

Aula03 – Textura e Interação

João Magalhães, 79923 e João Ferreira, 80041

Visualização de Informação, 2021

Mestrado em Engenharia Informática, Universidade de Aveiro

Utilização de uma textura num plano

Neste exemplo começamos por utilizar o código do primeiro exemplo da aula01, alterando a geometria de “BoxGeometry” para “PlaneGeometry”. Assim a forma apresentada é uma figura plana em vez do cubo representado anteriormente.

```
var geometry = new THREE.PlaneGeometry(5, 10, 12);
```

Posteriormente para darmos a textura pretendida ao plano realizamos o seguinte código.

```
var texloader = new THREE.TextureLoader();  
var tex=texloader.load("imagens/lena.jpg");
```

Sendo que no final deste exemplo resulta o seguinte output.



Textura num cubo

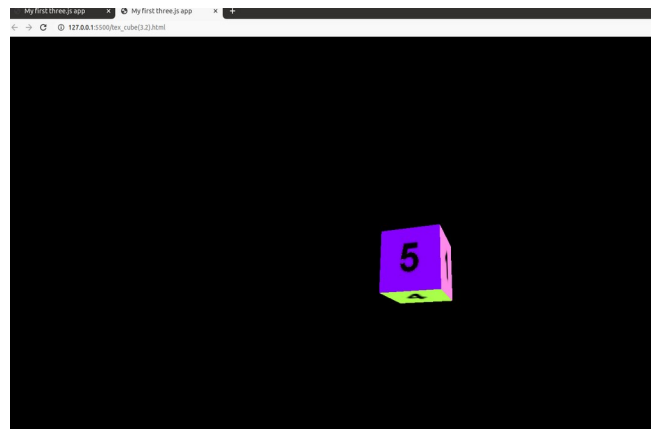
Para realizar este exercício começamos por regressar ao primeiro exemplo da aula01, só que desta vez o objetivo era por seis imagens, uma em cada face do cubo, para termos os números de um a seis, nas seis faces do cubo. Para isso tivemos que criar um agregado de materiais onde são colocadas todas as texturas que pretendemos.

```
var materials = [  
  new THREE.MeshBasicMaterial({map: loader.load('imagens/Im1.jpg')}),  
  new THREE.MeshBasicMaterial({map: loader.load('imagens/Im2.jpg')}),  
  new THREE.MeshBasicMaterial({map: loader.load('imagens/Im3.jpg')}),  
  new THREE.MeshBasicMaterial({map: loader.load('imagens/Im4.jpg')}),  
  new THREE.MeshBasicMaterial({map: loader.load('imagens/Im5.jpg')}),  
  new THREE.MeshBasicMaterial({map: loader.load('imagens/Im6.jpg')}),  
];
```

De seguida, como dito no enunciado tivemos que alterar o comando de criação da malha para usar texturas múltiplas associadas a cada face e usar o comando push.

```
var cube = new THREE.Mesh( geometry, materials );  
  
scene.add(cube);  
cubes.push(cube);
```

Obtemos assim o seguinte output:



Textura e iluminação

Neste exercício, começamos por criar uma esfera com raio 1 e com 32 segmentos de altura e largura. A textura usada foi uma imagem que já nos era fornecida.

```
var geometry = new THREE.SphereGeometry(1,32,32);

var texloader = new THREE.TextureLoader();
var tex=texloader.load("imagens/earth_surface_2048.jpg");
```

De seguida, mudámos o material de MeshBasicMaterial para MeshPhongMaterial de modo a conseguirmos obter a iluminação pedida no enunciado. Depois desta mudança já foi possível a criação da iluminação.

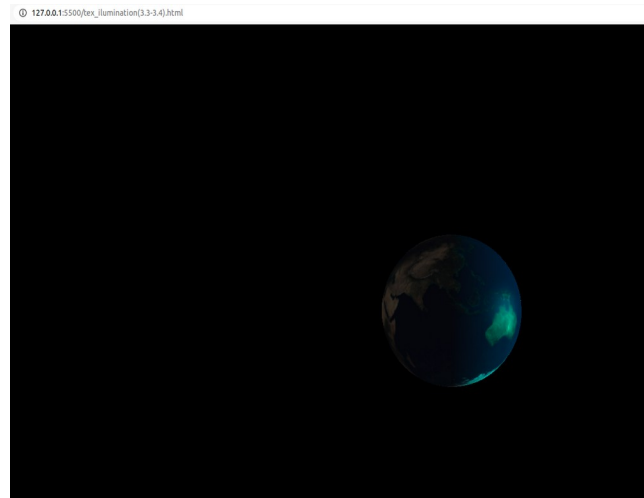
```
var light = new THREE.DirectionalLight( 0xfffff );
light.position.set(1,0,0);
scene.add(light);

var Amblight = new THREE.AmbientLight(0x333333);
scene.add(Amblight)

var material = new THREE.MeshPhongMaterial( {
  map: tex //ambient: 0x333333, color: 0x333333, specular: 0x555555, shininess: 100
});

var sphere = new THREE.Mesh( geometry, material );
```

