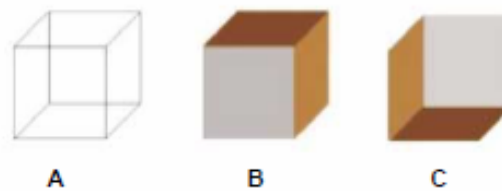


aula 9  
Fases do Realismo Visual  
IC/UFF - 2017



*Representação aramada em A e suas possíveis interpretações em B e C.*

# Fases do realismo

Geometria dos objetos da cena

Representação 3D (wire frame)

Eliminação de partes não visíveis

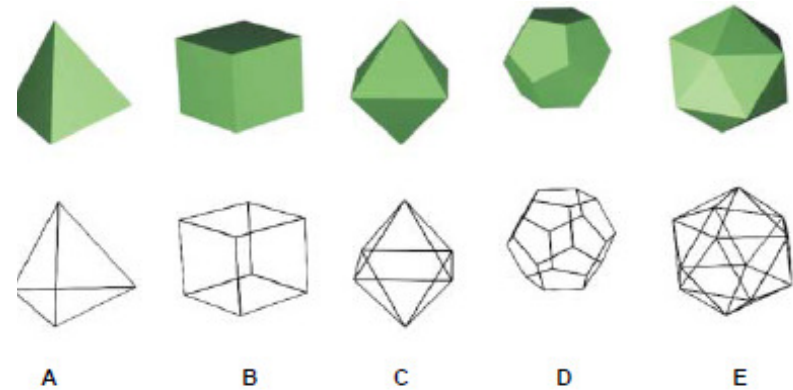
Shading (reflexão difusa)

Iluminação (reflexão especular)

Sombras (shadow)

