基本配置

1. 安装pnpm

```
npm install pnpm -g
```

2. 安装vite-pnpm

```
pnpm create vite
Project name: 输入 three-Vue3
选择vue
选择ts
```

3. pnpm i 安装配置包
4. pnpm dev 运行项目
5. pnpm add pinia 安装状态管理库

配置main.ts

```
import { createApp } from 'vue'
import { createPinia } from 'pinia'
import App from './App.vue'

const pinia = createPinia()
const app = createApp(App)

app.use(pinia)
app.mount('#app')
```

6. 安装three.js

```
pnpm add three
pnpm i --save-dev @types/three

import * as THREE from 'three';
import { OrbitControls } from 'three/examples/jsm/controls/OrbitControls'
import { GLTFLoader } from 'three/examples/jsm/loaders/GLTFLoader';
```

7. 安装dat.gui

```
pnpm add dat.gui --save-dev
pnpm i --save-dev @types/dat.gui
import { GUI } from "dat.gui";
```

8. 模型默认动画开启

```
var mixer:any=null
// 加载
const loader = new GLTFLoader();
loader.load('src/assets/glb/feiji.glb', function (gltf) {
  console.log('控制台查看加载gltf文件返回的对象结构', gltf);
  console.log('gltf对象场景属性', gltf.scene);
  gltf.scene.scale.set(3, 3, 3);
  gltf.scene.position.set(100,100,20)
  mixer=new THREE.AnimationMixer(gltf.scene)

  mixer.clipAction(gltf.animations[0]).play()
  // 返回的场景对象gltf.scene插入到threejs场景中
  scene.add(gltf.scene);
}, function (xhr) {
  if(xhr.loaded / xhr.total==1){
  }
}, function (error) {

  console.log('An error happened');

})

function render() {
  renderer.render(scene, camera); //执行渲染操作
  requestAnimationFrame(render)
  if (mixer) mixer.update(clock.getDelta())
}

render();
```

9. 封装第一个Class three