Final Project

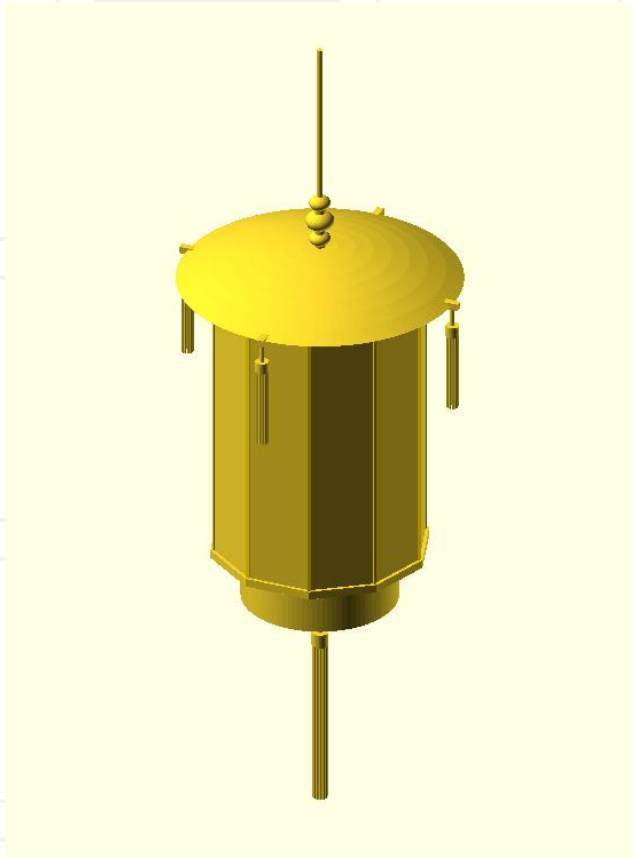
Group 3

2025.7.4

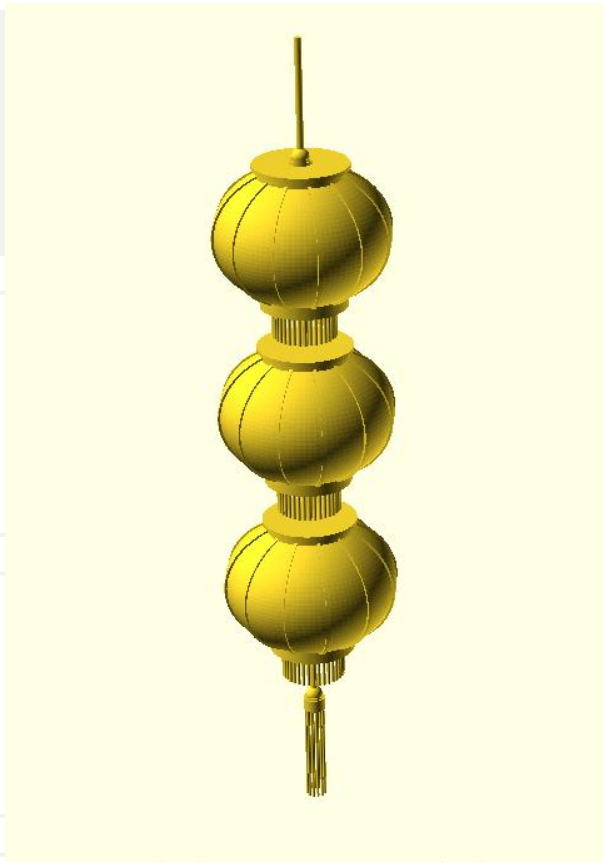
CONTENTS

MODELING

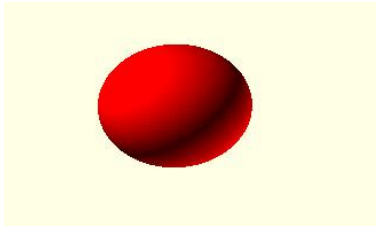
Zeo-Zhiyi HuangE

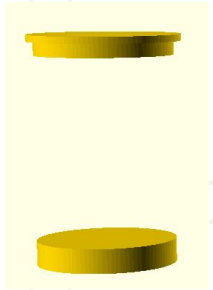



福
福
福



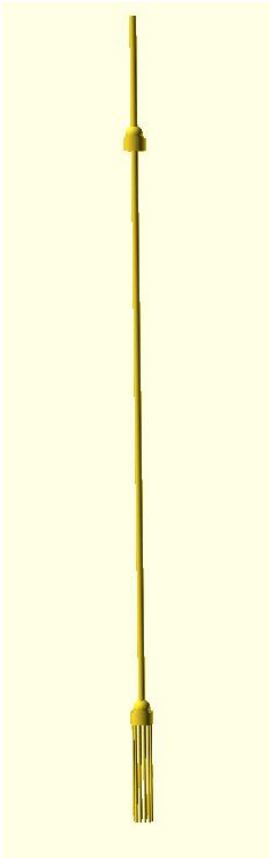
```
$fn=100;

module lantern() {
    color("red")
    scale([1,1,0.8])
    sphere(r=20);

    //上下盖子
    color("gold")
    translate([0,0,16])
    cylinder(h=1, r=11, center=true);

    color("gold")
    translate([0,0,15])
    cylinder(h=3, r=10, center=true);

    color("gold")
    translate([0,0,-15])
    cylinder(h=3.5, r=10, center=true);

    // 筋
      for (r=[0, 30, 60,90,120,150]) {
        color("gold")
        translate([0, 0, 0])
        rotate([90,0,r])
          scale([1,0.8,1])
    cylinder(h=0.8, r=20.5, center=true);
