# 1.环境准备

## 1.需要安装tweens，

### 使用命令：npm install @tweenjs/tween.js

## 2.安装后的导入方法

### import \* as TWEEN from '@tweenjs/tween.js'

## 3.TWEEN的使用步骤

#### 1）创建对象，可以使用链式调用，创建后做上面然后需要调用对象的onUpdate方法

#### 2）调用对象的start方法

#### 3）在animate函数里面（或者适当的地方）调用TWEEN的update方法

# 2.练习项目

## 结构如下

|  |
| --- |
|  |

## 这里有9个函数，副本演示9种情况，第一和第二种是最基本的，第3个快速我们设置它重复运动，从第六发函数开始，我们使用动作链，就是利用chain方法把两个或者以上的动作链接起来，注意：使用这种方式，每一个动作都不能够调用repeat方法，否则不会有效果

### appfunc1.js

|  |
| --- |
| import \* as THREE from 'three'  import ThreeApp from '../lib/threeapp2'  import \* as TWEEN from '@tweenjs/tween.js' //导入tweenjs  export function appFunc1(canvas){      let threeApp = new ThreeApp(canvas)      threeApp.initApp()      threeApp.animate()        let axesHelper = new THREE.AxesHelper(8)      threeApp.scene.add(axesHelper)        const boxGeometry = new THREE.BoxGeometry(4,4,4);      const boxMaterial = new THREE.MeshNormalMaterial();      const box = new THREE.Mesh(boxGeometry, boxMaterial);      threeApp.scene.add(box);      const groundGeometry = new THREE.BoxGeometry(24,1,24);      const groundMaterial = new THREE.MeshNormalMaterial();      const groundMesh = new THREE.Mesh(groundGeometry, groundMaterial);      groundMesh.position.y = -4;      threeApp.scene.add(groundMesh);        const animate = (t) => {        TWEEN.update(t)        window.requestAnimationFrame(animate);      };      animate();      let tween = new TWEEN.Tween({x:0})          .to({x:5})          .onUpdate((coords)=>{            box.position.x = coords.x          }).start()  } |

### appfunc2.js

|  |
| --- |
| import \* as THREE from 'three'  import ThreeApp from '../lib/threeapp2'  import \* as TWEEN from '@tweenjs/tween.js' //导入tweenjs  export function appFunc2(canvas){      let threeApp = new ThreeApp(canvas)      threeApp.initApp()      threeApp.animate()        let axesHelper = new THREE.AxesHelper(8)      threeApp.scene.add(axesHelper)        const boxGeometry = new THREE.BoxGeometry(4,4,4);      const boxMaterial = new THREE.MeshNormalMaterial();      const box = new THREE.Mesh(boxGeometry, boxMaterial);      threeApp.scene.add(box);      const groundGeometry = new THREE.BoxGeometry(24,1,24);      const groundMaterial = new THREE.MeshNormalMaterial();      const groundMesh = new THREE.Mesh(groundGeometry, groundMaterial);      groundMesh.position.y = -4;      threeApp.scene.add(groundMesh);        const animate = (t) => {        TWEEN.update(t)        window.requestAnimationFrame(animate);      };      animate();      let tween = new TWEEN.Tween({x:0,xRatation:0})          .to({x:5,xRatation:Math.PI/2},2000)          .onUpdate((coords)=>{            box.position.x = coords.x            box.rotation.x = coords.xRatation          }).start()  } |

