



WebGLRenderer - Les bases

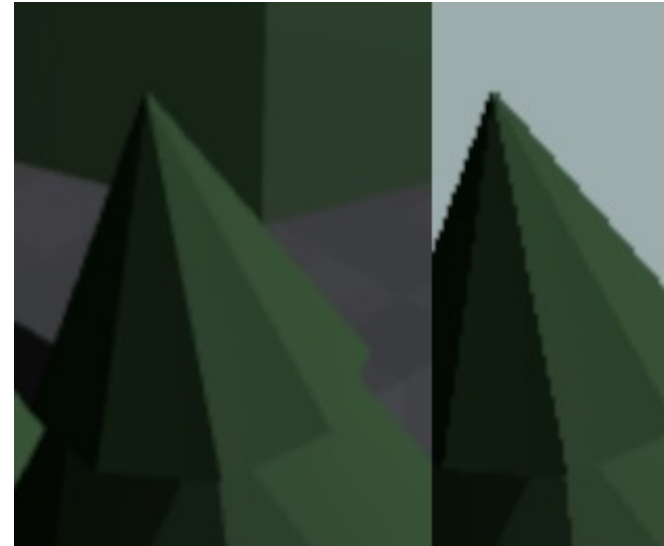
WebGLRenderer - antialias

La propriété **antialias** de **WebGLRenderer** est utilisée pour activer ou désactiver l'antirénelage (anti-aliasing) du rendu 3D

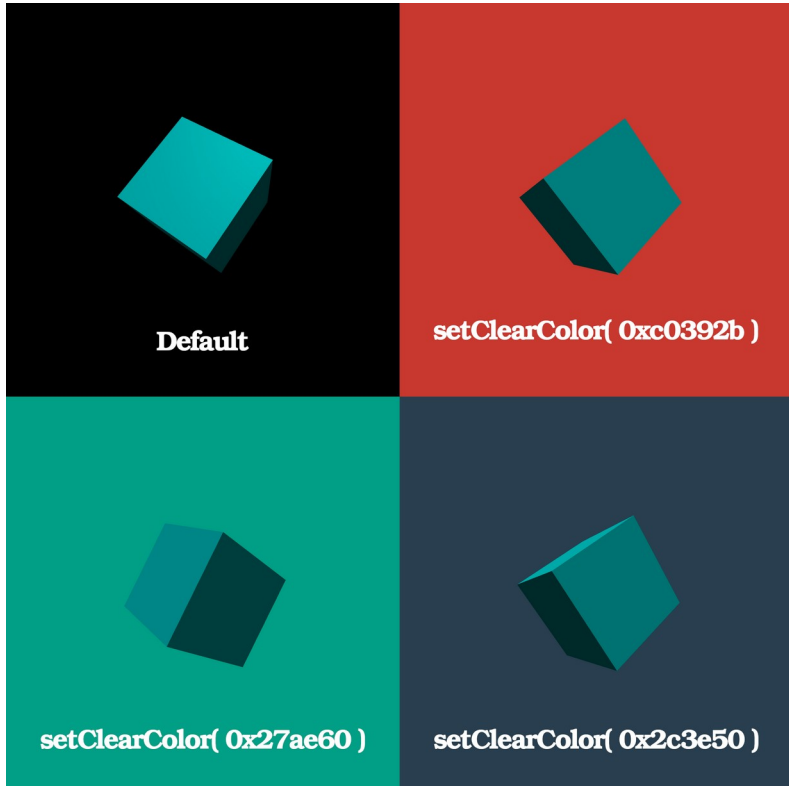
Cette propriété est utilisable dans le constructeur de **WebGLRenderer** (**Booléen**)

```
renderer = new THREE.WebGLRenderer( { antialias : true } );  
renderer.setPixelRatio( window.devicePixelRatio );  
renderer.setSize( window.innerWidth, window.innerHeight );  
renderer.setClearColor( 0x2c3e50, 0.8 );  
document.body.appendChild( renderer.domElement );
```

WebGLRenderer - antialias



WebGLRenderer - setClearColor



La méthode **setClearColor** de **WebGLRenderer** est utilisée pour définir la couleur de fond et l'opacité du fond du rendu 3D

Cette méthode accepte deux paramètres :

1. Couleur de fond du rendu 3D (**Hexadécimal**)
2. (**Facultatif**) - Opacité de la couleur de fond (Valeur **Float** entre **0** et **1**)

```
renderer = new THREE.WebGLRenderer( { antialias : true } );  
renderer.setPixelRatio( window.devicePixelRatio );  
renderer.setSize( window.innerWidth, window.innerHeight );  
renderer.setClearColor( 0x2c3e50, 0.8 );  
document.body.appendChild( renderer.domElement );
```