    }

    for (angle=[0:10:360]) {
        rotate([0,0,angle])
        translate([7,0,-24])
        cylinder(h=8, r=0.4);
    }


}
```





```
for (i = [0:2]) {
    translate([0, 0, i*43])
        lantern();
}
```

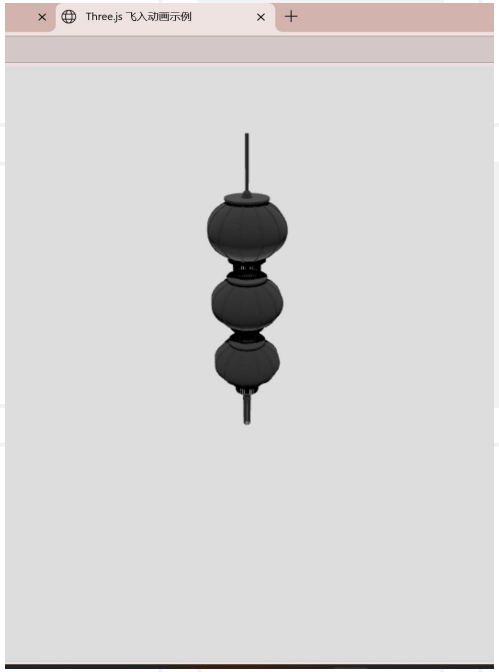
```
module bbb(){
    translate([0,0,-29])
    cylinder(h=160, r=0.8);

    translate([0,0,-30])
    union(){
    sphere(r=2);
    translate([0,0,-2.3])
    cylinder(h=2, r=2.2);
    translate([0,0,-3.8])
    cylinder(h=3, r=2.6);
    }

    for (angle=[0:30:330]) {
        rotate([0,0,angle])
        translate([2.1,0,-57])
        cylinder(h=25, r=0.3);
    }

    translate([0,0,104])
    union(){
    sphere(r=2);
    translate([0,0,-2.3])
    cylinder(h=2, r=2.2);
    translate([0,0,-3.8])
    cylinder(h=3, r=2.6);
    }
    }
```



