UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DCA
COMPUTAÇÃO GRÁFICA

* LISTA 04 *

DISCENTE: Tiago Felipe de Souza
MATRÍCULA: 20190153105

1. Defina o problema de visibilidade.

Dado que se tem vários objetos numa imagem, como fazer para se pintar os objetos na forma correta e na sequência correta de forma que seja coerente com o ponto de vista do observador em relação a uma cena, ou seja, como pintar os objetos de forma que a oclusão que possa/esteja acontecendo fique na forma correta quando chegar na imagem.

2. Quais são os 4 algoritmos básicos de visibilidade? Coloque o pseudo-código de cada um deles.

↳ Algoritmo do pintor:

Início {
- Criar uma lista de objetos;
- ordenar os objetos dessa lista de acordo a distância do mais distante para o mais próximo;
- Pintar os objetos na ordem de distância, do mais distante para o mais próximo;

Fim }.

Algoritmo z-Buffer:

Início {

- limpa o z-Buffer (coloca para o infinito);

Rasterização {

- faz um loop por todos os objetos;
- Verificar se o objeto que está acessando tem intersecção com a imagem, ou seja, Verifica se a distância em $z(x,y) < zBuf(x,y)$;
- se for menor, substitui z por zBuf;
- calcula a contribuição de iluminação do pixel do objeto em questão na cena;

fim - Rasterização }

Fim }

Algoritmo Ray casting:

Início {

- faz loop em y;
- faz loop em x; { Para passar por toda a imagem.
- Cria um raio do ponto focal ao pixel da imagem e encontra esse vetor;
- achar a intersecção desse vetor com todas as superfícies da imagem para encontrar a primeira que é tocada;

- calcula a contribuição de iluminação e cor para o pixel da imagem;

Fim }

↳ Algoritmo Ray Tracing recursivo:

Início {

- faz loop em x;
- faz loop em y;

-apartir de um ponto focal cria-se vários raios aos pixels da imagem e encontra esses vetores;

- acha a intersecção desses vetores com todas as superfícies da imagem e depois soma tudo para obter as contribuições;

Fim }

3. Explique em detalhes as vantagens e desvantagens de 3 dos algoritmos de visibilidade.

↳ Algoritmo do pintor: [redacted]

* Vantagens - simples de implementar; se os objetos já estiverem ordenados, ele é muito rápido.

Desvantagens - Todas as vezes que mudar a perspectiva, vai ter que recalcular (reordenando a lista de objetos.

↳ Algoritmo Z-Buffer:

Vantagens - rendimento superior para muitos objetos na cena em comparação ao pintor; rápida implementação.

Desvantagens - tem que ter um espaço reservado na memória, ou seja, depende de Hardware.

↳ Algoritmo Ray Tracing:

Vantagens - fácil implementação; a qualidade das imagens são as mais foto-realistas.

Desvantagens - lento quando há muitos objetos na cena

4. Qual dos algoritmos de visibilidade você usaria se tivesse apenas um objeto, modelado com muitas faces triangulares a ser visualizado? Explique o motivo.

Eu implementaria o algoritmo ray casting, pois é o mais fácil de implementar, a qualidade das imagens são as mais realistas e ele só é lento quando há muitos objetos na cena e não é esse caso.

5. Qual o princípio básico do "forward ray-tracing"? Explique.

São denominados raios de luz o caminho que esses fótons fazem ao serem emitidos por uma fonte de luz. Esse método de renderização seguindo os caminhos de fótons é denominado ray tracing.

O forward ray-tracing segue fótons na direção que a luz viaja.

6. Qual a diferença básica entre o "Forward Ray Tracing" e o "Backward Ray Tracing" (qual o principal problema do primeiro método, resolvido pelo segundo)? O forward ray tracing é muito caro computacionalmente pois ele calcula todos os raios de luz que contribuem para imagem, ou não, pois há raios que ao refletirem saem do frustum e não contribuem para a imagem, isso causa lentidão nos cálculos e alto custo em processamento. Já o backward ray tracing calcula e gasta processamento apenas com os raios que contribuem para imagem, fazendo o sentido contrário do forward ray tracing, dos raios que vão da imagem a fonte de luz.

7. Explique a ideia básica do "ray tracing" recursivo, incluindo quais os tipos de raios recursivos a se considerar, descrevendo sucintamente cada um deles.