Ficando assim o nosso output:



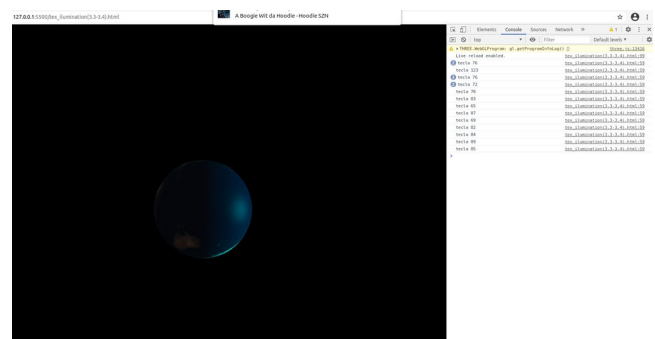
Interação

Neste exemplo, criamos um evento para quando o utilizador carregar nalguma tecla, esta ser imprimida na consola, isto é, quando o utilizador abre a página, tem que ir a consola usando a tecla F12 e de seguida aparece lá o key code da tecla que este primir.

```
document.addEventListener("keydown", onDocumentKeyDown, false);
```

```
function onDocumentKeyDown(event){
  // Get the key code of the pressed key
  var keyCode = event.which;
  console.log("tecla " + keyCode);
}
```

Dando o seguinte output:



Ativação iluminação

Este exercício consiste em modificar o código de modo a que quando o utilizador premir numa determinada tecla, neste caso o "L", a luz desliga/liga quando essa tecla é primida.

Bem como o aumento ou a diminuição da intensidade da luz premindo o "+" para aumentar e o "-" para diminuir.

```
function onDocumentKeyDown(event){
  // Get the key code of the pressed key
  var keyCode = event.which;
  console.log("tecla " + keyCode);
  if (keyCode == 76 && light.visible == false && Amblight.visible == false){ //keyCode 76 equivale ao L
    light.visible = true;
    Amblight.visible = true;
  }
  else if (keyCode == 76 && light.visible == true && Amblight.visible == true){
    light.visible = false;
    Amblight.visible = false;
  }
  else if (keyCode == 187){ //keyCode 187 equivale a "+"
    light.intensity += 0.1;
    Amblight.intensity += 0.1;
  }
  else if (keyCode == 189){ //keyCode 189 equivale a "-"
    light.intensity -= 0.1;
    Amblight.intensity -= 0.1;
  }
}
```

Alterar rotações e posições

Este exercício, é parecido com o anterior da iluminação mas é para as rotações e as posições que pretendemos mudar. As teclas de direção são utilizadas para aumentar/diminuir a velocidade de rotação à volta do eixo YY [direita e esquerda], a do eixo XX são as teclas de direção também mas desta vez são utilizadas as setas para cima e para baixo. Para inclinar mais ou menos o modelo à volta do eixo ZZ são utilizadas as teclas page Up/Down.

```
else if (keyCode == 38){ //keyCode 38 equivale "seta para cima"
    sphere.rotation.z += 0.1;
}

else if (keyCode == 39){ //keyCode 39 equivale "pg up"
    sphere.rotation.z += 0.1;
}

else if (keyCode == 40){ //keyCode 38 equivale "pg downs"
    sphere.rotation.z -= 0.1;
}

else if (keyCode == 37){ //keyCode 38 equivale "seta para cima"
    sphere.rotation.x += 0.1;
}

else if (keyCode == 38){ //keyCode 38 equivale "seta para baixo"
    sphere.rotation.x -= 0.1;
}

else if (keyCode == 33){ //keyCode 38 equivale "seta para baixo"
    sphere.rotation.y += 0.1;
}

else if (keyCode == 34){ //keyCode 38 equivale "seta para esquerda"
    sphere.rotation.y -= 0.1;
}
```

```
function rotacionarTerra()
{
    var matrizRotacaoY = new THREE.Matrix4();
    matrizRotacaoY.makeRotationY(calcularRadianos(1));
    sphere.applyMatrix(matrizRotacaoY);

    var matrizRotacaoEixo = new THREE.Matrix4(),
        matrizTranslacaoPosicao = new THREE.Matrix4(),
        matrizTranslacaoOrigem = new THREE.Matrix4(),
        posicao = sphere.position;

    matrizRotacaoEixo.makeRotationY(calcularRadianos(1.5));
    matrizTranslacaoPosicao.makeTranslation(posicao.x, posicao.y, posicao.z);
    matrizTranslacaoOrigem.makeTranslation(-posicao.x, -posicao.y, -posicao.z);

    sphere.applyMatrix(matrizTranslacaoOrigem);
    sphere.applyMatrix(matrizRotacaoEixo);
    sphere.applyMatrix(matrizTranslacaoPosicao);
}
```

