

Ficha 5 - Culling

①

Os 3 tipos de culling são:

- Backface culling
- View Frustum culling
- Occlusion culling

② ~~Os~~ Este

O BFC elimina os ~~tri~~ triângulos que estão "de costas" para a camera.

Este tipo de culling é realizado em hardware polígono, o que implica que os polígonos sejam enviados para a placa gráfica.

O ideal seria evitar esta comunicação

View Frustum culling identifica o que está dentro do frustum (parcialmente) e faz culling de todo o que não está dentro. Apenas o que está dentro do frustum ~~é enviado~~ (ou seja que parcialmente) ~~para todos os componentes~~ é enviado para todos os componentes gráficos de hardware. Assim o hardware só precisa de fazer render do que pertence ao visível.

②

Equação normalizada: $Ax + By + Cz + D = 0$

Dados: p_1, p_2, p_3

1º $\vec{u} = \vec{p_2} - \vec{p_1}$

2º $\vec{v} = \vec{p_3} - \vec{p_1}$

3º $\vec{n} = (\vec{u} \times \vec{v}) = \vec{v} \times \vec{u}$, onde $A = n_x, B = n_y, C = n_z$

4º $\vec{n} = \frac{\vec{n}}{|\vec{n}|}$

5º Como p_1 pertence ao plano, temos:

$$A p_{1x} + B p_{1y} + C p_{1z} + D = 0 \Rightarrow$$

$$\Rightarrow D = -A p_{1x} - B p_{1y} - C p_{1z} = -\vec{n} \cdot \vec{p_1}$$

③

Tem 2 fases:

- Fase de pré-processamento
- Fase de teste

Na fase de pré-processamento ^{são} ~~se~~ ~~é~~ obtidas todas as equações (uma por cada face).

Na fase de teste, para cada vértice/triângulo objeto tenta-se se está dentro com o frustum.

④

$$\vec{f_c} = \vec{p} + d \times \vec{farList}$$

$$\vec{f_{2D}} = \vec{f_c} - \left(\frac{(\vec{u_p} \times \vec{f_c}) \cdot \vec{f_c}}{2} \right) + \left(\frac{\vec{n} \times \vec{f_c}}{2} \right)$$

$$\vec{n} = \frac{\vec{d} \times \vec{u_p}}{|\vec{d}| \times |\vec{u_p}|} = \vec{n}$$

5

6

A aproximação geométrica realiza-se no espaço global e usa a rotação sobre a forma de view frustum para extrair os 6 que definem os limites do frustum

- near
- far
- top
- bottom
- left
- right

Os planos são definidos de forma que o vetor da normal fique virado para dentro do view frustum. Para ver se um objeto está dentro do view frustum fazem-se testes para ver de que lado do plano é que o objeto está.

7

Consideramos um ponto $p = (x, y, z, w)$ de
em 3D. Tendo uma matriz $modelView$ M e
uma matriz projetada no ponto P é transformada
pelas matrizes M e P , no ponto $p_c = (x_c, y_c, z_c, w_c)$
no clip space, usando $p_c = pMD$.

8

Escolhe-se um eixo, um plano perpendicular a
divide-se o "mundo" em 2 regiões
Seleciona-se um eixo diferente

9



10

possíveis situações:

- Incluir o filho na célula do pai ①
- Incluir filho em ambas as células ②
- Dividir o filho de forma a que cada parte ③
ocupe uma célula.

(1)

Vantagens: Não há polígonos com mais do que 1 célula.
Desvantagens: Célula pode ficar com mais polígonos.

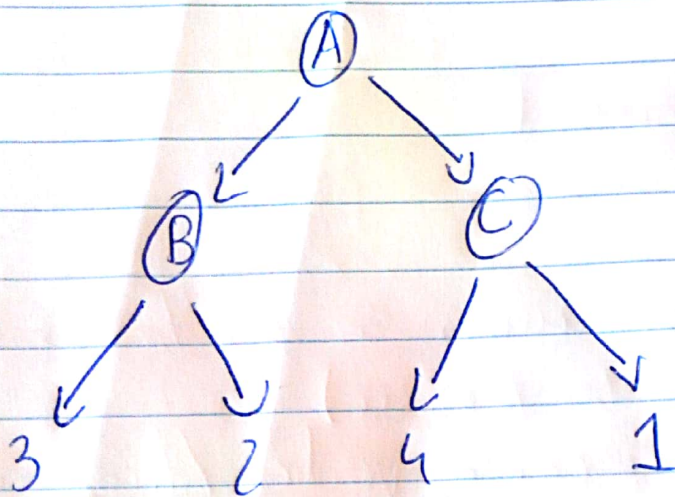
(2)

Vantagens: A célula pode não ficar com tanto polígono.
Desvantagens: Vai existir ambiguidade ao perceber
se um polígono existe em várias células.

(3)

Vantagens: Vai ser mais ~~pre~~ claro em termos de
localização do polígono se a distribuição for
mais precisa.
Desvantagens: Ter de dividir o polígono.

(11)



Ordem de desenho: 2, 3, 4, 1