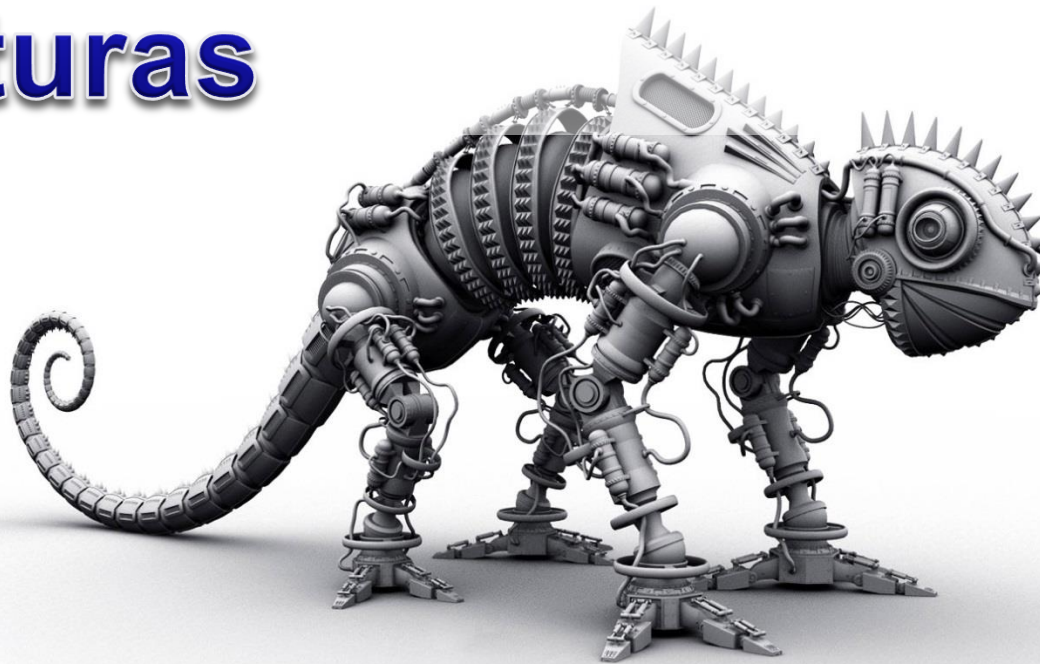


COMPUTAÇÃO GRÁFICA

Mapeamento de Texturas



```
var scene = new THREE.Scene();
```

```
var camera = new THREE.PerspectiveCamera( 100,  
                                          window.innerWidth / window.innerHeight,  
                                          0.1, 1000 );
```

```
var renderer = new THREE.WebGLRenderer({ antialiasing: true });
```

```
renderer.setSize( window.innerWidth, window.innerHeight );
```

```
renderer.setClearColor( 0x000000, 1 );
```

```
document.body.appendChild( renderer.domElement );
```

```
var geometry = new THREE.BoxGeometry( 8, 8, 8 );
```

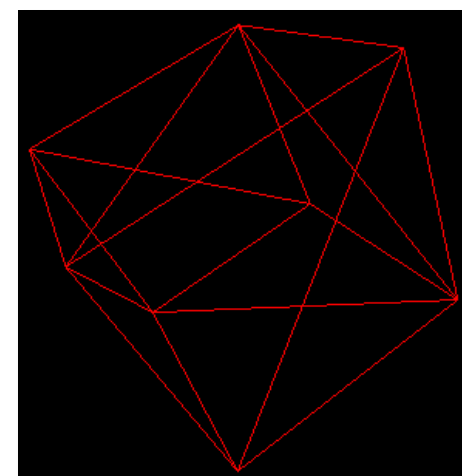
```
var textura = new THREE.MeshBasicMaterial({  
    color: 0xff0000, wireframe: true  
});
```

```
var mesh = new THREE.Mesh( geometry, textura );  
scene.add( mesh );
```

```
camera.position.z = 15;
```

```
function render(){  
    requestAnimationFrame( render );  
    mesh.rotation.x += .01;  
    mesh.rotation.y += .01;  
    mesh.rotation.z += .01;  
    renderer.render( scene, camera );  
}
```

```
render();
```





Mapeamento de Texturas



- **Mapeamento de textura ou texturização é a parte da computação gráfica, que se dedica em criar materiais e texturas a fim de simular superfícies mais realísticas.** Trata-se de um método gráfico computacional aplicado para 3D e foi desenvolvido na tese de doutorado de Edwin Catmull, em 1974. Atualmente diretor da Walt Disney Animation Studios e Pixar Animation Studios.

Computação Gráfica

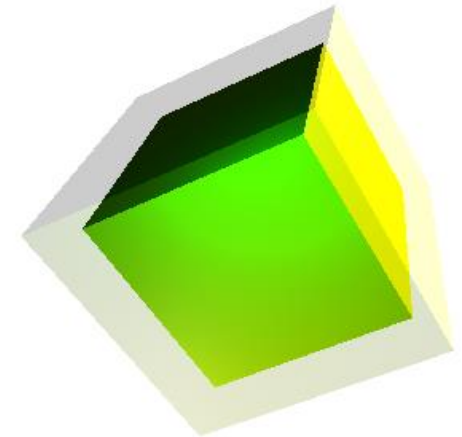
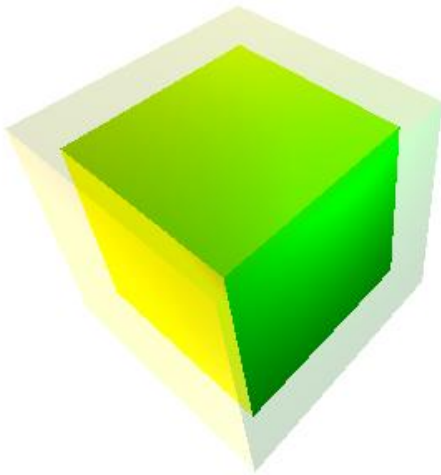




Mapeamento de Texturas



- Um mapa de textura é aplicado a um lado de um polígono no qual será renderizado.

