# 《 “精灵深林” 课程设计报告》

**一、选题背景与意义**

本次课程设计选题围绕幻想主题展开，结合我们小组对主题的想法与自我能力，我们准备构建一个“精灵深林“ 创建简单的三维”精灵“，他们将在“深林”当中”自由“舞动。运用webgl和javascrpit等语言来实现场景的构造。通过本次课程设计来让我们更能理解webGL的使用和对3D事物的理解。

**二、实现功能描述**

**（一）基础场景搭建**

1. **场景与相机创建**：利用 THREE.js 库，创建了一个 THREE.Scene 场景实例，作为所有 3D 对象的容器。同时，构建了一个 THREE.PerspectiveCamera 透视相机，视角设置为 75 度，根据窗口宽高比动态调整视锥体，相机初始位置在 (0, 10, 30)，并看向原点 (0, 0, 0)，确保合适的观察视角。
2. **渲染器配置**：实例化 THREE.WebGLRenderer 渲染器，使其渲染尺寸与浏览器窗口大小一致，背景色设定为天空蓝（0x87ceeb），并将渲染器的 domElement 添加到页面 body 元素中，实现场景在网页上的可视化展示。

**（二）光照效果**

1. **环境光与定向光添加**：为了营造逼真的光照环境，添加了 THREE.AmbientLight 环境光，强度为 0.6，颜色为白色（0xffffff），提供整体柔和的基础照明。同时，设置 THREE.DirectionalLight 定向光，强度 0.8，颜色白色，光源位置在 (10, 20, 10)，模拟类似太阳光的主光源方向，增强场景立体感与物体明暗对比。

**（三）场景元素建模**

1. **地面构建**：利用 THREE.PlaneGeometry 创建一个长、宽均为 500 的平面几何图形代表地面，材质选用 THREE.MeshStandardMaterial，颜色为墨绿色（0x228B22），并绕 X 轴旋转 -Math.PI / 2 使其平放，位置设置在 y = -1 处，贴合自然场景布局。

const groundGeometry = new THREE.PlaneGeometry(500, 500);

const groundMaterial = new THREE.MeshStandardMaterial({ color: 0x228B22 });

const groundMesh = new THREE.Mesh(groundGeometry, groundMaterial);

groundMesh.rotation.x = -Math.PI / 2;

groundMesh.position.y = -1;

scene.add(groundMesh);

1. **树木生成**：定义 createTree 函数实现树木建模，树干部分由 THREE.CylinderGeometry 生成圆柱几何体，材质为深棕色（0x8B4513）；树冠用 THREE.SphereGeometry 制作球体几何体，材质为墨绿色（0x228B22），通过随机位置在相机视点附近分布 100 棵树，丰富场景细节与自然氛围。

function createTree(positionX, positionZ) {

const trunkGeometry = new THREE.CylinderGeometry(0.2, 0.2, 3, 32);

const trunkMaterial = new THREE.MeshStandardMaterial({ color: 0x8B4513 });

const trunkMesh = new THREE.Mesh(trunkGeometry, trunkMaterial);

trunkMesh.position.set(positionX, 1, positionZ);

const foliageGeometry = new THREE.SphereGeometry(1.5, 32, 32);

const foliageMaterial = new THREE.MeshStandardMaterial({ color: 0x228B22 });

const foliageMesh = new THREE.Mesh(foliageGeometry, foliageMaterial);

foliageMesh.position.set(positionX, 3, positionZ);

scene.add(trunkMesh);

scene.add(foliageMesh);

}

1. **小精灵建模**：由多个几何体组合而成小精灵模型。身体用 THREE.CylinderGeometry，头部是 THREE.SphereGeometry，翅膀为 THREE.PlaneGeometry 并设置透明效果（透明度 0.5），还有金色光环 THREE.TorusGeometry。为每个小精灵赋予随机半径（20 - 50 之间）、初始角度和旋转速度，在场景中随机分布 30 个小精灵，增添奇幻色彩。

function createFairy() {

const radiusMin = 20;

const radiusMax = 50;

const bodyGeometry = new THREE.CylinderGeometry(0.2, 0.3, 1, 32);

const bodyMaterial = new THREE.MeshStandardMaterial({ color: 0xffc0cb });

const bodyMesh = new THREE.Mesh(bodyGeometry, bodyMaterial);

const headGeometry = new THREE.SphereGeometry(0.3, 32, 32);

const headMaterial = new THREE.MeshStandardMaterial({ color: 0xffe4b5 });

const headMesh = new THREE.Mesh(headGeometry, headMaterial);

headMesh.position.y = 0.8;

const wingGeometry = new THREE.PlaneGeometry(1, 1.5);

const wingMaterial = new THREE.MeshStandardMaterial({ color: 0xadd8e6, transparent: true, opacity: 0.5 });

