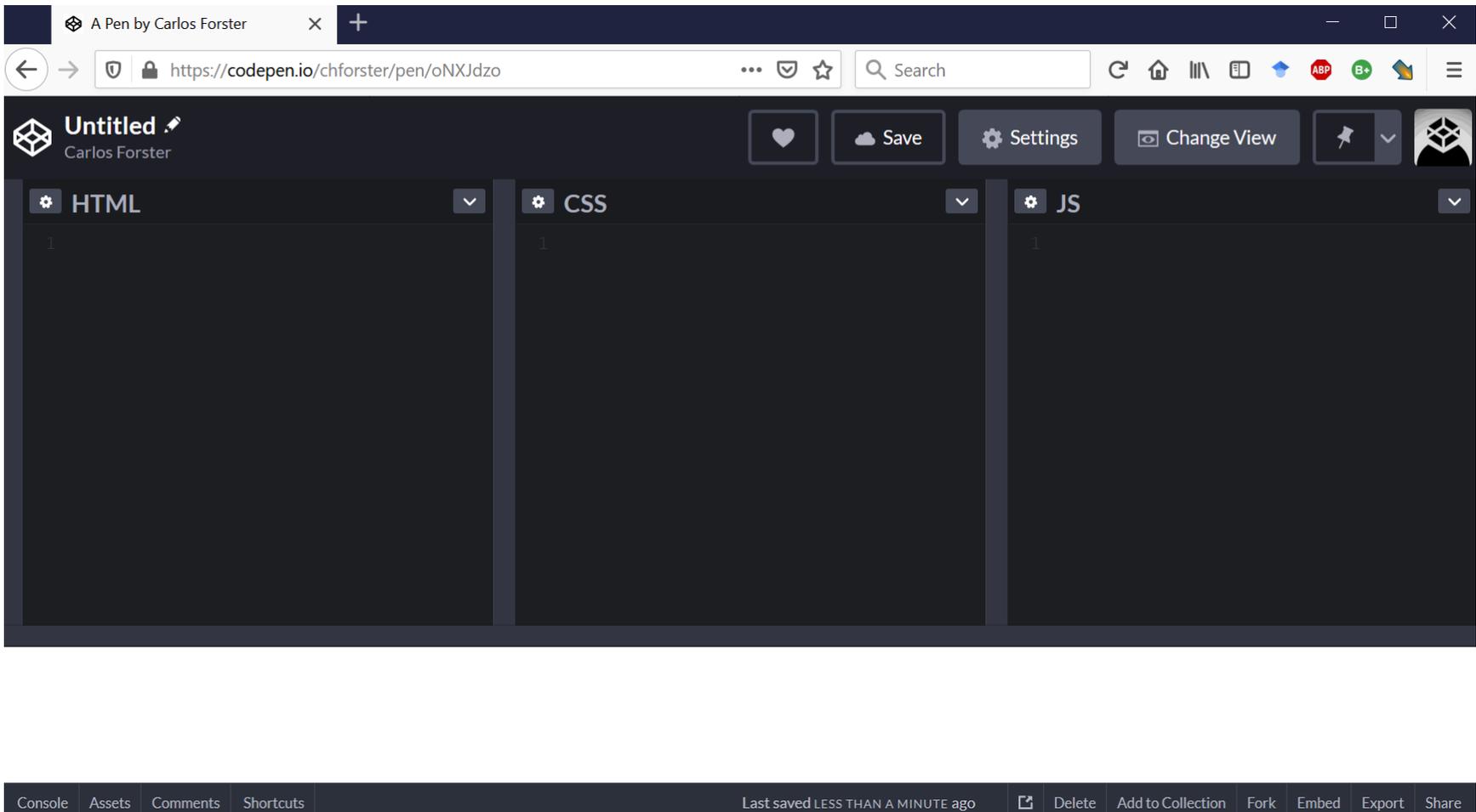


Introdução à atividade com o THREE.JS

Prof. Forster

Minha sugestão para vocês é utilizar inicialmente o Codepen:

<https://codepen.io/chforster/pen/rNVokNY>



O que vocês fizerem na versão gratuita do Codepen fica público, então acaba que temos muito exemplos feitos com THREE.JS pela "comunidade".

