

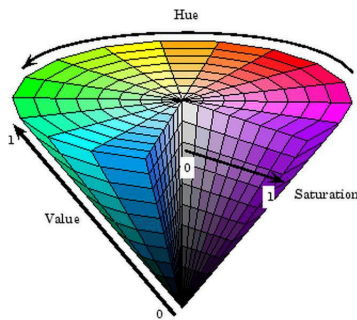
# CGR LISTA 1

Aluno: Gustavo de Souza

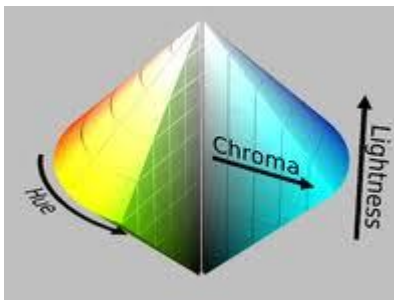
- 1.B
- 2.A
- 3.A
- 4.D
- 5.A

6. Os outros modelos foram desenvolvidos para terem uma representação mais compreensível para os seres humanos das divisões de cores.

HSV (Hue, Saturation, Value): É uma matriz que forma um cone/cilindro e tanto sua profundidade quanto a evolução dos ângulos contribuem para a saturação da cor:



HLS (Hue, Lightness, Saturation): Mesma ideia que a anterior, o controle da luz, saturação e a cor é feita por um cone duplo:



7. Oculos de realidade virtual: para projetar em uma tela ou lentes o ambiente virtual para um usuário;

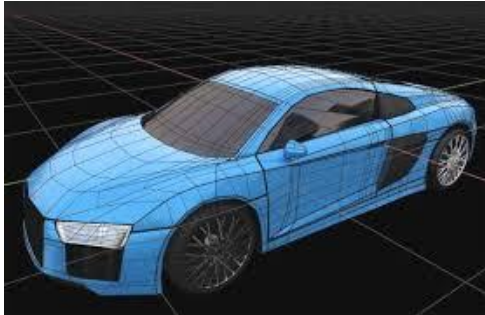
Controle: para permitir que o usuário interaja com o mundo virtual;

Câmera de profundidade: para identificar e ambientar o usuário dentro do mundo virtual de forma que condiz com seu espaço e movimentação física.

8. Objetos modeláveis: Objetos Geométricos, Objetos Orgânicos, Objetos Procedurais, Objetos Volumétricos, Objetos Físicos / Dinâmicos e etc.

Objetos não modeláveis: uma consciência humana.

9. Representações de superfícies são aquelas que representam uma malha de um objeto, sua casca externa (pele):



Já os sólidos são formas concretas que possuem características internas e externas e são capazes de serem utilizados em ambientes virtuais com físicas aplicadas.

10. **Boundary Representation (B-Rep)** é a forma mais completa e versátil para representar sólidos em aplicações que exigem precisão, controle topológico e integração com superfícies suaves.

11. Não, a fórmula de Euler não é suficiente sozinha. A Validação Completa deve ser feita em três partes:

Teste da Fórmula de Euler, Teste das Restrições Topológicas e Testes das Restrições Geométricas

Verificações geométricas e topológicas, como:

Verificar se todas as arestas pertencem a exatamente duas faces;

Checar orientação das normais;

Testar para buracos ou auto-interseções.

12. **L-Systems (Lindenmayer Systems):** É um sistema de gramática formal criado por Aristid Lindenmayer, originalmente para modelar crescimento de plantas reais. Com o tempo, passou a ser usado em computação gráfica para gerar vegetações realistas de forma procedural (automática). Ele reproduz bem o crescimento fractal de plantas reais, além de gerar variações infinitas com pequenas mudanças nas regras e ser leve computacionalmente (só texto e regras).