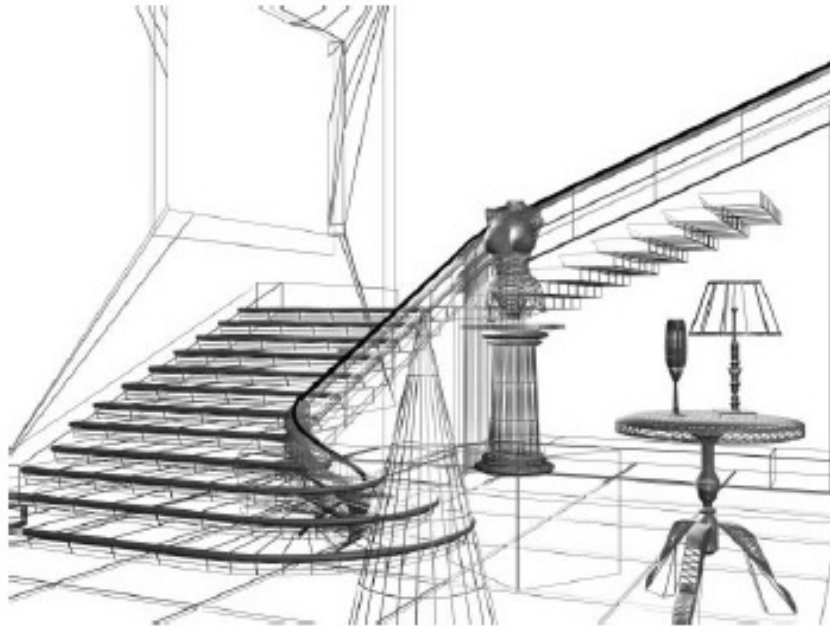
Reflexão, transparências, refração

Texturas



*Os 5 tipos possíveis de poliedros regulares.*

# Wire frame (modelagem, projeção, volume de visão)



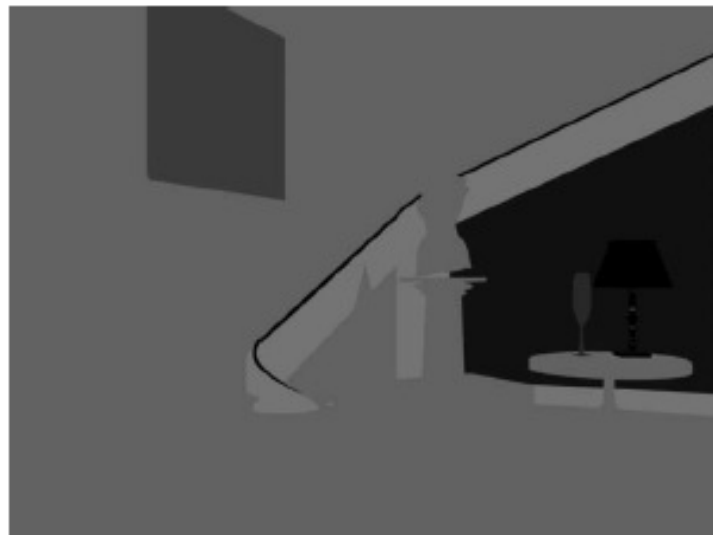
*Representação dos objetos em wire-frame.*

# Tratamento de hidden



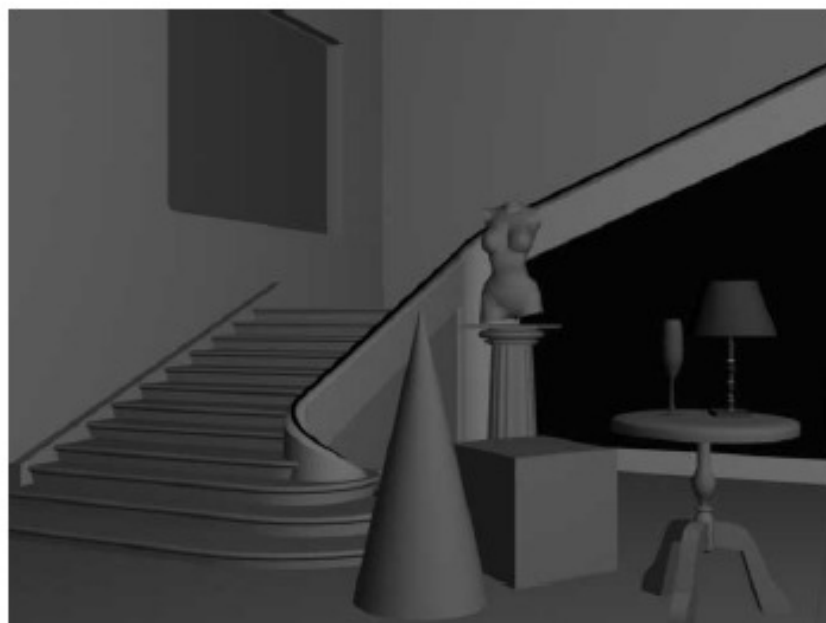
*Remoção das linhas não-visíveis.*

# Shading com luz ambiente



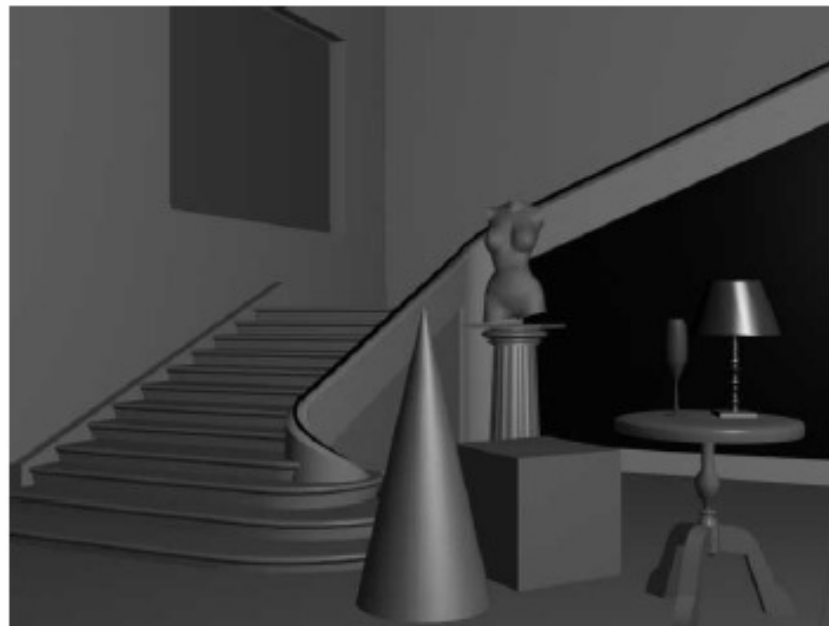
*Reflexão Ambiente aplicada em uma cena.*