const leftWing = new THREE.Mesh(wingGeometry, wingMaterial);

const rightWing = new THREE.Mesh(wingGeometry, wingMaterial);

leftWing.position.set(-0.6, 0.3, 0);

rightWing.position.set(0.6, 0.3, 0);

leftWing.rotation.y = Math.PI / 6;

rightWing.rotation.y = -Math.PI / 6;

const haloGeometry = new THREE.TorusGeometry(0.4, 0.05, 16, 100);

const haloMaterial = new THREE.MeshStandardMaterial({ color: 0xffff00, emissive: 0xffff00 });

const haloMesh = new THREE.Mesh(haloGeometry, haloMaterial);

haloMesh.position.y = 1.2;

haloMesh.rotation.x = Math.PI / 2;

const fairyGroup = new THREE.Group();

fairyGroup.add(bodyMesh);

fairyGroup.add(headMesh);

fairyGroup.add(leftWing);

fairyGroup.add(rightWing);

fairyGroup.add(haloMesh);

fairyGroup.userData = {

radius: Math.random() \* (radiusMax - radiusMin) + radiusMin,

angle: Math.random() \* Math.PI \* 2,

speed: Math.random() \* 0.5 + 0.2

};

return fairyGroup;

}

1. **白云制作**：createCloud 函数利用多个随机大小的球体 THREE.SphereGeometry 组合成云团，颜色为白色（0xffffff），通过随机位置在场景上空分布 20 朵白云，使场景更具灵动性与层次感。

function createCloud() {

const cloudGroup = new THREE.Group();

for (let i = 0; i < 5; i++) {

const cloudGeometry = new THREE.SphereGeometry(Math.random() \* 2 + 1, 32, 32);

const cloudMaterial = new THREE.MeshStandardMaterial({ color: 0xffffff });

const cloudMesh = new THREE.Mesh(cloudGeometry, cloudMaterial);

cloudMesh.position.set(

Math.random() \* 5 - 2.5,

Math.random() \* 2 - 1,

Math.random() \* 5 - 2.5

);

cloudGroup.add(cloudMesh);

}

cloudGroup.position.set(

Math.random() \* 100 - 50,

Math.random() \* 30 + 20,

Math.random() \* 100 - 50

);

cloudGroup.userData = {

direction: Math.random() > 0.5 ? 1 : -1

};

return cloudGroup;

}

**（四）动画效果**

1. **小精灵动画**：在 animate 函数中，基于时间更新小精灵状态实现动画逻辑。每个小精灵按自身速度绕随机半径圆形轨道旋转，更新角度与位置坐标，同时，利用三角函数根据时间控制翅膀 THREE.Mesh 的 rotation.z 角度，模拟翅膀拍动效果，速度感与灵动性十足。
2. **窗口自适应**：添加 window 的 resize 事件监听器，当窗口大小改变时，实时更新相机的 aspect 比例、投影矩阵，以及渲染器尺寸，确保场景在不同屏幕尺寸下正确显示、无变形。
3. **使用交互控件**：调节小精灵数量、速度，和改变背景颜色模拟实现天空变化。

**三、技术选型与工具使用**

1. **核心库**：选用 THREE.js（版本 r128）作为主要的三维图形处理库，基于 WebGL 底层能力，提供简洁易用的 API 实现 3D 场景搭建、渲染、动画等复杂功能，满足在主流浏览器环境运行需求。
2. **开发语言**：使用 JavaScript 语言编写代码，结合 THREE.js 库操控 3D 对象与逻辑实现，发挥其灵活高效、面向浏览器端开发特性，与课程允许使用的语言范围契合。

**四、团队分工**

本次课程设计小组成员共 2 人，分工如下：

1. **成员 刘洋**：负责整体场景架构设计、提出细节方案整改、撰写文档、以及协调各模块整合工作。
2. **成员 方纪陈**：专注于场景元素建模工作，包括地面、树木、小精灵、白云等模型几何体创建与材质选型，精心雕琢各元素外观细节，确保奇幻风格突出。

**五、进度安排**

1. **第一阶段**：小组内部讨论，确定大概方向、所用语言、技术等
2. **第二阶段**：开始行动，明确各成员间的分工，在完成自己任务同时与小组成员交流。
3. **第三阶段**：将各成员的成果结合，谈论与修改课程设计的效果与细节 。
4. **第四阶段**：整理项目代码、撰写详细课程设计报告，准备展示材料，确保项目可按提交要求完整呈现，按时提交源代码、文字材料、展示报告及相关附件到课程网站与指定个人网站展示。

**六、总结与展望**

本次课程设计，让我们对webgl的认识得到进一步的加深与理解，同时在完成课程设计途中，我们也获得了许多的乐趣，在面对最终的成品后；虽然可能不是很精美但是让我们感觉到所付出的努力得到了回报。