13. Há vários métodos, vou citar o principal que é a simulação de partículas, consiste basicamente na representação de milhares (ou milhões) de partículas pequenas. Cada partícula tem propriedades como posição, velocidade, cor, opacidade e tempo de vida além de que elas são afetadas por forças físicas: gravidade, vento, turbulência, colisão, etc.

14. A troca da nomenclatura muito provavelmente tem um caráter comercial e de fácil acesso, pois não é necessário manter o rigor acadêmico e matemático para um software comercial.

15. As octrees são árvores hierárquicas que são basicamente áreas que particionam o volume de um dado objeto, as octrees possuem sempre nos com áreas menores (8

subareas) para ir analisando e definindo todas as regioes de um dado objeto tridimensional, a diferenca das octrees para as BSP-tress é que as BSPs sempre subdividem seus nós na metade (2 subareas), abrindo assim um leque para maior representacoes geometricas, seu ponto positivo é que devido sua estrutura é possivel subdividir e tratar figuras com contornos mais complexos, porem o lado negativo é que a subdivisao pode escalonar a um ponto de elevada complexidade. Ja as Octrees sao otimas pois sua estrutura é previsivel e controlavel, porem nao permite segmentar todos os tipos de volumes, principalmente os mais complexos com furos.

## **16. semi entendi**

### **17. PULEI**

### **18. Etapas:**

Sistema de referencia do objeto (SRO): é voce definir como a imagem (objeto) será exibida com instancias (estigada, tamanho, parametros em geral).

Sistema de referencia do universo (SRU): definir como todos os objetos irão estar exibidos no grande montante da cena

Sistema de referencia da selecao (SRS): é a definicao da area que será exibida de fato (cortada)

Sistema de referencia de dispositivo (SRD): é a definicao da saida da imagem (em monitor, oculos VR, janela flutuante e por ai vai)

19. Metodo incremental (DDA) é uma pura aplicacao da equacao da reta onde  $y-y_0=m(x-x_0)$ , para cada x de uma equacao tera seu y plotado conforme a equacao definida apartir de dois pontos, é uma abordagem boa pois é de simples porém a necessidade de arredondamentos em virtude de divisoes faz com que o modelo perca um pouco a acuracia. Ja o metodo do ponto medio utiliza apenas operacoes aritméticas simples como subtracao e soma, ele se baseia no fato de fazer uma comparacao com o pixel adjacente mais proximo da linha ideal, ele sempre busca o ponto medio entre os pixels e determina se o mesmo esta acima ou abaixo da linha ideal, tornando o processo mais rapido em virtude de nao utilizar pontos flutuantes, porem é um metodo nao tao trivial de se implementar inicialmente.

### **20. PULEI**

### **21. O maximo de bits diferentes é 4.**

22. A reflexão é conceitualmente e matematicamente um subconjunto da operação de escala, especificamente o caso em que os fatores de escala ao longo de um ou mais eixos são -1. Utilizar uma escala negativa qualquer (e.g., -2) combina os efeitos de uma reflexão e uma mudança de tamanho. Embora relacionadas, em softwares de modelagem elas são

frequentemente apresentadas como ferramentas distintas porque servem a propósitos de design diferentes: uma para criar imagens espelhadas, a outra para alterar dimensões.

23.

Comutativas:

a (duas escalas em sequencia)

b (duas rotações em sequencia)

c (duas translações em sequencia)

Não-Comutativas:

d

e

f

Pois pode haver de eu querer redimensionar ou transladar em relação a um eixo, e se esse eixo for girado/movido, o resultado será diferente.

24. A principal vantagem do uso de coordenadas homogêneas em computação gráfica é a unificação da representação de todas as transformações geométricas afins (como translação, rotação, escala e cisalhamento) e da projeção perspectiva como uma única operação de multiplicação de matrizes.