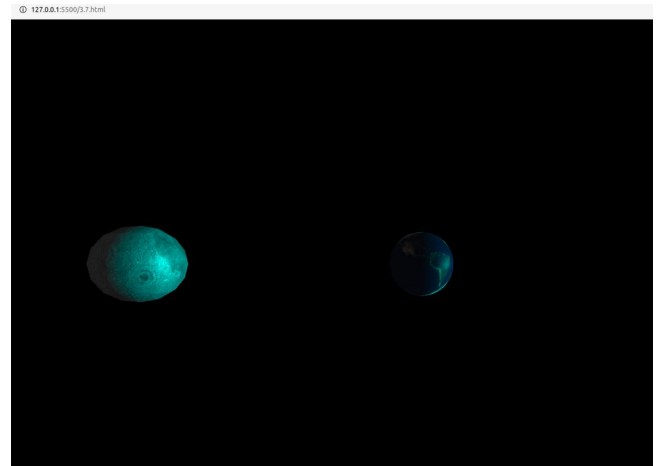
```
function rotacionarLua()
{
    var matrizRotacaoY = new THREE.Matrix4();
    matrizRotacaoY.makeRotationY(calcularRadianos(2));
    moon.applyMatrix(matrizRotacaoY);

    var matrizRotacaoEixo = new THREE.Matrix4(),
        matrizTranslacaoPosicao = new THREE.Matrix4(),
        matrizTranslacaoOrigem = new THREE.Matrix4(),
        posicao = moon.position;

    matrizRotacaoEixo.makeRotationY(calcularRadianos(0.6));
    matrizTranslacaoPosicao.makeTranslation(posicao.x, posicao.y, posicao.z);
    matrizTranslacaoOrigem.makeTranslation(-posicao.x, -posicao.y, -posicao.z);

    moon.applyMatrix(matrizTranslacaoOrigem);
    moon.applyMatrix(matrizRotacaoEixo);
    moon.applyMatrix(matrizTranslacaoPosicao);
}
```

Dando assim o seguinte output:



Concatenação de transformações / adição da lua

Neste exemplo, temos de criar outro modelo que irá representar a lua.

```
var geometry2 = new THREE.SphereGeometry(1,8,8);
var texloader2 = new THREE.TextureLoader();
var tex2=texloader2.load("imagens/moon_1024.jpg");
```

E posteriormente, para permitir que a lua rode à volta da terra, temos de criar a lua como um filho da terra para ser influenciada pelas transformações da terra, para isso temos que ir multiplicando as matrizes dos dois objetos.

Conclusão

Esta aula proporcionou-nos a capacidade de experimentarmos novos tipos de texturas em geometrias diferentes, bem como aumentar o nosso conhecimento sobre a iluminação que já tinha sido tratada na última aula.

Também aprendemos como ter alguma interação quando realizamos alguma ação com o modelo que foi criado, neste caso era quando uma tecla era pressionada.

Nessa sequência, aprendemos a ligar/desligar ou aumentar/diminuir a intensidade da luz ao clicar numa tecla, bem como alterar a sua rotação ou velocidade de rotação.

Por último, lidamos com questões de termos dois modelos e conseguirmos que um deles realizasse a rotação à volta do outro.

Em suma, consideramos que esta aula foi útil, para termos uma melhor compreensão de como mudar a textura da geometria utilizada, melhor o conhecimento sobre a iluminação e como tornar o programa interativo para o utilizador.