Company Name
Year / Quarter / Month

# Project kickoff presentation

Use this space for legal copy. Explain if the information is confidential, who can read this deck, etc.

Company Name

# Contents

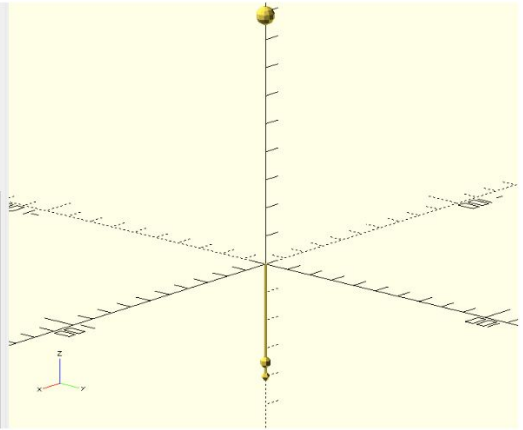
# 3D Model Designing

Creating a customizable 3D model of a traditional Chinese paper lantern with four three components:

1. Lantern Body
2. Fixing Ring System
3. Hanging System
4. Tassel System

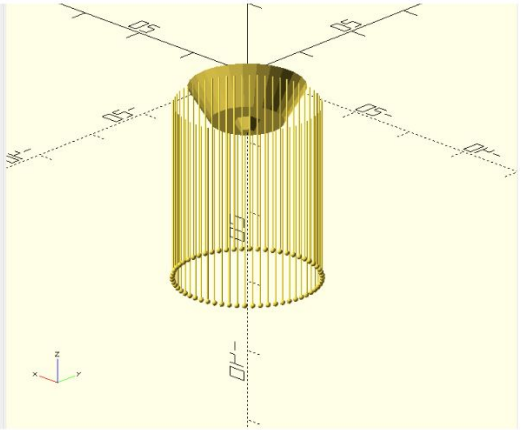
## 3) Hanging system

```
/* ================ Hanging System ================ */
// Top hook connection point
translate([0, 0, frame_height + 8.65])
sphere(d=bar_thickness * 5.5);

// Bottom hanging rod
translate([0,0,-20]) cylinder(h=40, r=0.5, center=true);
translate([0,0,-35]) sphere(2);
translate([0,0,-40]) sphere(1.5);
```



## 4) Tassel system

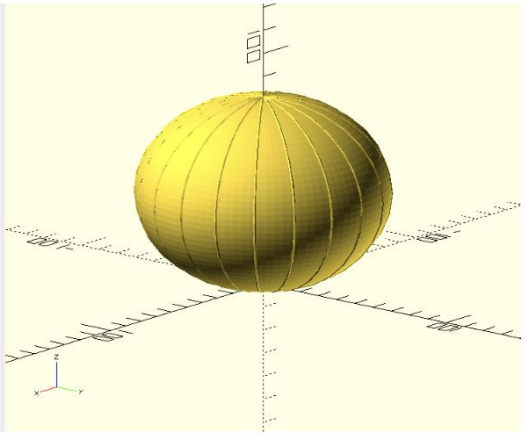
```
/* ================ Tassel System ================ */
// Tassel connector
translate([0, 0, -7]) {
    sphere(d=bar_thickness * 3);  // Central connection point
    cylinder(h=connector_height, r1=5, r2=8);  // Tassel support
}

// Create tassel array
for (i = [0:tassel_count-1]) {
    angle = i * (360 / tassel_count);
    distance = 10;  // Tassel starting distance from center

    translate([distance * cos(angle), distance * sin(angle), 0.5 -
    connector_height]) {
        // Tassel thread
        translate([0, 0, -tassel_length])
        cylinder(h=tassel_length, d=tassel_thickness, $fn=8);

        // Tassel end decorative bead
        translate([0, 0, -tassel_length])
        sphere(d=end_bead_diameter-2.3, $fn=20);
    }
}
```