25 e 26. PULEI

27. A escolha entre usar vetores linha ou coluna dita a ordem da operação de transformação (vetor linha pré-multiplica a matriz,  $V' = V \cdot M$ , enquanto vetor coluna pós-multiplica,  $V' = M \cdot V$ ) e exige que a própria matriz de transformação seja transposta para realizar a mesma operação geométrica em cada caso. Isso também inverte a ordem de concatenação de múltiplas transformações e torna essencial saber qual convenção (linha ou coluna, comum em DirectX ou OpenGL, respectivamente) está sendo usada para construir e aplicar as matrizes corretamente.

28. Sim, pois assim efetuamos uma limitação visual daquilo que será exibido em tela e pode ser aplicado lógicas para evitar a renderização de objetos que não pertencem ao view volume da câmera, tornando assim o sistema mais otimizado.

29. O Z-buffer (ou buffer de profundidade) é uma estrutura de dados, tipicamente uma matriz 2D com as mesmas dimensões da tela (ou do buffer de imagem), que armazena um valor de profundidade (coordenada Z, na visão da câmera) para cada pixel. Durante o processo de rasterização, ao desenhar um pixel de uma primitiva (como um polígono), a profundidade calculada para esse pixel é comparada com o valor já existente no Z-buffer naquela posição (x, y). Se a nova profundidade for menor (ou seja, mais próxima do observador, dependendo da configuração), o Z-buffer é atualizado com a nova profundidade e o buffer de cor é atualizado com a cor do pixel da nova primitiva. Caso contrário, o pixel da nova primitiva é descartado, pois está atrás de algo já desenhado.

**Conclusão:** Embora as otimizações modernas reduzam o impacto de primitivas ocultas, o trabalho fundamental do Z-buffer (leitura, comparações, escritas potenciais) escala com o número total de fragmentos gerados pelas primitivas visíveis e parcialmente visíveis na cena. Portanto, o desempenho do Z-buffer está, sim, ligado ao número de primitivas, especialmente àquelas que contribuem com pixels para a imagem final ou que estão próximas o suficiente da superfície visível para não serem descartadas por técnicas de culling muito agressivas antes da rasterização.

30. É uma técnica de otimização usada em computação gráfica para **eliminar polígonos (faces) que estão "de costas" para o observador (câmera)**, ou seja, cuja normal aponta para longe da direção de visão. Assume-se que esses polígonos pertencem a um objeto sólido e opaco, e portanto, não seriam visíveis de qualquer forma. A técnica funciona calculando o vetor normal de cada face (geralmente determinado pela ordem dos vértices, ou "winding order" - horário ou anti-horário) e comparando sua direção com a direção de visão da câmera (por exemplo, usando o produto escalar entre a normal da face e o vetor que vai da face para a câmera). Se o produto escalar indicar que o ângulo entre eles é maior que 90 graus (ou seja, a face aponta para longe), a face é descartada e não passa pelas etapas subsequentes do pipeline de renderização (como rasterização e sombreamento).

31. A condição necessária principal que o objeto (ou mais precisamente, sua malha poligonal) deve satisfazer para que a filtragem de faces traseiras (back-face culling) funcione corretamente é que ele seja modelado como um **objeto sólido e fechado (manifold)**, com uma **definição consistente da orientação das faces** (geralmente através da ordem dos vértices - winding order).

32. Ao comparar a direção da normal (N) com a direção de visão da câmera (V), geralmente através do produto escalar ( $N \cdot V$ ), o sistema identifica se a face está voltada para a câmera ou para longe dela. As faces que apontam para longe (faces traseiras, onde  $N \cdot V \leq 0$ ) são consideradas invisíveis em objetos sólidos e fechados e podem ser descartadas precocemente, evitando que passem por etapas mais custosas do pipeline de renderização, como rasterização e teste Z, otimizando significativamente o processo.

### 33. Dados:

- Posição do Observador (O) = (2, 5, 3)
- A condição para eliminação por back-face culling é  $N \cdot V \leq 0$ , onde N é o vetor normal e V é o vetor do ponto para o observador ( $V = O - P$ ).

