Transformações Geométricas

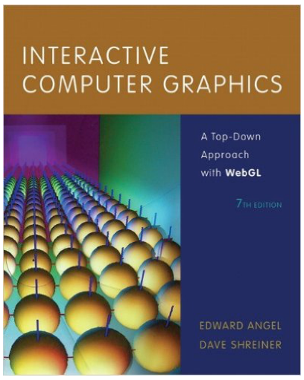
Prof. Antonio L. Apolinário Junior
Estagiária Docente: Rafaela Alcantara
UFBA/IM/DCC/BCC - 2018.1

Roteiro

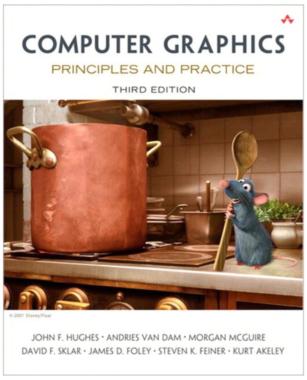
- Transformações Geométricas no contexto do Pipeline Gráfico
- Representação matemática
  - Sistema de Coordenadas Homogêneas
- Transformações
  - 2D
  - 3D
  - Composição de Transformações
- Transformações Geométricas em ***Three.js/WebGL***

Leitura de referencia

- Capitulo 3
  Interactive Computer Graphics -
  A top-down approach with OpenGL
  7th Edition
  Angel, Edward.
  Addison-Wesley. 2014.

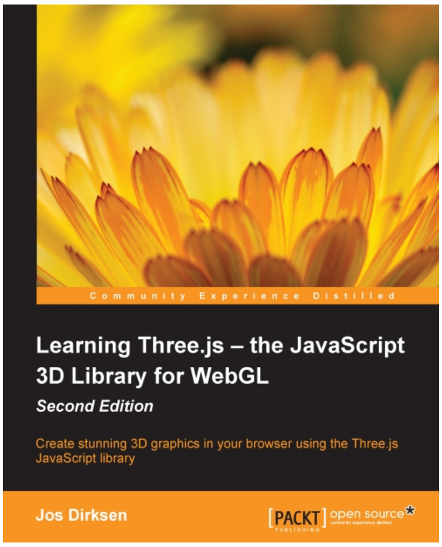


- Capítulos 11, 12 e seção 6.6
  Computer Graphics : Principles and Practice
  Third Edition in C
  John F. Hughes / Andries van Dam
  Morgan McGuire / David F. Sklar
  James D. Foley / Steven K. Feiner
  Addison-Weslley. 2013.



Leitura de referencia

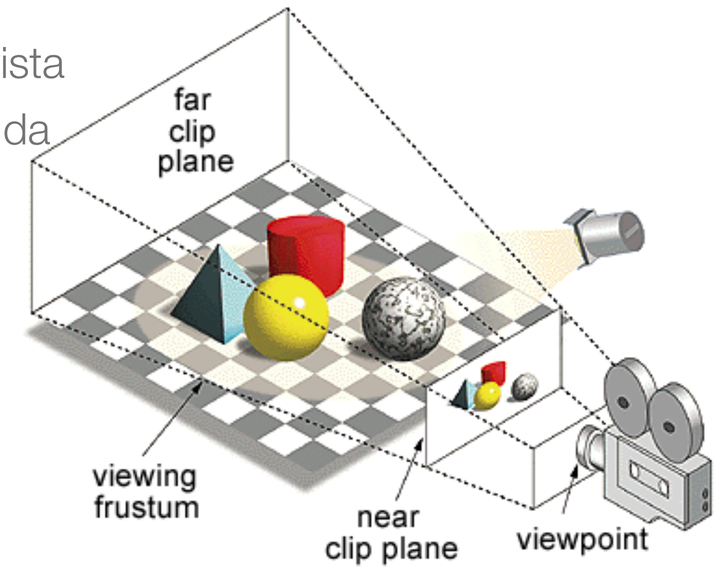
- Capitulo 2 e 8
  Learning Three.js: The JavaScript
  3D Library for WebGL
  Jos Dirksen
  2nd Edition.
  Packt Publishing - 2015.



Transformações Geométricas no contexto do Pipeline Gráfico

Pipeline Gráfico

- Conjunto de estágios em que o processo de formação de imagem em uma aplicação gráfica é dividido
- Envolve:
  - Montagem da cena;
  - definição de ponto de vista
  - Controle da Iluminação da cena
  - Formação da imagem



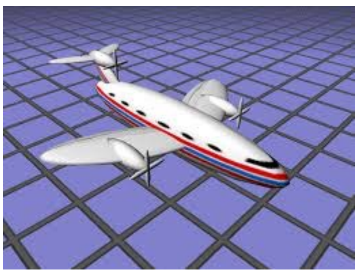
Malha Poligonal

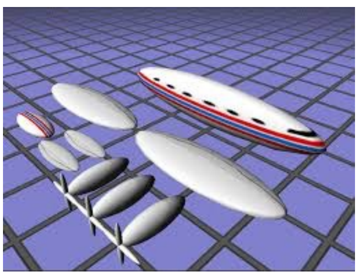
- Descreve a geometria de um objeto a partir de um sistema de referencia próprio

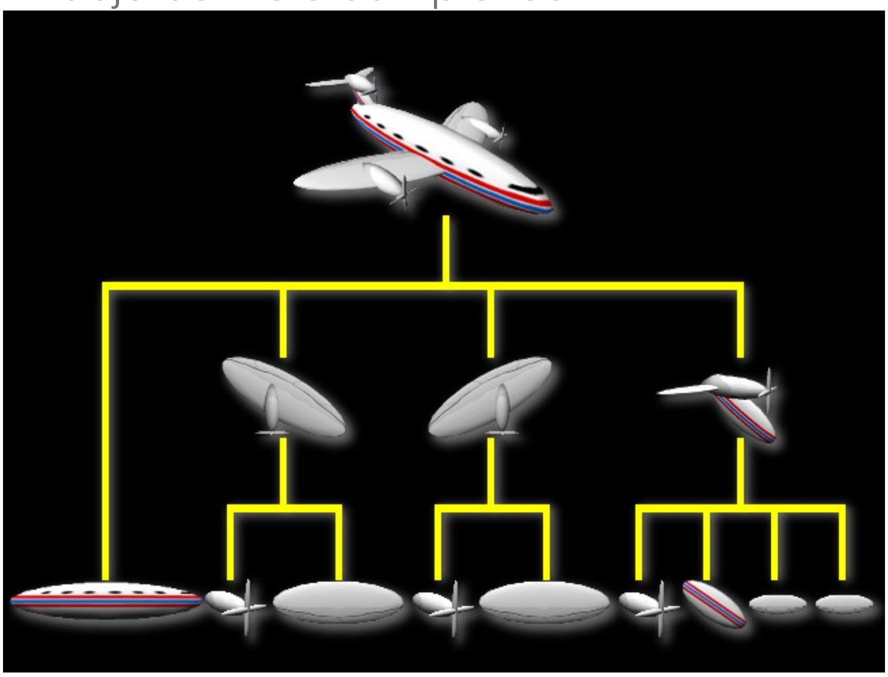
```
0.748952 -0.764952 -0.210132,
0.872246 -0.600882 -0.210132,
1.00016 -0.369696 -0.210132,
1.08939 -0.084084 -0.210132,
1.14496 0.324436 -0.210132,
1.15747 0.681712 -0.210132,
1.08016 0.793529 -0.210132,
0.98164 0.872032 -0.210132,
0.808263 0.929016 -0.210132,
0.442563 0.985585 -0.210132,
0.221794 1.00159 -0.210132,
0 1.0053 -0.210132,
-0.221794 1.00159 -0.210132,
-0.442563 0.985585 -0.210132,
-0.808263 0.929016 -0.210132,
```

Como montar cenas complexas baseadas em malhas poligonais?

- Objetos são descritos em coordenadas absolutas
  - Composição em objetos mais complexos







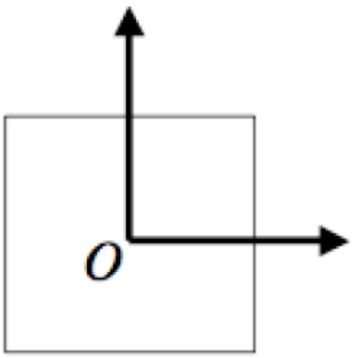
Como montar cenas complexas baseadas em malhas poligonais?