## 2. Component Implementation

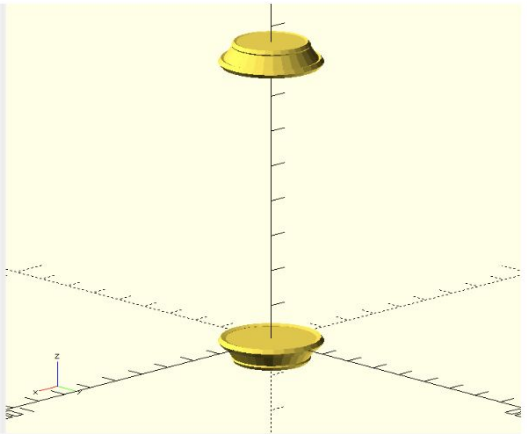
### 1) Lantern body

```
/* ================ Lantern Main Structure ================ */
// 1. Vertical elliptical frames
for (i = [0:vertical_count-1]) {
    angle = i * (360 / vertical_count);
    rotate([0, 0, angle])
    translate([0, 0, frame_height/2])
    rotate([90, 0, 0])
    scale([1, frame_height/(ellipse_major*3), 1])
    rotate_extrude(angle=360)
    translate([ellipse_minor, 0])
    circle(d=bar_thickness);
}


// 2. Ellipsoid skin shell
z_scale = frame_height/(ellipse_major*3);  // Z-axis scaling factor
translate([0, 0, frame_height/2])
difference() {
    // Outer surface
    scale([ellipse_minor, ellipse_minor, ellipse_minor * z_scale])
    sphere(1, $fn=96);

    // Inner surface (hollow out to form shell)
    scale([
        ellipse_minor - skin_thickness,
        ellipse_minor - skin_thickness,
        (ellipse_minor - skin_thickness) * z_scale
    ])
    sphere(1, $fn=96);
}


```



### 2) Fixing ring system

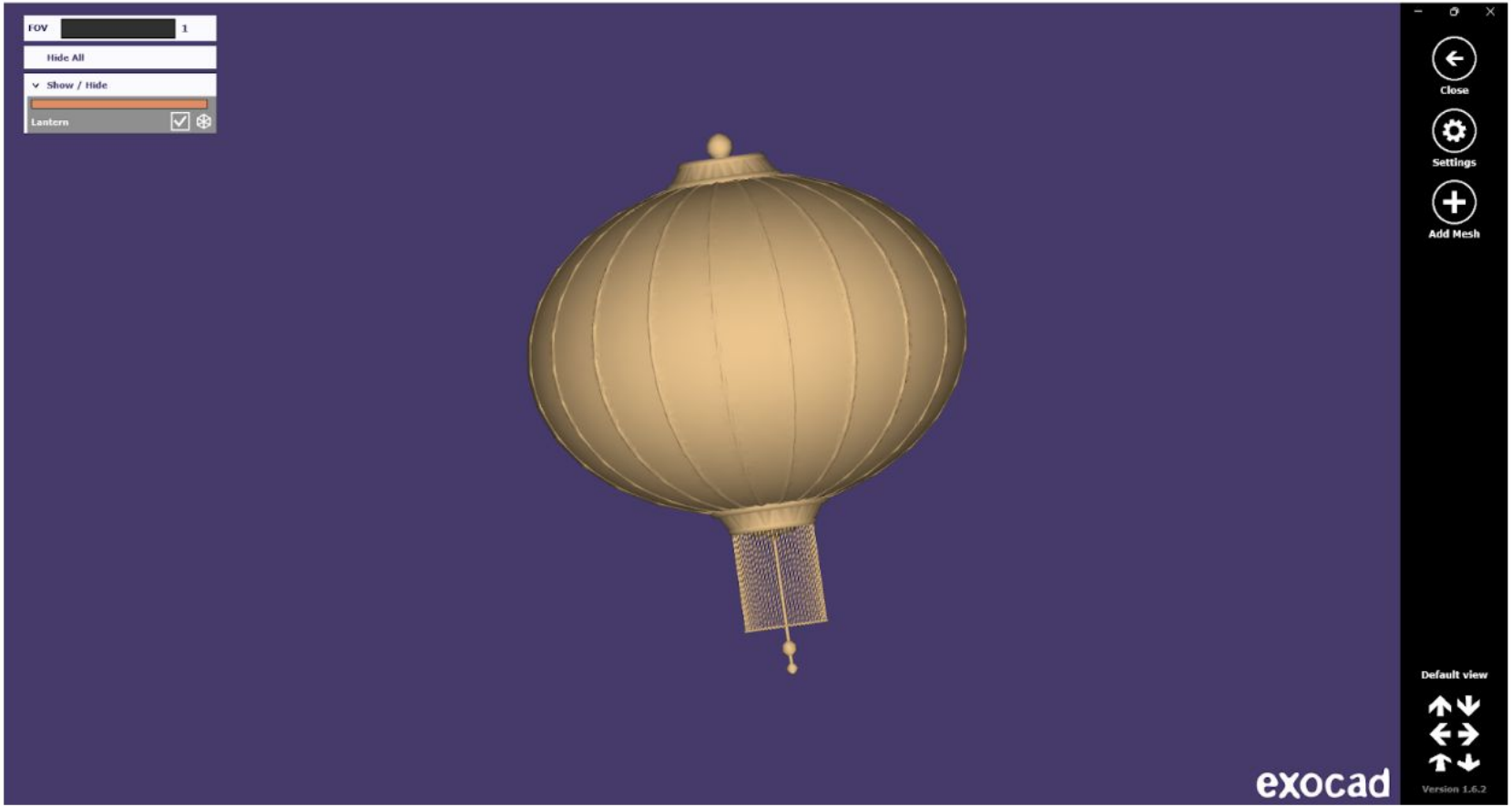
```
/* ================ Lantern Fixing Rings ================ */
// Top fixing ring system
translate([0, 0, frame_height]) {
    rotate_extrude(angle=360)
    translate([ellipse_minor * 0.25, 0])
    circle(d=bar_thickness * 2);

    translate([0,0,2.5]) cylinder(h=6.25, r1=14.25, r2=10, center=true);

    translate([0, 0, 4.85])
    rotate_extrude(angle=360)
    translate([ellipse_minor * 0.185, 0])
    circle(d=bar_thickness * 1.75);
}


// Bottom fixing ring system
translate([0, 0, 0]) {
    rotate_extrude(angle=360)
    translate([ellipse_minor * 0.25, 0])
    circle(d=bar_thickness * 2);

    translate([0,0,-2.5]) cylinder(h=6.25, r1=10, r2=14, center=true);

    translate([0, 0, -4.85])
    rotate_extrude(angle=360)
    translate([ellipse_minor * 0.185, 0])
    circle(d=bar_thickness * 1.75);
}