# Luz direcional



*Reflexão Difusa aplicada em uma cena.*

# Tratamento de Iluminação especular



*Reflexão Especular aplicada em uma cena.*

# Cores e texturas





# Sombras e reflexão



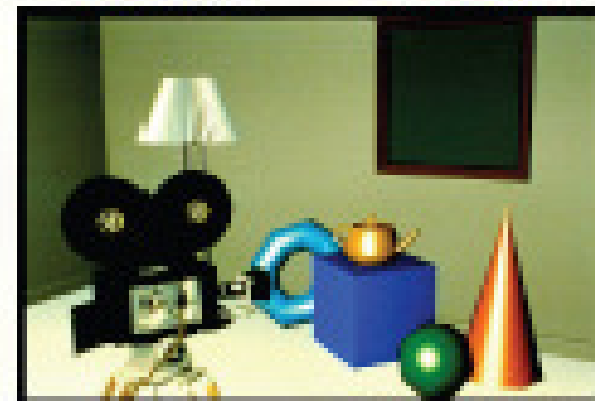
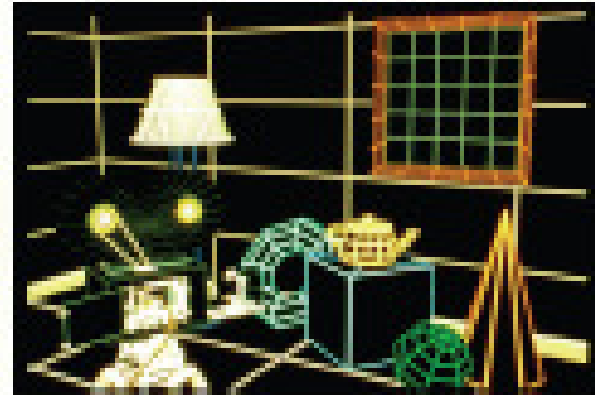
# Realismo em passos

## Objetivos

Melhorar o entendimento das cenas e objetos criados

Possibilidade de representação de dados, objetos e cenas complexas

Realismo até o nível desejado da forma adequada para a aplicação:  
real time x perfeição física da cena

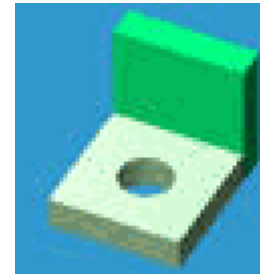


# Nível adequado do realismo

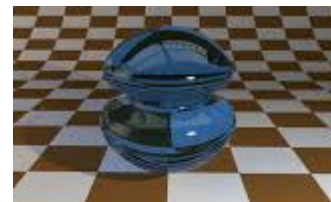
1- Remoção de partes invisíveis do objeto (linhas, superfícies e oclusões por outros objetos)



2 - Sombreamento das diversas superfícies ou *Shading* :  
reflexão difusa,  
reflexão especular



