Universidade Cruzeiro do Sul

Computação Gráfica

- TEXTURAS

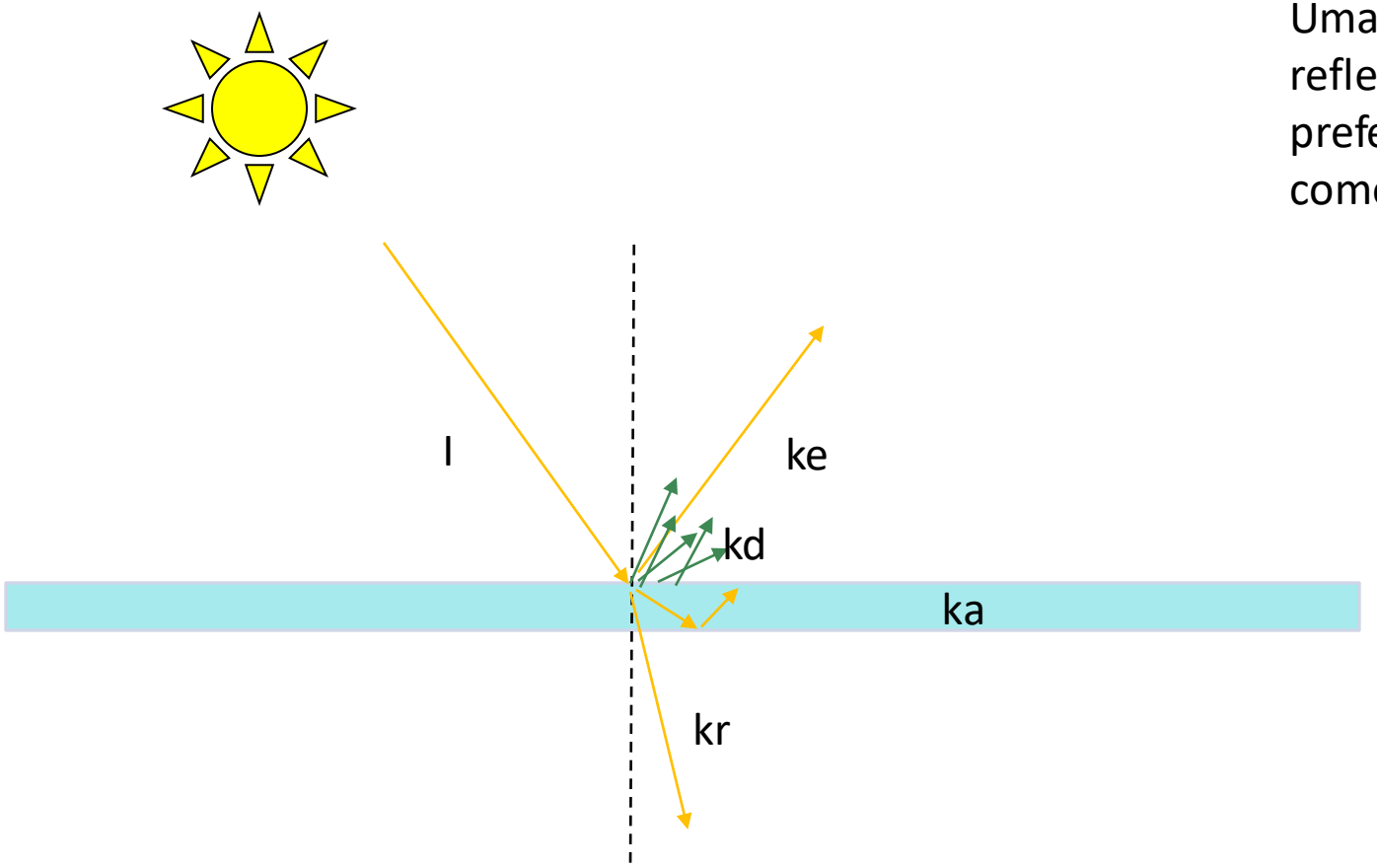
amlucena@cruzeirodosul.edu.br

Na última aula...

Como é equacionada a propagação da luz?

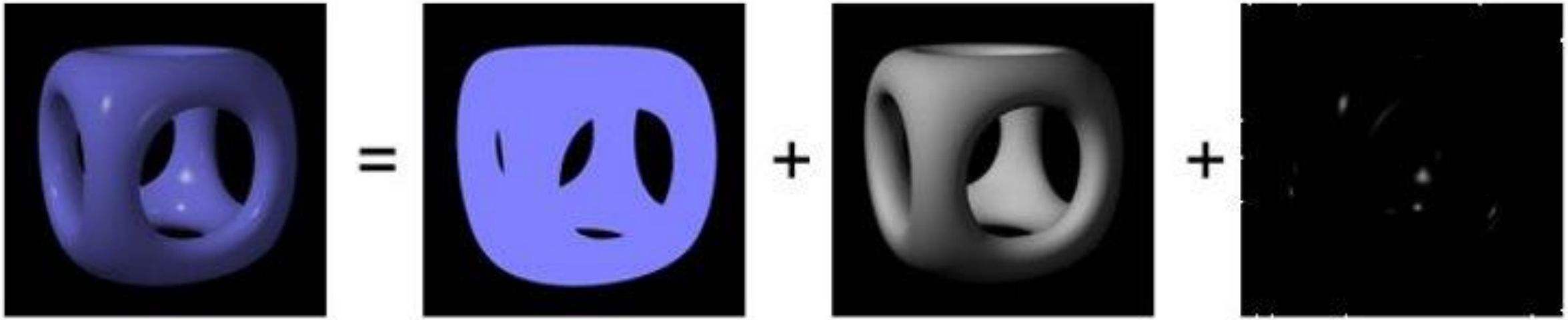
I - Intensidade
ke – refletido
ka – absorvido
kr – refratado
kd - difuso

