**简介**

FlyControls是相机的控件，飞行模拟器控件,用键盘和鼠标控制相机移动和旋转。这个控件使用可以把相机想象成是无人机的摄像头。

案例查看地址：<http://www.wjceo.com/blog/threejs/2018-02-12/32.html>

### 操作方式

A键和D键控制镜头左右移动   
W键|鼠标左键和S键|鼠标右键控制镜头前进后退   
R键和F键控制镜头的前进后退   
Q键和E键控制镜头沿Z轴进行顺时针和逆时针旋转   
向左键和向右键控制镜头沿Y轴旋转   
向上键和向下键控制镜头沿X轴旋转

**简单使用**

首先需要将飞行空间代码引入

<script src="examples/js/controls/FlyControls.js"></script>

* 1

然后实例化完了相机以后，就可以实例化FlyControls了

controls = new THREE.FlyControls( camera ); //传入相机对象

* 1

设置相关的属性

controls.movementSpeed = 100; //设置移动的速度

controls.rollSpeed = Math.PI / 6; //设置旋转速度

* 1
* 2

这里需要实例化一个clock对象来计算每一帧所花费的时间

clock = new THREE.Clock();

* 1

在每次重新渲染的时候，需要获取到这次的时间，并且调用update的时候传入

var delta = clock.getDelta();

controls.update(delta);

* 1
* 2

这样就完成了一个简单的案例

具体代码以下

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<style type="text/css">

html, body {

margin: 0;

height: 100%;

}

canvas {

display: block;

}

</style>

</head>

<body onload="draw();">

</body>

<script src="build/three.js"></script>

<script src="examples/js/controls/FlyControls.js"></script>

<script src="examples/js/libs/stats.min.js"></script>

<script src="examples/js/libs/dat.gui.min.js"></script>

<script>

var renderer;

function initRender() {

renderer = new THREE.WebGLRenderer({antialias: true});

renderer.setSize(window.innerWidth, window.innerHeight);

//告诉渲染器需要阴影效果

renderer.shadowMap.enabled = true;

renderer.shadowMap.type = THREE.PCFSoftShadowMap; // 默认的是，没有设置的这个清晰 THREE.PCFShadowMap

renderer.gammaInput = true;

renderer.gammaOutput = true;

document.body.appendChild(renderer.domElement);

}

var camera;

function initCamera() {

camera = new THREE.PerspectiveCamera(45, window.innerWidth / window.innerHeight, 0.1, 1000);

camera.position.set(0, 40, 0);

}

var scene;

function initScene() {

scene = new THREE.Scene();

}

var ambientLight,pointLight;

function initLight() {

ambientLight = new THREE.AmbientLight("#111111");

scene.add(ambientLight);

pointLight = new THREE.PointLight("#ffffff");

pointLight.position.set(-40, 60, -10);

//告诉平行光需要开启阴影投射

pointLight.castShadow = true;

scene.add(pointLight);

}

var cube,plane;

function initModel() {

//辅助工具

var helper = new THREE.AxisHelper(10);

scene.add(helper);

//球体

var sphereGeometry = new THREE.SphereGeometry(10,30,30);

var sphereMaterial = new THREE.MeshStandardMaterial({color:0xff00ff});

var sphere = new THREE.Mesh(sphereGeometry,sphereMaterial);

sphere.position.set(-20,20,0);

sphere.castShadow = true;

scene.add(sphere);

//立方体

var cubeGeometry = new THREE.CubeGeometry(10,10,10);

var cubeMaterial = new THREE.MeshLambertMaterial({color: 0x00ffff});

cube = new THREE.Mesh(cubeGeometry, cubeMaterial);

cube.position.x = 30;

cube.position.y = 5;

cube.position.z = -5;

//告诉立方体需要投射阴影

cube.castShadow = true;

scene.add(cube);

//底部平面

var planeGeometry = new THREE.PlaneGeometry(5000, 5000, 20, 20);

var planeMaterial = new THREE.MeshStandardMaterial({color: 0xaaaaaa});

plane = new THREE.Mesh(planeGeometry, planeMaterial);

plane.rotation.x = -0.5 \* Math.PI;

plane.position.y = -0;

//告诉底部平面需要接收阴影

plane.receiveShadow = true;

scene.add(plane);

}

//初始化性能插件

var stats;

function initStats() {

stats = new Stats();

document.body.appendChild(stats.dom);

}

//用户交互插件 鼠标左键按住旋转，右键按住平移，滚轮缩放

var controls,clock;

function initControls() {

clock = new THREE.Clock();

controls = new THREE.FlyControls( camera );

controls.movementSpeed = 100; //设置移动的速度

controls.rollSpeed = Math.PI / 6; //设置旋转速度

controls.autoForward = false;

controls.dragToLook = false;

}

//初始化dat.GUI简化试验流程

var param;

function initGui() {

}

function render() {

renderer.render(scene, camera);

}

//窗口变动触发的函数

function onWindowResize() {

camera.aspect = window.innerWidth / window.innerHeight;

camera.updateProjectionMatrix();

render();

renderer.setSize(window.innerWidth, window.innerHeight);

}

function animate() {

var delta = clock.getDelta();

//更新控制器

render();

//更新性能插件

stats.update();

controls.update(delta);

requestAnimationFrame(animate);

}

function draw() {

initRender();

initScene();

initCamera();

initLight();

initModel();

initControls();

initStats();

initGui();

animate();

window.onresize = onWindowResize;

}

</script>

</html>