单击以添加断点

```
<!DOCTYPE html>
<html lang="zh">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Three.js 飞入动画示例</title>
    <style>
        body { margin: 0; overflow: hidden; }
        canvas { display: block; }
    </style>
</head>
<body>
    <script src="https://cdn.jsdelivr.net/npm/three@0.132.2/build/three.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/three@0.132.2/examples/js/loaders/GLTFLoader.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/three@0.132.2/examples/js/controls/OrbitControls.js"></script>

    <script>
        // 场景相机渲染器
        const scene = new THREE.Scene();

        // 轨道控制器
        const controls = new THREE.OrbitControls(camera, renderer.domElement);    l000);
        controls.target.set(0,0,0);
        controls.update();

        // 光源
        const ambientLight = new THREE.AmbientLight(0xffffff, 0.5);
        scene.add(ambientLight);

        const directionalLight = new THREE.DirectionalLight(0xffffff, 0.8);
        directionalLight.position.set(10, 20, 10);
        directionalLight.castShadow = true;
        scene.add(directionalLight);

        // 点光源跟随模型
        const pointLight = new THREE.PointLight(0xffffff, 1, 100);
        scene.add(pointLight);
```

```
C:\Users\86138>cd C:\Users\86138\Desktop\final

C:\Users\86138\Desktop\final> http-server
Starting up http-server, serving ./

http-server version: 14.1.1

http-server settings:
CORS: disabled
Cache: 3600 seconds
Connection Timeout: 120 seconds
Directory Listings: visible
AutoIndex: visible
Serve GZIP Files: false
Serve Brotli Files: false
Default File Extension: none

Available on:
  http://192.168.1.13:8080
  http://127.0.0.1:8080
Hit CTRL-C to stop the server
```