PO:

- $P0 = (5, 2, -2)$
- $N0 = (-1, 0, 0)$
- $V0 = O - P0 = (2-5, 5-2, 3-(-2)) = (-3, 3, 5)$
- $N0 \cdot V0 = (-1)(-3) + (0)(3) + (0)(5) = 3 + 0 + 0 = 3$
- Como  $3 > 0$ , P0 **não** é eliminado (é uma face frontal).

P1:

- $P1 = (-1, 7, -5)$
- $N1 = (0, 1, 0)$
- $V1 = O - P1 = (2-(-1), 5-7, 3-(-5)) = (3, -2, 8)$
- $N1 \cdot V1 = (0)(3) + (1)(-2) + (0)(8) = 0 - 2 + 0 = -2$
- Como  $-2 \leq 0$ , P1 **é** eliminado (é uma face traseira).

P2:

- $P2 = (4, 6, 1)$
- $N2 = (0, 0, -1)$
- $V2 = O - P2 = (2-4, 5-6, 3-1) = (-2, -1, 2)$
- $N2 \cdot V2 = (0)(-2) + (0)(-1) + (-1)(2) = 0 + 0 - 2 = -2$
- Como  $-2 \leq 0$ , P2 é eliminado (é uma face traseira)

Conclusao: letra C.

34.

a) Precisão do Z-buffer com 1 byte e intervalo (0.3, 1):

- **Número de níveis:** 1 byte = 8 bits, o que permite representar  $2^8 = 256$  níveis distintos de profundidade.
- **Intervalo de profundidade:** O intervalo total é  $1.0 - 0.3 = 0.7$ .
- **Precisão:** A precisão é o tamanho de cada "passo" de profundidade que pode ser representado. Calculamos isso dividindo o intervalo total pelo número de intervalos disponíveis (que é o número de níveis - 1, pois 256 níveis definem 255 intervalos entre eles):  
 $Precisão = 0.7 / (256 - 1) = 0.7 / 255 \approx 0.002745$
- **Resposta:** A distância mínima (no espaço normalizado de Z) entre dois objetos para garantir que possam ser distinguidos é de aproximadamente 0.002745. Essa é a precisão da representação.

b) Mudança com intervalo (0.5, 1):

- **Novo intervalo:** O intervalo total agora é  $1.0 - 0.5 = 0.5$ .
- **Número de níveis:** Continua sendo 256 (1 byte).
- **Nova Precisão:**  $Precisão = 0.5 / (256 - 1) = 0.5 / 255 \approx 0.00196$
- **Resposta:** Sim, a resposta mudaria. A precisão **melhoraria** (o valor da distância mínima seria menor,  $\approx 0.00196$ ). Isso ocorre porque os mesmos 256 níveis de profundidade estão sendo distribuídos em um intervalo menor (0.5 em vez de 0.7), tornando cada passo menor e a distinção entre profundidades mais fina.

c) Mudança com precisão de 2 bytes:

- **Número de níveis:** 2 bytes = 16 bits, o que permite representar  $2^{16} = 65.536$  níveis distintos.
- **Intervalo:** Usando o intervalo do item 'a', 0.7.
- **Nova Precisão:**  $Precisão = 0.7 / (65536 - 1) = 0.7 / 65535 \approx 0.00001068$
- **Resposta:** Sim, a resposta mudaria drasticamente. A precisão seria  **muito maior**  (a distância mínima distinguível seria extremamente menor,  $\approx 0.00001068$ ). Ter muito mais bits permite representar o mesmo intervalo com uma granularidade significativamente maior.

35. GL fornece os blocos de construção essenciais e de baixo nível, enquanto GLU oferece ferramentas de conveniência de nível superior para tarefas comuns, utilizando as funções da GL por baixo dos panos.

36. PULEI

37. Branco para ambas.

38. PULEI

39. Em suma, a projeção é o passo crucial que "achata" a cena 3D de forma controlada para que ela possa ser representada em um dispositivo de exibição 2D.