

🟡 10. 查看外部模型骨骼动画

实际开发一般在三维建模软件中编辑骨骼动画，对于threejs而言，你只需要掌握如何通过代码获取你想要的骨骼信息即可。

下面给大家演示，加载一个外部骨骼动画模型，查看骨骼结构，并播放动画。

骨骼动画创建

实际开发时候，比如一个人物角色模型你想实现走路、奔跑等动作，一般需要美术通过三维建模软件骨骼动画功能实现。简单说就是模型表面顶点数据跟着虚拟的骨骼关节 `Bone` 发生变化，比如一个骨骼关节 `Bone` 转动，附近的皮肤(模型顶点)跟着旋转变形。

最后用关键帧动画控制骨骼关节运动，带动身体皮肤变形，就可以生成骨骼动画。

可视化外部模型骨骼关节

把加载的模型 `gltf.scene` 作为 `THREE.SkeletonHelper()` 参数，添加到场景中,就可以可视化 `gltf.scene` 里面骨骼网格模型包含的骨骼关节。

```
const loader = new GLTFLoader();
loader.load("../骨骼动画.glb", function (gltf) {
  console.log('控制台查看gltf对象结构', gltf);
  model.add(gltf.scene);
  // 骨骼辅助显示
  const skeletonHelper = new THREE.SkeletonHelper(gltf.scene);
  model.add(skeletonHelper);
})
```

js

根据骨骼名称读取骨骼关节

骨骼关节 `Bone` 的父类`Object3D`，自然会继承父类`Object3D`的方法 `.getObjectByName()`，就像普通网格模型一样可以通过节点名称获取骨骼关节对象。

js

```
// 根据骨骼关节名字获取骨关节Bone
// 在三维软件中，骨骼关节层层展开，可以看到下面三个骨骼关节
const bone1 = gltf.scene.getObjectByName('Bone1'); //关节1
const bone2 = gltf.scene.getObjectByName('Bone2'); //关节2
const bone3 = gltf.scene.getObjectByName('Bone3'); //关节3
```

代码测试骨骼关节 Bone 带动模型表面变化

几个骨骼关节旋转了一定角度，你可以打开代码，观察骨骼网格模型的外形变化。

js

```
bone2.rotation.x = Math.PI / 6; //关节2旋转
bone3.rotation.x = Math.PI / 6; //关节3旋转
```

GUI拖动条控制骨骼关节 Bone 参数变化

js

```
import {GUI} from 'three/addons/libs/lil-gui.module.min.js';
const gui = new GUI();
gui.add(bone1.rotation, 'x', 0, Math.PI / 3).name('关节1');
gui.add(bone2.rotation, 'x', 0, Math.PI / 3).name('关节2');
gui.add(bone3.rotation, 'x', 0, Math.PI / 3).name('关节3');
```

查看骨骼网格模型 SkinnedMesh

在threejs中，骨骼动画模型的外表面是用骨骼网格模型 `SkinnedMesh` 表示，`SkinnedMesh` 表示的父类就是网格模型 `Mesh`，`SkinnedMesh` 和网格模型 `Mesh` 一样都是网格模型，用来表达一个物体的外表面。

加载gltf模型，如果存在骨骼网格模型的话，可以通过加载返回gltf对象的场景属性 `.scene` 查看。浏览器控制层层展开gltf.scene，就可以找到 `SkinnedMesh` 的模型对象。

js

```
const loader = new GLTFLoader();
loader.load("../骨骼动画.glb", function (gltf) {
  console.log('控制台查看gltf对象结构', gltf);
  model.add(gltf.scene);
  // 根据节点名字获取某个骨骼网格模型
  const SkinnedMesh = gltf.scene.getObjectByName('身体');
  console.log('骨骼网格模型', SkinnedMesh);
});
```

```
})
```

访问骨骼网格模型的骨架 `SkinnedMesh.skeleton`

`SkinnedMesh` 相比较 `Mesh` 区别就是, 可以跟着自己的骨架 `.skeleton` 变化, 比如骨架里面的骨骼关节 `Bone` 旋转, 会带动附近骨骼网格模型 `SkinnedMesh` 跟着旋转。

```
// 根据节点名字获取某个骨骼网格模型
const SkinnedMesh = gltf.scene.getObjectByName('身体');
console.log('骨架', SkinnedMesh.skeleton);
```

js

骨架的骨骼关节属性 `.skeleton.bones`

骨架 `SkinnedMesh.skeleton` 的关节属性 `.bones` 是一个数组包含了所有骨骼关节, 你可以和三维软件中骨骼模型骨骼节点目录对照, 观察关节属性 `.bones` 里面骨骼关节构成规律。

在浏览器控制台, 你可以看到 `.bones` 数组里面第一个关节对应了Blender中根关节, 你点开该关节Bone, 可以层层展开下去, 查看它的子关节, 查看子关节的子关节...

```
console.log('骨架所有关节', SkinnedMesh.skeleton.bones);
console.log('根关节', SkinnedMesh.skeleton.bones[0]);
```

js

士兵练习案例(拖动条控制骨骼变化)

你可以参考上面测试, 把课件士兵骨骼动画文件, 作为练习题, 测试一遍。

播放骨骼网格模型的关键帧动画

骨骼网格模型的关键帧动画, 一般都是美术在三维软件中, 通过关键帧动画控制骨骼关节变化实现。换句话说, 播放骨骼动画, 就是前面给大家讲解过的播放关键帧动画。

```
//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);
// gltf.animations[0]休息
// gltf.animations[1]跑步
// gltf.animations[2]静止展开
// gltf.animations[3]走路
const clipAction = mixer.clipAction(gltf.animations[1]);
```

js

```
clipAction.play(); //播放动画
// 如果想播放动画,需要周期性执行`mixer.update()`更新AnimationMixer时间数据
const clock = new THREE.Clock();
function loop() {
    requestAnimationFrame(loop);
    //clock.getDelta()方法获得loop()两次执行时间间隔
    const frameT = clock.getDelta();
    // 更新播放器相关的时间
    mixer.update(frameT);
}
loop();
```

← 9. 骨骼关节Bone

11. 骨骼动画不同动作切换→