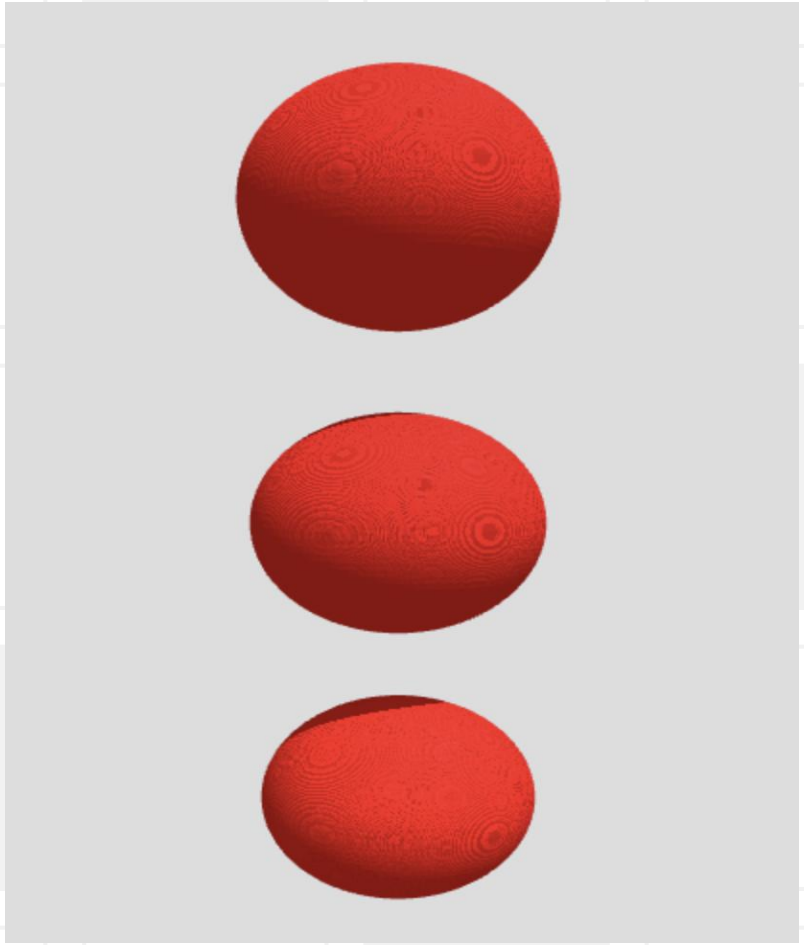
2

Render

Body part, Frame part, Tassel part

Lisa-Ziqi Ma

Body part



1 RGB

rgb: (0.729, 0.118, 0.118)
opacity: 0.93

2 TEXTURE

roughness: 0.8
metalness: 0.0

3 LIGTH

emissive: (0.3, 0.0, 0.0)
emissiveIntensity: 0.4

Frame part

1 RGB

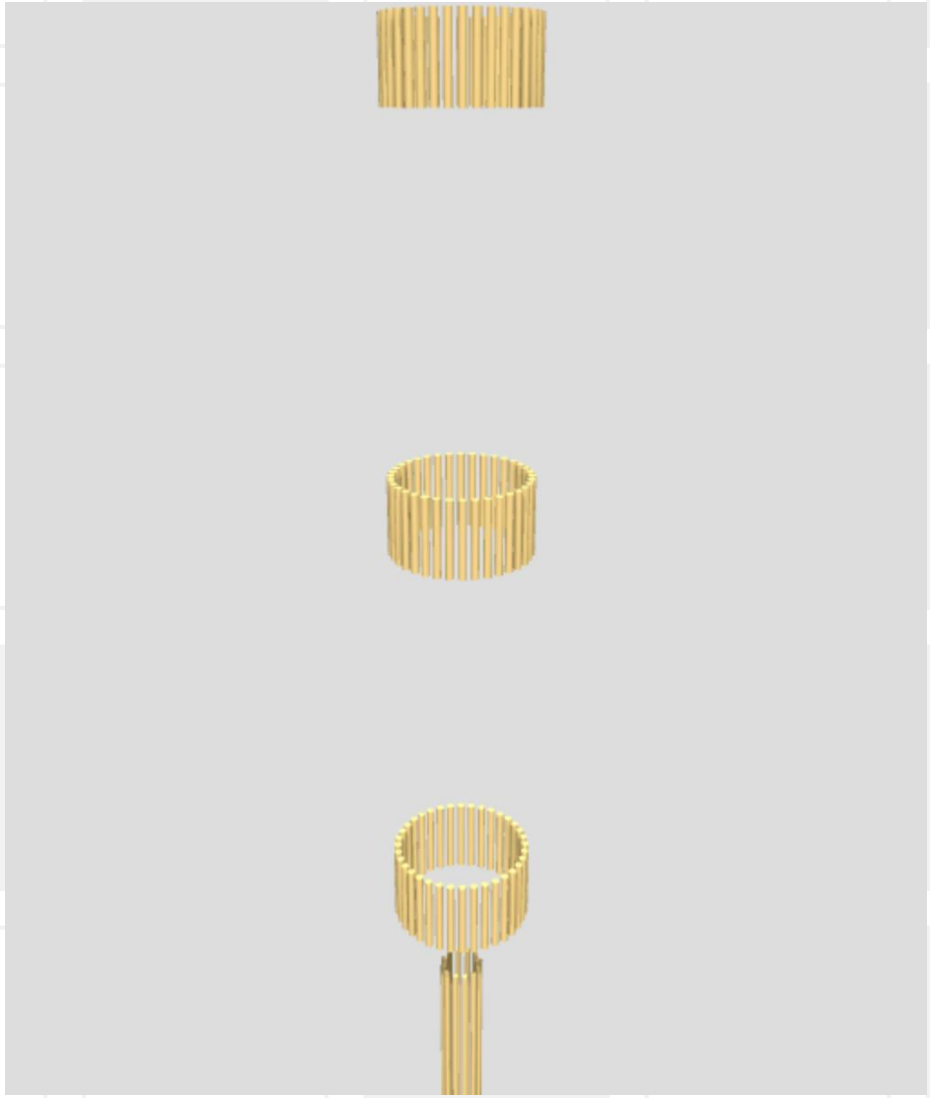
rgb: (0.71, 0.59, 0.0)

