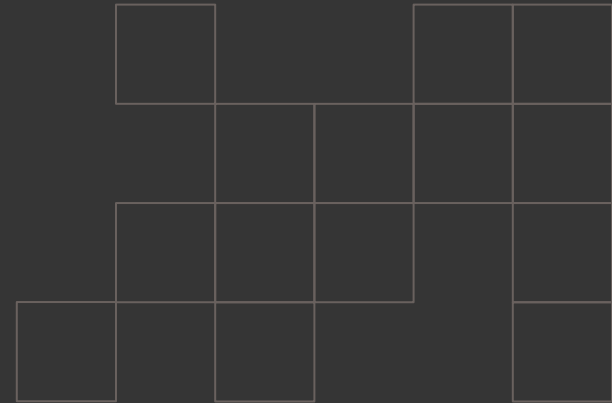


Company Name
Year / Quarter / Month

Project kickoff presentation

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



Contents

1. 3D Modeling
2. Shading/Texture
3. Animation
4. Github.io

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

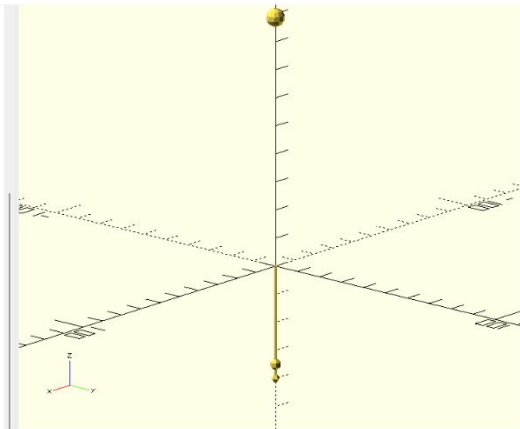
Tassel System

3) Hanging system

```

20
21 /* ===== Hanging System ===== */
22 // Top hook connection point
23 translate([0, 0, frame_height + 8.65])
24 sphere(d=bar_thickness * 5.5);
25
26 // Bottom hanging rod
27 translate([0,0,-20]) cylinder(h=40, r=0.5, center=true);
28 translate([0,0,-35]) sphere(2);
29 translate([0,0,-40]) sphere(1.5);
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

```

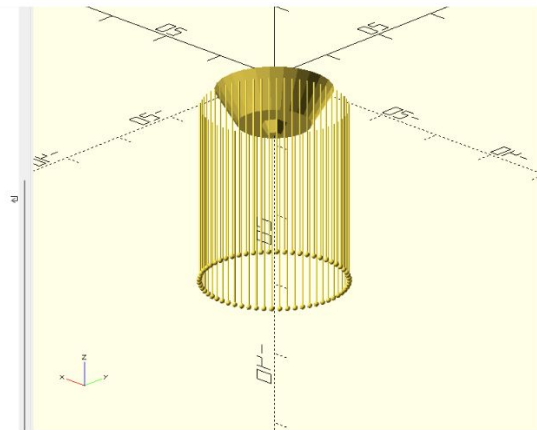


4) Tassel system

```

18
19 /* ===== Tassel System ===== */
20 // Tassel connector
21 translate([0, 0, -7]) {
22     sphere(d=bar_thickness * 3); // Central connection point
23     cylinder(h=connector_height, r1=5, r2=0); // Tassel support
24 }
25
26
27 // Create tassel array
28 for (i = [0:tassel_count-1]) {
29     angle = i * (360 / tassel_count);
30     distance = 10; // Tassel starting distance from center
31
32     translate([distance * cos(angle), distance * sin(angle), 0.5 *
33         connector_height]) {
34         // Tassel thread
35         translate([0, 0, -tassel_length])
36         cylinder(h=tassel_length, d=tassel_thickness, $fn=8);
37
38         // Tassel end decorative bead
39         translate([0, 0, -tassel_length])
40         sphere(d=end_bead_diameter-2.3, $fn=20);
41     }
42 }
43
44
45
46
47
48

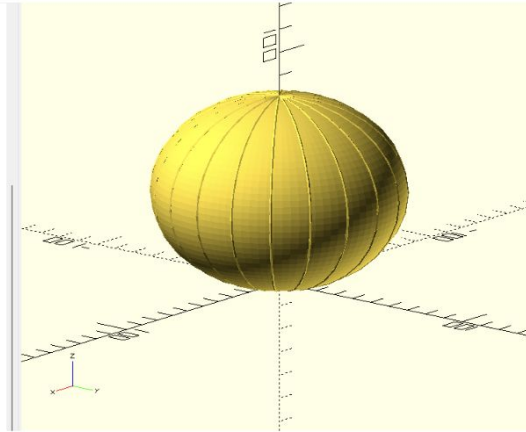
```