The image shows a browser window with the CodePen search results for 'three.js'. The browser's address bar shows the URL 'https://codepen.io/search/pens?q=three.js'. The CodePen logo is in the top left corner. Below the logo, there are navigation options: 'CREATE', 'Pen', 'Project', and 'Collection'. The 'Pen' option is selected. The search bar contains 'three.js'. Below the search bar, there are filters for 'SORT BY' (Relevance & Popularity), 'SEARCH DEPTH' (Everything), and 'FORKS' (Include forks). The search results are displayed in a grid. The first result is 'three.js canvas - particles - waves' by deathfang, featuring a blue background with white particles. The second result is 'CSS-Only Cube Pack' by David Khourshid, featuring a 3D cube structure. The browser's taskbar at the bottom shows several icons, including the CodePen logo and a 'PRO' badge.

Esse seria o HTML inicial feito fora do Codepen. Veja que ele acrescenta automaticamente esses detalhes.

```
HTML
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset=utf-8>
5 <title>Aprendendo THREE.JS</title>
6
7 </head>
8 <body>
9 </body>
10 </html>
```

Nesse HTML temos:

- O "doctype" que ajuda em algumas questões de compatibilidade entre programas. Fica registrado o tipo da formatação do dado no conteúdo.
- A estrutura básica: html / head / body
- O "charset" para podermos utilizar caracteres internacionais.
- O título que vai aparecer na janela do browser (navegador).

Seria bom ter também:

- Os estilos das tag, `<link>` para um CSS (automático no codepen) ou tag `<style>`
- Tags `<script>` para invocar as bibliotecas JavaScript e para colocar nosso código.

Como exemplo, fiz esse código JS dentro do HTML. O codepen já utiliza a janela própria do JS para isso.



```
HTML
```

```
1 <script>
2   // Escreva o programa JS aqui
3   // No codepen não precisa
4   console.log("Combinado!!");
5
6 </script>
```

```
Console
```

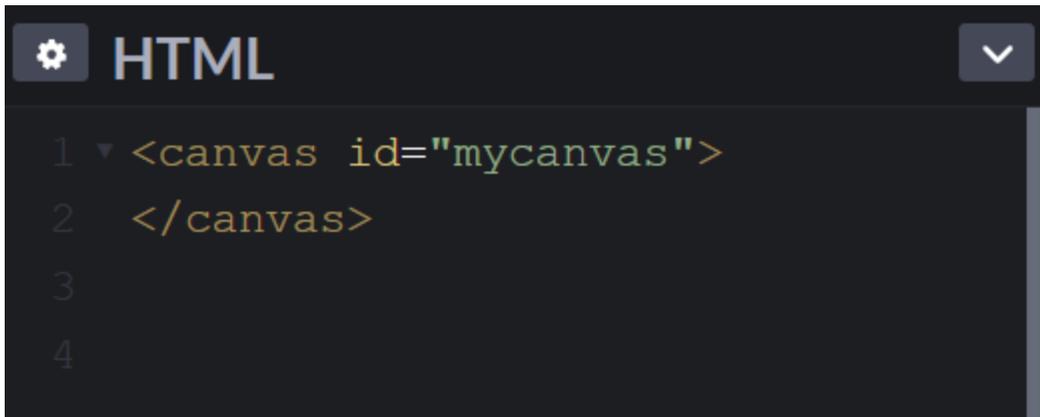
```
"Combinado!!"
```

No fim das contas é esse o HTML que eu vou usar, sabendo que o Codepen vai completá-lo com os outros detalhes.

No caso, estou criando a canvas no HTML. Pode ser criado alternativamente no código JS.

Essa canvas é obtido no código JS através de

```
document.getElementById("mycanvas")
```



The image shows a dark-themed code editor window titled "HTML". The code is as follows:

```
1 <canvas id="mycanvas">
2 </canvas>
3
4
```

O meu CSS ficou bem simples. Tiram-se as margens do documento. Seta a cor de fundo para uma cor neutra (para sabermos se tem alguma coisa ali). O `overflow` indica que não quero que as barras de rolagem apareçam.