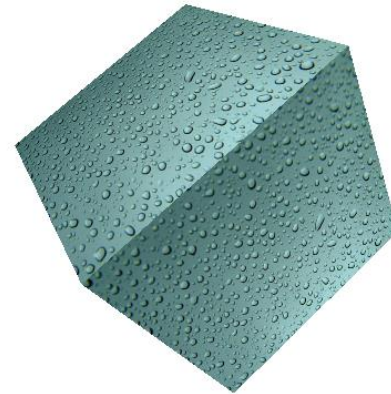




Mapeamento de Texturas



- A multitexturização é o uso de mais de uma textura ao mesmo tempo, em um mesmo polígono para criar texturas mais complexas.





Mapeamento de Texturas



Tipos de Mapas

- UV Mapping / UVW Mapping;
- Environment Mapping;
- Bump mapping;
- Parallax.





Mapeamento de Texturas



UV Map e UVW Mapping

- As coordenadas UV se referem a uma coordenada cartesiana em uma imagem de bitmap . Assim como X é a horizontal no espaço 3D, U é a horizontal em uma imagem plana 2D, ou seja, um bitmap. Da mesma forma que Y é vertical, V também é.
- A maioria dos modeladores 3D que faz mapeamento de texturas, geralmente usa UVW e não UV, pois o UVW possibilita um maior controle sobre o posicionamento da textura. O W é análogo ao Z, ele é a dimensão da profundidade. Pode-se pensar que uma dimensão de profundidade em uma imagem 2D é inútil, porém isso é um engano, já que esse valor pode controlar a orientação da textura numa face.

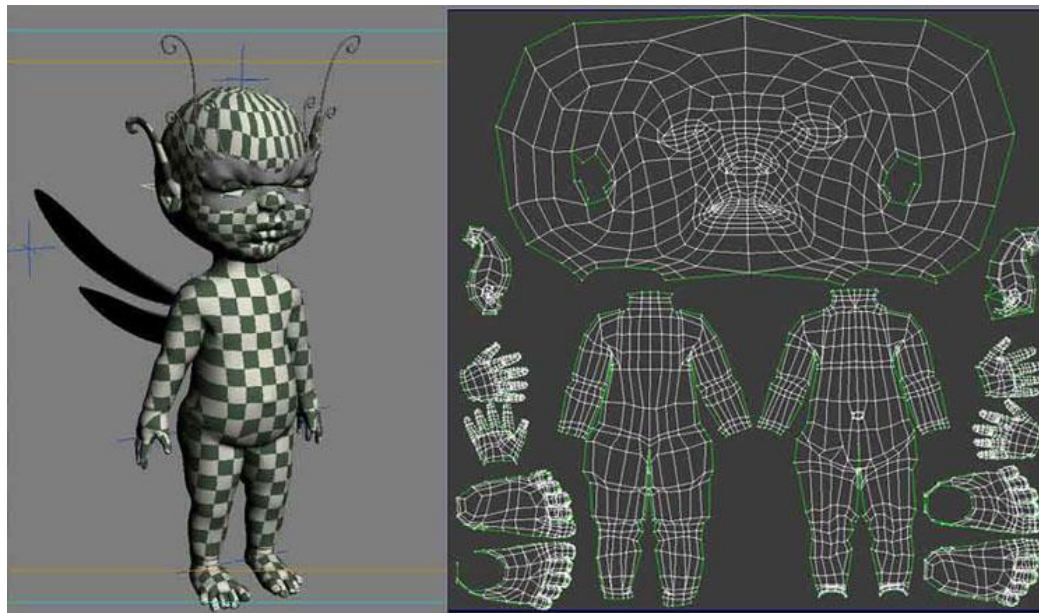


Mapeamento de Texturas



UV Map e UVW Mapping

- O UVW também é imprescindível no mapeamento de objetos complexos, como formas orgânicas, sólidos não-platônicos, primitivas não-geométricas e outros objetos de forma irregular.

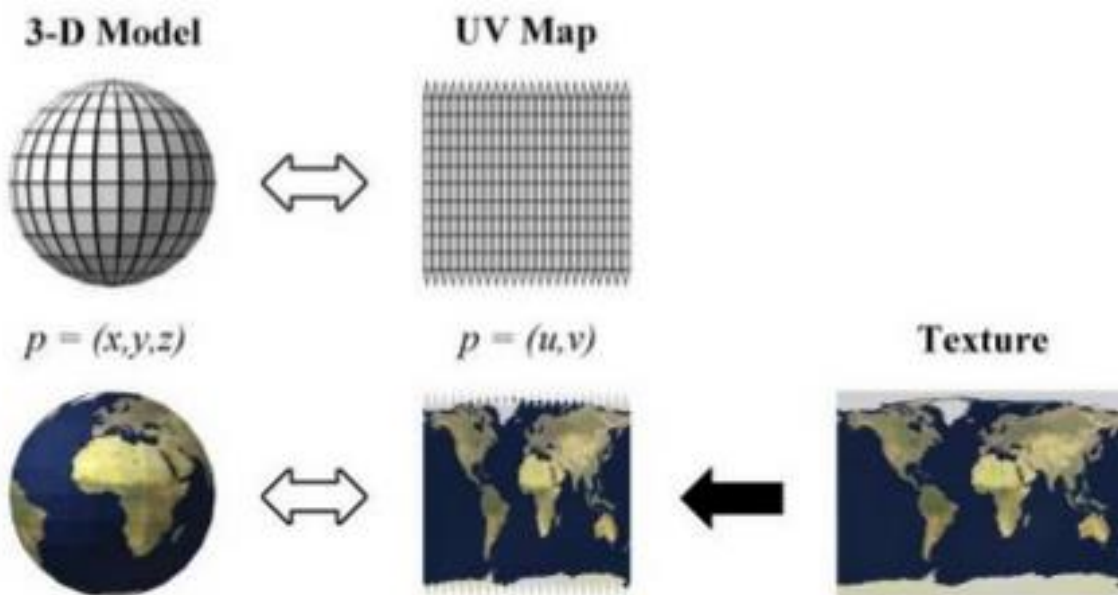




Mapeamento de Texturas



UV Map





Mapeamento de Texturas



- Texturização em Three.js.



```
var textura = new THREE.MeshLambertMaterial({  
    map: THREE.ImageUtils.loadTexture("img/pedra.jpg")  
});
```



Mapeamento de Texturas



- Texturização em Three.js.