2 TEXTURE

roughness: 0.5

metalness: 0.3

Tessel part

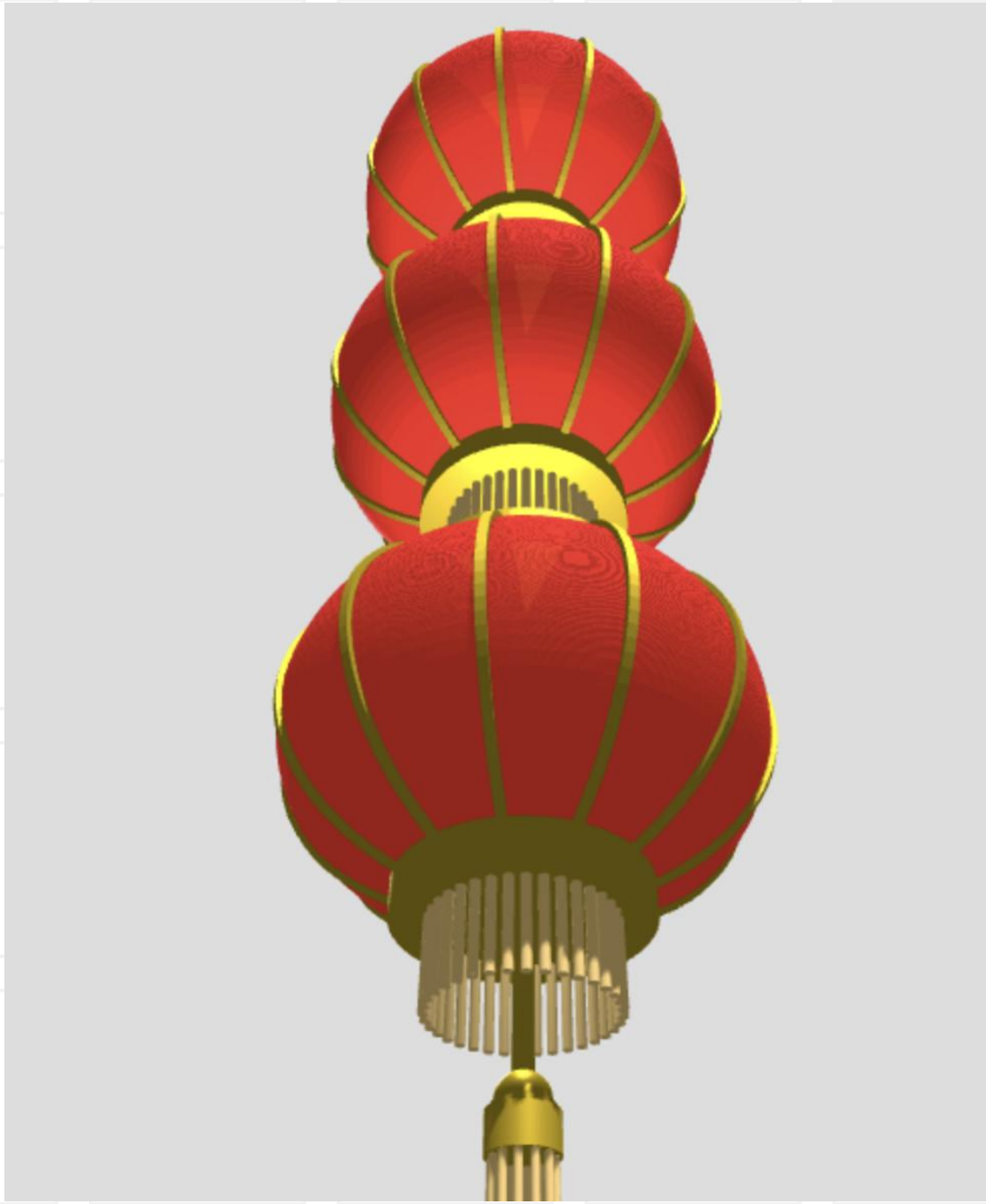


1 RGB

rgb: (0.71, 0.59, 0.35)