A canvas deva ocupar a janela por inteiro.

```

1  body {
2    margin: 0;
3    background: CornFlowerBlue;
4    overflow: hidden;
5  }
6  canvas {
7    width: 100%;
8    height: 100%;
9  }
```

Agora, no menu de configurações, pedimos para incluir o THREE.JS de forma que não precisamos fazê-lo manualmente.

The screenshot shows the 'Pen Settings' dialog box. On the left is a sidebar with menu items: HTML, CSS, JS (highlighted), Pen Details, Privacy PRO, Behavior, Editor, Template, and Screenshot PRO. The main area is titled 'JavaScript Preprocessor' and has a dropdown menu set to 'None'. Below this is the 'Add External Scripts/Pens' section, which includes a search bar with 'three' entered and a list of search results. The first result is 'three.js' with a score of 110 and the description 'JavaScript 3D library'. The second result is 'jquery.threedubmedia' with a score of 2.2. At the bottom of the dialog, there is a promotional banner for upgrading to PRO and a green 'Save & Close' button.

Search Results	Score
three.js JavaScript 3D library	110
jquery.threedubmedia	2.2

No caso que estejam fazendo fora do codepen, uma possibilidade é baixar o THREE.JS e copiar o arquivo `three.js` ou `three.min.js` para uma pasta `js` do projeto. Pode ser que precise de outros arquivos da biblioteca, por exemplo, a "câmera orbitante" ou algum carregador de arquivo formato ".obj", ".fbx", ".glTF" etc

No caso de acesso a arquivos locais, o Firefox não oferece problemas. Mas eu tenho usado localmente o "miniweb" (<http://miniweb.sourceforge.net/>) como servidor de páginas para Windows. Ou os arquivos podem ser hospedados em algum outro local...

Vamos ver o código do exemplo:

```
JS
1 // Our Javascript will go here.
2
3 var scene = new THREE.Scene();
4 var camera = new THREE.PerspectiveCamera(
5     75, window.innerWidth / window.innerHeight, 0.1, 1000 );
6 var renderer = new THREE.WebGLRenderer( {
7     canvas:document.getElementById("mycanvas"),
8     alpha:false,
9     antialias:true
10 });
11
12 renderer.setSize( window.innerWidth, window.innerHeight );
13
```

Para o THREE.JS precisamos definir:

- **scene** - que é o objeto raiz que vai ser pai de toda a geometria que será desenhada por sua página.
- **camera** - corresponde ao sistema de transformações de coordenadas de visualização
- **renderer** - indica qual a plataforma e qual a superfície para renderizar a cena

A cena inicialmente é apenas o objeto raiz e vamos completá-la depois.

A câmera é uma câmera de perspectiva com alguns parâmetros definidos. No caso, o campo de visão é "70". A razão de aspecto é calculada baseada nas dimensões da janela.

O renderer é um que usa a WebGL e foram definidas algumas opções:

O canvas a ser utilizado foi aquele criado no HTML.

O `alpha` e o `antialias` vocês devem modificar para ver o que acontece.

Nesta parte definimos a cena (e damos uma reposicionada na câmara)

```
JS 5 unsaved changes X
15
16 var geometry = new THREE.BoxGeometry( 1, 1, 1 );
17 var material = new THREE.MeshLambertMaterial( { color: 0x00ffff,
  wireframe: false } );
18 var cube = new THREE.Mesh( geometry, material );
19
20 var light = new THREE.DirectionalLight( 0xffffff, 1.0 );
21 light.position={ x: 3, y:2, z: 1};
22
23 scene.add( cube );
24 scene.add(light)
25 camera.position.z = 5;
26
```

É criado um objeto `BoxGeometry` que contém a definição dos vértices e faces de um cubo centrado na origem com as dimensões dadas pelos parâmetros. Pode ser redimensionado, transladado e rotacionado utilizando as variáveis próprias para isso (que o `THREE.JS` calcula a matriz para você). Pode ser também fornecida diretamente a matriz de transformação (mas o `THREE.JS` precisa ser avisado disso).

A rotação em relação à origem e eixos cartesianos será utilizada na animação.

É definido um material do tipo "Lambert", reflexão difusa, semelhante a um objeto de borracha. A cor é azul ciano (0x00FFFF vermelho=00, verde=FF, azul=FF). Para esse material funcionar, é necessária uma fonte de luz. A opção wireframe deve ser alterada para ver o que acontece.