2. Component Implementation

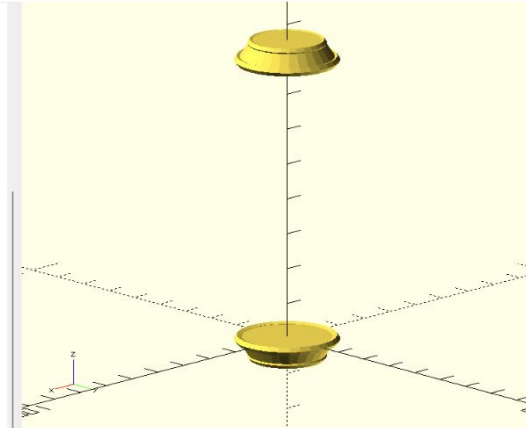
1) Lantern body

```
19 /* ===== Lantern Main Structure ===== */
20 // 1. Vertical elliptical frames
21 for (i = [0:vertical_count-1]) {
22     angle = i * (360 / vertical_count);
23     rotate([0, 0, angle])
24     translate([0, 0, frame_height/2])
25     rotate([90, 0, 0])
26     scale([1, frame_height/(ellipse_major*3), 1])
27     rotate_extrude(angle=360)
28     translate([ellipse_minor, 0])
29     circle(d=bar_thickness);
30 }
31
32
33 // 2. Ellipsoid skin shell
34 z_scale = frame_height/(ellipse_major*3); // Z-axis scaling factor
35 translate([0, 0, frame_height/2])
36 difference() {
37     // Outer surface
38     scale([ellipse_minor, ellipse_minor, ellipse_minor * z_scale])
39     sphere(1, $fn=96);
40
41     // Inner surface (hollow out to form shell)
42     scale([
43         ellipse_minor - skin_thickness,
44         ellipse_minor - skin_thickness,
45         (ellipse_minor - skin_thickness) * z_scale
46     ])
47     sphere(1, $fn=96);
48 }
49
50
```



2) Fixing ring system

```
20 /* ===== Lantern Fixing Rings ===== */
21 // Top fixing ring system
22 translate([0, 0, frame_height]) {
23     rotate_extrude(angle=360)
24     translate([ellipse_minor * 0.25, 0])
25     circle(d=bar_thickness * 2);
26
27     translate([0,0,2.5]) cylinder(h=6.25, r1=14.25, r2=10, center=true);
28
29     translate([0, 0, 4.85])
30     rotate_extrude(angle=360)
31     translate([ellipse_minor * 0.185, 0])
32     circle(d=bar_thickness * 1.75);
33 }
34
35
36 // Bottom fixing ring system
37 translate([0, 0, 0]) {
38     rotate_extrude(angle=360)
39     translate([ellipse_minor * 0.25, 0])
40     circle(d=bar_thickness * 2);
41
42     translate([0,0,-2.5]) cylinder(h=6.25, r1=10, r2=14, center=true);
43
44     translate([0, 0, -4.85])
45     rotate_extrude(angle=360)
46     translate([ellipse_minor * 0.185, 0])
47     circle(d=bar_thickness * 1.75);
48 }
49
50
51
```



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) => {
```

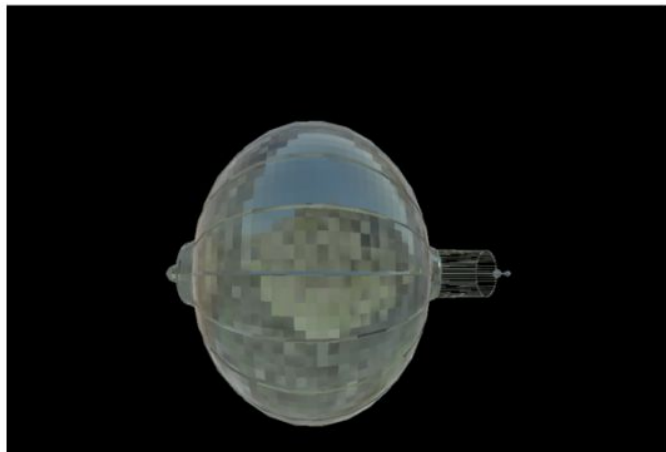

2. Changing the camera position to a further view

```
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
```

```
});
```



Github page

How it started

1. First we created a github page with instructions

2. second we started to upload and code the web page using html

3. Finally we start to test and work out bugs with the page

```
1  # RIDER-WQ.github.io
2  project files
3
4  [Animation](https://jack10079.github.io)
5
6  [pdf](https://github.com/RIDER-WQ/RIDER-WQ.github.io/blob/main/assets/files/Final%20Project%20PPT.pdf)
7
8  [scad](Chinese Lantern.scad)
9
10
11 [stl](Chinese Lantern.stl)
12
13 <video width="320" height="240" autoplay>
14   <source src="model.mp4" type="video/mp4">
15   <source src="model.ogg" type="video/ogg">
16   Your browser does not support the video tag.
17 </video>
18
```

```
7  d. create models/your-model.stl
8  e. edit fun.html, remove https://bytes.usc.edu/~saty/... in the loader.load() call, instead put in models/your-model.stl
9  f. load fun.html into your browser - ta-da!!
10 -->
11
12
13 <!DOCTYPE html>
14 <html lang="en">
15   <head>
16     <title>STL loader</title>
17     <meta charset="utf-8">
18     <meta name="viewport" content="width=device-width, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">
19     <script src="https://unpkg.com/three@0.146.0/build/three.min.js" crossorigin="anonymous" refererpolicy="no-referrer"></script>
20     <script src="https://unpkg.com/three@0.146.0/examples/js/loaders/STLLoader.js" crossorigin="anonymous" refererpolicy="no-referrer"></script>
21     <script src="https://unpkg.com/three@0.146.0/examples/js/controls/OrbitControls.js" crossorigin="anonymous" refererpolicy="no-referrer"></script>
22   </head>
23   <body>
24
25
26     <script>
27       const scene = new THREE.Scene();
28       scene.add(new THREE.AxesHelper(2)); // cmt this out if you like...
29
30       const light = new THREE.SpotLight();
31       light.position.set(20, 20, 20);
32       scene.add(light);
33
```



Thank you
for listening