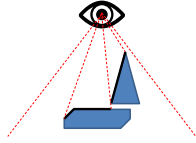


## Aula – Computação Gráfica



### Determinação de Superfície Visível

Slides para uso pessoal e exclusivo durante o período de aula. Distribuição ou qualquer uso fora do escopo da disciplina é expressamente proibido.

1

1

### O que é Determinação de Superfície Visível?

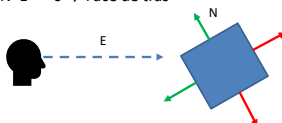
- Dado um conjunto de objetos 3D e uma câmera (ponto de vista)
  - Determinar quais arestas ou superfícies estão visíveis
- Por que um objeto não estaria visível?
  - “Está de costas”
  - Oclusão
- Requer saber a interação entre os objetos da cena

2

2

### Superfícies Contrárias (Back Face Culling)

- Pode-se eliminar as superfícies contrárias à câmera
- Como determinar se a face está vista de trás?
  - Assumindo  $E$  como o vetor da câmera para a face do polígono
  - $N \cdot E < 0 \rightarrow$  Face da frente
  - $N \cdot E \geq 0 \rightarrow$  Face de trás



3

3

## Algoritmo do Pintor (Painters Algorithm)

- É um algoritmo back-to-front (de trás para frente)
- Cria uma ordem de desenho dos polígonos
  - Cada polígono sobrescreve os anteriores se necessário
  - Garante a visibilidade correta em qualquer resolução de pixel
- Ache um forma de ordenar os polígonos por profundidade (z)
- Desenhe os polígonos nessa ordem
  - Os mais longe da câmera são desenhados primeiro
  - Cada polígono deve ser rasterizado na sequência
  - Os mais próximos da câmera serão os últimos
- Problema
  - Quando dois polígonos parcialmente escondem um ao outro



4

4

## Algoritmo Z-Buffer

- Z-Buffer é usado para armazenar a profundidade
  - Armazena profundidade por pixel
- Algoritmo
  - Inicializa o buffer com uma distância infinita
  - Quando o objeto é renderizado
    - A profundidade de cada porção (pixel) é comparada com o valor armazenado no "pixel" correspondente do z-buffer
      - Se o novo valor for menor (está mais próximo da câmera)
        - » Sua profundidade é colocada no z-buffer e sua cor no frame buffer
      - Caso contrário, não atualiza os buffers

5

5

## Algoritmo Z-Buffer

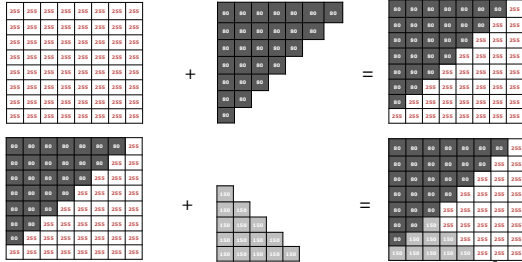
```
void z-Buffer() {
    int x, y;
    //Inicializa com valor de fundo e distancia maxima
    for (y = 0; y < YMAX; y++)
        for (x = 0; x < XMAX; x++) {
            WritePixelColor (x, y, BACKGROUND_VALUE);
            WriteZInBuffer (x, y, 255);
        }
    for cada polígono {
        for cada pixel do polígono projetado {
            //calcule profundidade-z para o pixel
            double pz = CalculaZValuePixel (x, y);
            if (pz < ReadZFromBuffer (x, y)) {
                // Novo ponto está mais perto do que o armazenado
                WritePixelColor (x, y, CalculaCorPixel (x, y))
                WriteZInBuffer (x, y, pz);
            }
        }
    }
}
```

6

6

## Algoritmo Z-Buffer

- Exemplo
  - Z-buffer inteiro com near = 0, far = 255



7

## Algoritmo Z-Buffer



A simple three dimensional scene



Z-buffer representation

T-bus  
<https://commons.wikimedia.org/wiki/File:s-Z-buffer.jpg>  
 CC BY 2.0

8

8

## Algoritmo Z-Buffer

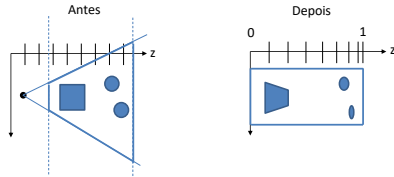
- Vantagens
  - Barato e fácil de implementar em hardware
  - Não requer pré-processamento (e.g. ordenação)
  - Pode usar a técnica de rasterização incremental para calcular o z

9

9

## Algoritmo Z-Buffer

- Desvantagens
  - Compressão em z pode causar problemas
    - Z-fighting
  - A informação de profundidade
    - Perde precisão rapidamente

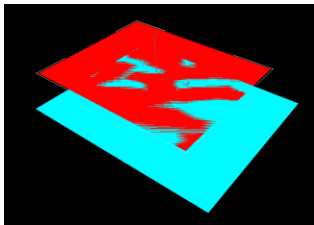


10

10

## Algoritmo Z-Buffer

- Z-fighting
  - Ocorre quando dois objetos estão com valores similares em z



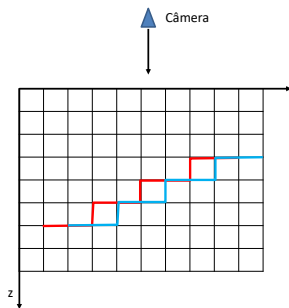
Mnoskins  
<https://commons.wikimedia.org/wiki/File:Z-fighting.png>  
 Public Domain

11

11

## Algoritmo Z-Buffer

- Z-fighting



12

12

## Algoritmo Z-Buffer

- Z-fighting
  - <http://www.youtube.com/watch?v=UVGdOFVbvBo/course/cs123/data/scenes/contest/ntom>:
  - [https://www.youtube.com/watch?v=9AcCrF\\_nX-I](https://www.youtube.com/watch?v=9AcCrF_nX-I)

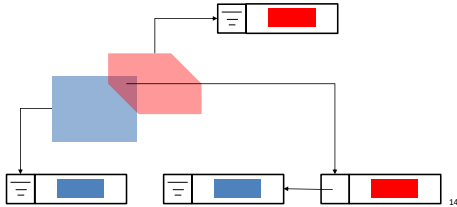


13

13

## Transparência

- Algoritmo A-Buffer
  - Permite considerar objetos transparentes
  - Armazena uma lista dos objetos de um pixel
  - Informações são combinadas para gerar a contribuição do pixel



14

14

## Perguntas ?????

15

15