I = ke+ka+kr+kd



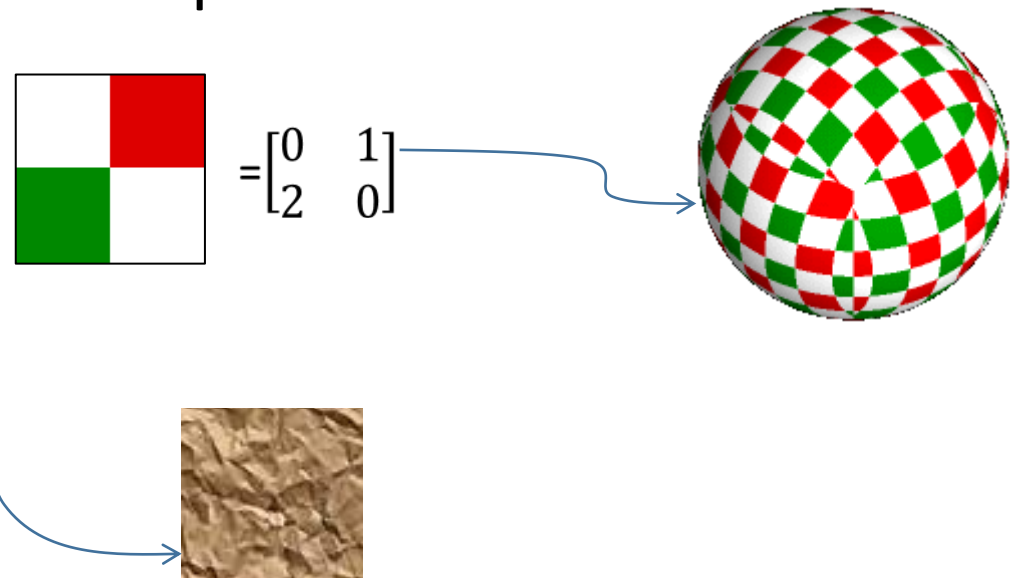
Uma parte kd que é refletida sem direção preferencial, conhecida como reflexão difusa.

Na última aula...



Para que texturas?

- Aplicações de materiais simples nem sempre são suficientes para representar elementos reais. Para isso, usamos mapeamento de texturas.
- Por quê **mapeamento**? → A idéeia é reproduzir sobre a superfície de um objeto as propriedades descritas por:
  - Uma matriz
  - Uma imagem
  - Uma função



- É necessário estudar:
  - Mapeamento
  - *Wrapping*

Função de Mapeamento

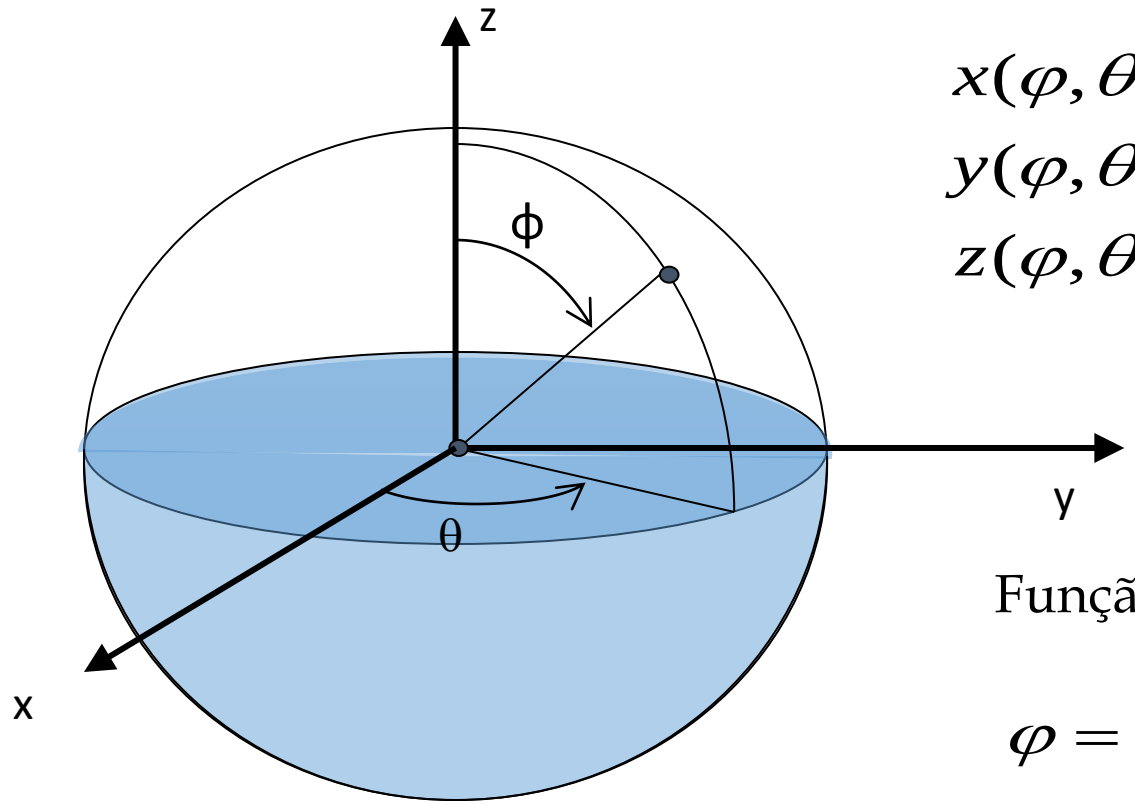
- Retorna o ponto do objeto correspondente a cada ponto do espaço de textura