- Objetos são descritos em coordenadas absolutas
  - Posicionam-se relativamente dentro de uma cena

Espaços

- Espaço do Objeto (***Object Space***)

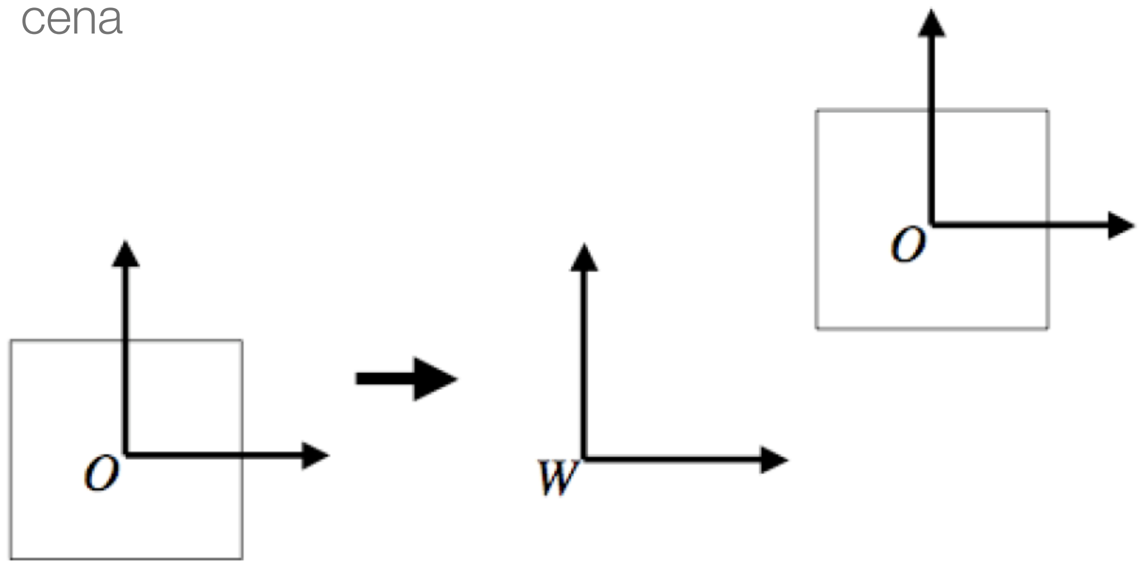
  - Sistema de referencia utilizado na descrição de um determinado objeto da cena



Espaços

Espaço do Mundo (***World Space***)

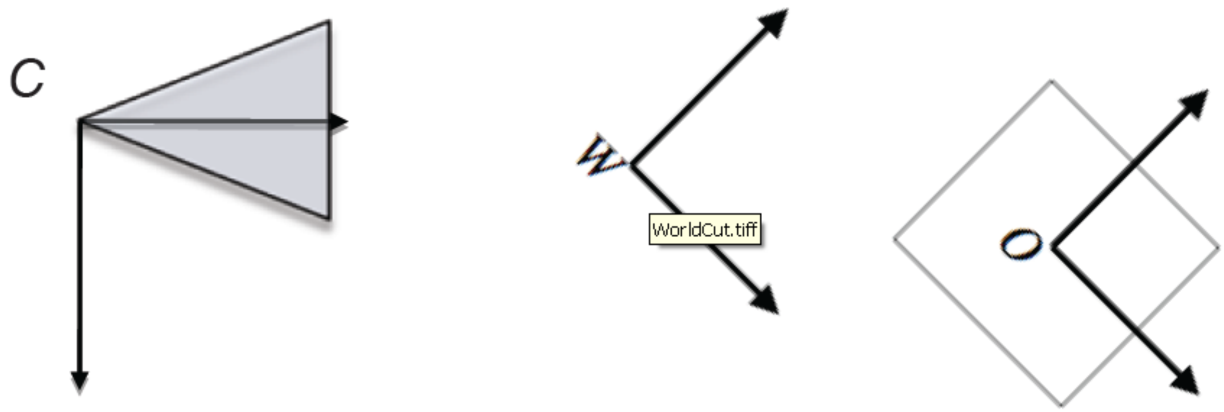
- Sistema de referência global para todos os objetos da cena



Espaços

Espaço da Camera (***Camera Space***)

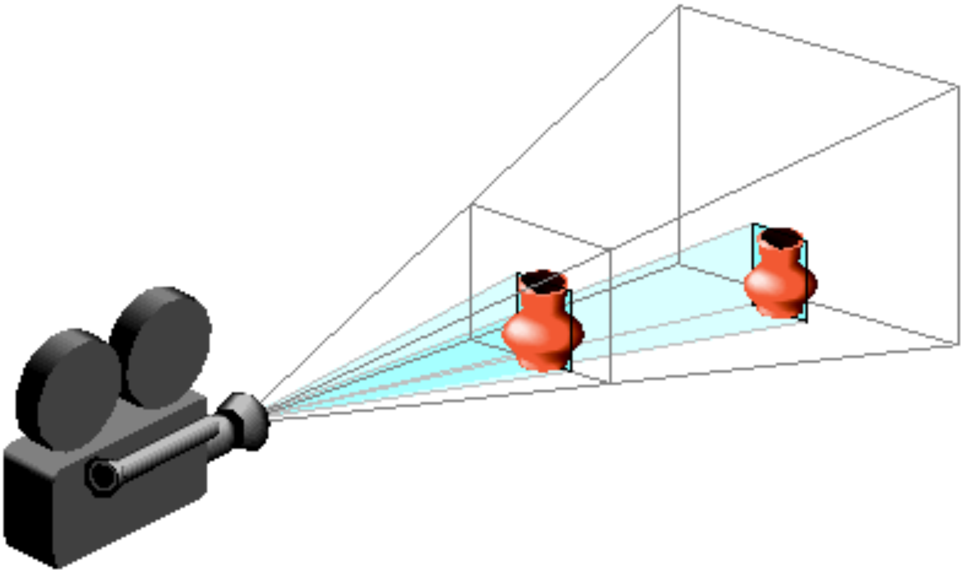
- Espaço referenciado pelo posicionamento da camera virtual
  - Ideal para visualização



Espaços

Espaço da Imagem Projetada (***Projection Image Space***)

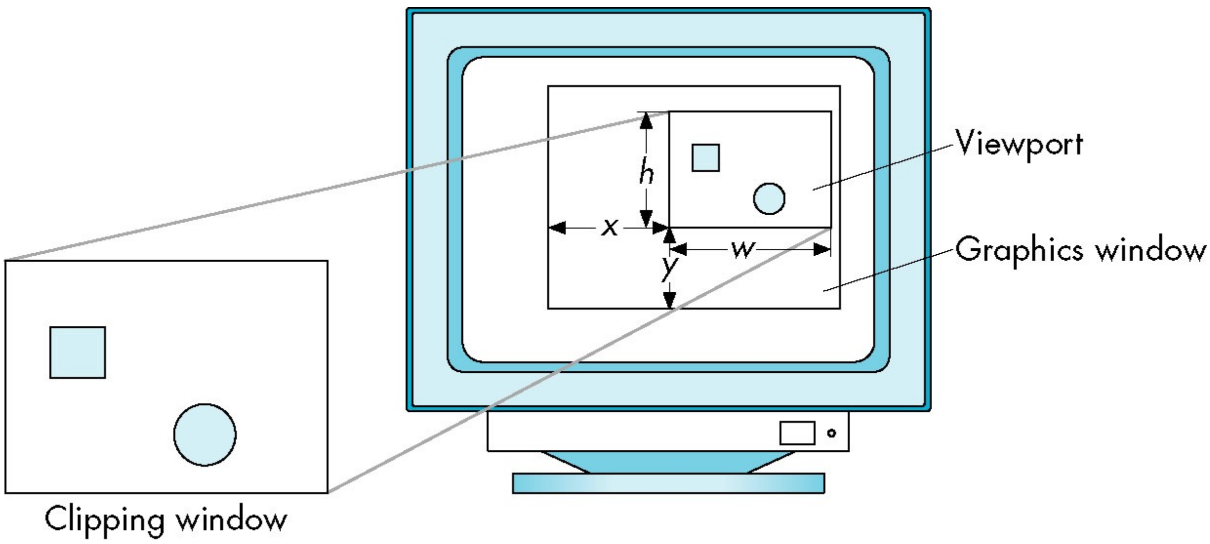
- Espaço bidimensional onde a imagem dos objetos são formadas



Espaços

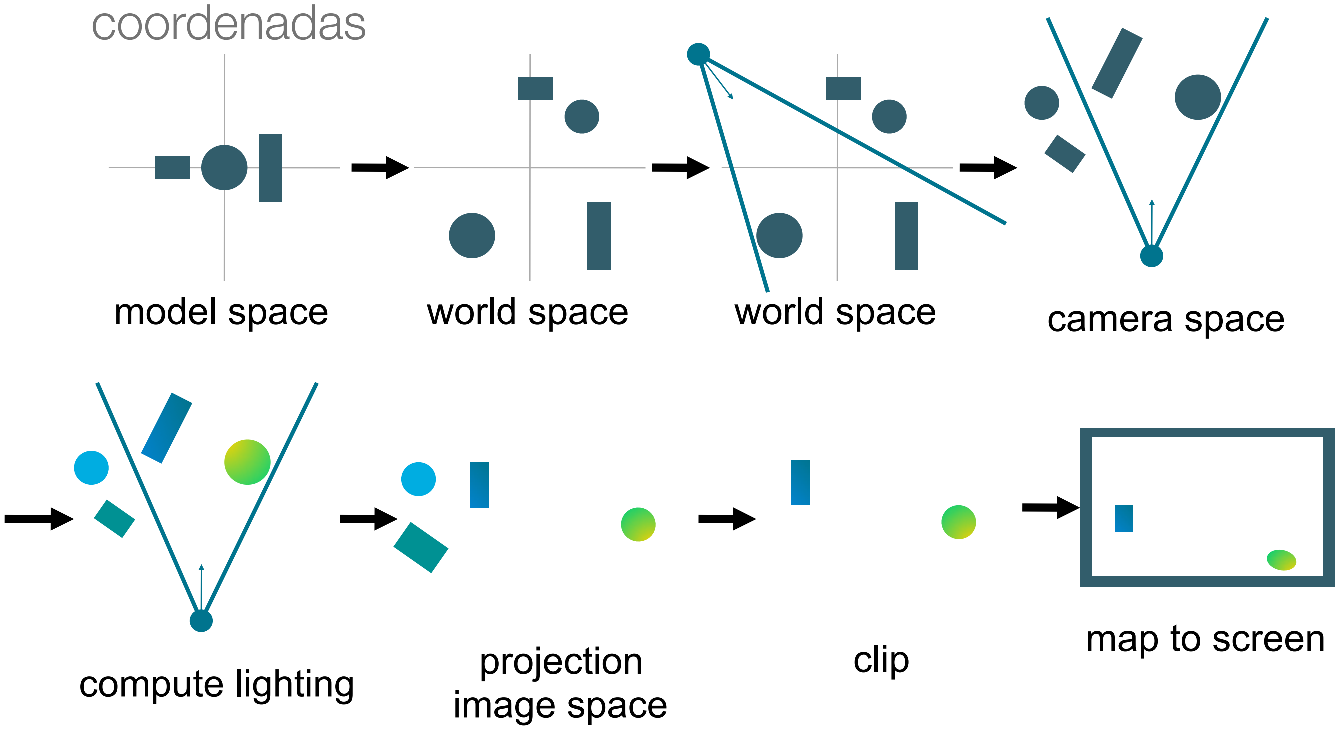
Espaço da Tela (***Window Space***)