2 TEXTURE

roughness: 0.5
metalness: 0.0



3

Animation production part

The animation part about lanterns was produced here

Eric-Qirui Zhai

Design goal

Elasticly fly in from the right side (x=20) to the center of the scene (x=0),

Gently float up and down during movement,

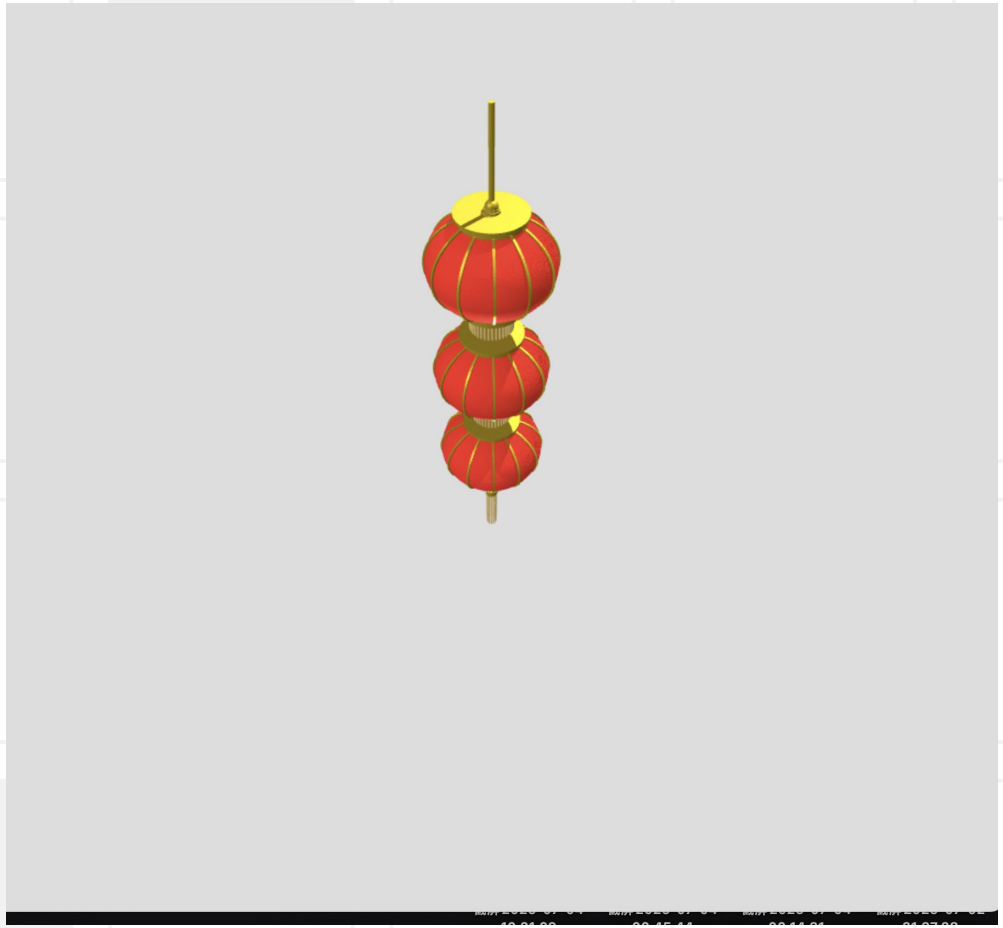
Continuously rotate at a slow pace,

Gradually slow their rotation speed when nearing the end of the flight,

Maintain floating and rotating states after completing the animation,

Have a point light source that follows the lanterns' movement,

Automatically adapt to browser window size changes



Code

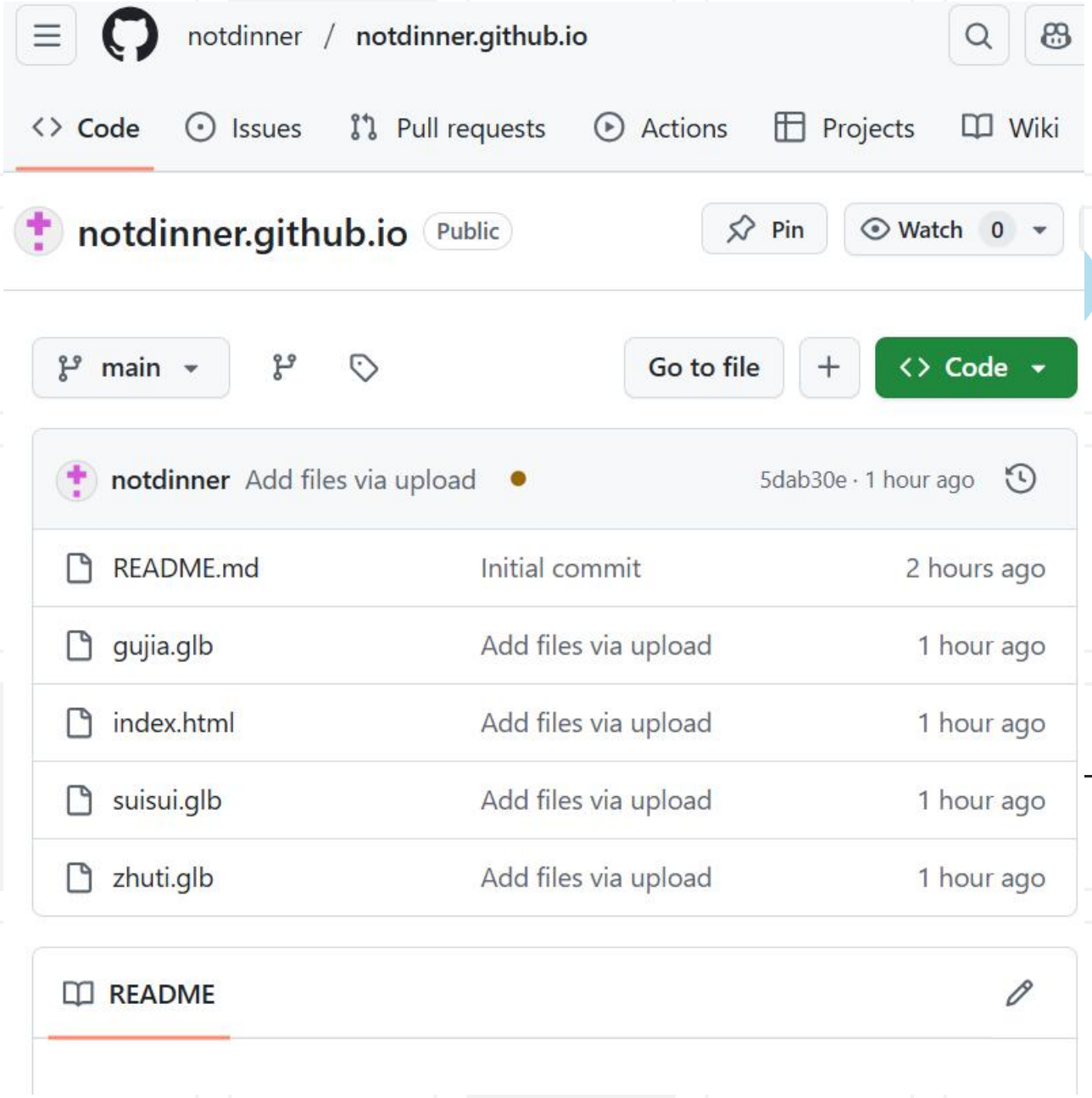
```
let isModelLoaded = false;
const animationState = {
    flyingProgress: 0,
    flyingDuration: 240,
    rotationSpeed: 0.01,
    floatIntensity: 0.3
};

function easeOutElastic(t){
    const p = 0.3;
    return Math.pow(2, -10 * t) * Math.sin((t - p / 4) * (2 * Math.PI) / p) + 1;
}

function animate(){
    requestAnimationFrame(animate);

    if(isModelLoaded && lanternGroup.children.length > 0){
        if(animationState.flyingProgress < animationState.flyingDuration){
            animationState.flyingProgress++;
            const progress = animationState.flyingProgress / animationState.flyingDuration;
            const easedProgress = easeOutElastic(progress);

            lanternGroup.position.x = 20 * (1 - easedProgress);
            lanternGroup.position.y = Math.sin(Date.now() * 0.003) * animationState.floatIntensity;
```