  $(x, y, z) = F(s, t)$

- Corresponde à forma com que a textura é usada para “embrulhar” (*wrap*) o objeto
  - Na verdade, na maioria dos casos, precisamos de uma função que nos permita “desembrulhar” (*unwrap*) a textura do objeto, isto é, a inversa da função de mapeamento
- Se a superfície do objeto pode ser descrita em forma paramétrica esta pode servir como base para a função de mapeamento

Fonte: Slides do prof. Paulo Roma - UFRJ

Parametrização da Esfera



Função de mapeamento

$$x(\varphi,\theta) = \sin\varphi\cos\theta$$
$$y(\varphi,\theta) = \sin\varphi\sin\theta$$
$$z(\varphi,\theta) = \cos\varphi$$

$$\varphi = \pi \cdot t$$
$$\theta = 2\pi \cdot s$$

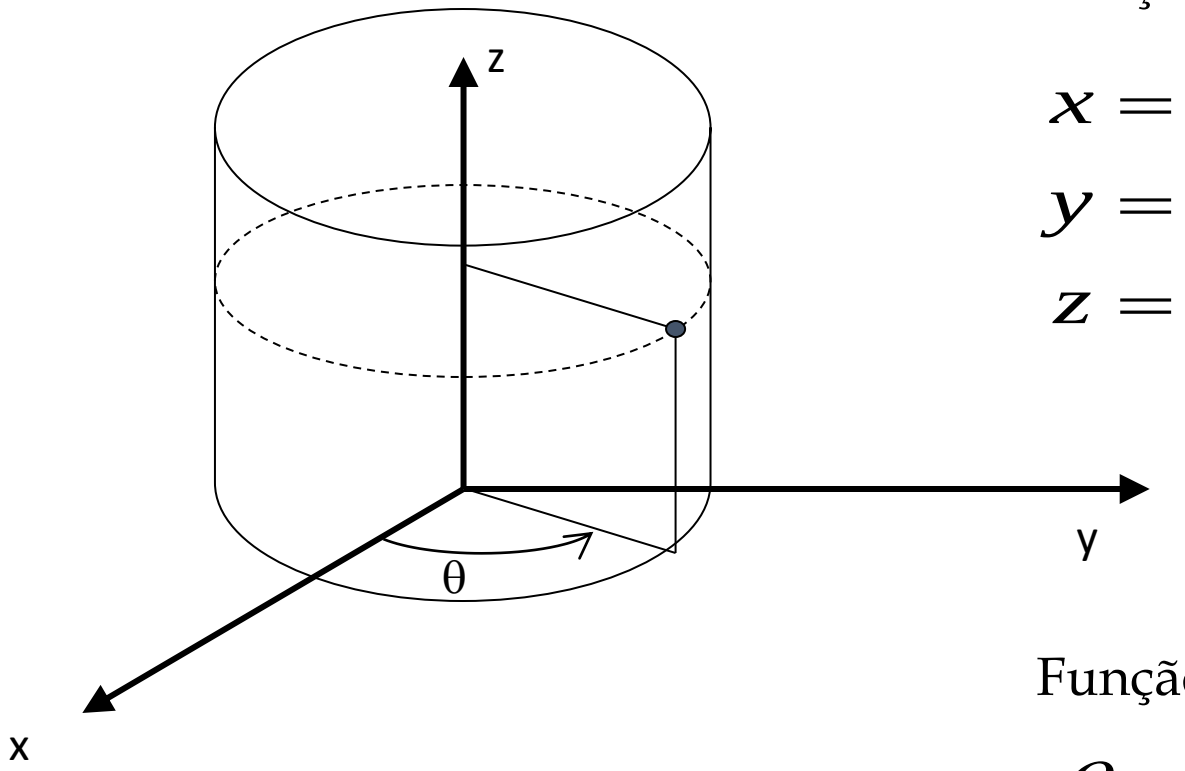
Função de mapeamento inversa

$$\varphi = \arccos z$$
$$\theta = \arctan\frac{y}{x}$$

$$t = \frac{\arccos z}{\pi}$$
$$s = \frac{\arctan\frac{y}{x}}{2\pi}$$

Fonte: Slides do prof. Paulo Roma - UFRJ

Parametrização do Cilindro



Função de mapeamento

$$x = \cos\theta$$
$$y = \sin\theta$$
$$z = z$$

$$\theta = 2\pi \cdot s$$
$$z = t$$

Função de mapeamento inversa

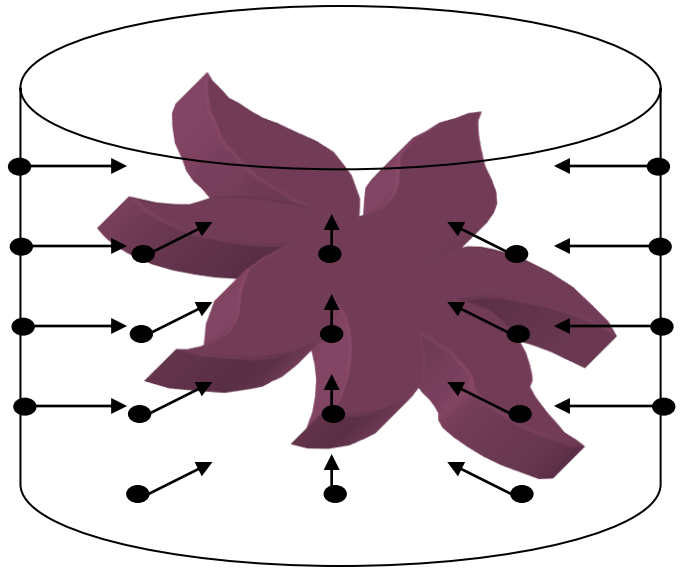
$$\theta = \arctan\frac{y}{x}$$
$$z = z$$

