

Tutorial 7 – Importação de objetos

O *ThreeJS* permite a importação de objetos tais como texturas, objetos 3D e animações de forma a potenciar o desenvolvimento de aplicações gráficas para a *web* tirando partido de formatos de conteúdos populares tais como *FBX*, *OBJ* ou *COLLADA*.

1. Objetivos de aprendizagem

Neste tutorial, irás aprender a importar objetos 3D para a tua cena em *ThreeJS*. Para além disso, irás aprender a criar e configurar luzes na cena e a criar hierarquias entre objetos..

2. Tutorial

Para a importação de objetos, será necessário recorrer a duas bibliotecas externas, a biblioteca *FBXLoader* que permite importar objetos 3D no formato *FBX* e a biblioteca *zlib* que permite descodificação de dados. Segue os passos seguintes de forma a preparar os ficheiros necessários para a realização do tutorial, incluindo a importação das duas bibliotecas externas referidas.

2.1. Ficheiros necessários

1. Copia a pasta do tutorial anterior e muda o nome para “*Tutorial 7*” e remove todos os comentários do código.
2. Dentro da pasta “*Tutorial 7*”, cria uma nova pasta com o nome “*Objetos*”.
3. Accede ao Virtual e descarrega os ficheiros localizados na pasta “Ficheiros Adicionais” dentro da pasta “Aulas Práticas”.
4. Guarda os ficheiros descarregados dentro da pasta JavaScript
5. Por fim, descarrega o objeto 3D a importar. Accede ao link seguinte: <https://github.com/mrdoob/three.js/blob/master/examples/models/fbx/Samba%20Dancing.fbx?raw=true>. Ser-te-á apresentada uma caixa de diálogo que te irá perguntar onde pretendes descarregar o ficheiro. Selecciona a pasta “*Objetos*” localizada na pasta “*Tutorial 8*”.

2.2. Importação e configuração de objetos

1. Abre o *VSCode* e, através do programa, abre a pasta “*Tutorial 8*”.
2. abre o ficheiro “*index.html*” e adiciona as as linhas assinaladas a vermelho, tal como mostra a imagem abaixo.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>
5       WebGL - Three.js
6     </title>
7   </head>
8   <body>
9     <!-- Biblioteca necessária para FBXLoader -->
10    <script src="../JavaScript/inflate.min.js"></script>
11    <script src="../JavaScript/three.js"></script>
12    <!-- Biblioteca necessária para importação de objetos FBX -->
13    <script src="../JavaScript/FBXLoader.js"></script>
14    <script src="../JavaScript/app.js"></script>
15  </body>
16 </html>
```

3. Agora que já as bibliotecas necessárias para a importação de objetos no ficheiro *HTML* foram importadas, é necessário modificar o ficheiro “*app.js*” para importar o objeto pretendido. Assim, abre o ficheiro “*app.js*” e adiciona as seguintes variáveis:

```
18
19 // Variável que guardará o objeto importado
20 var objetoImportado;
21
22 // Variável que irá guardar o controlador de animações do objeto importado
23 var mixerAnimacao;
24
25 // Variável que é responsável por controlar o tempo da aplicação
26 var relógio = new THREE.Clock();
27
28 // Variável com o objeto responsável por importar ficheiros FBX
29 var importer = new THREE.FBXLoader();
30
```

4. Abaixo das variáveis, adiciona o código abaixo. Este código irá ser executado assim que a página comece a ser carregada e tratará de importar o objeto passado por parâmetro.

```
30
31 // Função utilizada para importar objetos FBX. O primeiro parâmetro é a localização do .fbx
32 // e o segundo parâmetro é uma função que apenas é chamada se conseguir importar o objeto.
33 importer.load('./Objetos/Samba Dancing.fbx', function (object) {
34
35     // O mixerAnimação é inicializado tendo em conta o objeto importado
36     mixerAnimacao = new THREE.AnimationMixer(object);
37
38     // object.animations é um array com todas as animações que o objeto trás quando é importado.
39     // O que nós fazemos, é criar uma ação de animação tendo em conta a animação que é pretendida.
40     // De seguida é inicializada a reprodução da animação.
41     var action = mixerAnimacao.clipAction(object.animations[0]);
42     action.play();
43
44     // object.traverse é uma função que percorre todos os filhos desse mesmo objeto.
45     // O primeiro e único parâmetro da função é uma nova função que deve ser chamada para cada
46     // filho. Neste caso, o que nós fazemos é ver se o filho tem uma mesh e, no caso de ter,
47     // é indicado a esse objeto que deve permitir projetar e receber sombras, respetivamente.
48     object.traverse(function (child) {
49
50         if (child.isMesh) {
51
52             child.castShadow = true;
53             child.receiveShadow = true;
54
55         }
56
57     });
58
59     // Adiciona o objeto importado à cena
60     cena.add(object);
61
62     // Quando o objeto é importado, este tem uma escala de 1 nos três eixos(XYZ). Uma vez que
63     // este é demasiado grande, mudamos a escala deste objeto para ter 0.01 em todos os eixos.
64     object.scale.x = 0.01;
65     object.scale.z = 0.01;
66     object.scale.y = 0.01;
67
68     // Mudamos a posição do objeto importado para que este não fique na mesma posição que o cubo.
69     object.position.x = 3;
70
71     // Guardamos o objeto importado na variável objetoImportado.
72     objetoImportado = object;
73
74 });
```

5. Se executares o teu código agora, apenas verás o cubo. Isto acontece porque não existem luzes na cena e porque o material do cubo que criaste no tutorial anterior não é influenciado pela luz presente na cena. Vamos então mudar o material do cubo para que este seja afetado pela luz. Para isso altera a variável “material”, tal como se segue:

```
8
9 var geometria = new THREE.BoxGeometry(1, 1, 1);
10 // Mudamos o material do cubo para este receba luz por parte de uma fonte de luz.
11 var material = new THREE.MeshStandardMaterial({ color: 0xff0000});
12 var cubo = new THREE.Mesh(geometria, material);
13
```

6. De seguida, vamos proceder à criação de uma luz na cena. Para tal, atualiza a função “*Start()*” da seguinte forma:

```
127
128 function Start() {
129
130     cena.add(cubo);
131
132     // Criação de um foco de luz com a cor branca (#ffffff) e intensidade a 1 (intensidade normal)
133     var light = new THREE.SpotLight('#ffffff', 1);
134
135     // Mudar a posição da luz para ficar 5 unidades a cima de onde a câmara se encontra.
136     light.position.y = 5;
137     light.position.z = 10;
138
139     // Dizemos a light para ficar a apontar para a posição do cubo.
140     light.lookAt(cubo.position);
141
142     // Adicionamos a light à cena.
143     cena.add(light);
144
145     camara.position.z = 10;
146
147     requestAnimationFrame(update);
148 }
149
```

7. Se agora executares o código, verás que o cubo já é afetado pela luz assim como podes verificar que o boneco importado nos passos anteriores já aparece. O objeto que importarmos tem uma animação, a qual nós definimos para ser reproduzida (através do código *action.play()*;) mas que não está a ser reproduzida. Isto acontece porque não estamos a atualizar o tempo de animação. Para fazê-lo, dirige-te à função “*update()*” e adiciona o seguinte código:

```
150 function update() {
151
152     if (cuboCoordRotation != null) {
153         cubo.rotation.x += cuboCoordRotation.y * 0.1;
154         cubo.rotation.y += cuboCoordRotation.x * 0.1;
155     }
156
157     if (camaraAndar != null) {
158         camara.position.x += camaraAndar.x;
159         camara.position.z += camaraAndar.z;
160     }
161
162     // Necessário atualizar o mixerAnimação tendo em conta o tempo desde o ultimo update.
163     // relogio.getDelta() indica quanto tempo passou desde o último frame renderizado.
164     if(mixerAnimacao) {
165         mixerAnimacao.update(relogio.getDelta());
166     }
167
168     camaraAndar = { x: 0, y: 0, z: 0 };
169
170     renderer.render(cena, camara);
171
172     requestAnimationFrame(update);
173 }
174
```

Agora, se executares o código, verás que o objeto importado estará a executar a animação (a dançar).

Nota: Por vezes, um objeto é constituído por vários subobjectos, mantendo uma relação hierárquica. Imaginemos que queremos que o objeto importado seja filho do cubo, isto é, quando o cubo sofrer uma translação, rotação ou escala o objeto importado também sofrerá essas mesmas alterações. Para isso basta mudares uma linha de código na função `importer.load()`:

```
// Adiciona o objeto importado como filho do cubo
cena.add(object);
```

para

```
// Adiciona o objeto importado à cena
cubo.add(object);
```

Se voltares a correr o código verás que o objeto importado rodará juntamente com o cubo.

Desafios

- Desafio 1.** Neste momento, ao utilizares as teclas A, S, D ou W, a câmara move-se. Faz com que, ao utilizar essas teclas, seja o boneco importado a se deslocar pela cena ao invés da câmara.
- Desafio 2.** Atualiza as definições da luz para a cor azul.
- Desafio 3.** Configura as definições da luz, para que aponte para o objeto importado

ENTREGA

O trabalho deve ser submetido no MOODLE até dia **14/05/2021**.

A submissão é individual e deve conter todos os ficheiros necessários à correta execução da aplicação: uma pasta contendo o ficheiro .html e a pasta JavaScript com os respetivos ficheiros.