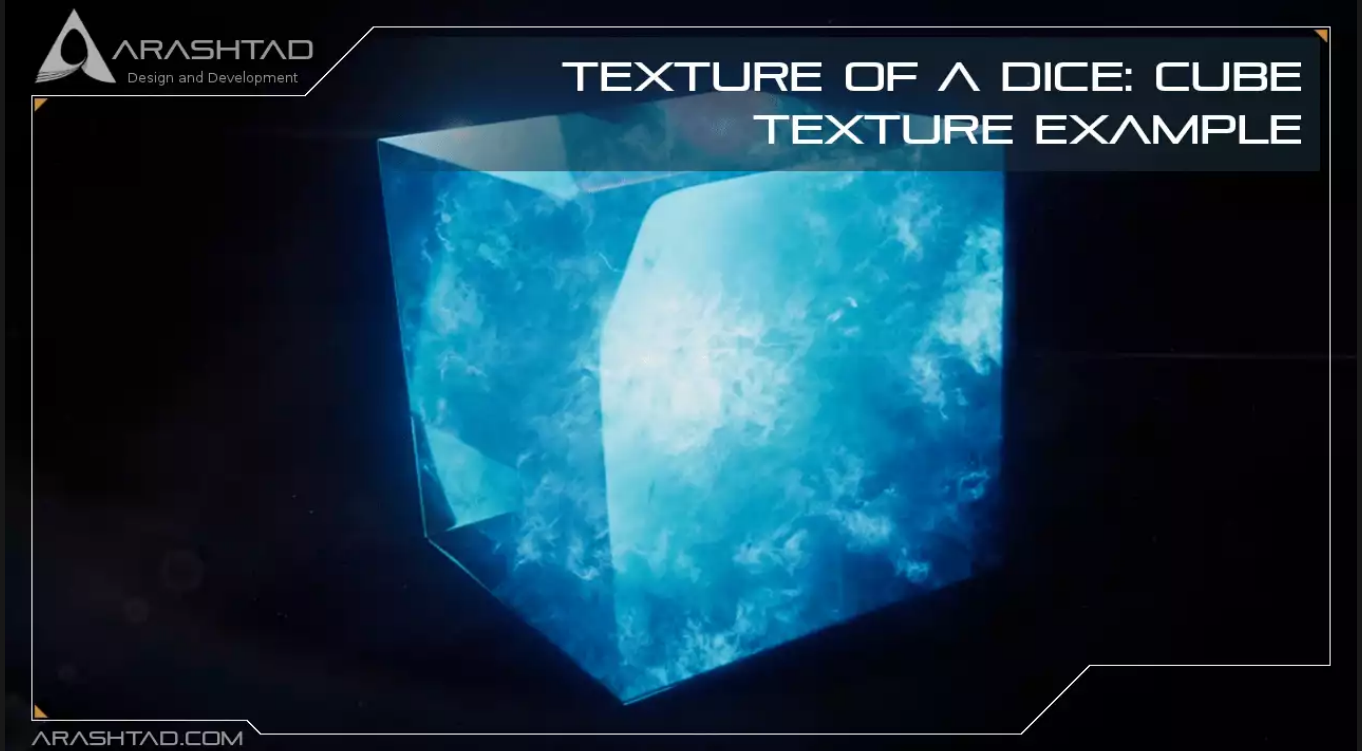


Texture of A Dice: Cube Texture example

No comments



We have provided a lot of tutorials about the textures on our blog. Some of which, we have directly worked on the textures, and others that we have used the texture in our three.js examples. However, in none of these examples we have had multi-texture objects, or in other words, we have never faced a situation in which we had to place different textures for the different sides or part of it. In this tutorial, we will work on an example that covers this situation. This example includes designing a dice or a cube with different textures on each of the sides. To make this happen, we should first download the image of each of the faces and then map them in the materials onto the object. As always, we begin with the basics and get started from creating the Vite plugin, to have all all the necessary files inside our project folder and then will add the script related creating and animating a dice. The project can be used when you want to style your website in a beautiful way at the time the user is waiting for some process to get completed.