Funcionou?

```
var geometry = new THREE.BoxGeometry( 8, 8, 8 );

var textura  = new THREE.MeshLambertMaterial(
  { map: THREE.ImageUtils.loadTexture("img/pedra.jpg") }
);

var mesh = new THREE.Mesh( geometry, textura);
scene.add( mesh );
```



Mapeamento de Texturas



- Texturização em Three.js.



Funcionou?

```
var geometry = new THREE.BoxGeometry( 8, 8, 8 );

var textura  = new THREE.MeshLambertMaterial(
    { map: THREE.ImageUtils.loadTexture("img/pedra.jpg") }
);

var mesh = new THREE.Mesh( geometry, textura);
scene.add( mesh );
```

```
var ambientLight = new THREE.AmbientLight( 0xFFFFFF );
scene.add(ambientLight);
```



Mapeamento de Texturas



amarelo.jpg



azul.jpg



globo.jpg



gotas.jpg



granitoazul.jpg



listrado.jpg



madeira.jpg



metal.jpg



pano.jpg



pedra.jpg



planetaterra.jpg



planetaterrabum
p.jpg



planetaterraspec.j
pg



verde.jpg



vermelho.jpg



vidro.jpg





Mapeamento de Texturas



- Com base no código disponível, realize as alterações necessárias para modelar o seguinte cenário.





Mapeamento de Texturas



- Multitexturização em Three.js.

```
var geometry = new THREE.BoxGeometry( 8, 8, 8 );

var textura  = new THREE.MeshLambertMaterial(
    { map: THREE.ImageUtils.loadTexture("img/pedra.jpg") }
);

var material = new THREE.MeshLambertMaterial({
    map: THREE.ImageUtils.loadTexture("img/gotas.jpg"),
    opacity: 0.5,
    transparent: true
});

var mesh = new THREE.SceneUtils.createMultiMaterialObject(
    geometry,
    [textura, material]
);
scene.add( mesh );
```





Mapeamento de Texturas



- Multitexturização em Three.js.

```
var geometry = new THREE.PlaneGeometry(500, 500, 10, 10);
```



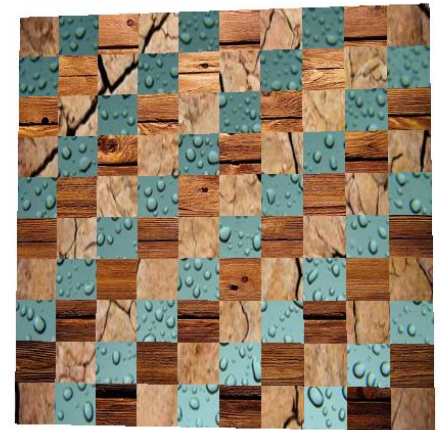


Mapeamento de Texturas



- Multitexturização em Three.js.

```
var materiais = [];  
  
materiais.push(  
  new THREE.MeshLambertMaterial(  
    { map: THREE.ImageUtils.loadTexture("img/pedra.jpg"),  
      side:THREE.DoubleSide }  
  )  
);  
  
materiais.push(  
  new THREE.MeshLambertMaterial(  
    { map: THREE.ImageUtils.loadTexture("img/gotas.jpg"),  
      side:THREE.DoubleSide }  
  )  
);  
  
materiais.push(  
  new THREE.MeshLambertMaterial(  
    { map: THREE.ImageUtils.loadTexture("img/madeira.jpg"),  
      side:THREE.DoubleSide }  
  )  
);
```

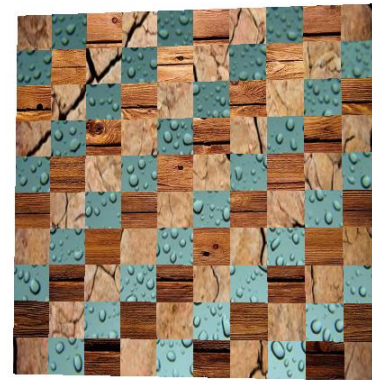




Mapeamento de Texturas



- Multitexturização em Three.js.



```
var l = geometry.faces.length / 2;

for (var i = 0; i < l; i++) {
    var j = 2 * i;
    geometry.faces[j].materialIndex = i % 3;
    geometry.faces[j + 1].materialIndex = i % 3;
}

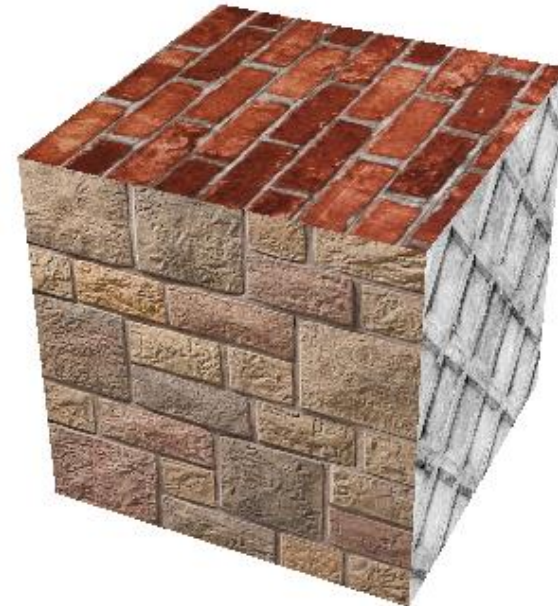
mesh = new THREE.Mesh(geometry, new THREE.MeshFaceMaterial(materials));
scene.add(mesh);
```




Mapeamento de Texturas



Com base no código disponível, realize as alterações necessárias para modelar um cubo que possua 6 texturas diferentes.