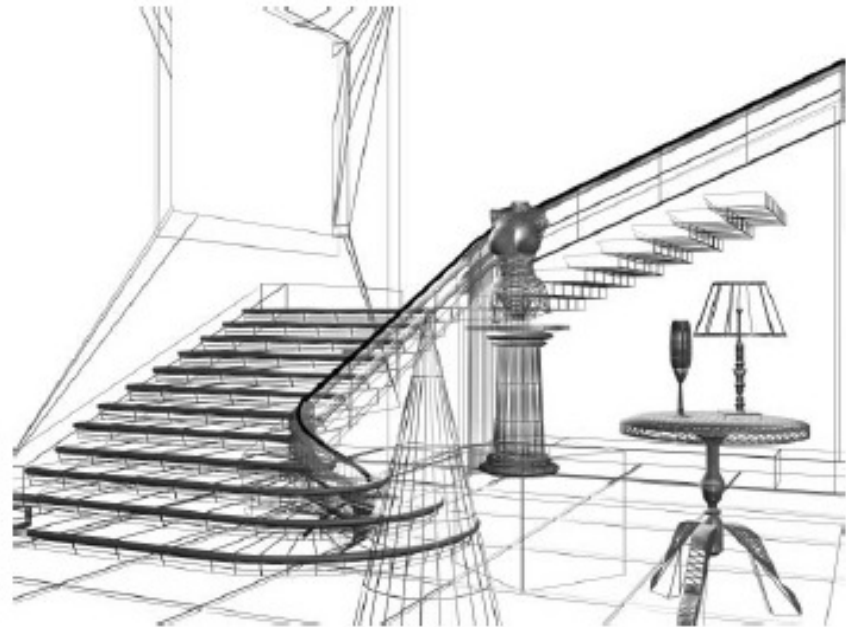
3 - Detalhes:  
Sombras (*shadows*)  
Reflexão,  
transparências,  
refração,  
Texturas



*Wire frame* : adequado para posicionamentos e desenho, mas não realístico

Todas as linhas são mostradas.

Passo seguinte do realismo eliminar **partes da cena que não são vistas quando objetos opacos são vistos de determinada direção.**



# Tratamento de *hiddens* ou *Hidden Line/surface problem*

Eliminação de linhas:  
caso particular da  
definição de que faces  
ou superfícies são  
ocultas por outras do  
objeto ou cena.