PLEASE ADD TITLE

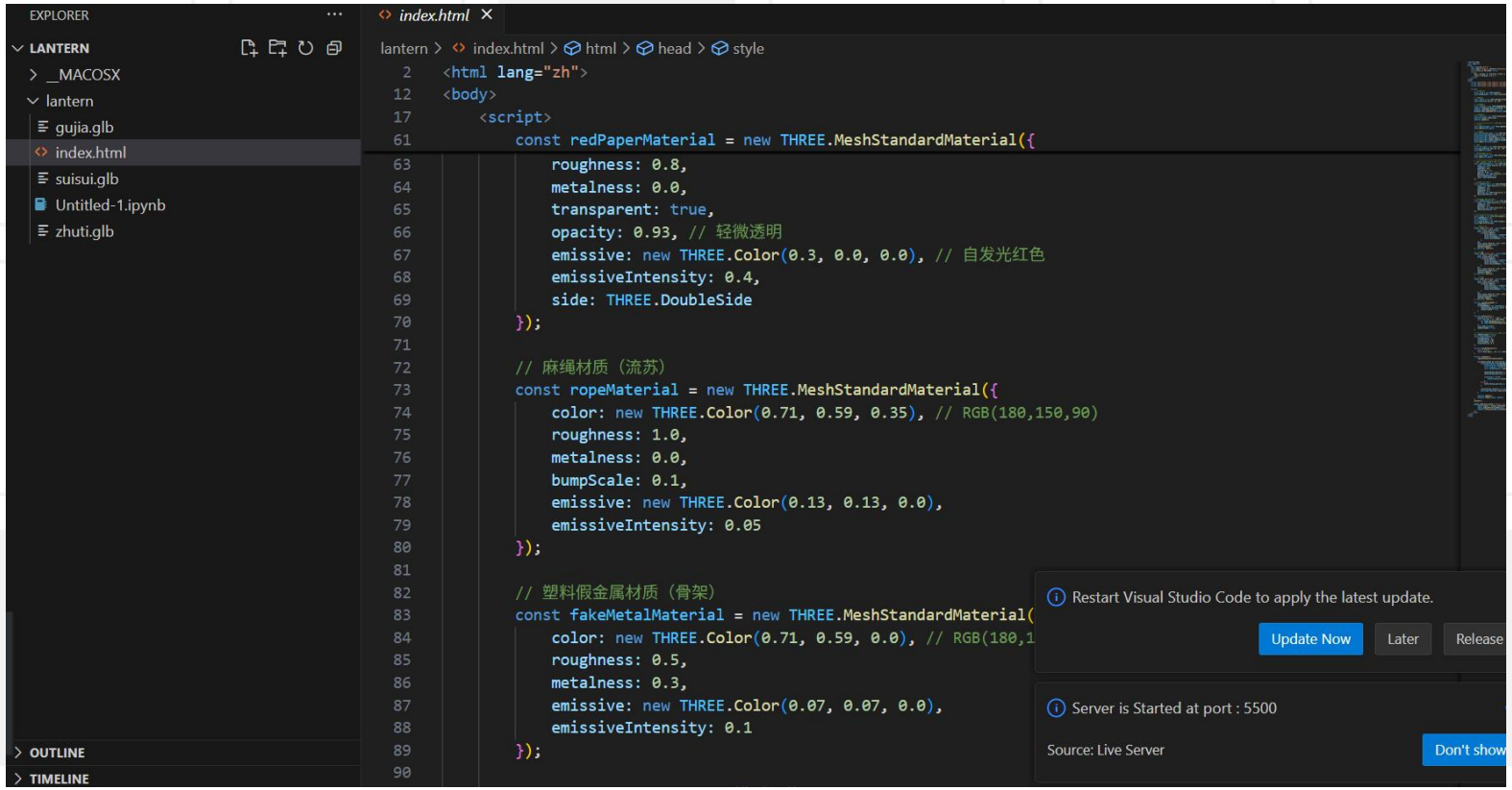
```
        if(isModelLoaded && lanternGroup.children.length > 0){
            if(animationState.flyingProgress < animationState.flyingDuration){
                animationState.flyingProgress++;
                const progress = animationState.flyingProgress / animationState.flyingDuration;
                const easedProgress = easeOutElastic(progress);

                lanternGroup.position.x = 20 * (1 - easedProgress);
                lanternGroup.position.y = Math.sin(Date.now() * 0.003) * animationState.floatIntensity;

                if(progress > 0.7){
                    animationState.rotationSpeed = 0.005;
                }
            } else {
                lanternGroup.position.y = Math.sin(Date.now() * 0.003) * animationState.floatIntensity;
            }

            lanternGroup.rotation.y += animationState.rotationSpeed;
            pointLight.position.copy(lanternGroup.position);
        }

        controls.update();
        renderer.render(scene, camera);
    }
    animate();

    window.addEventListener('resize', () => {
        camera.aspect = window.innerWidth / window.innerHeight;
        camera.updateProjectionMatrix();
```

Team Report and GitHub Website Production

Norton_Yuehao Liu



GitHub website



teaching assistant's help

P

My group members' help

P


THANK YOU