Getting started with the basics:

We are going to get started with the simple elements of a Three.js scene including the camera, the renderer, the scene, the object and the light source (if necessary). Before we do that, we'd rather use the Vite plugin to be able to easily create all the folders and file that you need to run the Three.js code. First off, create a folder in the directory of your projects by using the following commands:

```
mkdir DiceTexture
```

`cd DiceTexture` Then, inside of the your project folder, create the necessary files and folders by simply running the Vite plugin command:

```
npm create vite@latest
```

Then enter the name of the project. You can write the name of your project as the name. And also the package (the name is arbitrary and you can choose anything that you want). Then select vanilla as the framework and variant. After that, enter the following commands in the terminal:

```
cd DiceTexture
```

`npm install` `npm install three` Afterward, you can enter the javascript code that you want to write in the main.js file. So, we will enter the base or template code for running every project with an animating object, such as a sphere. Also do not forget to install the Three.js package library every time create a project:

The code:

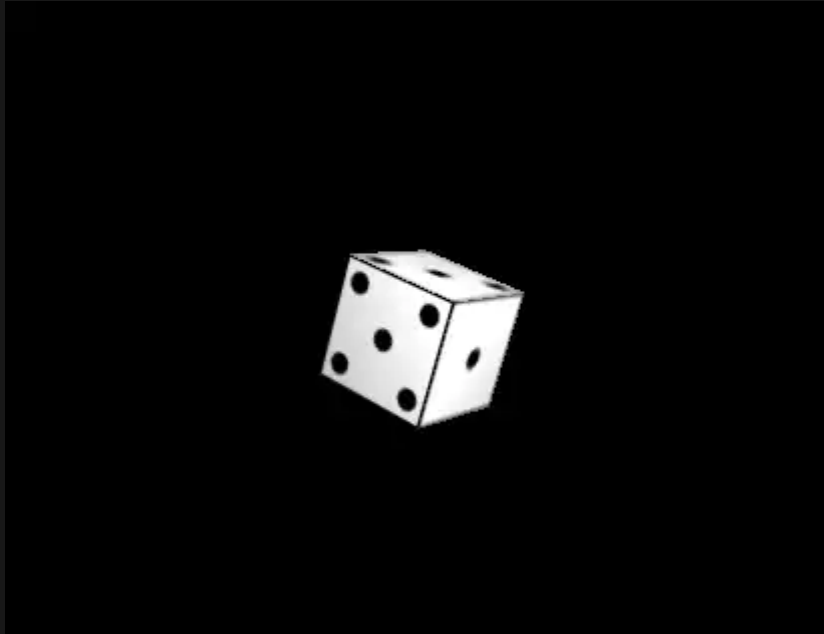
Now, enter the following script in the main.js file:

```
import * as THREE from '/node_modules/three/build/three.module.js'
import { Mesh } from 'three'
var camera, scene, renderer, dice, cube;
init();
animate();
function init() {
  renderer = new THREE.WebGLRenderer();
  renderer.setSize(window.innerWidth, window.innerHeight);
  document.body.appendChild(renderer.domElement);
  scene = new THREE.Scene();
  camera = new THREE.PerspectiveCamera(70, window.innerWidth
/ window.innerHeight, 1, 1000);
  camera.position.set(110, 110, 250);
  camera.lookAt(scene.position);
  scene.add(camera);
  var materials = [
    new THREE.MeshStandardMaterial({
      color: 0xffffffff,
      map: new THREE.TextureLoader().load('./img/1.jpg')
    }),
    new THREE.MeshStandardMaterial({
      color: 0xffffffff,
      map: new THREE.TextureLoader().load('./img/2.jpg')
    }),
    new THREE.MeshStandardMaterial({
      color: 0xffffffff,
      map: new THREE.TextureLoader().load('./img/3.jpg')
    }),
    new THREE.MeshStandardMaterial({
```

```
        color: 0xffffffff,
        map: new THREE.TextureLoader().load('./img/4.jpg')
    });
    new THREE.MeshStandardMaterial({
        color: 0xffffffff,
        map: new THREE.TextureLoader().load('./img/5.jpg')
    }),
    new THREE.MeshStandardMaterial({
        color: 0xffffffff,
        map: new THREE.TextureLoader().load('./img/6.jpg')
    })
];
cube = new THREE.BoxGeometry(40, 40, 40);
dice = new THREE.Mesh(cube, materials );
scene.add(dice);
const width = 20;
const height = 20;
const intensity = 2500;
const rectLight = new THREE.RectAreaLight( 0xffffffff, intensity,
width, height );
rectLight.position.set( 250, 250, 250 );
rectLight.lookAt(scene.position );
scene.add( rectLight )
}
function animate() {
    requestAnimationFrame(animate);
    dice.rotation.x += .05;
    dice.rotation.y += .05;
    dice.rotation.z += .05;
    renderer.render(scene, camera);
}
```