Espaço discreto e bidimensional vinculado a imagem armazenada no *framebuffer*



Pipeline Gráfico

- Envolve uma sequencia de mudanças de sistemas de coordenadas



Representação Matemática

Transformações Geométricas

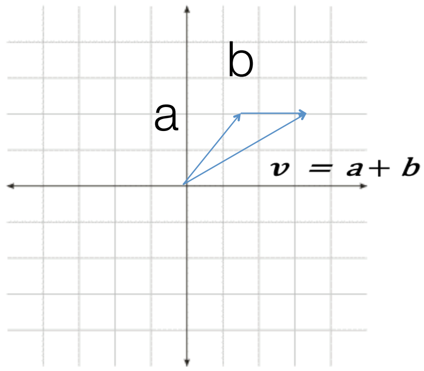
- Objetos com a mesma forma básica podem variar em:
  - Posição
  - Dimensão
  - Orientação

Transformações Geométricas

- Motivação
  - Uma única representação para objetos complexos
    - Visualização em diferentes posições
      - Economia de memória
      - Processamento pode ser transferido para GPU
    - Animação
- Transformações Geométricas são aplicadas em todos os vértices de um objeto

Sistema de Coordenadas

- Qualquer vetor pode ser gerado a partir da combinação linear dos vetores da base
    - Constantes escalares podem ajustar a dimensão do vetor resultante



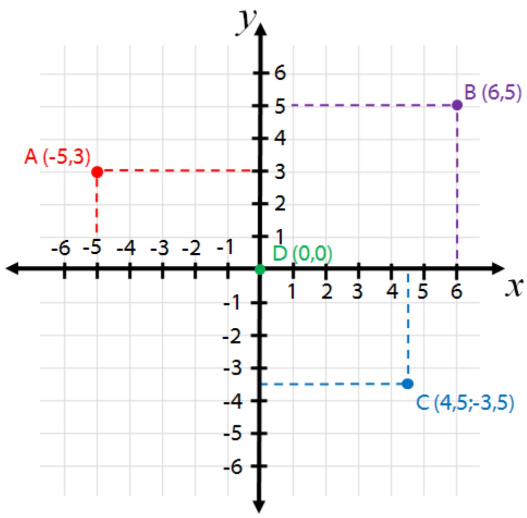
Sistema de Coordenadas

- Pontos podem ser descritos a partir da origem do sistema de coordenadas (O) mais um vetor

$$P = O + \vec{v}$$

$$P = O + x.\vec{i} + y.\vec{j}$$

$$P = \begin{bmatrix} x \\ y \end{bmatrix} \quad \vec{i} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \vec{j} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$



Transformações Lineares

- Na forma matricial:

$$T \times P = P'$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a.x + by \\ c.x + d.y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

Transformações Geométricas

- Transformações mais gerais incluem a soma de um vetor deslocamento (componente afim)
  - Modificam a origem do sistema de coordenadas

$$\vec{v}' = A\vec{v} + \vec{b}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Coordenadas Homogêneas

- Como representar uma transformação geral na forma matricial?
  - Sistema de Coordenadas Homogêneo

$$T.P + d =$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} a.x + b.y + e \\ c.x + d.y + f \end{bmatrix}$$

Coordenadas Homogêneas

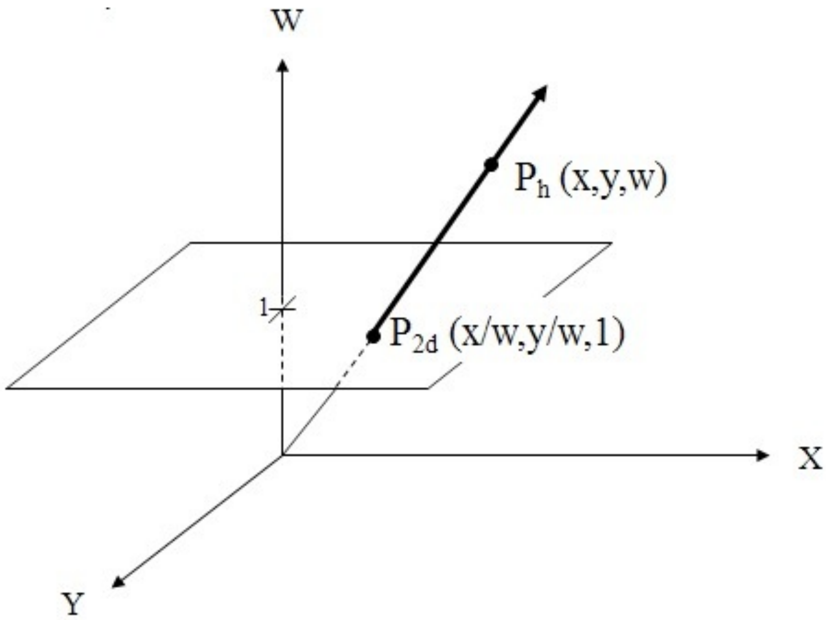
- Em coordenadas homogêneas:

$$T.P = P'$$

$$\begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a.x + b.y + e \\ c.x + d.y + f \\ 1 \end{bmatrix}$$

Coordenadas Homogêneas

- Interpretação Geométrica



Transformações Geométricas 2D

Escala

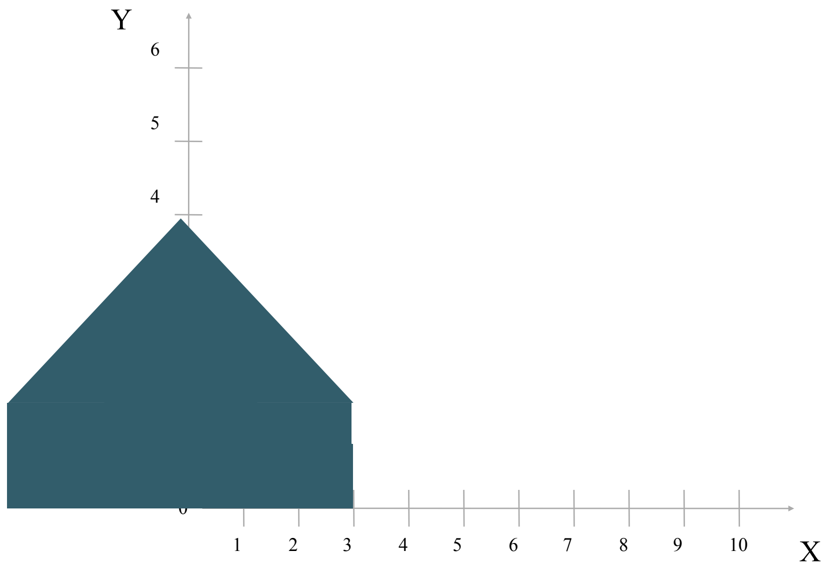
- Permite alterar a dimensão do objeto no espaço
  - Aumento ou diminui as coordenadas x e y
- Definida por uma matriz de transformação
- Multiplica as coordenadas dos pontos

$$\begin{bmatrix} E_x & 0 \\ 0 & E_y \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x.E_x \\ y.E_y \end{bmatrix}$$

Escala

$$E_x = 3$$

$$E_y = 2$$



Escala

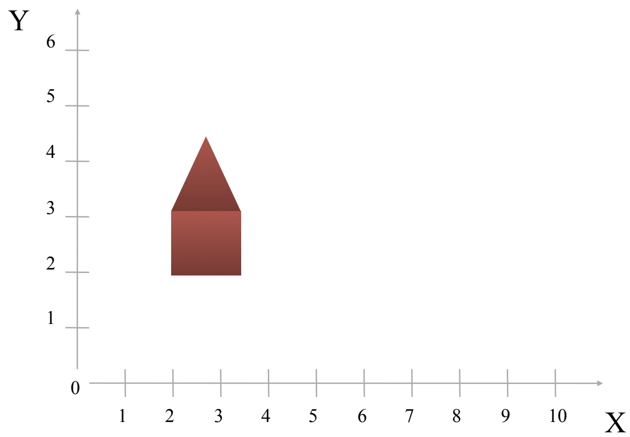
- Fator de escala define o tipo de mudança na dimensão
  - 0<|E| ≤1 ==> Reduz
  - |E| > 1 ==> Aumenta
  - |E| = 1 ==> Mantém
  - E < 0 ==> Reflexão