### appfunc3.js

|  |
| --- |
| import \* as THREE from 'three'  import ThreeApp from '../lib/threeapp2'  import \* as TWEEN from '@tweenjs/tween.js' //导入tweenjs  export function appFunc3(canvas){      let threeApp = new ThreeApp(canvas)      threeApp.initApp()      threeApp.animate()        let axesHelper = new THREE.AxesHelper(8)      threeApp.scene.add(axesHelper)        const boxGeometry = new THREE.BoxGeometry(4,4,4);      const boxMaterial = new THREE.MeshNormalMaterial();      const box = new THREE.Mesh(boxGeometry, boxMaterial);      threeApp.scene.add(box);      const groundGeometry = new THREE.BoxGeometry(24,1,24);      const groundMaterial = new THREE.MeshNormalMaterial();      const groundMesh = new THREE.Mesh(groundGeometry, groundMaterial);      groundMesh.position.y = -4;      threeApp.scene.add(groundMesh);        const animate = (t) => {        TWEEN.update(t)        window.requestAnimationFrame(animate);      };      animate();      let tween = new TWEEN.Tween({x:0,y:0,xRotation:0})          .to({x:5,y:8,xRotation:Math.PI/2},2000)          .onUpdate((coords)=>{            box.position.x = coords.x            box.position.y = coords.y            box.rotation.x = coords.xRotation          }).repeat(Infinity)          .delay(500)          .start()  } |

### appfunc4.js

|  |
| --- |
| import \* as THREE from 'three'  import ThreeApp from '../lib/threeapp2'  import \* as TWEEN from '@tweenjs/tween.js' //导入tweenjs  export function appFunc4(canvas){      let threeApp = new ThreeApp(canvas)      threeApp.initApp()      threeApp.animate()        let axesHelper = new THREE.AxesHelper(8)      threeApp.scene.add(axesHelper)        const boxGeometry = new THREE.BoxGeometry(4,4,4);      const boxMaterial = new THREE.MeshNormalMaterial();      const box = new THREE.Mesh(boxGeometry, boxMaterial);      threeApp.scene.add(box);      const groundGeometry = new THREE.BoxGeometry(24,1,24);      const groundMaterial = new THREE.MeshNormalMaterial();      const groundMesh = new THREE.Mesh(groundGeometry, groundMaterial);      groundMesh.position.y = -4;      threeApp.scene.add(groundMesh);        const animate = (t) => {        TWEEN.update(t)        window.requestAnimationFrame(animate);      };      animate();      let tween = new TWEEN.Tween({x:0,y:0,xRotation:0})          .to({x:5,y:8,xRotation:Math.PI/2},2000)          .onUpdate((coords)=>{            box.position.x = coords.x            box.position.y = coords.y            box.rotation.x = coords.xRotation          }).repeat(Infinity)          .easing(TWEEN.Easing.Exponential.InOut)          .delay(500)          .start()  } |

### appfunc5.js

|  |
| --- |
| import \* as THREE from 'three'  import ThreeApp from '../lib/threeapp2'  import \* as TWEEN from '@tweenjs/tween.js' //导入tweenjs  export function appFunc5(canvas){      let threeApp = new ThreeApp(canvas)      threeApp.initApp()      threeApp.animate()        let axesHelper = new THREE.AxesHelper(8)      threeApp.scene.add(axesHelper)        const boxGeometry = new THREE.BoxGeometry(4,4,4);      const boxMaterial = new THREE.MeshNormalMaterial();      const box = new THREE.Mesh(boxGeometry, boxMaterial);      threeApp.scene.add(box);      const groundGeometry = new THREE.BoxGeometry(24,1,24);      const groundMaterial = new THREE.MeshNormalMaterial();      const groundMesh = new THREE.Mesh(groundGeometry, groundMaterial);      groundMesh.position.y = -4;      threeApp.scene.add(groundMesh);        const animate = (t) => {        TWEEN.update(t)        window.requestAnimationFrame(animate);      };      animate();      let tween = new TWEEN.Tween({x:0,y:0,xRotation:0})          .to({x:5,y:8,xRotation:Math.PI/2},2000)          .onUpdate((coords)=>{            box.position.x = coords.x            box.position.y = coords.y            box.rotation.x = coords.xRotation          }).repeat(Infinity)          .easing(TWEEN.Easing.Exponential.InOut)          .delay(500)      tween.start()      let box2 = new THREE.Mesh(boxGeometry,boxMaterial)      threeApp.scene.add(box2)      let tween2 = new TWEEN.Tween({x:0,y:0,xRotation:0})      .to({x:-5,y:8,xRotation:Math.PI/2},2000)      .onUpdate((coords)=>{        box2.position.x = coords.x        box2.position.y = coords.y        box2.rotation.x = coords.xRotation      }).repeat(Infinity)      .easing(TWEEN.Easing.Linear.None)      .delay(500)      tween2.start()  } |