Now if we save the code, and enter the following command in the terminal:
the following result:

`npm run dev` We should see



Which is a rotating dice with a beautiful animation on all three axes.

Explaining the code:

In the above code, we set the camera, scene, and renderer. Then, we defined the material of all the sides of the dice. The materials, load the texture related to any of the sides. You can enter any photo with any design in the directory of the images. Afterward, we created a cube using the `BoxGeometry()` function and added the material to it. Notice that all of these codes up to here are written inside the `init()` function. We wrote another function for the animation. And in the end, we executed the 2 functions.

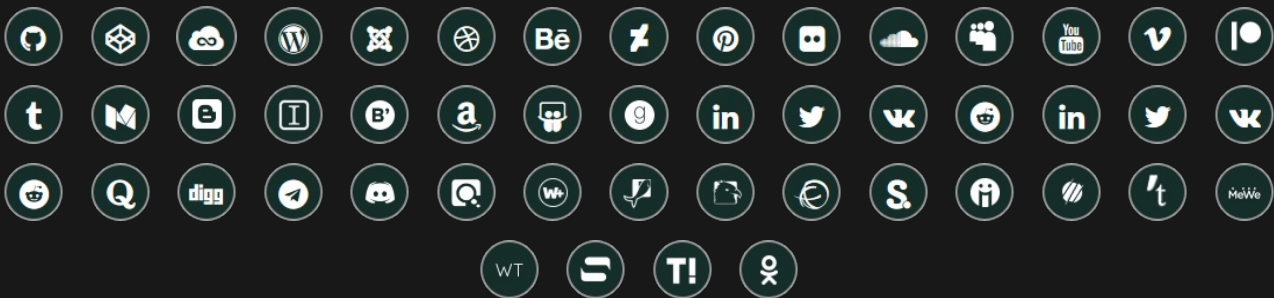
Conclusion

In this article, we have managed to create a dice with a certain kind of animation in Three.js. This example required a certain kind of applying texture. Meaning that we had to add different textures for different sides of the cube or the dice. To do so, we added these texture images to six materials concerning the six sides of the cube. The example we worked on in this article was very simple but at the same time was the question of many of those who had learned how to use different kinds of textures but did not know how to apply different textures on different sides of the same cube.

Join Arashtad Community

Follow Arashtad on Social Media

We provide variety of content, products, services, tools, tutorials, etc. Each social profile according to its features and purpose can cover only one or few parts of our updates. We can not upload our videos on SoundCloud or provide our eBooks on Youtube. So, for not missing any high quality original content that we provide on various social networks, make sure you follow us on as many social networks as you're active in. You can find out Arashtad's profiles on different social media services.



Get Even Closer!

Did you know that only one universal Arashtad account makes you able to log into all Arashtad network at once? Creating an Arashtad account is free. Why not to try it? Also, we have regular updates on our newsletter and feed entries. Use all these beneficial free features to get more involved with the community and enjoy the many products, services, tools, tutorials, etc. that we provide frequently.

[SIGN UP](#)[NEWSLETTER](#)[RSS FEED](#)