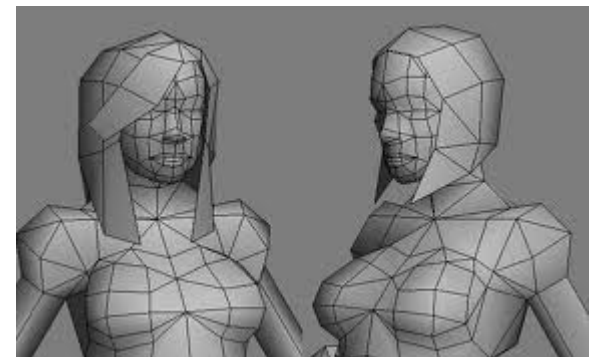
# Técnicas de visibilidade

*Back face culling*

*Priority fill ou painter's algorithm*

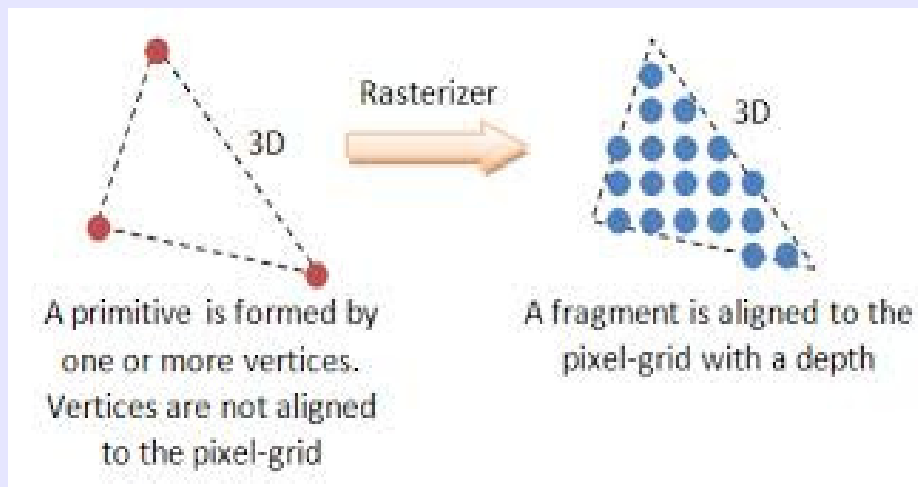
*Z- buffer*  
*(min Max)*

*Ray casting*  
*(Ray tracing simplificado*  
*ou aproximado)*



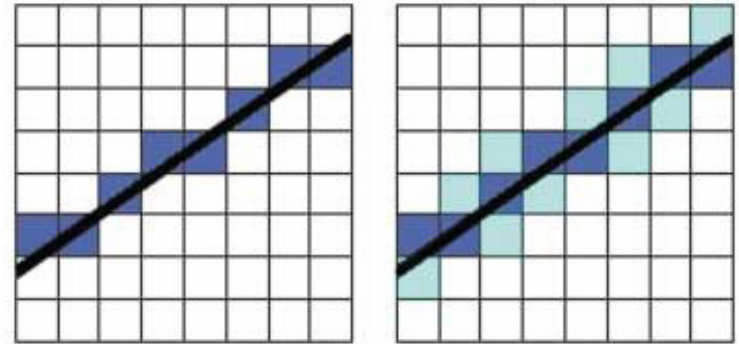
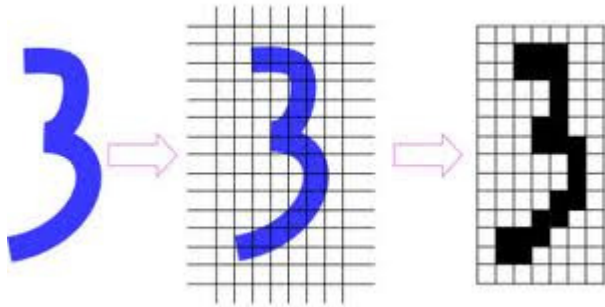
## HÁ ALGORITMOS NA FORMA **VETORIAL** E **RASTER**

**RASTER:** o objeto em 3D é tratado na forma final quando já “*discretizado*” em pixels.

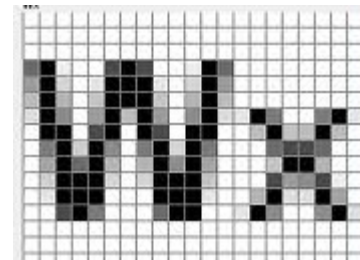
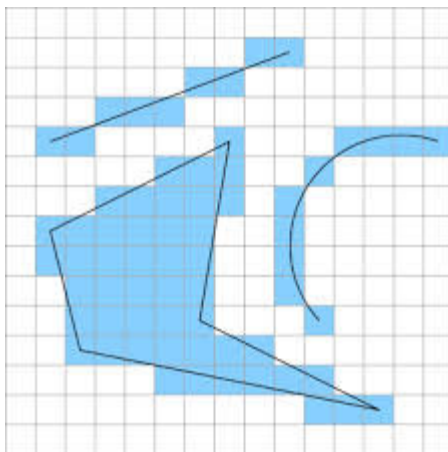


**Rasterisation**  
(ou **rasterization**)  
converte uma imagem descrita como **vector format** para a forma de pixels ( dots ) para representação em video, **printer** ou **storage in a bitmap file format**.

*Aliasing* → *antialiasing*



*Rasterizar* = Usar a malha de pixels para descrever os objetos!





# *Back face culling*

Demo: em javascript:

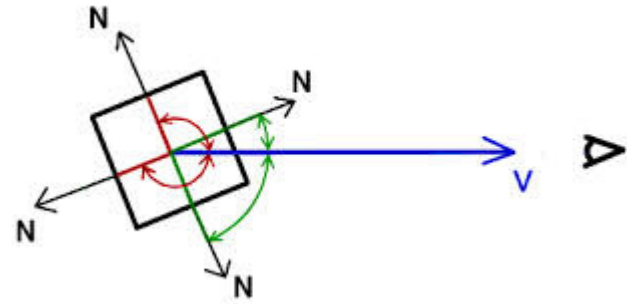
<http://echolot-1.github.io/back-face-culling-demo/>  
[echolot-1/back-face-culling-demo](http://echolot-1/back-face-culling-demo)



*Em CG back-face culling determina quando a face de um objeto será visível.*

*Esse processo torna o rendering mais eficiente pois reduz o número de polígonos a ser desenhado.*

## *Back face culling*



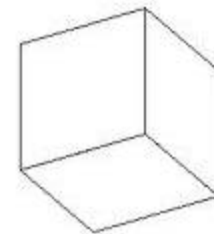
Idéia básica:

**Remover faces traseiras dos objetos em relação ao observador**

Adequadas para objetos convexos.