Escala

- Escala em objetos sem “referencial” na origem
  - Deslocamento em relação a origem como “efeito colateral”

$$E_x = 2$$

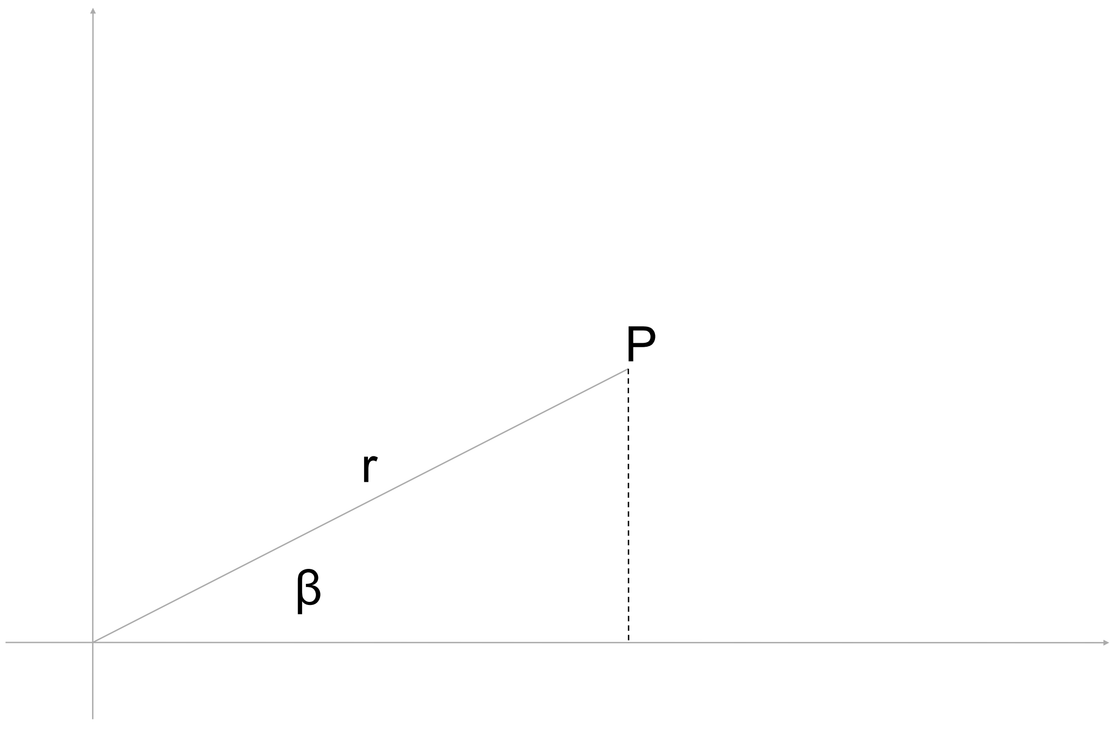
$$E_y = 2$$



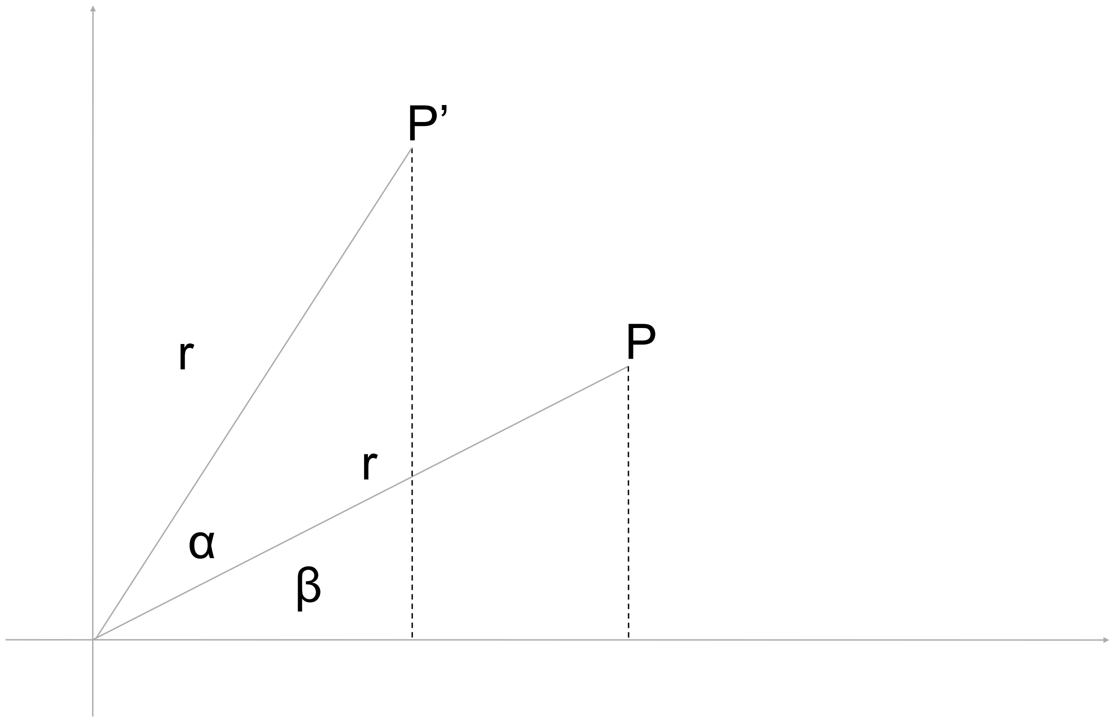
Rotação

- Modifica a orientação do objeto
- Rotação ao redor de um referencial
    - Origem do sistema de coordenadas

