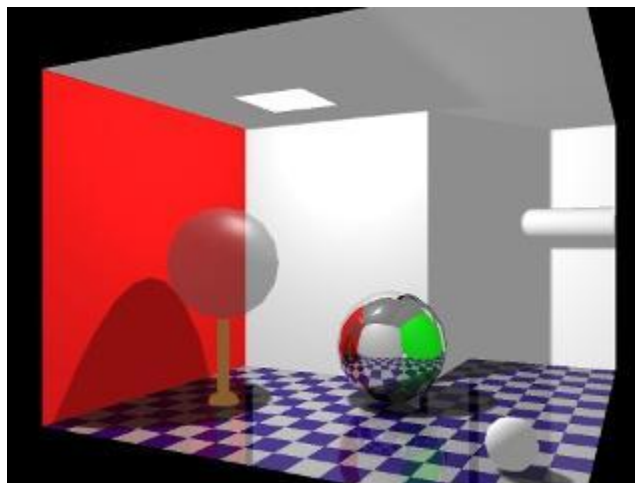
51) Qual a vantagem de definir a textura em um espaço unitário?

52) Mipmapping (MIP - *Multum In Parvo* = muito em pouco espaço) é um método de aceleração do cálculo da cor média para áreas de amostragem da textura. Explique como é realizado e mostre uma figura comparativa de uma mesma cena renderizada com e sem mipmapping.

53) A figura abaixo (Rosalee Wolfe / Siggraph Education) mostra imagens geradas por *Ray Tracing* (esquerda) e pela técnica de *Environment mapping* (direita). Cite erros nas imagens e explique porque ocorreram as falhas.



54) Na imagem abaixo (disponível em http://www.winosi.onlinehome.de/Gallery_t14_03.htm) renderizadas utilizando OpenGL, cite quais tipos de mapeamento de textura foram utilizados. Explique qual o motivo (efeito) da sua utilização.



55) Pesquise e explique sobre alternativas ao uso de *Shadow Mapping* para geração de sombras em cenas renderizadas por algoritmos de iluminação local (em jogos).

56) Explique a diferença entre *Bump Mapping* e *Normal Mapping*.

57) Como *Displacement Mapping* e *Relief Mapping* diferem da técnica de *Normal Mapping*? Defina as vantagens e desvantagens de cada uma.

58) Qual a relação entre as sub-áreas Síntese de Imagens e Visão Computacional? Defina as principais etapas no processo de Visão Computacional, explicando cada uma delas.

59) Explique como é calculada a componente especular no algoritmo de Ray Tracing (dica: raio primário).

60) Explique como é calculada a componente difusa no algoritmo Ray Tracing. Mostre uma árvore de intersecções e explique como a componente difusa é composta.

61) Quais as possíveis otimizações para redução do número de raios no algoritmo de Ray Tracing?

62) Para uma cena realista (filme de animação, por exemplo), qual o melhor algoritmo de iluminação global?

63) A fim de otimizar o processamento do traçado de raios (Ray Tracing), como podemos diminuir do número de intersecções a testar?

64) Para otimizar o Ray Tracing, como podemos diminuir do número de raios a processar?

65) Como melhorar os efeitos de iluminação difusa no Ray Tracing? (dica: infinitas direções?)

66) Explique o funcionamento do algoritmo de iluminação Radiosidade. (livros!)

67) Como o método do hemicubo auxilia no aumento de desempenho no cálculo da Radiosidade?

68) O método Progressive Refinement Radiosity gera um resultado melhor ou pior em relação ao algoritmo tradicional para cálculo da Radiosidade?

69) Como é realizada a criação da imagem (*rendering*) através de Radiosidade?

70) Quais problemas o algoritmo de Path Tracing tenta resolver em relação ao algoritmo tradicional de Ray Tracing?

71) Descreva as diferenças e semelhanças entre o algoritmo de Monte Carlo Path Tracing e o Ray Tracing proposto por Greg Ward, Francis Rubinstein e Robert Clear no

SIGGRAPH de 1988 (que dispara um número N de raios difusos no ambiente em direções aleatórias ao redor da normal)?

72) Cite alguns programas (comerciais ou não) para *rendering* baseados em iluminação global e explique quais técnicas cada um utiliza.