OBS :

Ser **não convexo**  $\neq$  ser **côncavo**

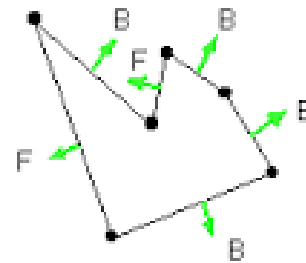
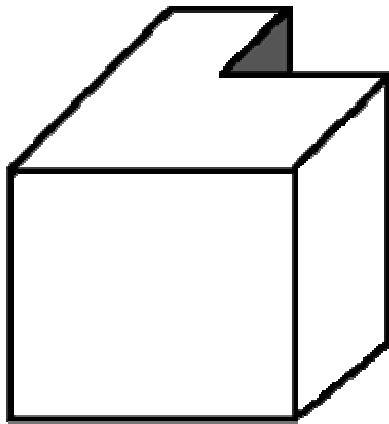


# Objetos convexos

## Definição:

Formado por faces convexas.

*i.e.* Formado por polígonos convexas: nos quais a **ligação entre quais quer 2 pontos** internos nunca passa por uma parte externo a face:

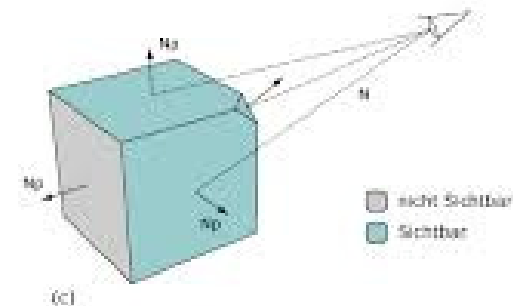
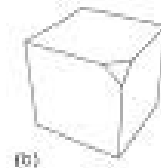
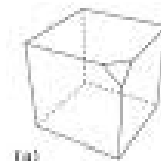
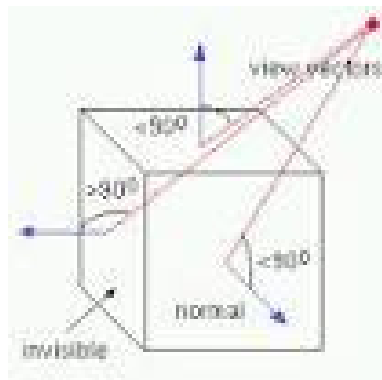


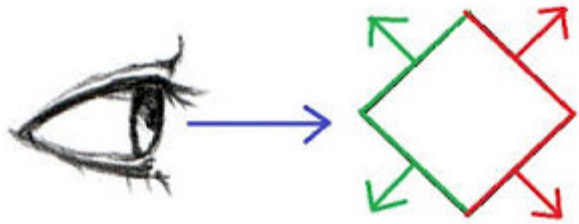
# Algoritmo no espaço do objeto

Usa-se a **direção que as normais** às faces fazem com a direção de visualização.

Entre **-90** graus e **90** graus a **face é visível** pelo observador (ou a face é de frente) .

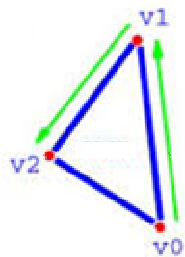
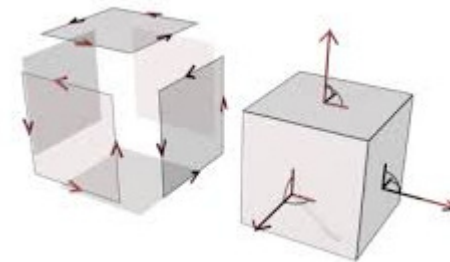
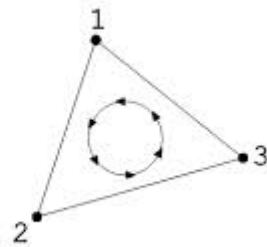
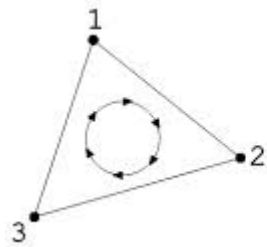
*(Back face culling, método de Roberts ou teste da normal)*