lado01.jpg



lado02.jpg



lado03.jpg



lado04.jpg



lado05.jpg



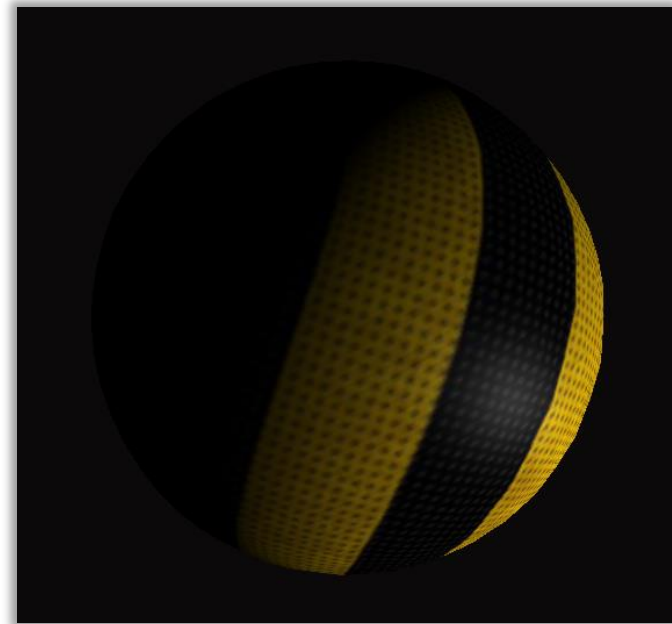
lado06.jpg



Mapeamento de Texturas



- Texturização com realismo



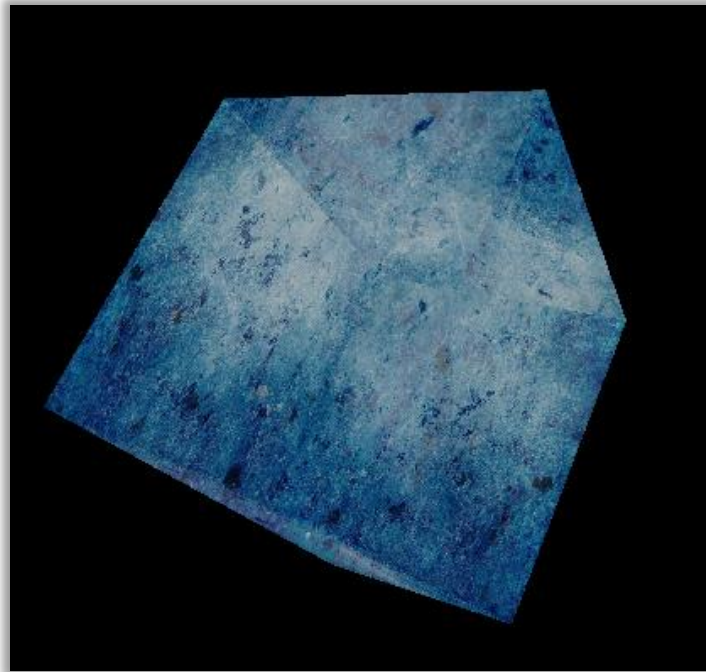
Qual tipo de material utilizado na construção de cada malha de polígonos?



Mapeamento de Texturas



- Texturização com realismo



Quais técnicas foram utilizadas?



Mapeamento de Texturas



Environment Mapping

- Uma necessidade crescente de games modernos é o realismo e a autenticidade na representação de ambientes reais ou completamente fantásticos. Para dar uma sensação real de que o jogador faz parte do cenário, por mais fantasioso que seja, efeitos presentes no mundo real devem ser simulados
- Além de sombreamento, texturização de qualidade e um ambiente bem desenvolvido, uma necessidade que pode passar despercebida, mas que pode fazer toda a diferença é a reflexão do ambiente por partes cromadas ou polidas da roupa ou veículo do personagem sendo controlado. Para produzir este efeito, uma opção atual é o chamado environment mapping, ou mapeamento de ambiente.



Mapeamento de Texturas



Environment Mapping

- Em aplicações que não são em tempo real, a técnica do raytracing é muito mais indicada, graças à sua precisão e qualidade superiores, por tratar a reflexão como um efeito físico, e analisar o posicionamento de raios de luz ao redor do objeto refletor. Porém, analisar cada um destes raios de luz e representá-los em tempo real requer um grande poder de processamento.
- Note como a colher reflete o ambiente ao seu redor. Uma alternativa mais leve



Computação Gráfica



Mapeamento de Texturas



Environment Mapping

- Para solucionar este problema, o environment mapping é uma técnica que, apesar de sacrificar um pouco da qualidade produzida por raytracing, garante um efeito similar, mas que exige muito menos do computador. Existem várias formas de mapeamento de ambiente, mas as mais difundidas e utilizadas são o mapeamento cúbico ou esférico. Basicamente, uma imagem pré-renderizada do ambiente é armazenada em um corpo oco, e então esta textura é projetada no objeto refletor.
- O mapeamento esférico está sendo abandonado graças a efeitos de distorção que acabam acontecendo, graças ao formato no qual o mapa de ambiente é armazenado. A perda de perspectiva faz com que o realismo seja prejudicado. Com o mapeamento cúbico, o ambiente pode ser representado através de seis imagens diferentes, unidas de modo que as divisões não ficam evidentes, e distorções sejam imperceptíveis.



Mapeamento de Texturas



Environment Mapping

- Uma desvantagem é que, como o ambiente ao redor do objeto é o único aspecto mapeado, não existe reflexo do próprio corpo refletor, através desta técnica. Caso isto seja necessário, deve ser feito separadamente ou através de raytracing.