```



# Main Parameter Table

| Parameter Name | Function Description | Substitution Value |
|---|---|---|
| frame_height | Lantern height | 80 |
| ellipse_major | Major axis radius of ellipse | 35 |
| ellipse_minor | Minor axis radius of ellipse | 54 |
| vertical_count | Number of vertical frames | 20 |
| bar_thickness | Frame bar thickness | 1.2 |
| skin_thickness | Skin thickness | 0.4 |
| tassel_count | Number of tassels | 60 |
| tassel_length | Tassel length | 25 |
| tassel_thickness | Tassel thread thickness | 0.2 |
| end_bead_diameter | End bead diameter | 3 |
| connector_height | Tassel connector height | 5 |

# 4. Model Rendering Image



# Animation

1. Initial Approach- Github repository -> Online html editor
2. Second Approach- VS code -> access the file locally

1. Giving the code access to the file of the lantern model

```
const loader = new THREE.STLLoader();
loader.load(
    'models/Lantern.stl',
    function (geometry) {
        const mesh = new THREE.Mesh(geometry, material)
        scene.add(mesh)
    },
    (xhr) => {
```

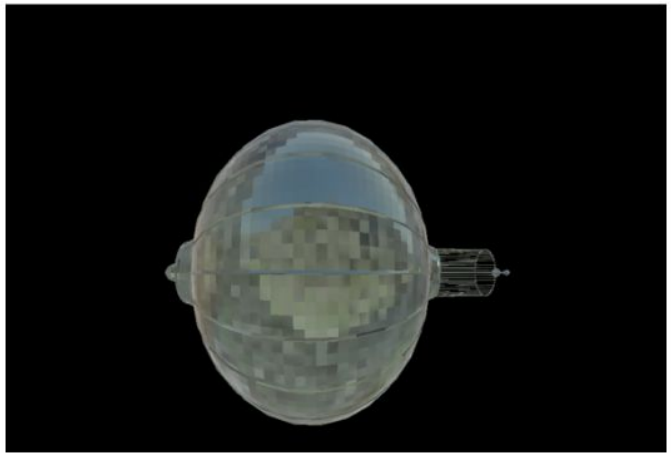
## 2. Changing the camera position to a further view

```
function render() {


    const timer = Date.now()*0.0005;
    camera.position.x = Math.cos(timer)*4;
    camera.position.z = Math.sin(timer)*4;

    renderer.render(scene, camera);
}
```

```
function render() {


    const timer = Date.now()*0.0005;
    camera.position.x = Math.cos(timer)*150;
    camera.position.z = Math.sin(timer)*150;

    renderer.render(scene, camera);
}
```

3. Flipping the 3D model

```
const loader = new THREE.STLLoader();
loader.load(
    'models/Lantern.stl',
    function (geometry) {
        const mesh = new THREE.Mesh(geometry, material)
        mesh.rotation.x = -Math.PI / 2;
        scene.add(mesh)
    },
```



# Material

```
envTexture.mapping = THREE.CubeReflectionMapping;

let material;

material = new THREE.MeshPhysicalMaterial({
    color: 0xFF0000,
    envMap: envTexture,
    metalness: 1,
    roughness: 1,
    opacity: 1.0,
    transparent: true,
    transmission: 0,
    clearcoat: 1.0,
    clearcoatRoughness: 0.3
});
```