$$s = \frac{\theta}{2\pi}$$
$$t = z$$

Fonte: Slides do prof. Paulo Roma - UFRJ

Parametrizando Objetos Genéricos

- O que fazer quando o objeto não comporta uma parametrização natural?
- Uma sugestão é usar um mapeamento em 2 estágios [Bier e Sloan]:
  - Mapear textura sobre uma superfície simples como cilindro, esfera, etc aproximadamente englobando o objeto
  - Mapear superfície simples sobre a superfície do objeto. Pode ser feito de diversas maneiras
    - Raios passando pelo centróide do objeto
    - Raios normais à superfície do objeto
    - Raios normais à superfície simples
    - Raios refletidos (*environment mapping*)

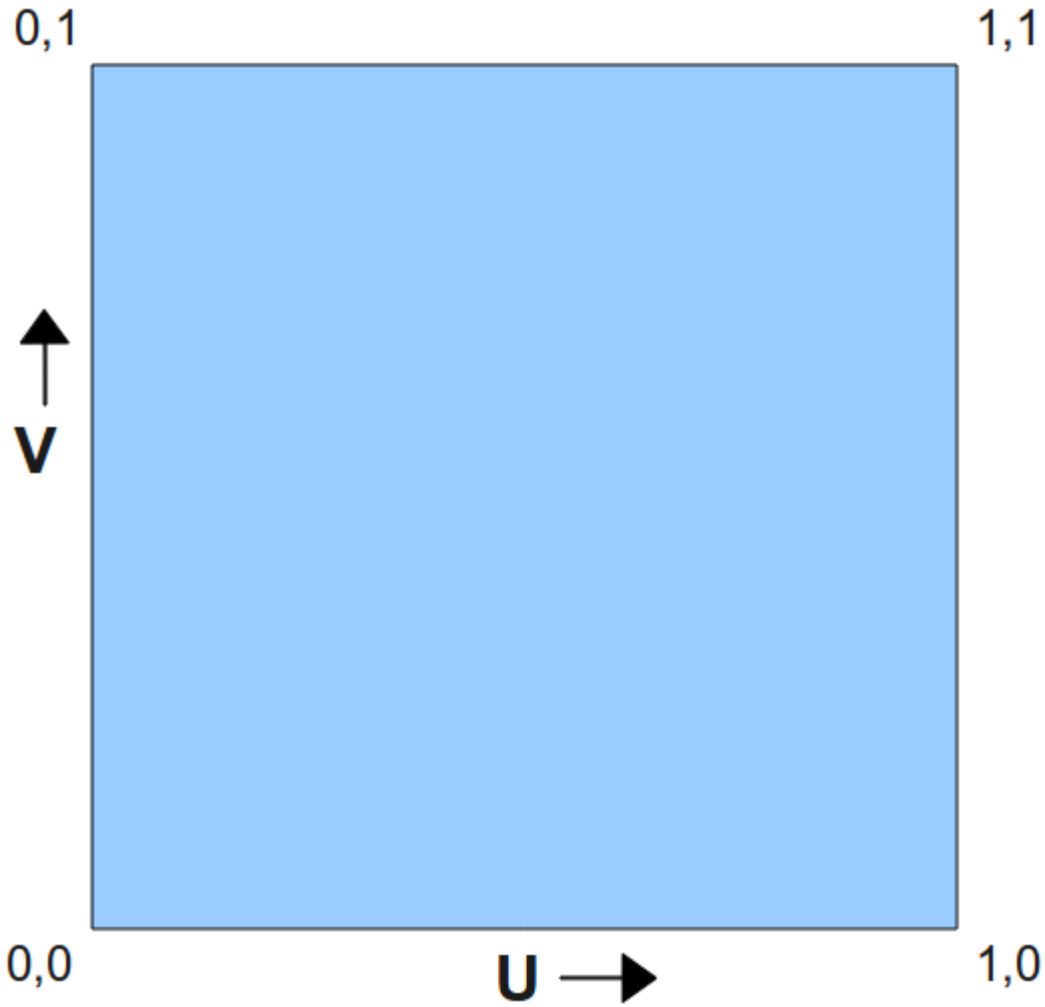


Fonte: Slides do prof. Paulo Roma - UFRJ

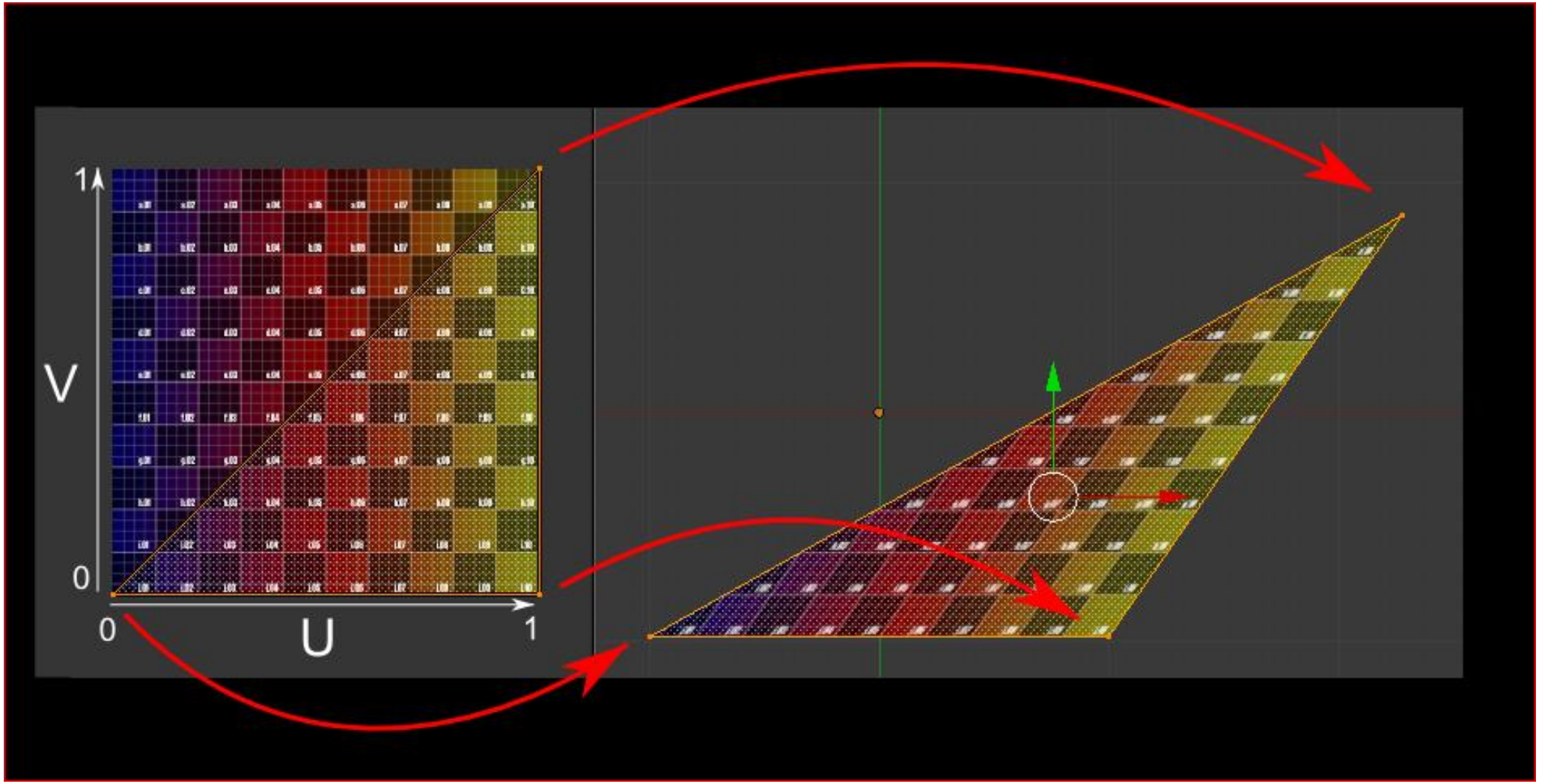
Mapeamento UV

A convenção usual é usar U e V como o eixo do espaço de textura em que U corresponde a X no sistema de coordenadas cartesianas 2D e V corresponde a Y.

A OpenGL/WebGL trata os valores dos eixos UV como indo da esquerda para a direita na U eixo e para baixo no eixo V.