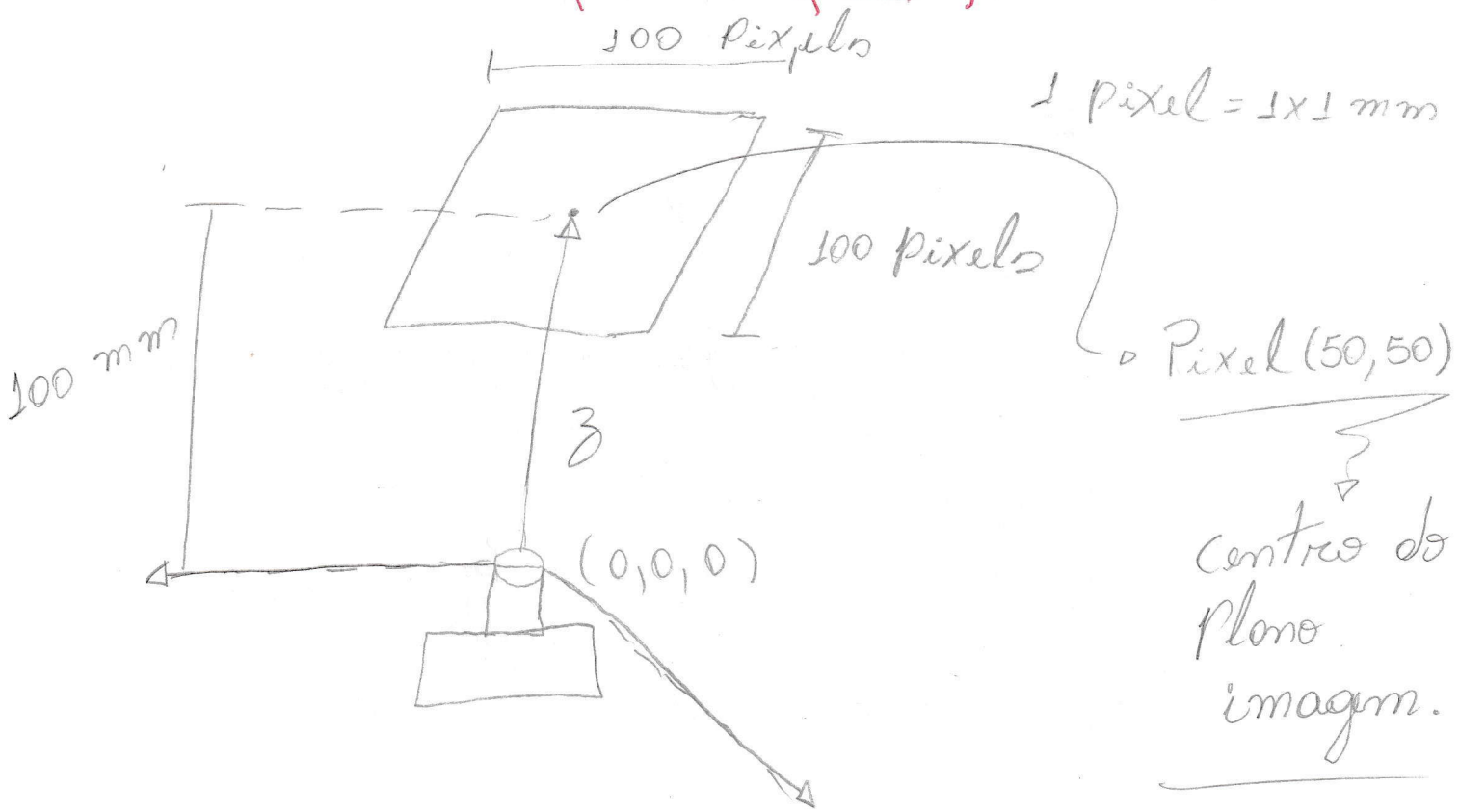
O ray tracing recursivo: traça os raios que influenciam na imagem até o objeto, vê qual a intersecção dele com o objeto, calculando também com os raios que foram refletidos e refratados, calcula a contribuição do shadowray e por fim, calcular recursivamente a contribuição de todos. (└ e traçar)

└ Tipos de raio:

* Eye rays - origina-se no olho.
* Shadow Rays - do ponto na superfície na direção da luz.
* Reflection Rays - do ponto na superfície na direção refletida.
* Transmission Rays - do ponto na superfície na direção refratada.