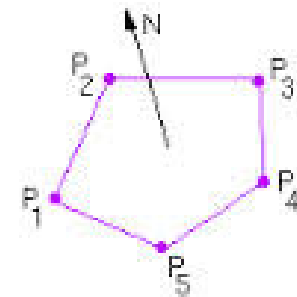
# 1-Obtêm a normal às faces

Através do cálculo do **produto vetorial** de dois vetores da face: a ordem dos vértices é importante!



$$N = (V_1 - V_0) \times (V_2 - V_0)$$

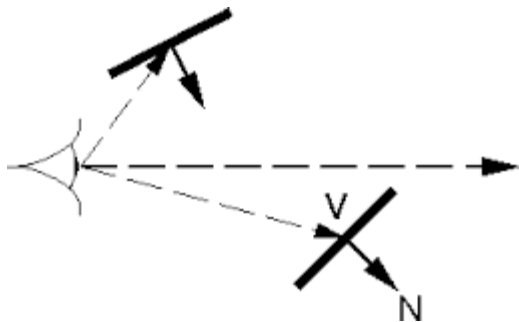
$$(V_1 - V_0) \times (V_2 - V_0) = -(V_2 - V_0) \times (V_1 - V_0)$$



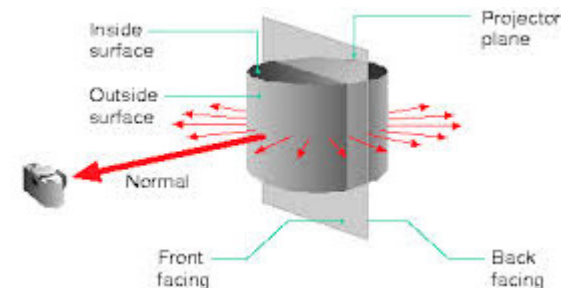
2 - Define-se o vetor da direção de visão

### 3- Verifica-se o ângulo!

Através do **produto interno** entre as normais e a direção de visão, (não é preciso calcular o ângulo) apenas ver se o resultado **é maior que zero** → ângulo entre  $-90^\circ$  e  $90^\circ$  !



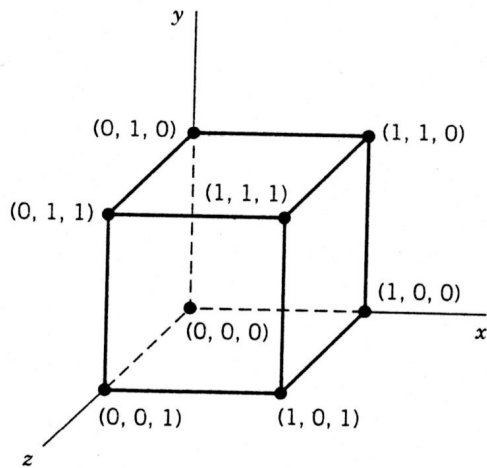
$$(V_0 - P) \cdot N \geq 0$$



# Dúvida em como calcular as

Diversas etapas do método de Robert?

