

Chinese Biscuit

Zhuohao Chen (Cisco)
Dishi Yuan (Jolute)
Yibing Chen (Cookie)
Haoyu Qin (Sonny)





Team

A good project is inseparable from a
good team.

Team Member-Round



Zhuohao Chen(Cisco)
Modeling, Preparation work
for shading



Dishi Yuan(Jolute)
Plate Modeling, Shading



Yibing Chen(Cookie)
Animation



Haoyu Qin(Sonny)
HTML





Introduction

Our biscuit is inspired by the picture on the left. It is full of the festive atmosphere of the Spring Festival and is composed of biscuit layers, cream layers, and text layers.

Table of contents

01

Modeling

The construction process of biscuit models.

02

Shading

Give the geometry the right pattern and color.

03

Animation

Make it move!

04

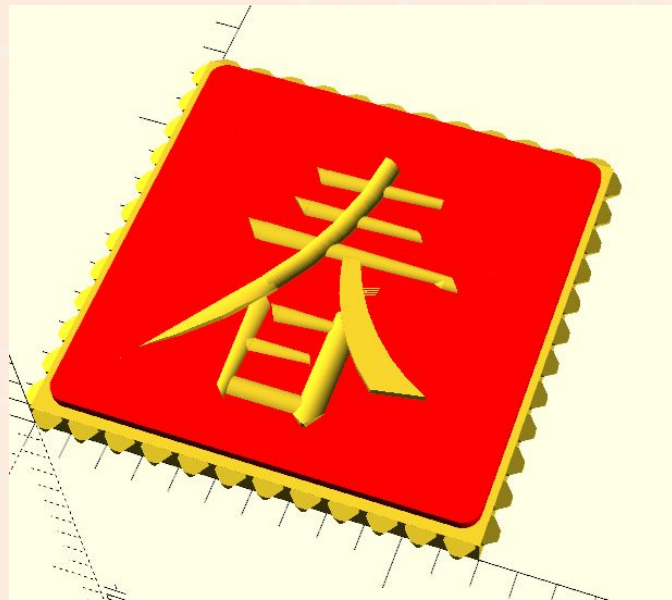
HTML

You can describe the topic of the section here

01

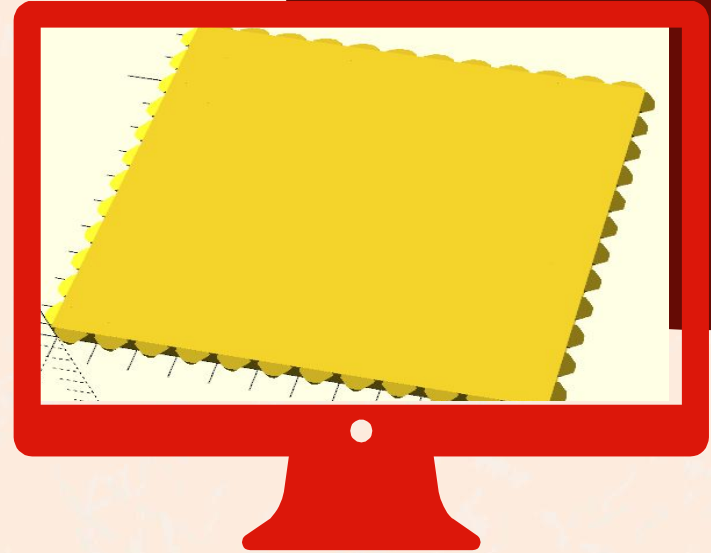
Modeling

how to make a biscuit with the word “Chun” using OpenSCAD.



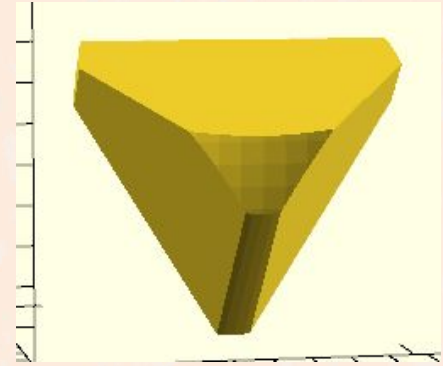
Biscuit Layers

The cookie layer is made up of a square base and many cookie cones

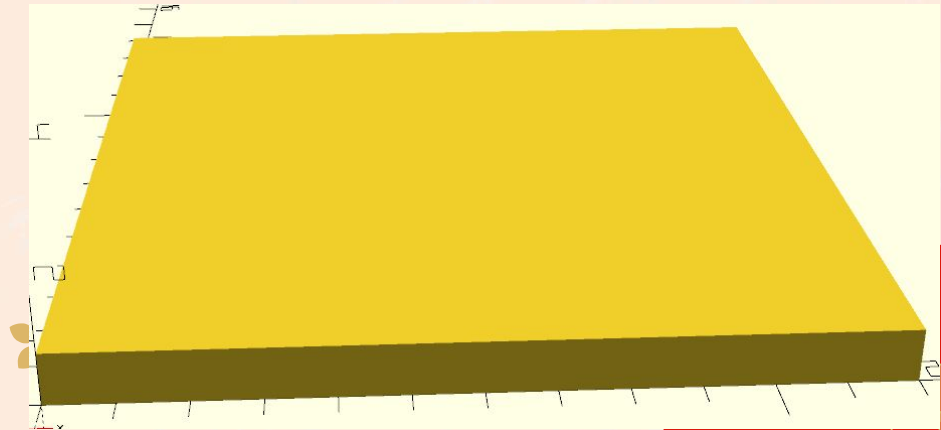




Biscuit Cone



Biscuit Body





```
9 module Interlayer(){  
0  
1   translate([0.8,0.8,0.8])  
2   color(c=[1,0,0,1])  
3  
4   minkowski()  
5   {  
6     cube([cookieLength-1.5,cookieLength-  
7       1.5,0.3]);  
8     cylinder(r=0.5,h=0.01);  
9   }  
0 }
```

the biscuit's middle layer



text layers

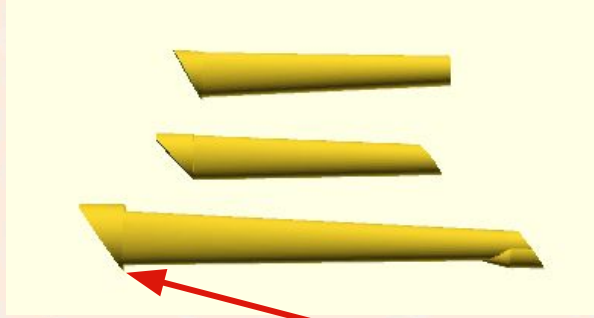
The text layer is composed of
the Chinese character ‘春’



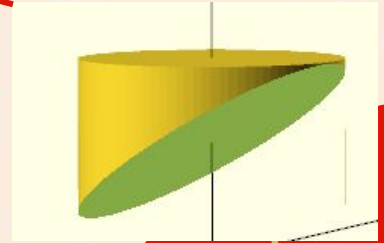
Reference

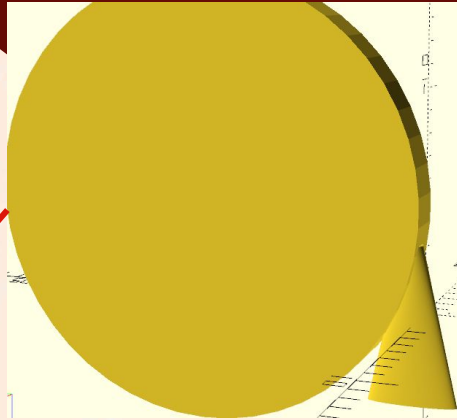
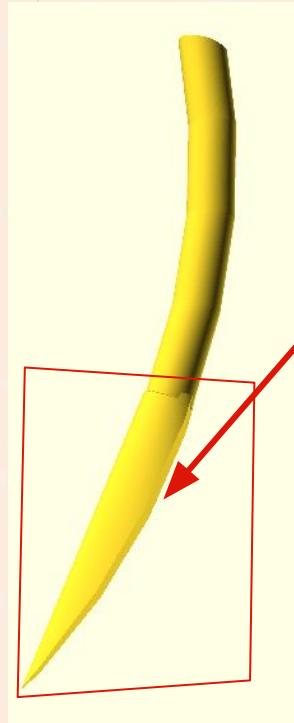


The design of the character '春' was inspired by the '春' character of Yan Zhenqing, a great calligrapher in ancient times



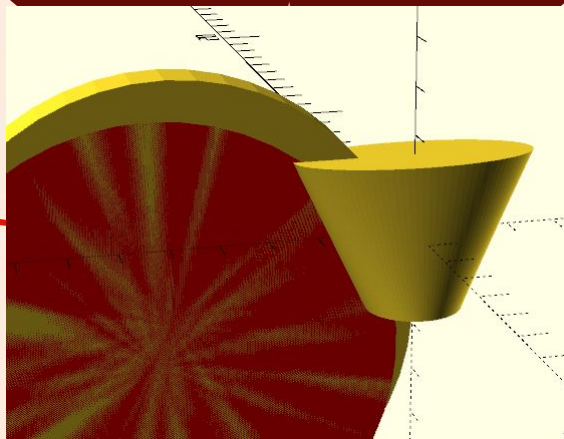
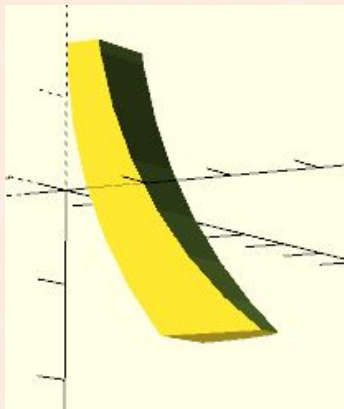
The horizontal stroke in Chinese characters, pronounced as 'Heng'





The left-falling stroke in Chinese characters,
pronounced as 'Pie'



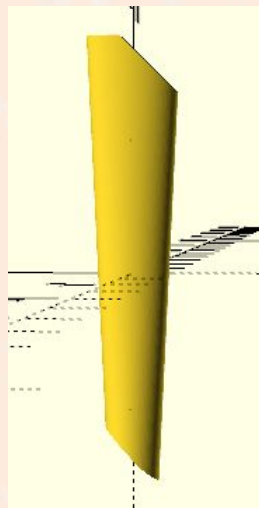


The right-falling stroke in Chinese characters, pronounced as 'Na'

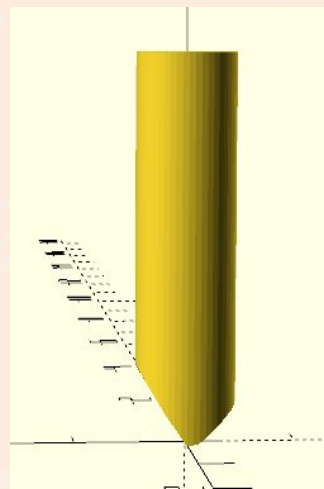




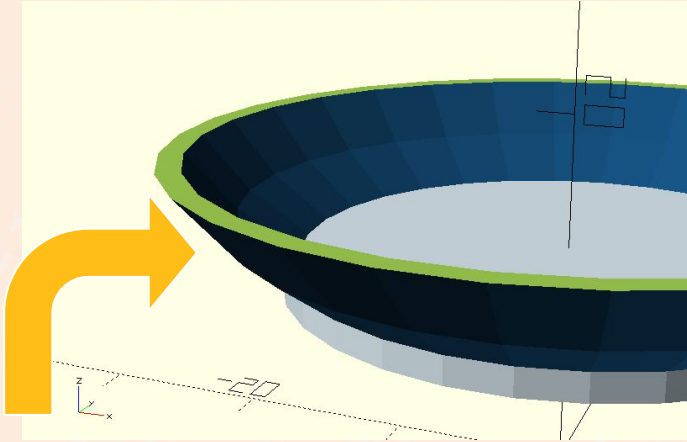
The vertical stroke in Chinese characters,
pronounced as 'Shu'



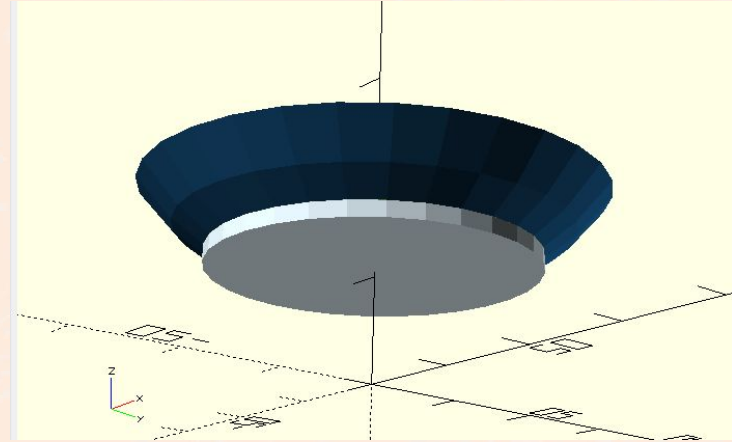
The horizontal
turning



★ Modeling-Plate ★



Curved edges



Fill the hole by using a disc
at the bottom



★ Modeling-Plate ★



BigPlate.scad* - OpenSCAD

文件(F) 编辑(E) 模型(D) 视图(V) Window 帮助(H)

编辑器

```
1 difference()
2 {
3   translate([0,0,35])
4   {
5     difference()
6     {
7       color([20/255,74/255,116/255],alpha=1)sphere(28);
8       color(c=[20/255,74/255,116/255],alpha=1) sphere(27);
9       translate([-50,-50,-17])
10      {
11        cube(100);
12      }
13    }
14  }
15  translate([-30,-30,2])
16  {
17    cube([60,60,10]);
18  }
19 }
20 translate([0,0,11.5])
21 {
22   color([208/255,223/255,230/255],alpha=1)
23   cylinder(h=2,r=15.7,center=true);
24 }
```

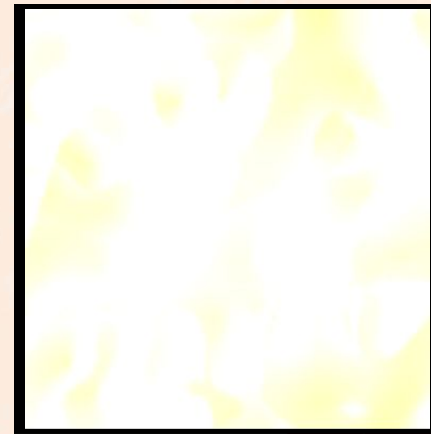
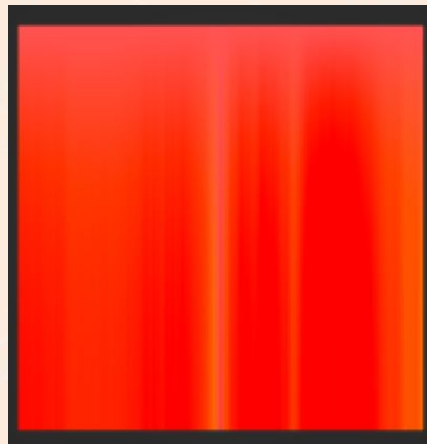
02

Shading



✦ Shading

ShaderFrog



✦ Shading

The screenshot displays the ShaderFrog Editor interface. At the top, there are navigation tabs: "Browse", "Editor" (highlighted), "Store", and "Create". The "Editor" tab shows a workspace with a grid and a sphere node. The sphere node is labeled "Sphere" and has a "Multiplier" slider set to 1. Below the workspace, there is a section for "SubShader" settings, including a "Preset" dropdown set to "Default" and a "Multiplier" slider. The "color" section shows RGB values: r 0.996078431372, g 0.882352941176, b 0.803921568627. The "Description" section is visible on the right, showing a "Screenshot" and a "Link" to the public view.

ShaderFrog

Browse Editor Store Create

All About ShaderFrog yds

LastPlate ✕ +

Public link <http://shaderfrog.com/view/6587> Fork Export... Publish

Zoom All Spring!

A SubShader. Internally this is a regular GLSL program that ShaderFrog has analyzed and made accessible to the graph. The entire shader output is being used. Compose it with other nodes to create complex effects.

"triplanar" SubShader Uniforms

Preset Default

Multiplier -1 0 1 1

color r 0.996078431372 g 0.882352941176 b 0.803921568627

vec3

Channel0 Building block

Sphere FPS 58.3

Screenshot Name

LastPlate

Link <http://shaderfrog.com/app/view/6587>

Description

Shader description

✦ Shading



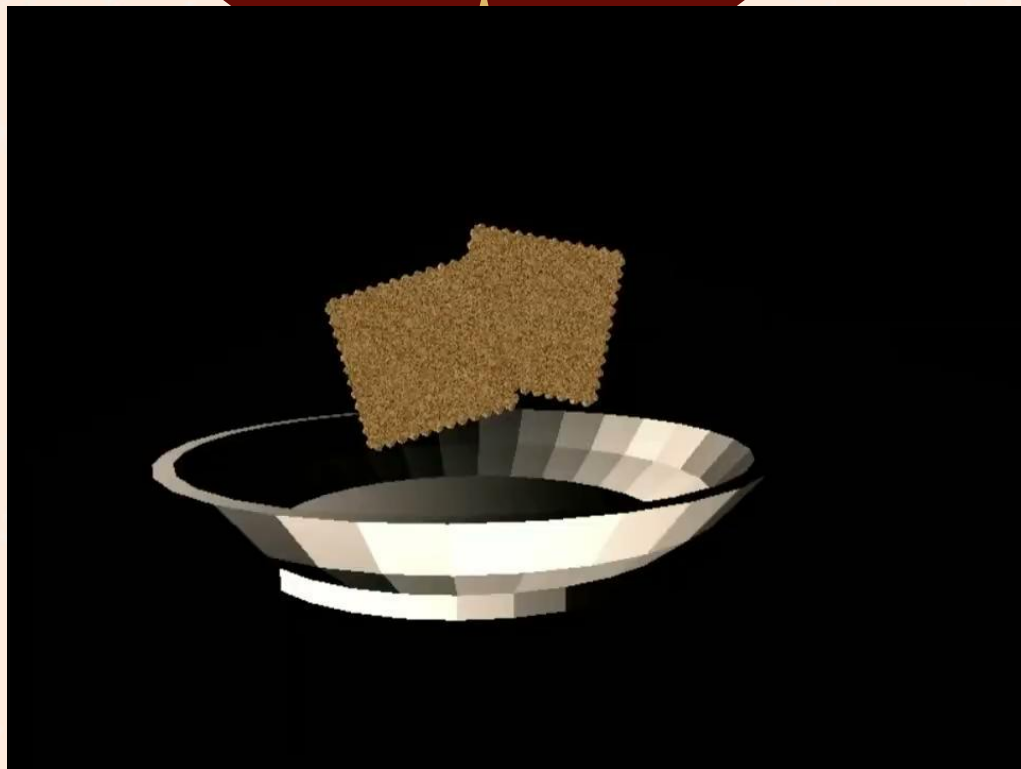
03

Animation

Animate the biscuit with
three.js



```
function animate() {  
    requestAnimationFrame( animate );  
  
    const mesh2 = scene.getObjectByName('bottom');  
    //write the reference of the object  
    //since all of the object's part are on the same position,  
    //we can just call one of them, then we chose the bottom part of the object  
  
    const cameraX = Math.sin(Date.now()*0.0008)*65;  
    //asking the camera on the X-coordinate make the circling motion  
    const cameraZ = Math.cos(Date.now()*0.0008)*65;  
    //asking the camera on the Z-coordinate make the circling motion  
  
    camera.position.set(mesh2.position.x + cameraX, mesh2.position.y, mesh2.position.z + cameraZ);  
    camera.lookAt(mesh2.position);  
    //keep the camera always stick with the object  
    camera.position.y += 1;  
    //since the camera is a little bit lower than the object, so we move it up by adding 1 in Y-coordinate  
    render();  
}
```

春

04

HTML



```
<html lang="en">
<head>
<title>Project show</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="mainTemplate.css">
</head>
<body class="body1" onload="changeImageSystem()">
  <script type="text/JavaScript" src="scriptformainpage.js"></script>
  <div class="header">
    <h1>Project Showcase</h1>
  </div>
  <div class="menu">
    <ul>
      <h3>Menu</h3>
      <a href="About Us.html"><li>About Us</li></a>
      <a href="http://localhost:8000/Mysf/"><li>Project</li></a>
      <li>Code</li>
      <a href="AnimationVideo.html"><li>Video</li></a>
      <li>Presentation</li>
    </ul>
  </div>

  <div class="MainContent" id="workingImage">
```

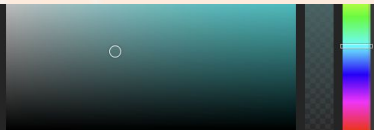
```
const sleep = (delay) => new Promise((resolve) => setTimeout(resolve, delay))

async function changeImageSystem(){
  images=new Array('image3.png','image4.png','image1.jpg','image2.jpg','image3.jpg');
  imageLen=5;
  count=imageLen-1;
  //changeImage(images[count]);
  for(;;)
  {
    count--;
    if(count<0)
    {
      count=imageLen-1;
    }
    changeImage(images[count]);
    await sleep(5000);
  }
}

function changeImage(imageName)
{
  var x=document.getElementById('workingImage').style;
  var name="url('"+imageName+"')"
  x.backgroundImage=name;
  x.backgroundRepeat="no-repeat";
}
```



```
text-shadow: 10px;  
font-size: filled;  
border: 0px;  
text-decoration-sty  
line-height: 100px;
```



```
div.menu{  
  width: 23%;  
  float:left;  
  background-color: rgb(75, 120, 120);  
  border-radius: 4px;
```

Sets the background color of an element.
(Edge 12, Firefox 1, Safari 1, Chrome 1, IE 4, Opera 3)

```
body.body1{  
  background-image: url(BackGround.png);  
  background-repeat:no-repeat;  
  background-size: 100%;
```

```
menu li{  
  background-color: aquamarine;  
  font-family: 'Times New Roman', Times, serif;  
  font-weight: bold;  
  text-align: center;  
  line-height: 30px;  
  font-size: 20px;  
  list-style: none;  
  height: 50px;  
  position: relative;  
  transition: 0.34s;
```

```
menu li:hover{  
  background-color: antiquewhite;  
  font-size: 30px;
```



Thanks!

Any questions?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