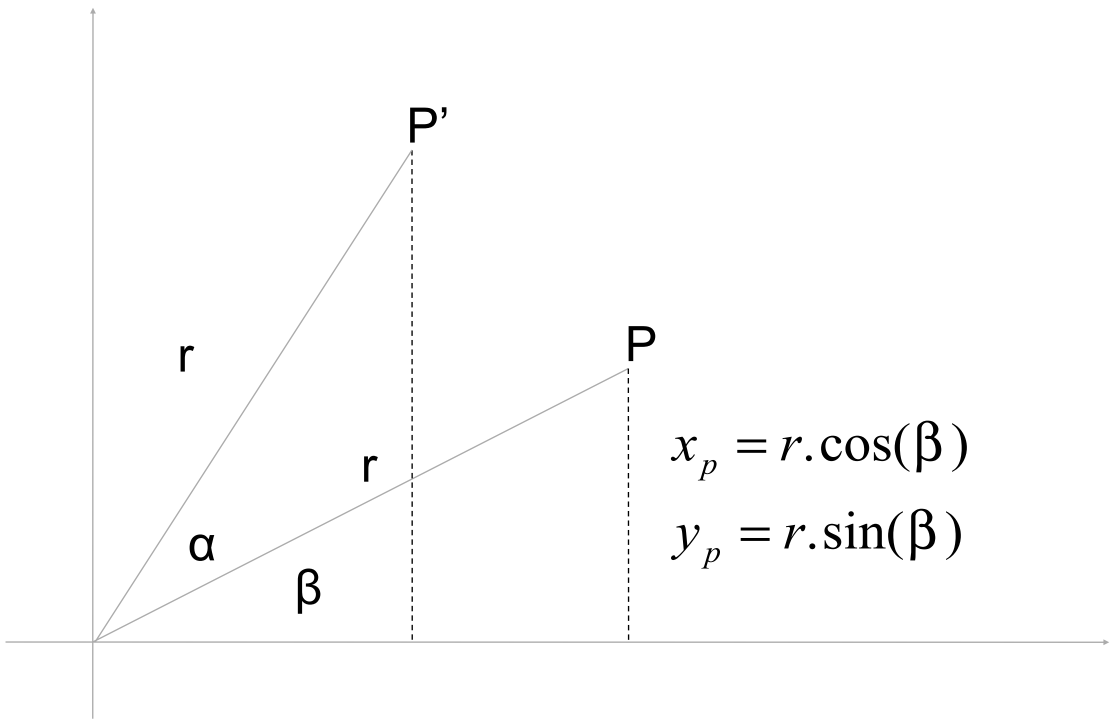
Rotação



Rotação

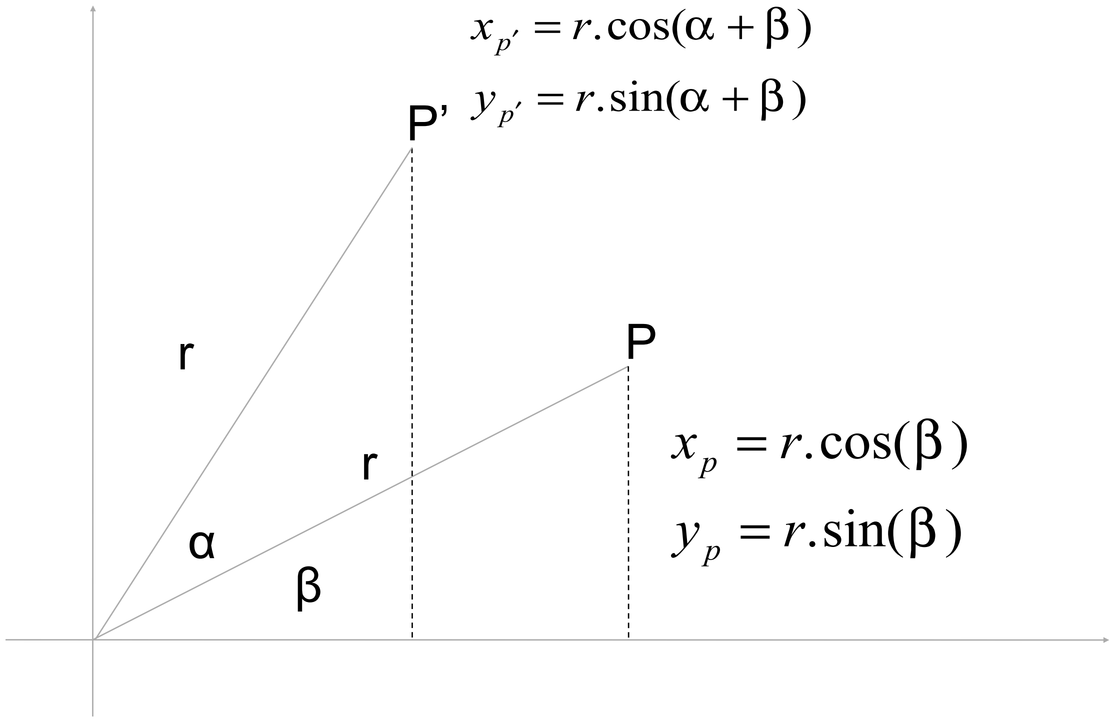


Rotação



$$x_p = r.\cos(\beta)$$
$$y_p = r.\sin(\beta)$$

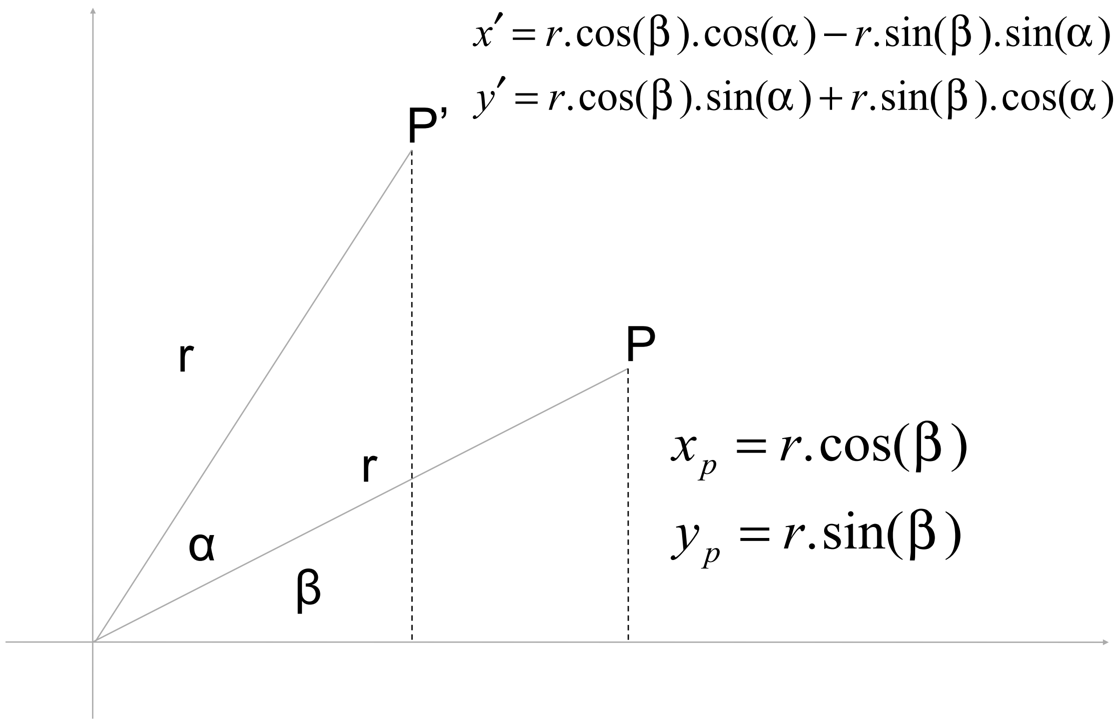
Rotação



$$x_{p'} = r.\cos(\alpha + \beta)$$
$$y_{p'} = r.\sin(\alpha + \beta)$$

$$x_p = r.\cos(\beta)$$
$$y_p = r.\sin(\beta)$$

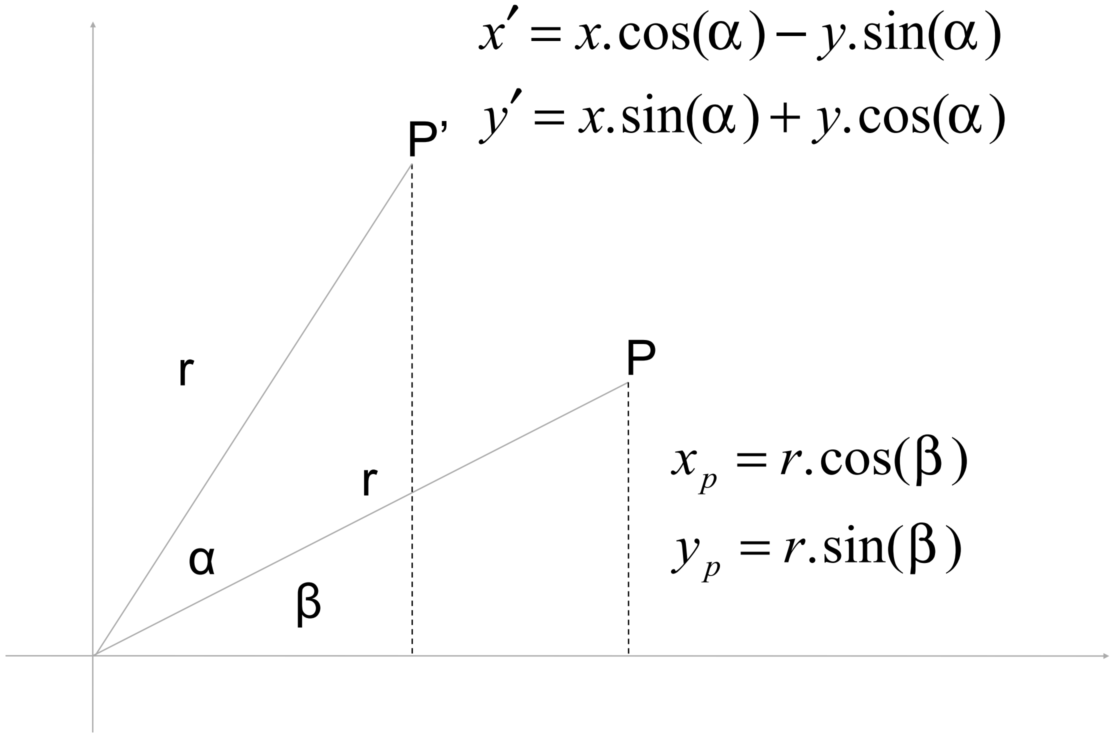
Rotação



$$x' = r.\cos(\beta).\cos(\alpha) - r.\sin(\beta).\sin(\alpha)$$
$$y' = r.\cos(\beta).\sin(\alpha) + r.\sin(\beta).\cos(\alpha)$$

$$x_p = r.\cos(\beta)$$
$$y_p = r.\sin(\beta)$$

Rotação



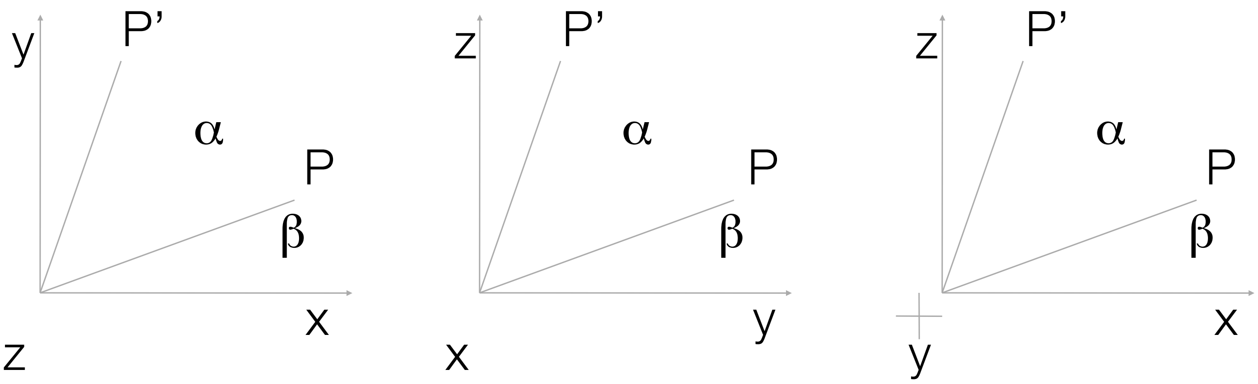
Rotação

- Matriz de Rotação

$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x.\cos(\alpha) - y.\sin(\alpha) \\ x.\sin(\alpha) + y.\cos(\alpha) \end{bmatrix}$$

$$\begin{bmatrix} x & y \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} = \begin{bmatrix} x.\cos(\alpha) - y.\sin(\alpha) & x.\sin(\alpha) + y.\cos(\alpha) \end{bmatrix}$$

Rotação



Rotações em torno dos eixos z, x e y, respectivamente.

Translação

- Permite alterar a posição do objeto no espaço
  - Deslocamentos ao longo das direções x e y
- Definida por um vetor deslocamento
- Adiciona os deslocamentos as coordenadas dos pontos

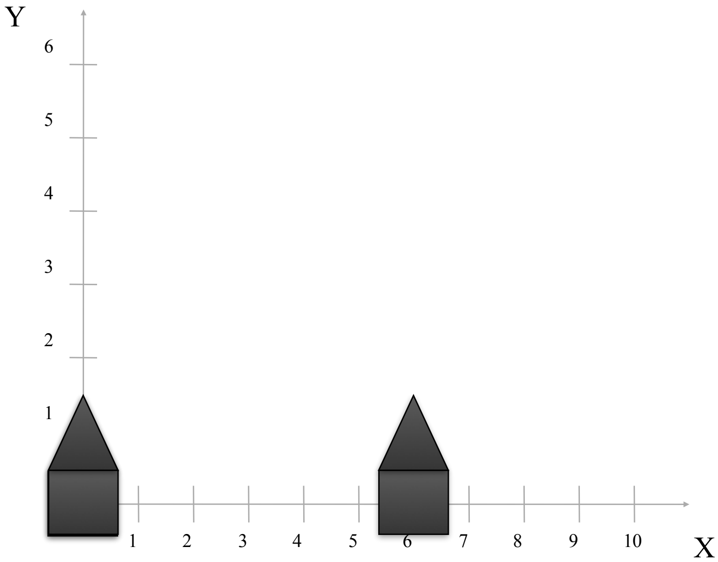
$$\begin{bmatrix} dx \\ dy \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x+dx \\ y+dy \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