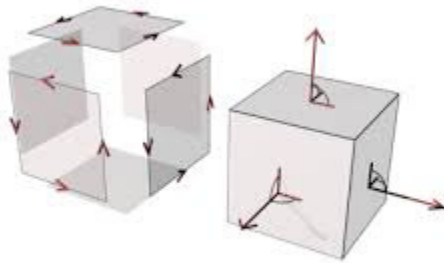
Peque uma folha de papel e vamos fazer isso para esse objeto:



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

1- Calcule a normal a cada uma das 6 faces

2- inclua essa direção como mais um elemento da sua lista de faces:



(Os vértices limites das faces devem ser descritos **sempre no mesmo sentido horário** (ou anti-horário) do exterior do objeto, **para todas as faces.**)



# Estrutura de dados baseada em Vértice

*vertex*      *coordinates*

$v_1$        $x_1 \ y_1 \ z_1$

$v_2$        $x_2 \ y_2 \ z_2$

$v_3$        $x_3 \ y_3 \ z_3$

$v_4$        $x_4 \ y_4 \ z_4$

$v_5$        $x_5 \ y_5 \ z_5$

$v_6$        $x_6 \ y_6 \ z_6$

$v_7$        $x_7 \ y_7 \ z_7$

$v_8$        $x_8 \ y_8 \ z_8$

*face*

*vertices*

$f_1$        $v_1 \ v_2 \ v_3 \ v_4$

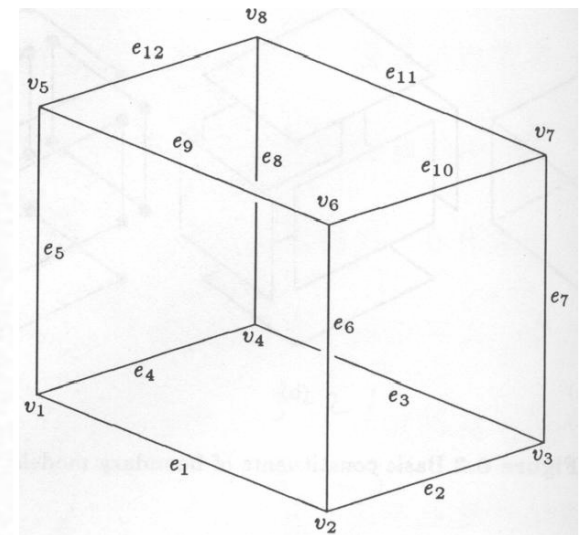
$f_2$        $v_6 \ v_2 \ v_1 \ v_5$

$f_3$        $v_7 \ v_3 \ v_2 \ v_6$

$f_4$        $v_8 \ v_4 \ v_3 \ v_7$

$f_5$        $v_5 \ v_1 \ v_4 \ v_8$

$f_6$        $v_8 \ v_7 \ v_6 \ v_5$



(como voce pode incluir essa normal na lista?  
as coordenados do vetor noral? Ou ... )

1- Calcule a normal a cada uma das 6 faces;

2- inclua essa direção como mais elementos da sua lista de faces;

3- Define-se o vetor da direção de visão;

4 - **Verifique se a fase será visível desta direção de visão:**

Através do sinal do **produto interno** entre as normais e a direção de visão, se o resultado **é maior que zero** → ângulo entre  $-90^\circ$  e  $90^\circ$  → Visível.

5- Reponda fazendo uma lista booleada se para esse direção uma dada fase é visível.

6- No seu programa vice vai só pintar as fases visíveis!

# Entregue as respostas destes itens

Em sua folha de papel até o fim da aula para o

Professora:

Esse é o nosso texto 4!!

## Bibliografia:

D. F. Rogers, J. A. Adams. Mathematical Elements for Computer Graphics, 2dn Ed. , Mc Graw Hill, 1990

E. Azevedo, A. Conci, [Computação Gráfica: teoria e prática](#), [Campus](#) ; - Rio de Janeiro, 2003

J.D.Foley,A.van Dam,S.K.Feiner,J.F.Hughes. Computer Graphics- Principles and Practice, Addison-Wesley, Reading, 1990.

Y. Gardan. Numerical Methods for CAD , MIT press, Cambridge, 1985.

A. H. Watt, F. Policarpo - [The Computer Image](#) , Addison-Wesley Pub Co (Net); 1998

[https://noppa.oulu.fi/noppa/kurssi/521493s/luennot/521493S\\_3-d\\_graphics\\_vi.pdf](https://noppa.oulu.fi/noppa/kurssi/521493s/luennot/521493S_3-d_graphics_vi.pdf)

<http://graphics.stanford.edu/papers/rad/>