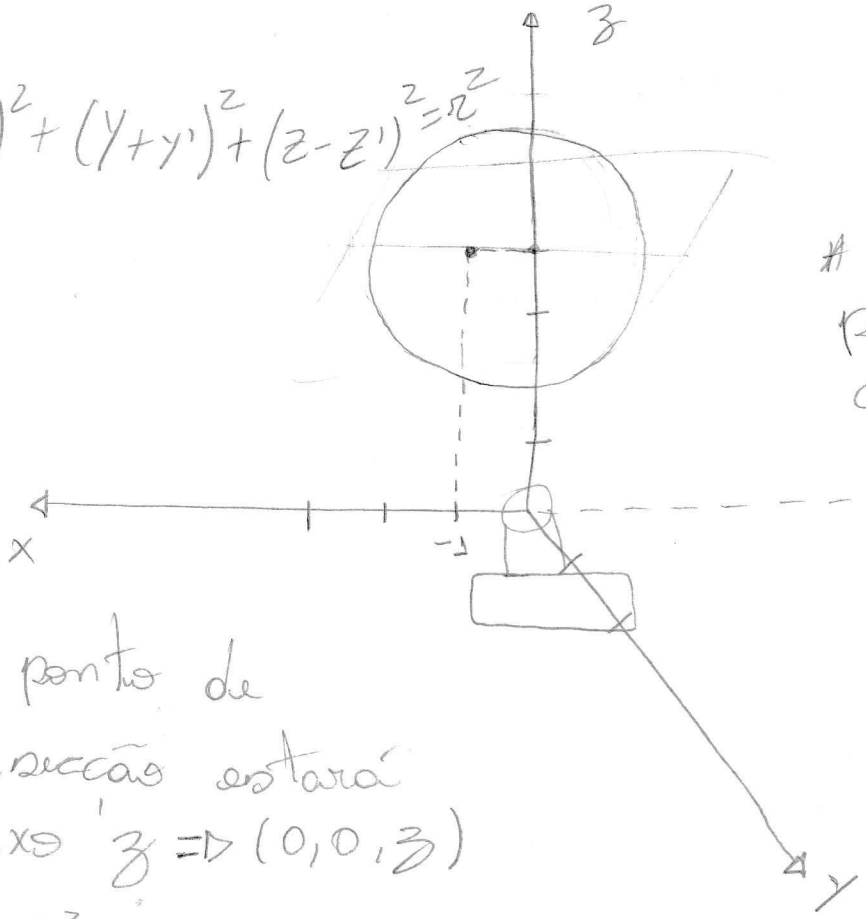
8. (Desafio - bônus) Uma câmera com distância focal de 100 mm, cujo plano imagem é formado por 100 x 100 pixels com dimensões de 1x1 mm cada um, encontra-se na origem do sistema de coordenadas com sua objetiva (eixo focal) apontado para o eixo Z do sistema. Dê o que se pede:

1. Calcule as coordenadas do ponto visível (ou Ponto de intersecção) na esfera de raio 2 que se encontra centrada em (-1, 0, 4), cuja projeção na imagem é o pixel (50, 50), sabendo que a origem do sistema de imagem é como no OpenGL (canto inferior esquerdo).



$(X+x')^2+(Y+y')^2+(z-z')^2=r^2$

$(-1,0,4)$; $r=2$



* A câmera aponta para o pixel central do plano imagem, a partir de $(0,0,0)$

* A esfera está deslocada em 1 unidade de distância de z.

* O ponto de intersecção estará no eixo z $\Rightarrow (0,0,z)$

$(x+1)^2+y^2+(z-4)^2=2^2$

$(0+1)^2+0^2+(z^2-8z+16)=4$

$z^2-8z+16-4+1=0$

$$\boxed{z^2-8z+13=0}$$

$\Delta=(-8)^2-4.1.13$

$\Delta=64-52$

$$\boxed{\Delta=12}$$

$$z=\frac{8\pm\sqrt{12}}{2.1}$$

$$z_1=\frac{8+\sqrt{12}}{2}\cong 5{,}73 \quad \times$$

$$z_2=\frac{8-\sqrt{12}}{2}\cong 2{,}27 \quad \text{MAIS PRÓXIMO!} \quad \checkmark$$

* O ponto $(0,0,2.27)$ é o ponto visível ou ponto de intersecção.