Mapeamento de Texturas



Environment Mapping



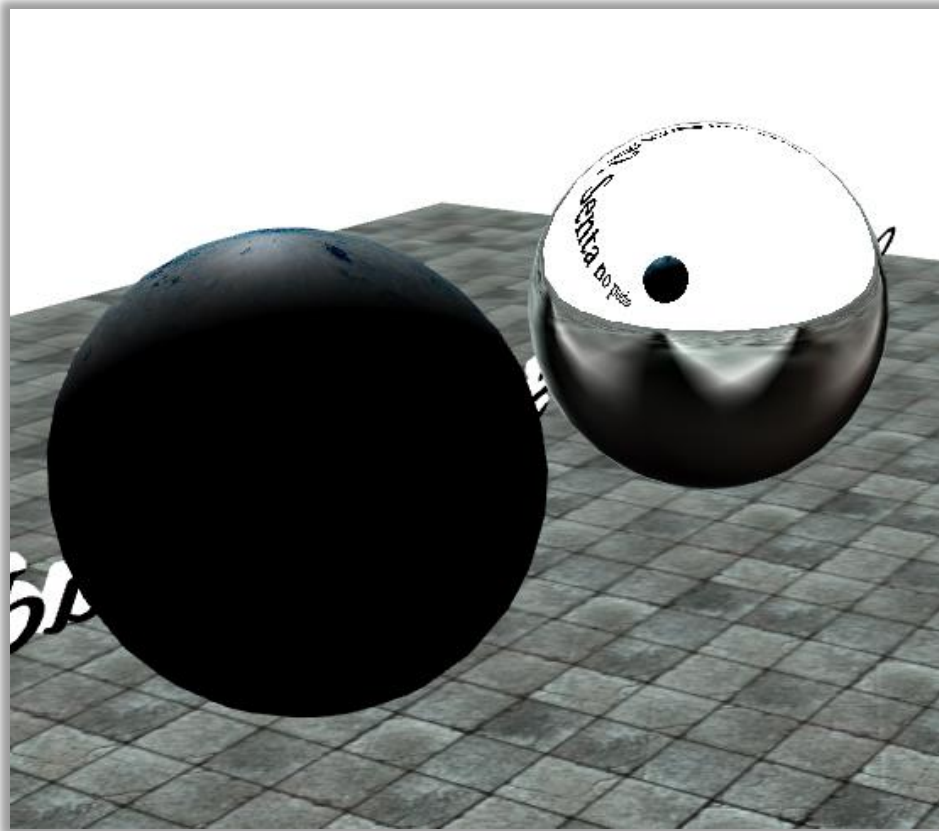
Computação Gráfica



Mapeamento de Texturas



- Texturização com realismo



Quais técnicas foram utilizadas?



Mapeamento de Texturas



- Texturização com realismo

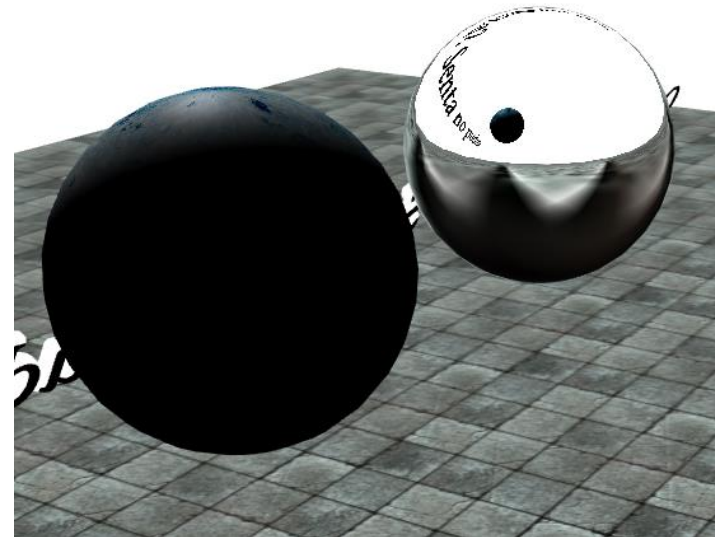
Constructor

```
CubeCamera(near, far, cubeResolution)
```

near -- The near clipping distance.

far -- The far clipping distance

cubeResolution -- Sets the width of the cube.



```
var sphereGeom = new THREE.SphereGeometry( 50, 32, 16 );

var mirrorSphereCamera = new THREE.CubeCamera( 0.1, 5000, 512 );
scene.add( mirrorSphereCamera );

var mirrorSphereMaterial = new THREE.MeshBasicMaterial( { envMap: mirrorSphereCamera.renderTarget } );

var mirrorSphere = new THREE.Mesh( sphereGeom, mirrorSphereMaterial );
mirrorSphere.position.set(15,100,0);

mirrorSphereCamera.position = mirrorSphere.position;
scene.add(mirrorSphere);
```



Mapeamento de Texturas



- Texturização com realismo

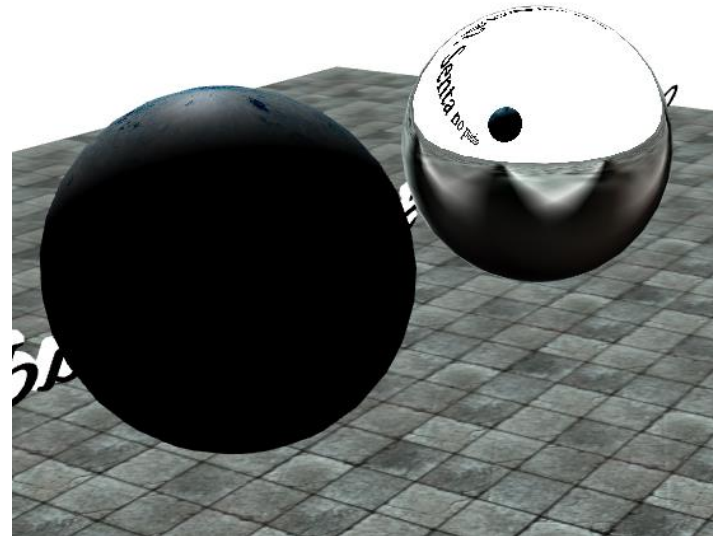
Constructor

```
CubeCamera(near, far, cubeResolution)
```

[near](#) -- The near clipping distance.

[far](#) -- The far clipping distance

[cubeResolution](#) -- Sets the width of the cube.



```
function render()
```

```
{
```

```
    mirrorSphere.visible = false;
```

```
    mirrorSphereCamera.updateCubeMap( renderer, scene );
```

```
    mirrorSphere.visible = true;
```



Mapeamento de Texturas



Curiosidade... Como funciona o Ray Tracing?