Translação

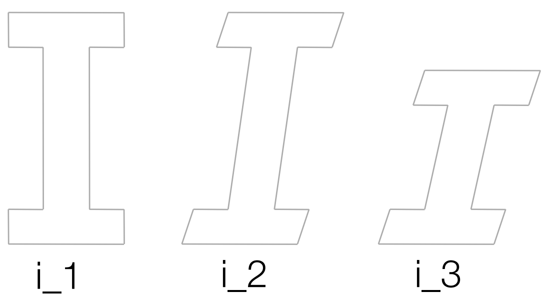
$$d = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$$

$$d = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$$

$$d = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$



Cisalhamento (shear)



$$MTS\_1 = \begin{bmatrix} 1 & sh_x \\ 0 & 1 \end{bmatrix} \qquad MTS\_2 = \begin{bmatrix} 1 & sh_x \\ 0 & e_y \end{bmatrix}$$

Transformações em Coordenadas Homogêneas

- Transformações:
    - Translação
$$\begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix}$$
    - Rotação
$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
    - Escala
$$\begin{bmatrix} E_x & 0 & 0 \\ 0 & E_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Transformações Geométricas 3D

Transformações Geométricas em 3D

- Transformações:
  - Translação

$$\begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

  - Escala

$$\begin{bmatrix} E_x & 0 & 0 & 0 \\ 0 & E_y & 0 & 0 \\ 0 & 0 & E_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

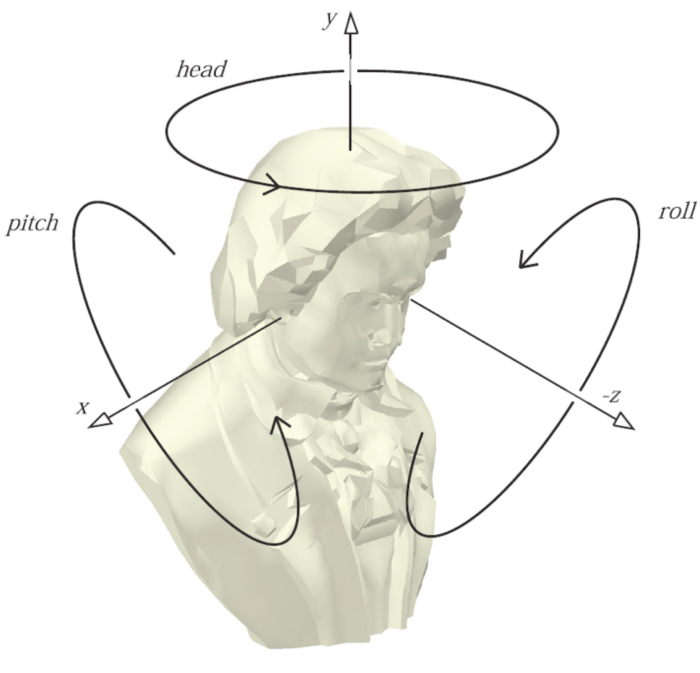
Transformações Geométricas em 3D

- Rotação em cada eixo coordenado:

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\phi) & 0 & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 & 0 \\ -\sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Transformações Geométricas em 3D

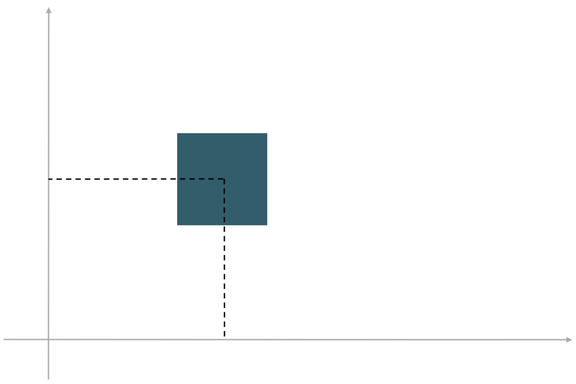
- Rotação em torno de um eixo genérico:
  - ***u = (ux, uy, uz)***

$$R = \begin{bmatrix} \cos\theta + u_x^2\left(1-\cos\theta\right) & u_x u_y\left(1-\cos\theta\right) - u_z\sin\theta & u_x u_z\left(1-\cos\theta\right) + u_y\sin\theta \\ u_y u_x\left(1-\cos\theta\right) + u_z\sin\theta & \cos\theta + u_y^2\left(1-\cos\theta\right) & u_y u_z\left(1-\cos\theta\right) - u_x\sin\theta \\ u_z u_x\left(1-\cos\theta\right) - u_y\sin\theta & u_z u_y\left(1-\cos\theta\right) + u_x\sin\theta & \cos\theta + u_z^2\left(1-\cos\theta\right) \end{bmatrix}.$$

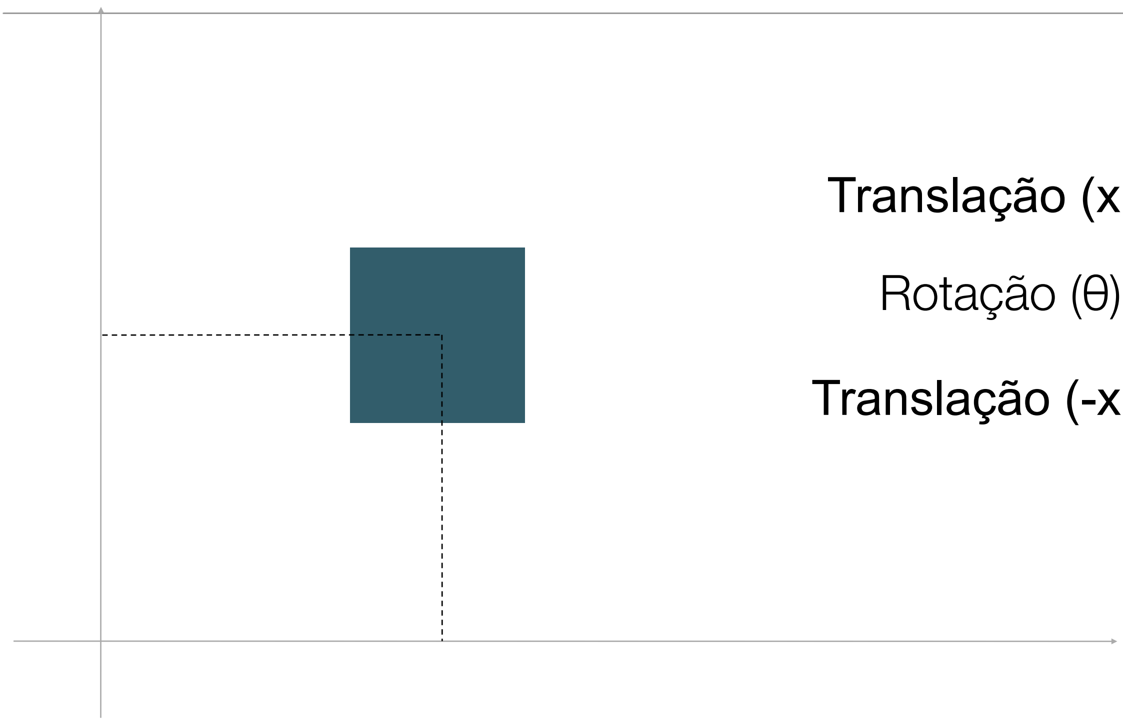
Composição de Transformações

Composição de Transformações

- Transformações mais gerais podem ser obtidas pela aplicação em seqüência das transformações básicas
  - Exemplo: rotação em um ponto qualquer



Composição de Transformações



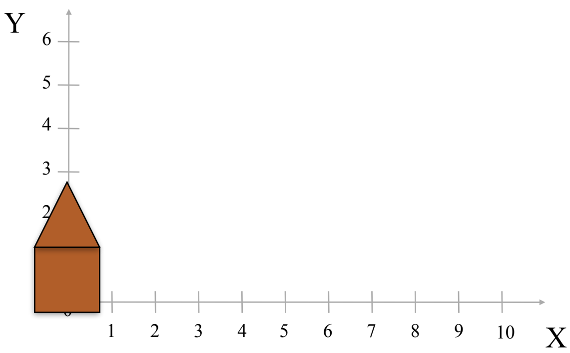
Translação (x,y)

Rotação (θ)

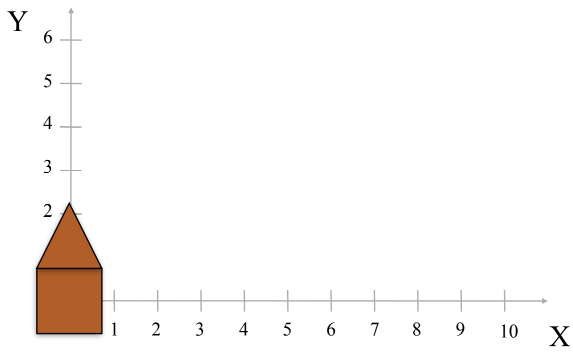
Translação (-x,-y)

