

Information Visualization W15: Final Tasks

177x202x

2017 年 6 月 17 日

プログラム概要

ブラウザ上で BoxGeometry のプロパティ値を変更できるプログラムを作成した。パラメータを変化させるために dat.GUI を使用している。

ソースコード

今回使用したソースコードを掲載する。使用しているライブラリは細かな作業のための jQuery と、ThreeJS からは three.min.js, Detector.js, OrbitControls.js, dat.gui.min.js を読み込んでいる。

```
(function($){  
    var width = 640;  
    var height = 480;  
    var scene, camera, renderer, controls, geoObj, mesh;  
  
    function main(){  
        // WebGL環境確認  
        if(!Detector.webgl) Detector.addGetWebGLMessage();  
        // GUI導入  
        var gui = new dat.GUI();  
  
        scene = new THREE.Scene();  
  
        // カメラ: 透視投影  
        camera = new THREE.PerspectiveCamera( 60, width/height, 0.001, 2000);  
        scene.add(camera);  
        camera.position.set( 0, 20, 40);  
  
        // レンダラー  
        renderer = new THREE.WebGLRenderer({antialias: true});  
        renderer.setSize(width,height);  
        renderer.setPixelRatio( window.devicePixelRatio);  
        renderer.shadowMap.enabled = true;  
        document.body.appendChild( renderer.domElement );  
  
        // ジオメトリー  
        var material = new THREE.MeshLambertMaterial ({color: 0x22ABCD});  
        mesh = new THREE.Mesh();  
        mesh.material = material;  
        mesh.position.set(0,0,0);  
        mesh.castShadow = true;  
        scene.add(mesh);  
  
        // パラメーター設置  
        geoObj = new geoCtrl();  
        var folder = gui.addFolder('BoxGeometry')  
        folder.add( geoObj, 'x', 10, 80 ).onChange( setGeoVal);  
        folder.add( geoObj, 'y', 10, 80 ).onChange( setGeoVal);  
        folder.add( geoObj, 'z', 10, 80 ).onChange( setGeoVal);  
        setGeoVal();  
    }  
})
```

```

// 自然光
var ambientLight = new THREE.AmbientLight( 0xDDDDCC, 0.8);

// スポットライト
var spotLight = new THREE.SpotLight(0xFFFFFF,1.2,0);
spotLight.castShadow = true;
spotLight.position.set( 10, 30, 30);
scene.add(ambientLight,spotLight);

//コントローラー
controls = new THREE.OrbitControls(camera);
controls.maxDistance = 100;
controls.maxPolarAngle = Math.PI * 0.48;

resizeSet();
setTimeout(resize, 1);

//自動回転
controls.autoRotate = true;
controls.autoRotateSpeed = 2.0;

rendering();
}

//GUIパラメータの初期値
var geoCtrl = function(){
    this.x = 20;
    this.y = 20;
    this.z = 20;
};

//GUIパラメータの準備
function setGeoVal(){
    mesh.geometry.dispose();
    mesh.geometry = new THREE.BoxGeometry( geoObj.x, geoObj.y ,geoObj.z );
}

function rendering(){
    requestAnimationFrame(rendering);
    controls.update();
    renderrer.render( scene, camera);
}

function resizeSet(){
    var queue = null; // キューをストック
    var wait = 300; // 0.3秒後に実行
    window.addEventListener( 'resize', function() {
        clearTimeout( queue );
        queue = setTimeout(function() {
            resize();
        }, wait );
    }, false );
};

function resize(){
    var width = window.innerWidth;
    var height = window.innerHeight;
    camera.aspect = width/height;
    camera.updateProjectionMatrix();
    renderrer.setSize(width, height);
}

$(function(){
    main();
});
})(jQuery);

```

実行結果

プログラムを実行させると、図 1 のような画面が表示される。開始直後の BoxGeometry の形状は $(x : y : z) = (20 \times 20 \times 20)$ の立方体である。

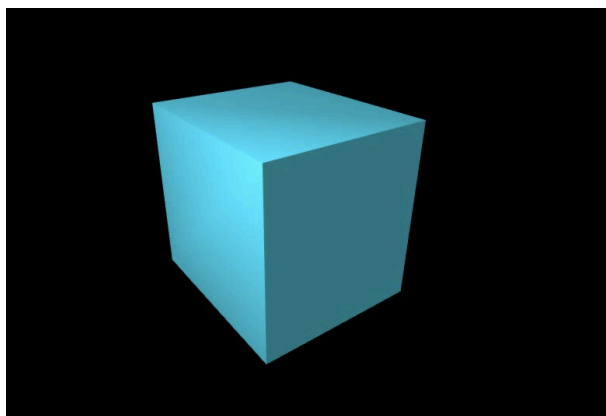


図 1 実行直後の BoxGeometry

画面右上のボックスのシークバーを前後させると、各軸方向の辺のサイズを 10 ～80 の範囲でコントロールできる。図 2 は、 $(x,y,z) = (25,25,63)$ のBoxGeometry。

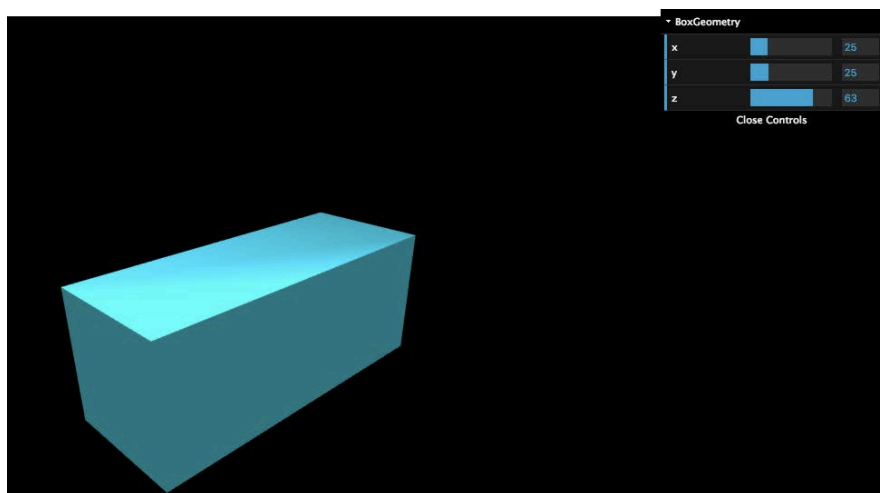


図 2 辺のサイズを変えた BoxGeometry