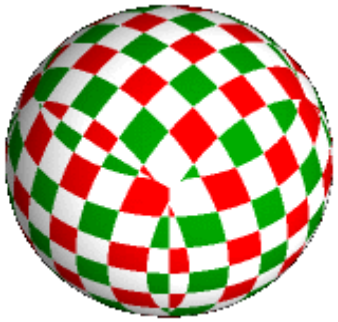
Mapeamento UV



Exemplos

Parametrização cúbica

Projetada em uma esfera



Projetada em um cilindro

Fonte: Slides do prof. Paulo Roma - UFRJ

Exemplos

Parametrização cilíndrica

Projetada em uma esfera

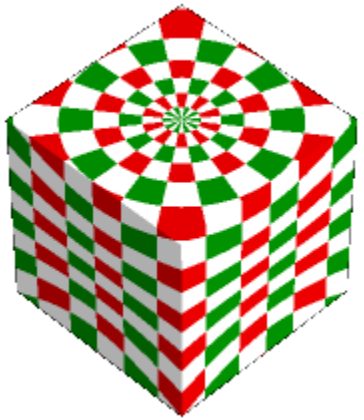
Projetada em um cubo

Fonte: Slides do prof. Paulo Roma - UFRJ

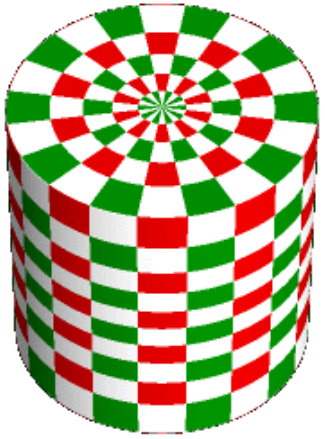
Exemplos

Parametrização esférica

Projetada em um cubo



Projetada em um cilindro



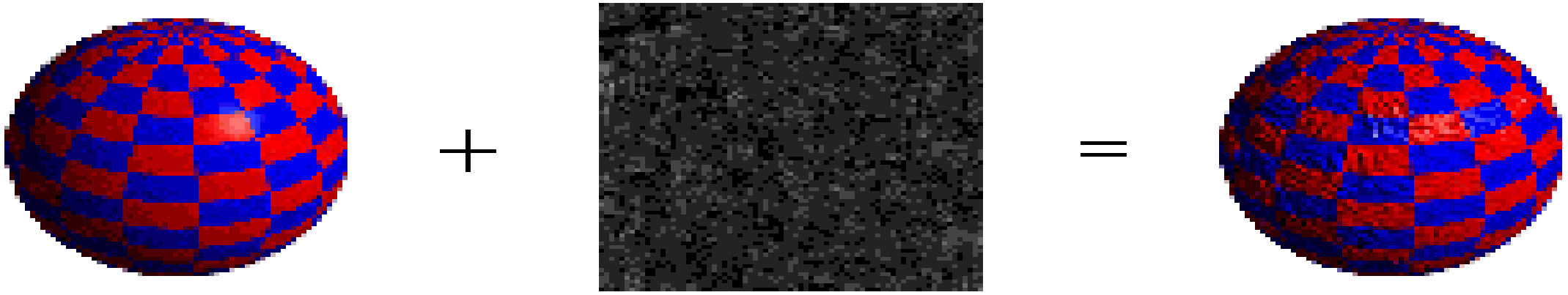
Fonte: Slides do prof. Paulo Roma - UFRJ

Bump mapping

Mapeamento básico de textura numa superfície suave.

- Nesta técnica a imagem mapeada é utilizada para fazer uma perturbação do vetor normal à superfície antes de calcular a iluminação, resultando em um efeito visual de superfície rugosa.

A superfície não muda realmente, sombreamento faz parecer mudada.

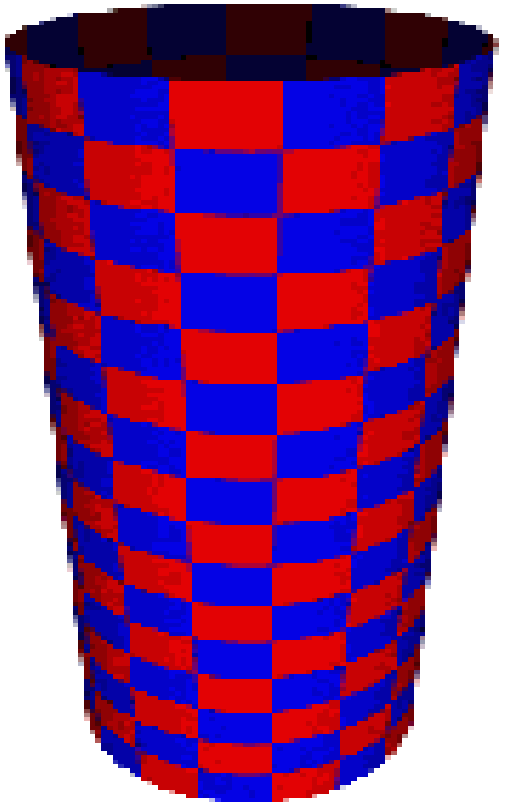


Esfera com mapa de texturas difuso

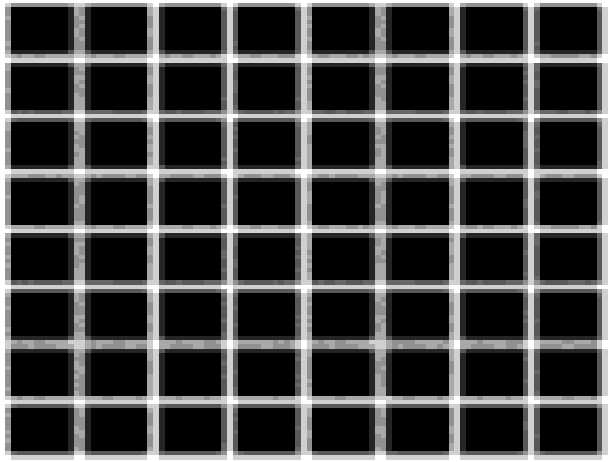
Bump map

Esfera com mapa de texturas difuso + bump map

Exemplo de “Bump mapping”

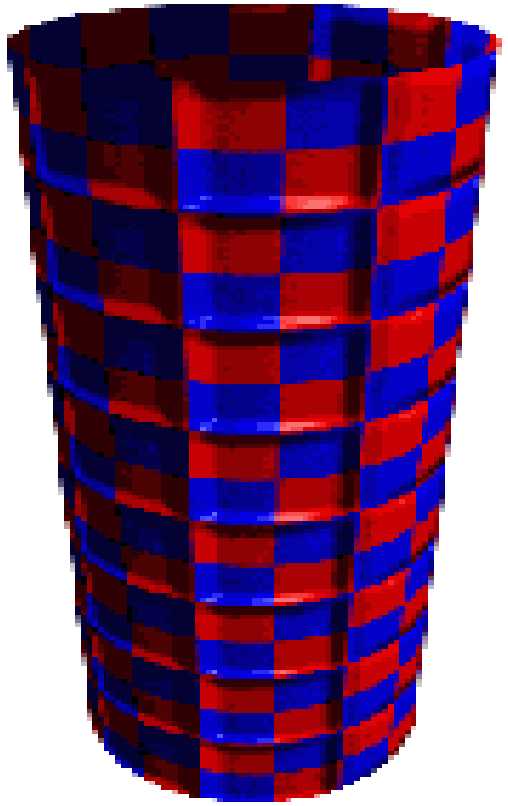


+



Bump Map

=



Cilindro c/ mapa de texturas difuso

Cilindro c/ mapa de texturas difuso + bump map

Mapa de deslocamentos (displacement mapping)