- Ele funciona quase como os raios de luz de verdade, mas ao contrário. Normalmente a luz vem de uma fonte e percorre um certo espaço até atingir algo e refletir o objeto de acordo com suas características. No ray tracing, quem está observando se torna a fonte de luz, para que somente a luz visível seja processada. Ou seja, o ray tracing projeta um raio de luz que sai de quem está observando e deve atingir todos os objetos da imagem, determinando o que está mais próximo e o que está mais longe dele.





Mapeamento de Texturas



Curiosidade... Como funciona o Ray Tracing?

- Assim, quando o raio atingir um objeto em seu caminho, ele irá verificar que aquele é realmente o ponto mais próximo do observador e assim irá determinar qual a cor. Para isso, a iluminação é calculada em relação também às outras fontes de luz, que podem ser originárias de reflexos de outros objetos, por exemplo. Esses são os chamados raios de luz secundários, que incluem: os raios de reflexão (criam os reflexos dos objetos nos outros), os raios de refração (dão “vida” à luz, criando a passagem da luz de um lugar para outro) e os raios de sombra.
- Isso tudo ajuda para que a imagem fique muito mais realista. Como várias outras técnicas utilizadas em computação gráfica, essa também é muito usada no cinema e na TV, em efeitos especiais.



Mapeamento de Texturas



Bump Mapping

- Bump mapping é uma técnica de processamento de texturas que vem se tornando cada vez mais um requisito básico de gráficos de games modernos, devido à demanda crescente de realismo ou detalhamento visual em jogos. Com a capacidade de processamento crescente do hardware disponível no mercado, está se tornando cada vez mais corriqueira a implementação de novas técnicas de texturização.
- Para dar uma ilusão de pequenas profundidades na superfície de corpos simples ou complexos, o bump mapping segue um processo simples: ele altera em termos computacionais a orientação normal da superfície, criando uma falsa profundidade, que reage à iluminação produzindo um pequeno nível de sombreamento, criando assim uma impressão de “buracos” ou irregularidades na textura sendo processada.

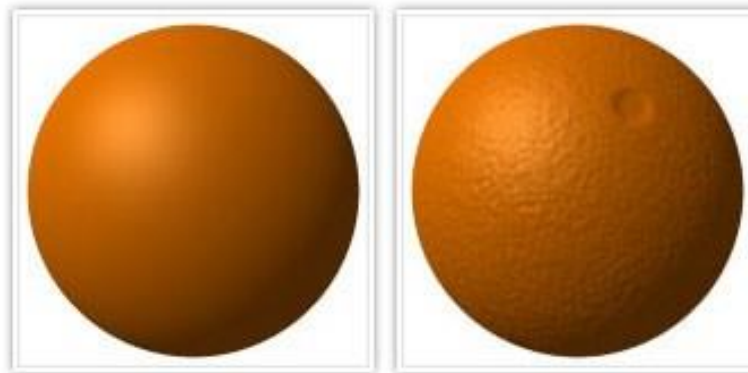


Mapeamento de Texturas



Bump Mapping

- É um sistema bastante simples, que produz resultados muito interessantes. Um exemplo clássico de como o bump mapping influencia a percepção da identidade de um objeto simples é a modelagem de uma laranja: é fácil representar grosseiramente a fruta através de uma simples esfera com uma textura da cor adequada, mas a falta de detalhes é gritante. A aplicação de bump mapping dá uma impressão de realismo crucial, que pode ser um grande diferencial em um jogo ou em um filme de animação tridimensional.





Mapeamento de Texturas



Bump Mapping

- Como o procedimento depende da iluminação do ambiente para proporcionar a ilusão de profundidade, o bump mapping é um sistema dinâmico, que alterna suas propriedades conforme o objeto texturizado se move através de uma cena com iluminação complexa. Isso garante uma riqueza visual bastante grande, e um nível de realismo que não seria possível sem a aplicação da técnica.
- Com esta simples tecnologia, que consiste apenas em um processamento matemático de um parâmetro da superfície de um objeto, a qualidade da representação visual aumenta de uma maneira impressionante. Com a evolução do hardware, esta técnica se torna cada vez mais simples de ser utilizada.



Mapeamento de Texturas



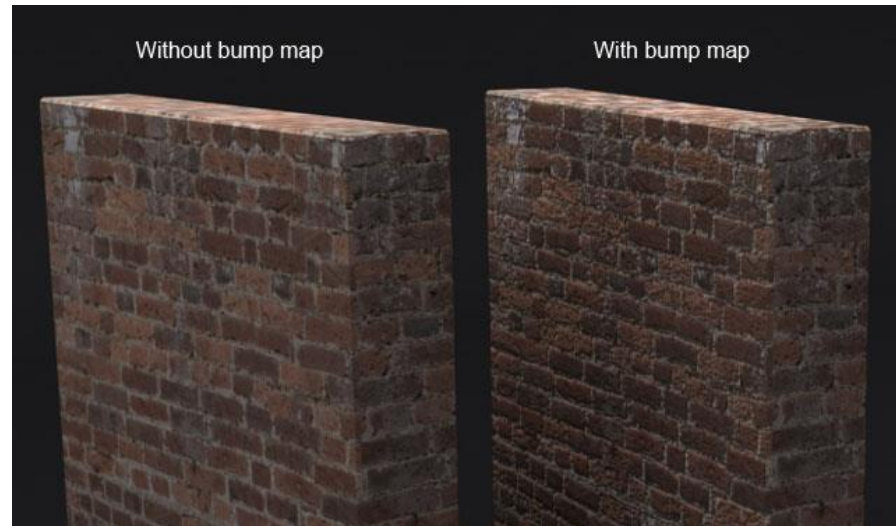
Bump Mapping



Base
Model



Bump
Mapping





Mapeamento de Texturas



Normal Mapping

É uma técnica utilizada para simular a iluminação de relevos.

Muito usado para adicionar detalhes sem usar mais polígonos. Um mapa normal é geralmente uma imagem RGB que corresponde ao X, Y e Z as coordenadas de uma superfície normal a partir de uma versão mais detalhada do objeto.

Um uso comum desta técnica é melhorar muito a aparência e os detalhes de um modelo de baixos polígonos.



