郭隆邦 2023-03-24

7. 变形动画原理

下面给大家介绍下变形动画的基本原理。

`.morphAttributes` 设置几何体变形目标顶点数据

`BufferGeometry` 属性 `.morphAttributes` 的功能就是用来设置几何体变形目标顶点数据。

```js
//几何体两组顶点一一对应，位置不同，然后通过权重系数，可以控制模型形状在两组顶点之间变化
const geometry = new THREE.BoxGeometry(50, 50, 50);
// 为geometry提供变形目标的顶点数据(注意和原始geometry顶点数量一致)
const target1 = new THREE.BoxGeometry(50, 200, 50).attributes.position;//变高
const target2 = new THREE.BoxGeometry(10, 50, 10).attributes.position;//变细
// 几何体顶点变形目标数据，可以设置1组或多组
geometry.morphAttributes.position = [target1, target2];

const mesh = new THREE.Mesh(geometry, material);
```

注意：给一个几何体geometry设置顶点变形数据 `.morphAttributes` 时候要注意，在执行代码 `new THREE.Mesh()` 之前设置，否则报错。

`.morphTargetInfluences` 权重系数控制变形程度

查看文档，你可以看到网格模型 `Mesh` 、点模型、线模型都有一个权重属性 `.morphTargetInfluences` ，该权重的作用是，控制geometry自身顶点和变形目标顶点分别对模型形状形象程度。

设置变形目标影响权重，范围一般0~1。

mesh在geometry原始形状和变形目标1顶点对应形状之间变化。

mesh的几何体变形目标是放在一个数组 `.morphAttributes.position` 中的，设置权重系数 `morphTargetInfluences` 的时候，需要设置索引值， 比如 `.morphTargetInfluences[0]` 影

响的变形目标是`.morphAttributes.position[0]`，`.morphTargetInfluences[1]`影响的变形目标是`.morphAttributes.position[1]`。

```js
//权重0：物体形状对应geometry.attributes.position表示形状
mesh.morphTargetInfluences[0] = 0.0;
//权重1：物体形状对应target1表示形状
mesh.morphTargetInfluences[0] = 1.0;
//权重0.5：物体形状对应geometry和target1变形中间状态
mesh.morphTargetInfluences[0] = 0.5;
```

mesh在geometry原始形状和变形目标2顶点对应形状之间变化

```js
mesh.morphTargetInfluences[1] = 0.5;
```

多个变形目标综合影响模型形状

一个网格模型的几何体geometry可以有多个变形目标，只要对应权重不为0，每个变形目标的形状都会影响模型的形状。

```js
// 两个变形目标同时影响模型形状
mesh.morphTargetInfluences[1] = 0.5;
mesh.morphTargetInfluences[0] = 0.5;
```

GUI控制变形权重系数`.morphTargetInfluences`

```js
import {GUI} from 'three/addons/libs/lil-gui.module.min.js';
const gui = new GUI();
// GUI拖动条可视化改变变形目标权重系数
const obj = {
    t1: 0,
    t2: 0,
}
gui.add(obj, 't1', 0, 1).name('变形目标1').onChange(function (v) {
    // 变形目标1对物体形状影响权重
    mesh.morphTargetInfluences[0] = v;
});
gui.add(obj, 't2', 0, 1).name('变形目标2').onChange(function (v) {
    // 变形目标2对物体形状影响权重
    mesh.morphTargetInfluences[1] = v;
```

```
});
```

生成变形动画

生成变形动画的方法非常简单，你只是需要通过关键帧动画，改变模型的变形权重系数即可。

变形动画目标：0~5秒，物体变高，5~10秒，物体变细。

```js
// 创建变形动画权重系数的关键帧数据
mesh.name = "Box";//关键帧动画控制的模型对象命名
// 设置变形目标1对应权重随着时间的变化
const KF1 = new THREE.KeyframeTrack('Box.morphTargetInfluences[0]', [0, 5], [0,
// 设置变形目标2对应权重随着时间的变化
const KF2 = new THREE.KeyframeTrack('Box.morphTargetInfluences[1]', [5, 10], [0,
// 创建一个剪辑clip对象
const clip = new THREE.AnimationClip("t", 10, [KF1, KF2]);
```

播放变形模型对应的关键帧动画，你可以看到一个变形动画效果，下面代码和前面播放关键帧动画的代码一样。

```js
// 播放变形动画
const mixer = new THREE.AnimationMixer(mesh);
const clipAction = mixer.clipAction(clip);
clipAction.play();
clipAction.loop = THREE.LoopOnce; //不循环播放
clipAction.clampWhenFinished = true // 物体状态停留在动画结束的时候

const clock = new THREE.Clock();

function loop() {
    requestAnimationFrame(loop);
    const frameT = clock.getDelta();
    // 更新播放器时间
    mixer.update(frameT);
}
loop();
```

解析外部变形动画模型

项目开发，大部分情况下，不需要你代码编辑变形动画的几何体变形数据，通常是在三维建模软件中，比如Blender，编辑好变形数据，你只需要在代码中播放变形动画即可。

播放外部变形动画模型，和你播放其它外部模型的关键帧动画一样，只要美术在三维软件中设置好变形的关键帧动画，你只需要播放关键帧动画即可，不同管内部的变形过程。

```js
loader.load("./鸟.glb", function (gltf) {
    model.add(gltf.scene);

    //包含关键帧动画的模型作为参数创建一个播放器
    const mixer = new THREE.AnimationMixer(gltf.scene);
    //  获取gltf.animations[0]的第一个clip动画对象
    const clipAction = mixer.clipAction(gltf.animations[0]);
    clipAction.play();

    const clock = new THREE.Clock();
    function loop() {
        requestAnimationFrame(loop);
        const frameT = clock.getDelta();
        // 更新播放器相关的时间
        mixer.update(frameT);
    }
    loop();
})
```

← 6. 虚拟装配(任意时间定位)　　8. 变形动画(定制人物胖瘦)→