- Uma desvantagem do mapeamento da rugosidade é o que ao observarmos a silhueta da superfície não vemos os detalhes da geometria que foram mapeados.
- Uma solução consiste em deslocar realmente a superfície
- Uso do mapa de texturas para deslocar cada ponto na superfície
  - valor de textura diz quanto mover na direção normal à superfície

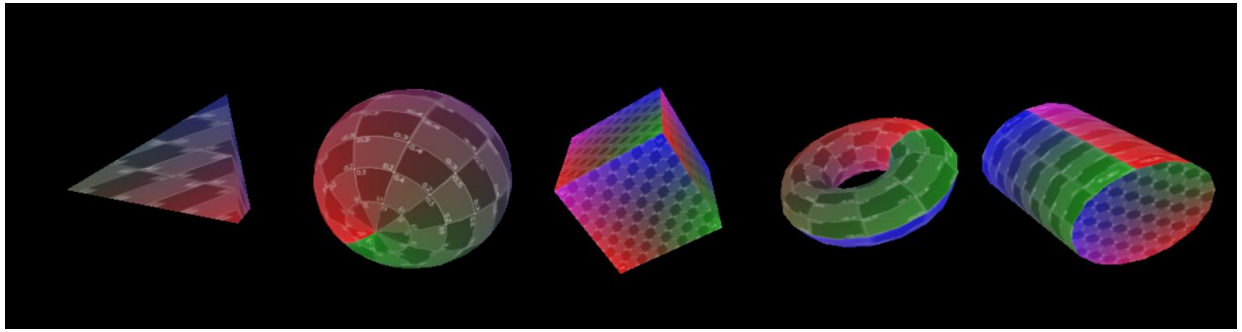
Praticando no Three.js

Exercício 1:

A) Utilizando o arquivo disponível para a "Aula06_Ex1", crie outras geometrias, e verifique o comportamento do mapeamento UV em cada uma:

Geometrias:

- Cubo (CubeGeometry)
- Esfera (SphereGeometry)
- Anel (TorusGeometry)
- Tetraedro (TetrahedronGeometry)
- Cilindro (CylinderGeometry)



Confira algumas propriedades das textures em:
Demo: Texture Parameters
Demo: Texture Rotation

Leia mais e consulte em:
https://threejs.org/manual/#en/materials

B) Modifique as texturas pela textura "uv.png" e verifique o resultado.

C) Modifique as texturas pela textura "earth.jpg" e verifique o resultado.

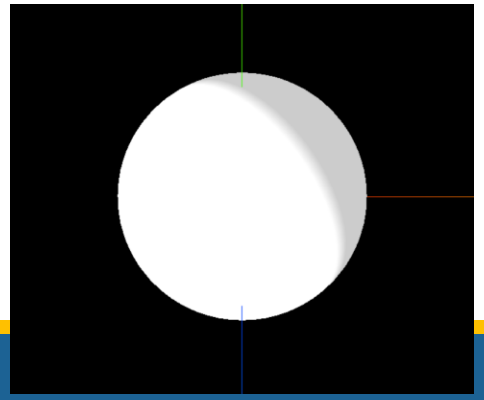
Praticando no Three.js

Exercício 2:

Utilizando os códigos elaborados em sala, , siga o roteiro proposto e modifique o arquivo 'Aula06_Ex2', para chegar no resultado da imagem ao lado.

Parte 1:

1. Modifique o "material1" para o tipo Phong.
2. Crie uma iluminação do tipo ambiente com intensidade de 0.8 e cor branca. (Não esqueça do "scene.add").
3. Crie uma fonte de luz do tipo direcional com intensidade 2 e coordenadas (-10,0,10) e target para a origem da cena.



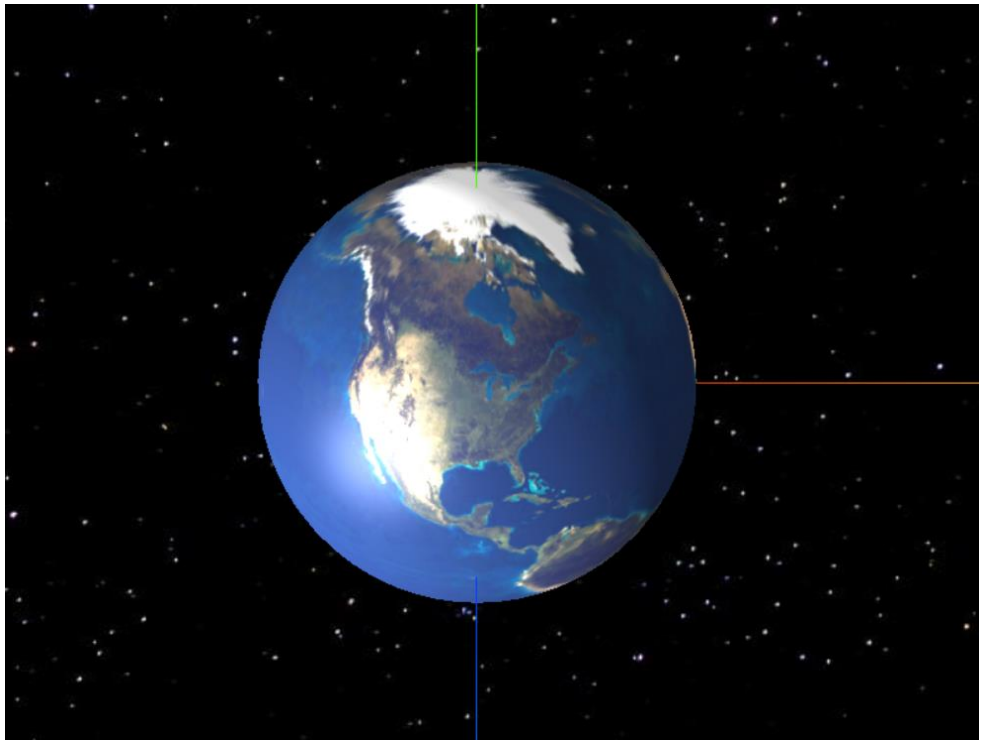
Praticando no Three.js

Parte 2:

1. Utilizando a classe "THREE.TextureLoader" crie um texture loader na variável "loader".
2. Agora na variável "colorMap" carregue a imagem "earth.jpg" da pasta "img" através do método "load()".
3. Na declaração do "material1" passe o parâmetro "map: colorMap" junto com a cor e veja o resultado.

4. Novamente, utilizando o texture loader, carregue a imagem "space.jpg" na variável "spacebg".
5. Atribua a variável "spacebg" ao atributo "background" do objeto "scene" (scene.background = spacebg).

6. Dentro da função "animate", atribua uma rotação no eixo y "sphere1", com incrementos de 0.01.



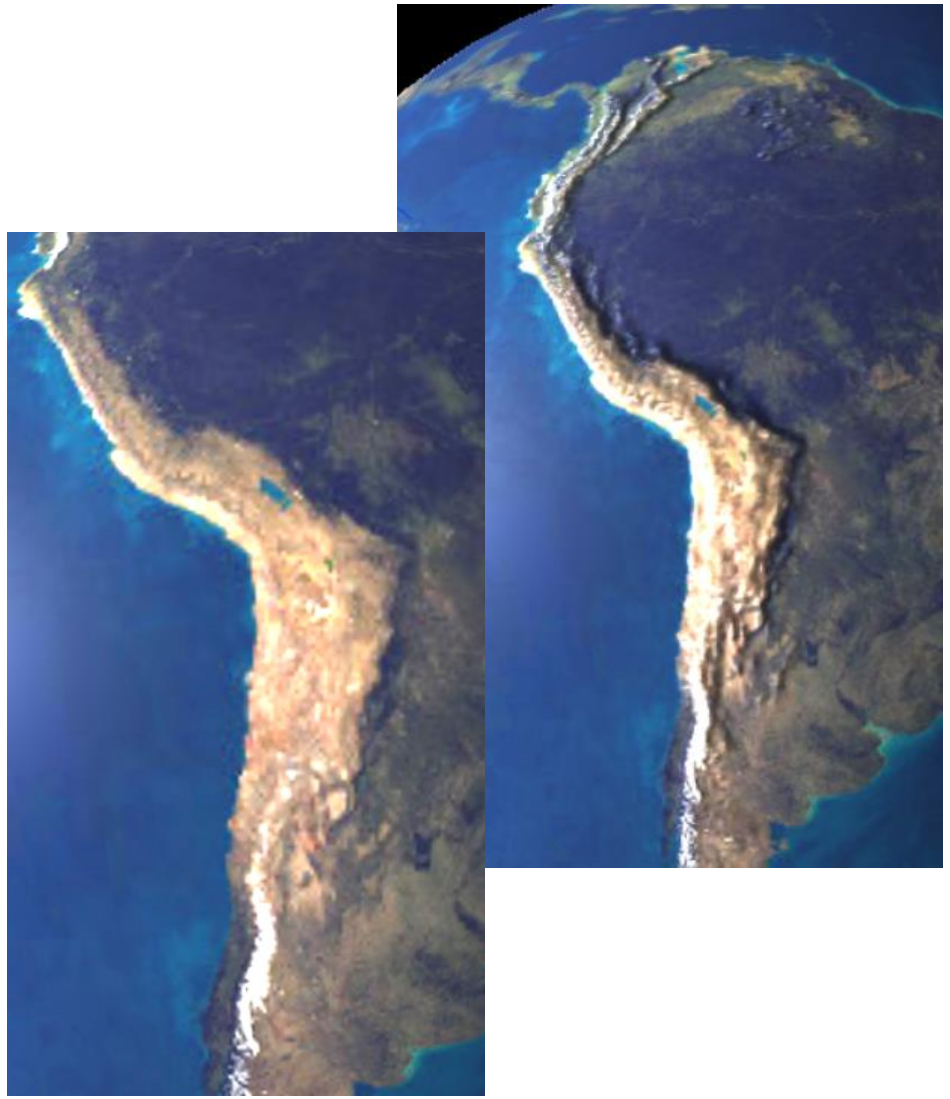
Praticando no Three.js

Parte 3:

1. Utilizando o "loader" carregue a imagem "normal.jpg" na variavel "normalMap" e passe-a na declaração do "material1" no parâmetro "normalMap".

2. Aproxime o zoom com os controle do mouse: É possível notar alguma diferença? (repare a região das américas e cordilheira dos andes)

3. Romova o "colorMap" para perceber a diferença.



Praticando no Three.js

Parte 4:

1. No "material1", adicione os seguinte parametros
specular: 0x333333, shininess: 15

2 Novamente, utilizando o "loader" carregue a imagem "specular.jpg" na variavel "specMap" e passe-a na declaração do "material1" no parâmetro "specularMap".

3 Aproxime o zoom com os controle do mouse: É possível notar alguma diferença? (repare no reflexo do mar e continentes).

4 Remova o "colorMap" para perceber a diferença.

5 Adicione todos os maps simultaneamente e remova o "AxisHelper" para gerar a cena final