Composição de Transformações

- Ordem das transformações é relevante



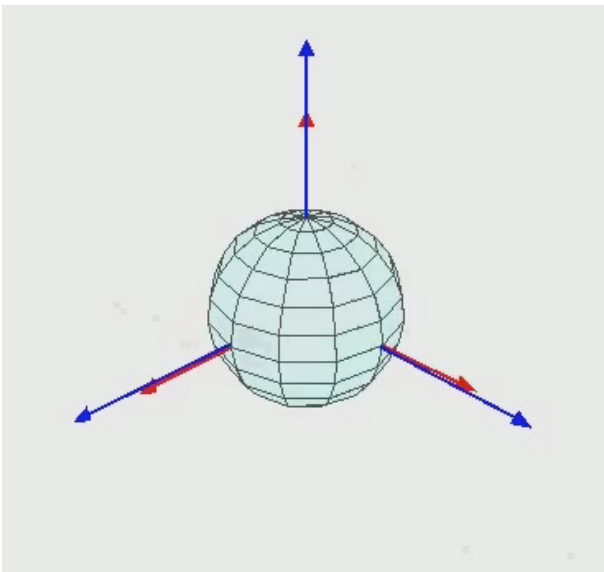
Translação → Rotação



Rotação → Translação

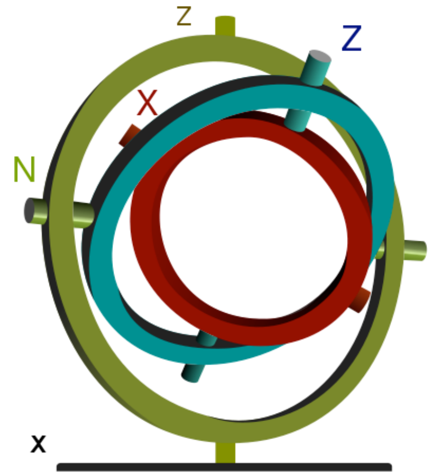
Composição de Transformações

- Rotações genéricas
  - Podem ser geradas por composição de 3 rotações elementares



Composição de Transformações

- Rotações genéricas
  - Equivalente a um "giroscópio"



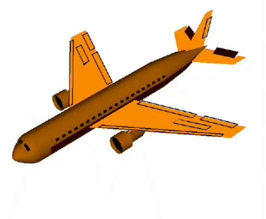
Composição de Transformações

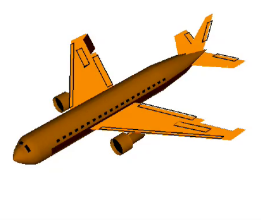
- Os 3 ângulos da rotação são chamados de ângulos de Euler

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\phi) & 0 & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

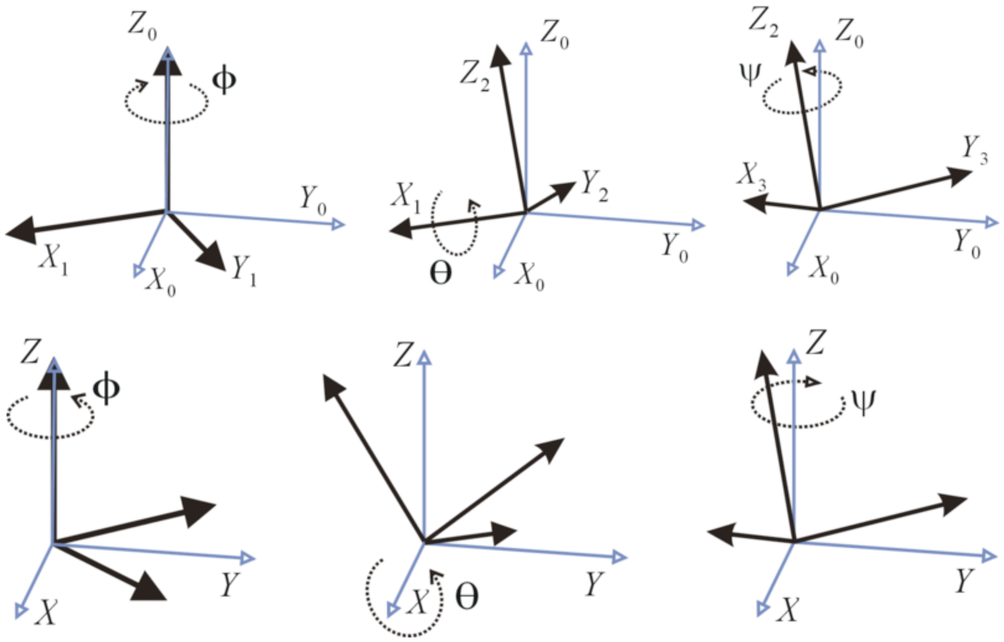
$$R_z = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 & 0 \\ -\sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$





Composição de Transformações

- Composição das rotações não é única:

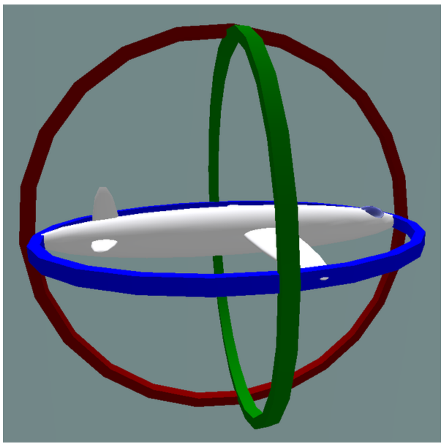


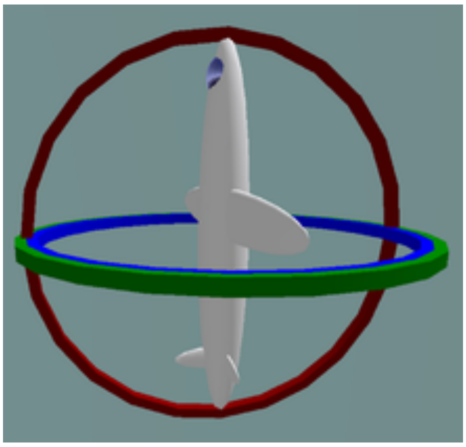
Composição de Transformações

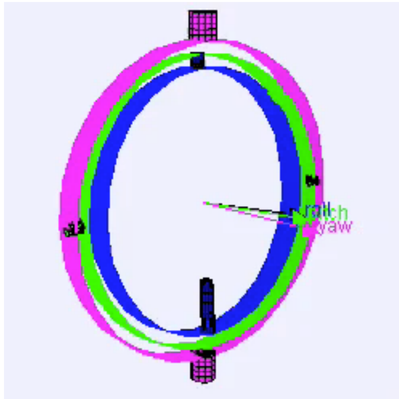
- Problema na representação por ângulos de Euler:
  - Controle de rotação em animação
    - Interpolação dos ângulos em separado não gera um resultado visualmente interessante.

Composição de Transformações

- Problema na representação por ângulos de Euler:
  - *Gimbal Lock*
    - Perda de um grau de liberdade para rotação



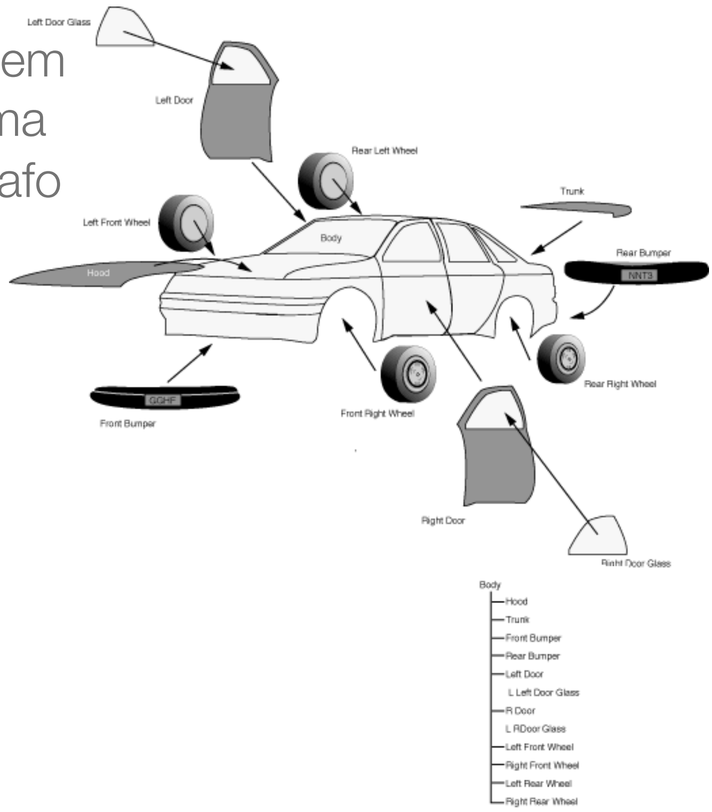




Grafo de Cena

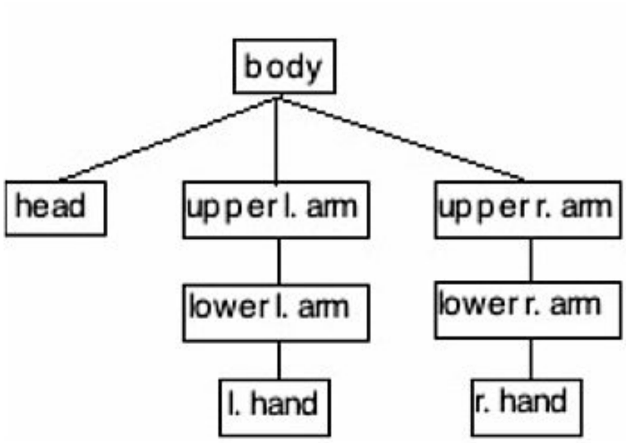
Objetos Hierárquicos

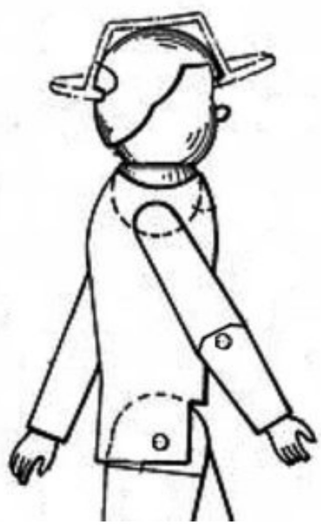
- Objetos complexos podem ser organizados de forma hierárquica como um grafo