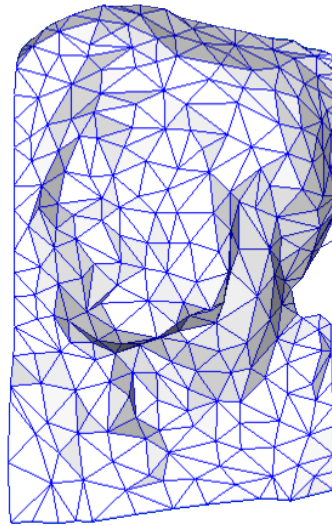
Mapeamento de Texturas



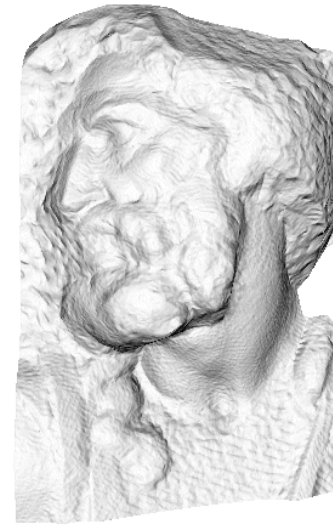
Normal Mapping



original mesh
4M triangles



simplified mesh
500 triangles



simplified mesh
and normal mapping
500 triangles



Mapeamento de Texturas



Parallax Mapping

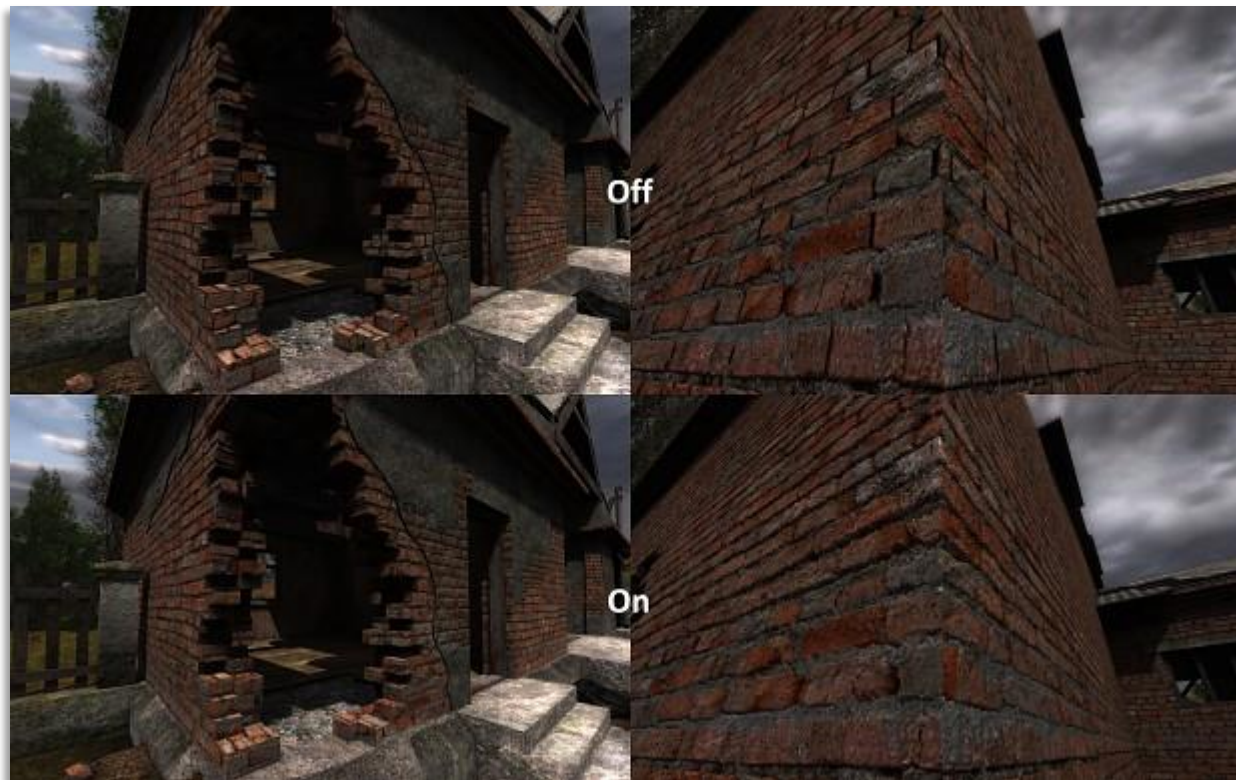
- Parallax mapping conhecido também como mapeamento de deslocamento virtual é um aprimoramento do bump map ou normal map.
- Para o usuário final, isso significa que as texturas, como muros de pedra terá mais profundidade e realismo aparente, portanto, maior influência sobre o desempenho da simulação. Parallax mapping foi introduzido por Tomomichi Kaneko em 2001.
- Parallax mapping é implementado através do deslocamento das coordenadas da textura em um ponto no polígono por uma função do ângulo de visão no espaço tangente e o valor do map de altura desse ponto.



Mapeamento de Texturas



Parallax Mapping

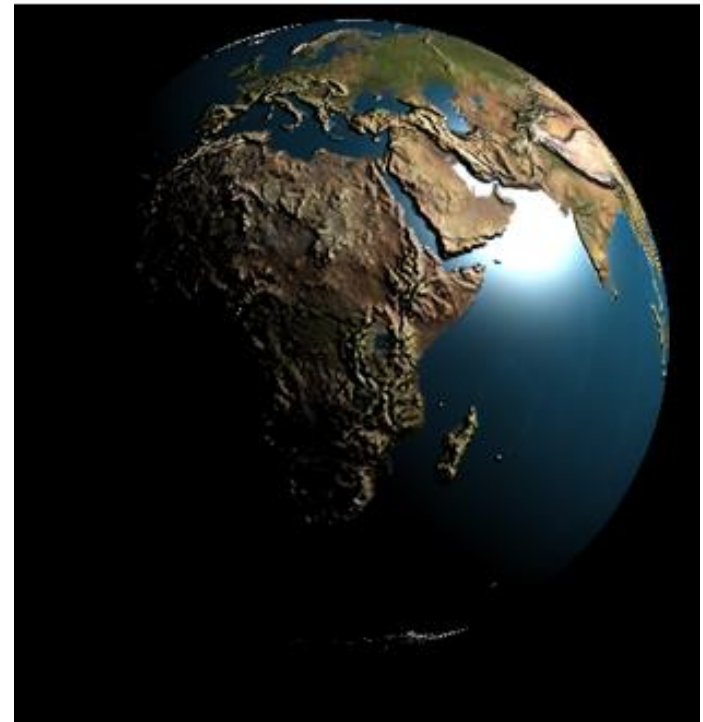
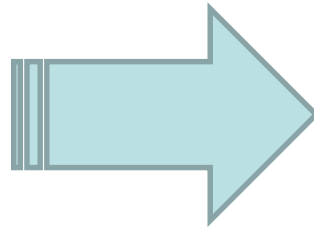




