

# Exemplos Criando um Cubo

## [Primeiros passos](#)

## [Criando um cubo](#)

## [Criando uma esfera](#)

## [Usando transformações](#)

## [Adicionando luz na cena](#)

## [Criando um componente](#)

## [Grafo de cena](#)

A primeira coisa a ser feita é **declarar** um elemento de **<canvas>** na sua página HTML. Ele será responsável por renderizar a sua aplicação.

```
<canvas id="myCanvas" width="640" height="640">
```

Depois disso no seu código javascript importe os seguintes arquivos:

```
import { Scene } from "../game/Scene";
import { Point3D } from "../geometric/Point3D";
import { PerspectiveCamera } from "../game/PerspectiveCamera";
import { Game } from "../game/Game";
import { Color } from "../geometric/Color";
import { CubeGameObject } from "../gameObject/CubeGameObject";
```

Após isso recupere o elemento canvas:

```
let canvas = document.getElementById("myCanvas");
```

E defina qual tipo de contexto você quer:

```
let context = canvas.getContext("webgl");
```

Com esses **arquivos importados comece** criando uma cena:

```
let scene = new Scene();
```

Depois crie uma câmera. Nesse tutorial optou-se por uma câmera em perspectiva porque ela se assemelha mais com a visão humana.

```
let camera = new PerspectiveCamera({near: 0.1, far : 500, aspect : 1, fovy
: 45 * Math.PI / 180, position : new Point3D(0, 0, 15)});
```

Com tudo isso pronto você pode instanciar um Game. Ele é a parte principal do motor de jogos e você terá apenas um por aplicação.

```
let game = new Game(context, scene, camera);
```

Tudo que foi visto até esse ponto é basicamente comum para qualquer aplicação que você irá fazer utilizando esse motor de **jogos daqui** para frente será mostrado a criação do cubo.

Para se criar um cubo você deve instanciar um objeto da classe CubeGameObject. No construtor dessa classe você pode ou não informar a **cor(color)** e a posição (**position**), caso não sejam informados será atribuído valores **padrão que para for será** preto e para a **positive** (0,0,0). No exemplo será criado um cubo vermelho na posição (0,0,0).

Criando a cor:

```
let red = new Color({r : 1});
```

Criando o cubo:

```
let cube = new CubeGameObject({color : red});
```

Para que você visualize o cubo basta por fim adicionar ele na cena:

```
scene.addGameObject(cube);
```

Agora com código pronto **pasta** indicar o caminho dele no arquivo **webpack.config.js** dessa forma:

```
entry :{  
  ...  
  cubes : "./app/nomeDoArquivo.js",  
  ...  
}
```

Com isso feito execute o comando:

```
npm run webpack
```

E por fim coloque na sua página HTML o arquivo no formato **nomeDoArquivo.bundle.js** localizado na pasta **/build**.



Aqui está resultado final:



