Final Report(Render)

In our final project, we planned to create a realistic render of 4 pieces of unique Chinese pastries sitting on a wooden plate, and that plate is sitting on a darker wooden base. In order to realize that plan, we utilized three major technologies — OpenScad, three.js API, and traditional web technologies — to tackle the three major components of this project respectively — modeling, shading and animation, and site-building.

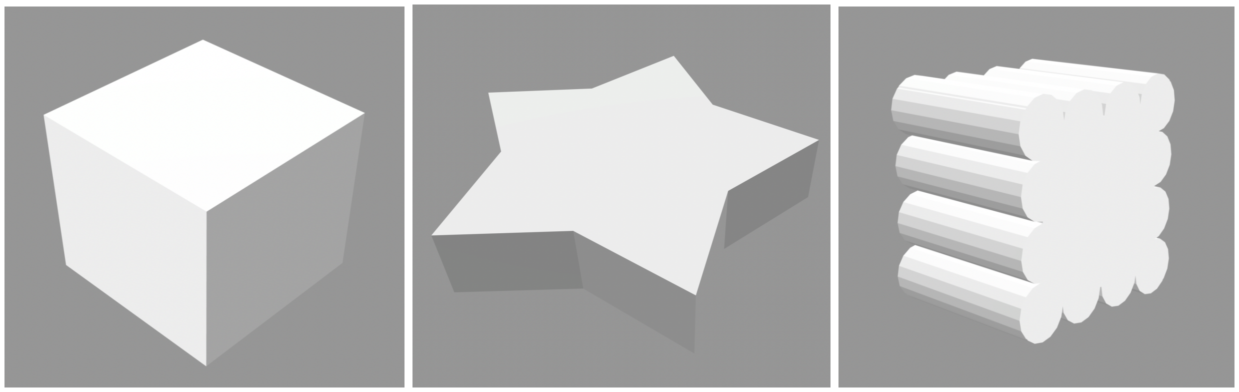

Figure 1-3

First of all, the modeling part. Since we are working on a tight schedule, we keep the geometry of all of our models to be relatively simple: a cube model for the red dates cake, a star-shaped model for the sweet osmanthus cake, and a more complicated square-shaped model for the mooncake (Figure 1-3).

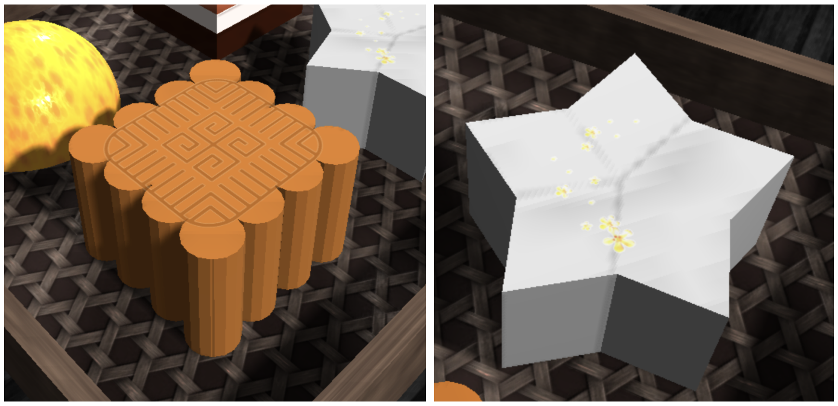

Figure 4-5

The next section is shading. This part is extremely challenging, and our group has met with so many problems that we have to try many different approaches and completely redo our project several times. The current final result is achieved mainly by 2 types of shading techniques: texture mapping and MeshPhysicalMaterial provided by three.js API. The mooncake and sweet osmanthus cake are made with similar techniques (Figure 4-5). Both of them were first loaded by the imported STL loader. And their materials are declared by MeshPhysicalMaterial. By manipulating the specific values within the MeshPhysicalMaterial, including color, roughness, and reflectivity, we managed to give unique textures to each of the renderings. However, by the MeshPhysicalMaterial itself, the mesh is still a bit bland. So we decided to add additional layers of texture maps to the respective meshes. To overcome the difficulty of STL files not supporting texture mapping, we created new basic plane geometries so that the texture map can be attached.

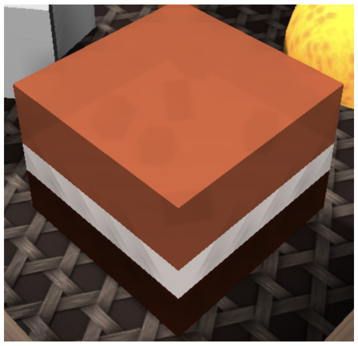

Figure 6

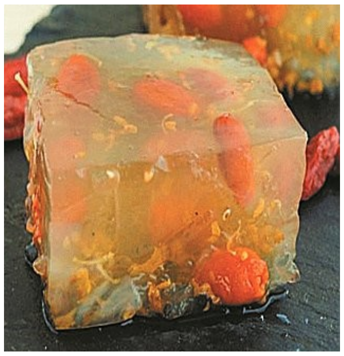

Figure 7

Red dates cake is composed of three separate STL cube models and each of them has a slightly different MeshPhysicalMaterial (Figure 6). The top layer has its transmission value set to 0.1, so the material can be semi-transparent. Moreover, to add more details, we drew inspiration from another type of Chinese pastry that has chunks of fruit inside it (Figure 7). As a result, we made multiple tiny cube objects and added them into the transparent layer, so the viewers can see the little details. Last but not least, the Su Pi Bing render was made with a different approach. The base model this time is a basic sphere geometry from three.js, which is not an STL file. In other words, we can apply texture maps. First, we applied the base color map, so the sphere has a base yellowish color. And then we applied 3 additional maps — normal, roughness and ambient occlusion map — which makes the surface texture more detailed and real. And here is the result (Figure 8).

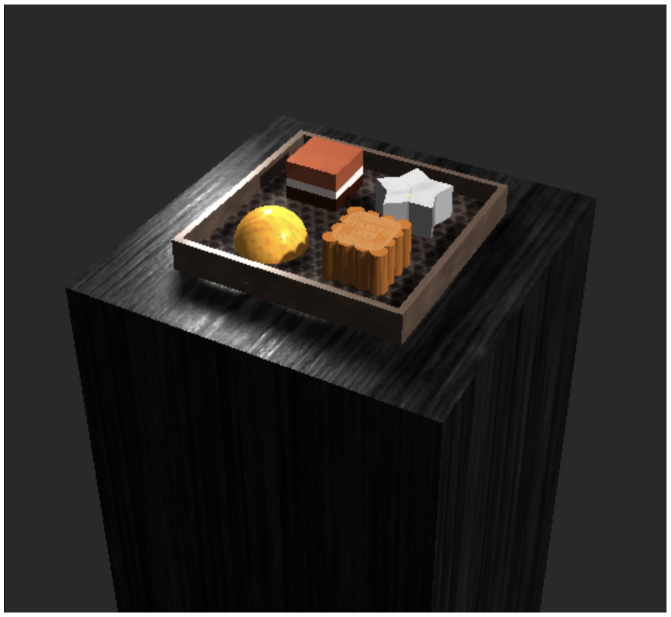
Figure 8

For the animation part, we took advantage of the built-in OrbitControls object. Simply importing and initializing it grants the viewer the ability to drag the screen and look around the screen. We decided to disable the panning functionality, however, because we want to keep our viewers' attention on the rendered objects. Furthermore, to let the camera spin by itself when the viewer is not dragging the window, we set the autoRotate attribute to be true on default and set it to false when the event listener detects the 'start' signal. On the other hand, when the event listener gets the 'end' signal, it will start a one-second countdown to enable auto-spinning again.

Lastly, in the site building stage we bring everything together by HTML, CSS and JS.