Objetos Hierárquicos

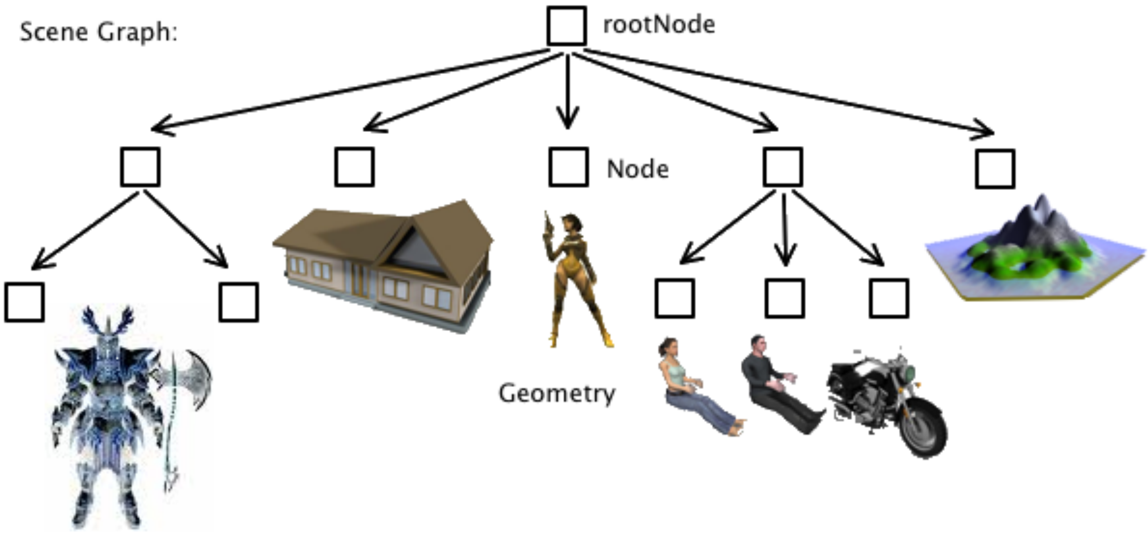
- Aplicação de transformações em cada nível da hierarquia
  - propagadas para os objetos filhos.





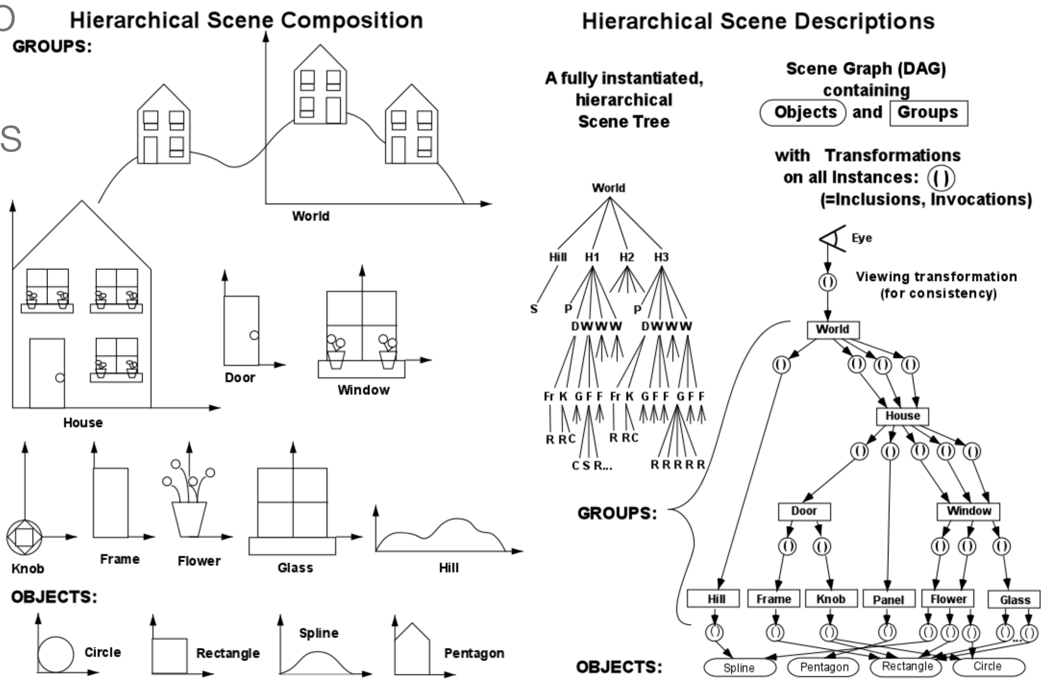
Grafo de Cena (*Scene Graphs*)

- Generalização do conceito de objetos hierárquicos para uma cena completa



Grafo de Cena (*Scene Graphs*)

- Generalização do conceito de objetos hierárquicos para uma cena completa



Aplicações em ***Three.JS/WebGL***

Grafos de Cena

- Baseados no objeto **`Object3D`**
  - Principais propriedades
    - **`name`**
    - **`parent`**
    - **`children`**
    - **`position`**
    - **`rotation`**
    - **`scale`**
    - **`matrix`**
    - **`matrixAutoUpdate`**
    - **`matrixWorldNeedsUpdate`**
    - **`rotationAutoUpdate`**
    - **`matrixWorld`**

Grafos de Cena

- Baseados no objeto **`Object3D`**
  - Principais métodos
    - `applyMatrix ( matrix)`
    - `translateX ( distance )`
    - `translateY ( distance )`
    - `translateZ ( distance )`
    - `localToWorld ( vector )`
    - `worldToLocal ( vector )`
    - `add ( object, ... )`
    - `remove ( object, ... )`
    - `traverse ( callback )`
    - `traverseVisible ( callback )`
    - `traverseAncestors ( callback )`
    - `updateMatrix ()`
    - `updateMatrixWorld ( force )`
    - `clone ()`
    - `getObjectByName (name)`
    - `getObjectById (id)`
    - `translateOnAxis (axis, distance)`
    - `rotateOnAxis (axis, angle)`

Grafos de Cena

- Exemplo simples:
  - Posicionamento de objetos em uma cena

```
(......)
    // Global Axis
    var globalAxis = new THREE.AxisHelper( 1.0 );
    scene.add( globalAxis );
    // Box
    var box = new THREE.BoxGeometry( 0.2, 0.2, 0.2 );
    var boxMat = new THREE.MeshBasicMaterial( {color: 0x0000ff,wireframe:true} );
    var cube = new THREE.Mesh( box, boxMat );
    cube.position.set(-0.7, 0.7, 0.0);
    cube.rotateOnAxis(new THREE.Vector3(1, 1, 1).normalize(), -Math.PI/4);
    scene.add( cube );
```

Grafos de Cena

- Exemplo simples:
  - Agrupando objetos

(……)

```
// Box
var groupBox = new THREE.Object3D();
var box = new THREE.BoxGeometry( 0.2, 0.2, 0.2 );
var boxMat = new THREE.MeshBasicMaterial( {color: 0x0000ff,wireframe:true} );
var cube = new THREE.Mesh( box, boxMat );
groupBox.add( cube );
var boxAxis = new THREE.AxisHelper( 0.3 );
groupBox.add( boxAxis );
groupBox.position.set(-0.7, 0.7, 0.0);
groupBox.rotateOnAxis(new THREE.Vector3(1, 1, 1).normalize(), -Math.PI/4);
scene.add( groupBox );
// Sphere
var groupSphere = new THREE.Object3D();
var sphereGeometry = new THREE.SphereGeometry( 0.2, 10, 10 );
var sphereMat = new THREE.MeshBasicMaterial( {color: 0xff0000, wireframe:true} );
var sphere = new THREE.Mesh( sphereGeometry, sphereMat );
groupSphere.add( sphere );
var sphereAxis = new THREE.AxisHelper( 0.4 );
groupSphere.add( sphereAxis );
groupSphere.position.set(0.0, 0.7, 0.0);
scene.add( groupSphere );
```

Grafos de Cena

- Exemplo simples:
  - Agrupando objetos (cont.)

```
// Group Third Line
var groupThirdLine = new THREE.Object3D();
var thirdLineAxis = new THREE.AxisHelper( 0.5 );
groupThirdLine.add( thirdLineAxis );
// Plane
var planeGeometry = new THREE.PlaneBufferGeometry(0.2, 0.5, 10, 10);
var planeMat = new THREE.MeshBasicMaterial( {color: 0xff00ff, wireframe:true} );
var plane = new THREE.Mesh( planeGeometry, planeMat );
plane.rotateOnAxis(new THREE.Vector3(1, 1, 1).normalize(), -Math.PI/4);
groupThirdLine.add( plane );
// Tetrahedron
var tetrahedronGeometry = new THREE.TetrahedronGeometry(0.2);
var tetrahedronMat = new THREE.MeshBasicMaterial( {color: 0x70aa70, wireframe:true} );
var tetrahedron = new THREE.Mesh( tetrahedronGeometry, tetrahedronMat );
tetrahedron.position.set( -0.7, 0.0, 0.0);
tetrahedron.rotateOnAxis(new THREE.Vector3(1, 1, 1).normalize(), -Math.PI/6);
groupThirdLine.add( tetrahedron );
// Dodecahedron
var dodecahedronGeometry = new THREE.DodecahedronGeometry(0.2);
var dodecahedronMat = new THREE.MeshBasicMaterial( {color: 0xaa7070, wireframe:true} );
var dodecahedron = new THREE.Mesh( dodecahedronGeometry, dodecahedronMat );
dodecahedron.position.set( -1.4, 0.0, 0.0);
dodecahedron.rotateOnAxis(new THREE.Vector3(1, 1, 1).normalize(), -Math.PI/6);
groupThirdLine.add( dodecahedron );
groupThirdLine.position.set(0.7, -0.7, 0.0);
scene.add( groupThirdLine );
```

A seguir... Sistema de Visualização