### appfunc6.js

|  |
| --- |
| import \* as THREE from 'three'  import ThreeApp from '../lib/threeapp2'  import \* as TWEEN from '@tweenjs/tween.js' //导入tweenjs  export function appFunc6(canvas){      let threeApp = new ThreeApp(canvas)      threeApp.initApp()      threeApp.animate()        let axesHelper = new THREE.AxesHelper(8)      threeApp.scene.add(axesHelper)        const boxGeometry = new THREE.BoxGeometry(4,4,4);      const boxMaterial = new THREE.MeshNormalMaterial();      const box = new THREE.Mesh(boxGeometry, boxMaterial);      threeApp.scene.add(box);      const groundGeometry = new THREE.BoxGeometry(24,1,24);      const groundMaterial = new THREE.MeshNormalMaterial();      const groundMesh = new THREE.Mesh(groundGeometry, groundMaterial);      groundMesh.position.y = -4;      threeApp.scene.add(groundMesh);        const animate = (t) => {        TWEEN.update(t)        window.requestAnimationFrame(animate);      };      animate();      const tween1 = new TWEEN.Tween({ x: 0, y: 0, xRotation: 0 })      .to({ x: 5, y: 8, xRotation: Math.PI / 2 }, 2000)      .onUpdate((coords) => {        box.position.x = coords.x;        box.position.y = coords.y;        box.rotation.x = coords.xRotation;      })      .easing(TWEEN.Easing.Exponential.InOut)      .delay(100);    const tween2 = new TWEEN.Tween({ x: 5, y: 8, xRotation: Math.PI / 2 })      .to({ x: 0, y: 0, xRotation: 0 }, 2000)      .onUpdate((coords) => {        box.position.x = coords.x;        box.position.y = coords.y;        box.rotation.x = coords.xRotation;      })      .easing(TWEEN.Easing.Linear.None)      .delay(100);    tween1.chain(tween2);    tween2.chain(tween1);    tween1.start();  } |

### appfunc7.js

|  |
| --- |
| import \* as THREE from 'three'  import ThreeApp from '../lib/threeapp2'  import \* as TWEEN from '@tweenjs/tween.js' //导入tweenjs  export function appFunc7(canvas){      let threeApp = new ThreeApp(canvas)      threeApp.initApp()      threeApp.animate()        let axesHelper = new THREE.AxesHelper(8)      threeApp.scene.add(axesHelper)        const boxGeometry = new THREE.BoxGeometry(4,4,4);      const boxMaterial = new THREE.MeshNormalMaterial();      const box = new THREE.Mesh(boxGeometry, boxMaterial);      threeApp.scene.add(box);      const groundGeometry = new THREE.BoxGeometry(24,1,24);      const groundMaterial = new THREE.MeshNormalMaterial();      const groundMesh = new THREE.Mesh(groundGeometry, groundMaterial);      groundMesh.position.y = -4;      threeApp.scene.add(groundMesh);        const animate = (t) => {        TWEEN.update(t)        window.requestAnimationFrame(animate);      };      animate();      const tween1 = new TWEEN.Tween({ x: 0, y: 0, xRotation: 0 })      .to({ x: 5, y: 8, xRotation: Math.PI / 2 }, 2000)      .onUpdate((coords) => {        box.position.x = coords.x;        box.position.y = coords.y;        box.rotation.x = coords.xRotation;      })      .easing(TWEEN.Easing.Exponential.InOut)      .delay(100);    const tween2 = new TWEEN.Tween({ x: 5, y: 8, xRotation: Math.PI / 2 })      .to({ x: -5, y: 8, xRotation: 0 }, 2000)      .onUpdate((coords) => {        box.position.x = coords.x;        box.position.y = coords.y;        box.rotation.x = coords.xRotation;      })      .easing(TWEEN.Easing.Linear.None)      .delay(100);    tween1.chain(tween2);    tween2.chain(tween1);    tween1.start();  } |