A fonte de luz é definida mais abaixo. No caso, essa fonte criada é direcional e branca, parâmetros que podem ser mudados.

A combinação de uma geometria já criada e de um material define um objeto Mesh, que pode ser adicionado na cena.

Até agora não desenhamos nada!

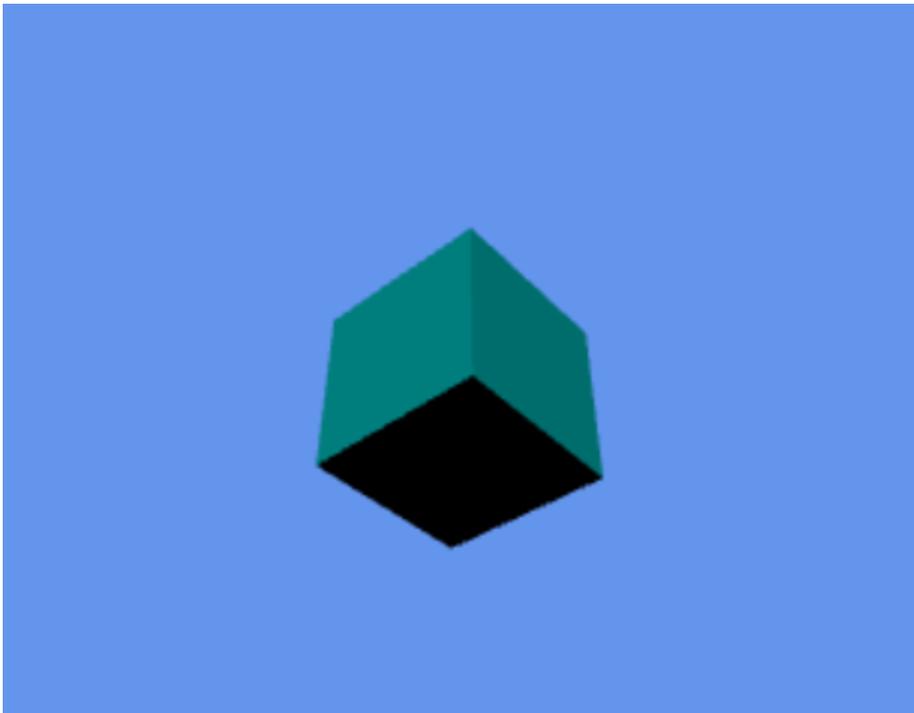
O método que vai desenhar é o `renderer.render`

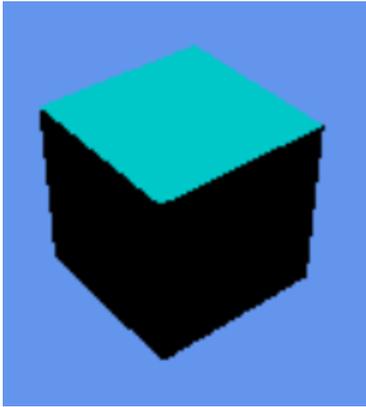
Vamos fazer isso na forma de animação.

```
JS
27 function animate() {
28     requestAnimationFrame( animate );
29     cube.rotation.x += 0.01;
30     cube.rotation.y += 0.01;
31     renderer.render( scene, camera );
32 }
33
34 animate();
35
36
```

Toda a vez que o navegador for precisar de um novo frame para renderizar, vai achar o `animate()` .

Isto porque na primeira vez que invocamos o `animate` , a função se registrou com `requestAnimationFrame` para que na próxima vez seja chamada para desenhar.





(exemplo sem antialias)

E é nessa função que podemos fazer alterações no grafo de cenas, modificando objetos, materiais, seus posicionamento e relações. Alterar posicionamento de câmera dinamicamente, entre outras diversas coisas.

Conseguindo fazer esse tutorial, já temos um bom pontapé inicial para a atividade. Agora é preciso explorar as opções que a ferramenta oferece e associar ao estudo da teoria.

Manual do THREE.JS (inglês) <https://threejs.org/docs/>

Mais uma introdução: <http://davidscottlyons.com/threejs-intro/>