Mapeamento de Texturas



- Texturização com realismo



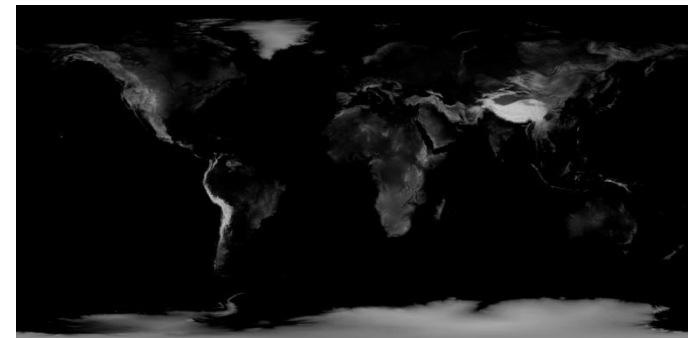
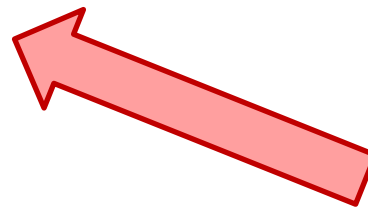
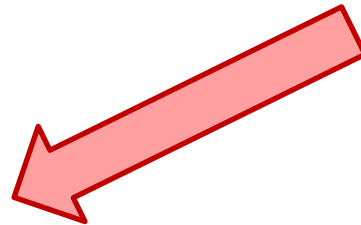
Quais técnicas foram utilizadas?



Mapeamento de Texturas



- Texturização com realismo





Mapeamento de Texturas



- Texturização com realismo

```
//Definição do Objeto  
var sphere = new THREE.SphereGeometry(0.5, 32, 32);  
//Definição do Material  
var material = new THREE.MeshPhongMaterial();
```





Mapeamento de Texturas



- Texturização com realismo

```
//Definição do Objeto  
var sphere = new THREE.SphereGeometry(0.5, 32, 32);  
//Definição do Material  
var material = new THREE.MeshPhongMaterial();
```



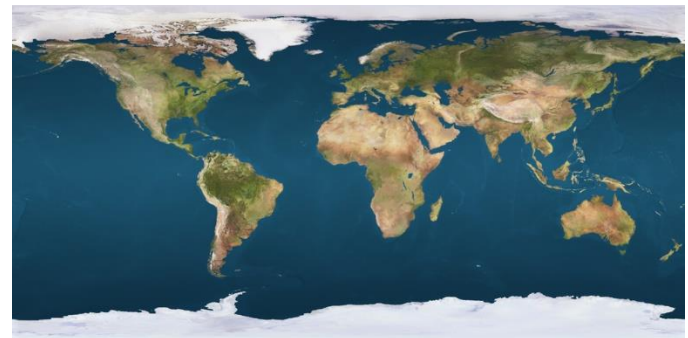


Mapeamento de Texturas



- Texturização com realismo

```
material.map = THREE.ImageUtils.loadTexture('img/planetaterra.jpg');
```



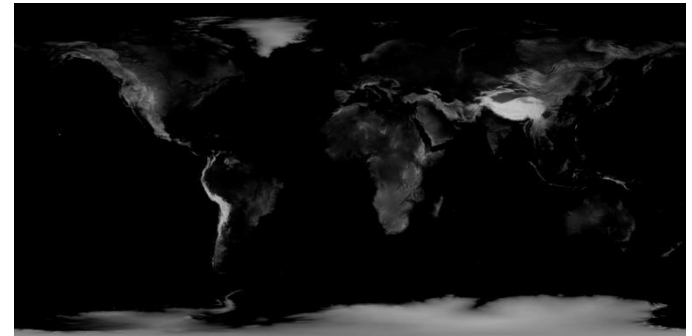
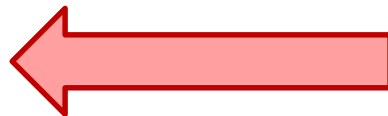


Mapeamento de Texturas



- Texturização com realismo

```
material.bumpMap    = THREE.ImageUtils.loadTexture('img/planetaterrabump.jpg');  
material.bumpScale = 0.04;
```





Mapeamento de Texturas



- Texturização com realismo

```
material.specularMap    = THREE.ImageUtils.loadTexture('img/planetaterraspec.jpg');  
material.specular      = new THREE.Color('grey');
```





Mapeamento de Texturas



- <https://www.youtube.com/watch?v=nPEYdw2Ssa8>
- http://threejs.org/examples/css3d_periodictable.html

H																			He
Li	Be											B	C	N	O	F	Ne		
Na	Mg											Al	Si	P	S	Cl	Ar		
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr		
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe		
Cs	Ba		Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn		
Fr	Ra		Rf	Db	Sg	Bh	Hs	Mt	Ds	Rg	Cn	Uut	Fl	Uup	Lv	Uus	Uuo		
		La	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu			
		Ac	Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr			



Bibliografia



- TORI, Romero et al. **Fundamentos de Computação Gráfica**. Rio de Janeiro: Livros Técnicos e Científicos Editora Ltda, 1987.
- ROCHA, Prof. Me. João Carlos. **Cor Luz, Cor Pigmento e os Sistemas RGB e CMY.**, São Paulo, v. 1, n. 1, p.2-2, dez. 2012.