WebGLRenderer - alpha

Il est possible d'utiliser la propriété **alpha** de **WebGLRenderer** (**Booléen**) pour préciser l'existence d'un canal **alpha** dans le **canvas**

Il est ainsi possible de rendre la couleur de fond transparente

```
renderer = new THREE.WebGLRenderer( { canvas : DOMcanvas, antialias : true, alpha : true } );
```

HTML ThreeJS Div

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi imperdiet tempus libero non eleifend. Sed ultricies rutrum felis, a dapibus ipsum porta a. Aenean eu tellus turpis. Mauris sollicitudin ornare felis, eget hendrerit lacus ultricies ut. Vestibulum vehicula ante augue, id pharetra lectus sagittis sit amet. Donec augue enim, auctor nec hendrerit trincidunt, dapibus dapibus lorem. Proin eget tellus nisi. Curabitur quis nisi malesuada, laculis arcu in, trincidunt est. Praesent in risus ligula. Mauris vulputate ut urna id sollicitudin. Morbi eget nisl nulla.

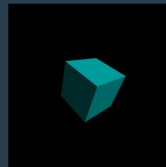


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi imperdiet tempus libero non eleifend. Sed ultricies rutrum felis, a dapibus ipsum porta a. Aenean eu tellus turpis. Mauris sollicitudin ornare felis, eget hendrerit lacus ultricies ut. Vestibulum vehicula ante augue, id pharetra lectus sagittis sit amet. Donec augue enim, auctor nec hendrerit trincidunt, dapibus dapibus lorem. Proin eget tellus nisi. Curabitur quis nisi malesuada, laculis arcu in, trincidunt est. Praesent in risus ligula. Mauris vulputate ut urna id sollicitudin. Morbi eget nisl nulla.

Alpha : true

HTML ThreeJS Div

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi imperdiet tempus libero non eleifend. Sed ultricies rutrum felis, a dapibus ipsum porta a. Aenean eu tellus turpis. Mauris sollicitudin ornare felis, eget hendrerit lacus ultricies ut. Vestibulum vehicula ante augue, id pharetra lectus sagittis sit amet. Donec augue enim, auctor nec hendrerit trincidunt, dapibus dapibus lorem. Proin eget tellus nisi. Curabitur quis nisi malesuada, laculis arcu in, trincidunt est. Praesent in risus ligula. Mauris vulputate ut urna id sollicitudin. Morbi eget nisl nulla.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi imperdiet tempus libero non eleifend. Sed ultricies rutrum felis, a dapibus ipsum porta a. Aenean eu tellus turpis. Mauris sollicitudin ornare felis, eget hendrerit lacus ultricies ut. Vestibulum vehicula ante augue, id pharetra lectus sagittis sit amet. Donec augue enim, auctor nec hendrerit trincidunt, dapibus dapibus lorem. Proin eget tellus nisi. Curabitur quis nisi malesuada, laculis arcu in, trincidunt est. Praesent in risus ligula. Mauris vulputate ut urna id sollicitudin. Morbi eget nisl nulla.

Alpha : false

WebGLRenderer - canvas

HTML ThreeJS Div

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi imperdiet tempus libero non eleifend. Sed ultricies rutrum felis, a dapibus ipsum porta a. Aenean eu tellus turpis. Mauris sollicitudin ornare felis, eget hendrerit lacus ultricies ut. Vestibulum vehicula ante augue, id pharetra lectus sagittis sit amet. Donec augue enim, auctor nec hendrerit tincidunt, dapibus dapibus lorem. Proin eget tellus nisi. Curabitur quis nisi malesuada, iaculis arcu in, tincidunt est. Praesent in risus ligula. Mauris vulputate ut urna id sollicitudin. Morbi eget nisi nulla.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi imperdiet tempus libero non eleifend. Sed ultricies rutrum felis, a dapibus ipsum porta a. Aenean eu tellus turpis. Mauris sollicitudin ornare felis, eget hendrerit lacus ultricies ut. Vestibulum vehicula ante augue, id pharetra lectus sagittis sit amet. Donec augue enim, auctor nec hendrerit tincidunt, dapibus dapibus lorem. Proin eget tellus nisi. Curabitur quis nisi malesuada, iaculis arcu in, tincidunt est. Praesent in risus ligula. Mauris vulputate ut urna id sollicitudin. Morbi eget nisi nulla.

Si nous souhaitons utiliser un **canvas** existant du DOM, il est possible d'utiliser la propriété **canvas** de **WebGLRenderer**

Dans le cas où la propriété **canvas** n'est pas utilisée, un élément **canvas** sera créé dans **domElement**

```
<canvas id="ThreejsCanvas"></canvas>

var DOMcanvas = document.getElementById("ThreejsCanvas");
renderer = new THREE.WebGLRenderer( { canvas : DOMcanvas, antialias : true } );
```

WebGLRenderer - Les bases

Si vous êtes prêts, passons à la pratique !