### appfunc8.js

|  |
| --- |
| import \* as THREE from 'three'  import ThreeApp from '../lib/threeapp2'  import \* as TWEEN from '@tweenjs/tween.js' //导入tweenjs  export function appFunc8(canvas){      let threeApp = new ThreeApp(canvas)      threeApp.initApp()      threeApp.animate()        let axesHelper = new THREE.AxesHelper(8)      threeApp.scene.add(axesHelper)        const boxGeometry = new THREE.BoxGeometry(4,4,4);      const boxMaterial = new THREE.MeshNormalMaterial();      const box = new THREE.Mesh(boxGeometry, boxMaterial);      threeApp.scene.add(box);      const groundGeometry = new THREE.BoxGeometry(24,1,24);      const groundMaterial = new THREE.MeshNormalMaterial();      const groundMesh = new THREE.Mesh(groundGeometry, groundMaterial);      groundMesh.position.y = -4;      threeApp.scene.add(groundMesh);        const animate = (t) => {        TWEEN.update(t)        window.requestAnimationFrame(animate);      };      animate();      const tween1 = new TWEEN.Tween({ x: 0, y: 0, xRotation: 0 })      .to({ x: 5, y: 8, xRotation: Math.PI / 2 }, 2000)      .onUpdate((coords) => {        box.position.x = coords.x;        box.position.y = coords.y;        box.rotation.x = coords.xRotation;      })      .easing(TWEEN.Easing.Exponential.InOut)      .delay(100);    const tween2 = new TWEEN.Tween({ x: 5, y: 8, xRotation: Math.PI / 2 })      .to({ x: -5, y: 8, xRotation: 0 }, 2000)      .onUpdate((coords) => {        box.position.x = coords.x;        box.position.y = coords.y;        box.rotation.x = coords.xRotation;      })      .easing(TWEEN.Easing.Linear.None)      .delay(100);    let tween3 = new TWEEN.Tween({x:-5,y:8,xRotation:0})        .to({x:0,y:0,xRotation:- Math.PI / 2 },2000)        .onUpdate(coords=>{          box.position.x = coords.x;          box.position.y = coords.y;          box.rotation.x = coords.xRotation;        }).easing(TWEEN.Easing.Linear.None)        .delay(100)    tween1.chain(tween2);    tween2.chain(tween3);    tween3.chain(tween1)    tween1.start();  } |

## 然后需要在App.jsx里面导入并且调用对应的函数，第8个是这个练习里面最完美的

|  |
| --- |
| import { useEffect } from 'react'  import './App.css'  import { appFunc1 } from './appFuncs/appfunc1'  import { appFunc2 } from './appFuncs/appfunc2'  import { appFunc3 } from './appFuncs/appfunc3'  import { appFunc4 } from './appFuncs/appfunc4'  import { appFunc5 } from './appFuncs/appfunc5'  import { appFunc6 } from './appFuncs/appfunc6'  import { appFunc7 } from './appFuncs/appfunc7'  import { appFunc8 } from './appFuncs/appfunc8'    function App() {    useEffect(() => {      let canvas = document.getElementById("myThreeJSCanvas") //在外面创建canvas      // appFunc1(canvas)      // appFunc2(canvas)      // appFunc3(canvas)      // appFunc4(canvas)      // appFunc5(canvas)      // appFunc6(canvas)      // appFunc7(canvas)      appFunc8(canvas)    }, [])    return (      <>        <div>          <canvas id="myThreeJSCanvas"></canvas>        </div>      </>    )  }  export default App |

### 效果：物体会呈三角形轨迹移动并且旋转，原点-》右边上面-》左边上面=》原点

## 总结TWEEN库提供了补间动画功能，有很多方法如repeat，delay，to，onUpdate,easing，chain方法等等可以把重复进行同一个动作，也可以把一连串的动作链接起来，完成特殊的效果