计算机图形学

之

旋转的三角形

**学 院： 计算机与控制工程学院**

**班 级： 计181-2**

**学 号： 201858501210、201858501239**

**姓 名： 王宇航、丁智颖**

**指导老师： 马朝青**

**2021 年 6月 25 日**

# .概述

webgl：

WebGL完美地解决了现有的Web交互式三维动画的两个问题：第一，它通过 HTML脚本本身实现Web交互式三维动画的制作，无需任何 浏览器插件支持;第二，它利用底层的图形 硬件加速功能进行的图形渲染，是通过统一的、标准的、跨平台的OpenGL接口实现的。

使用鼠标选择窗口内任意位置画N个等腰三角形

要画的三角形大小及颜色通过右键弹出菜单选定(任选M个颜色)

设定三角形绕其中心旋转的开始、停止、旋转方向的功能键，并通过相应按键控制三角形旋转。

# 代码分析

## 三角形大小及颜色的选择

右键菜单：

  <ul id="rightMenu" style="width: 100px;">

  <li style="text-align: center"><input type="button" value="珊瑚红" onclick="drawTriangle('darksalmon')"></li>

  <li style="text-align: center"><input type="button" value="明黄色" onclick="drawTriangle('yellow')"></li>

  <li style="text-align: center"><input type="button" value="青绿色" onclick="drawTriangle('aquamarine')"></li>

  <li style="text-align: center"><input type="button" value="海蓝色" onclick="drawTriangle('dodgerblue')"></li>

  <li style="text-align: center"><input type="button" value="浅蓝色" onclick="drawTriangle('azure')"></li>

</ul>

属性设置为隐藏：

#rightMenu {

          list-style: none;

          background: gainsboro;

          border: solid 1px darkgrey;

          margin: 0px;

          padding: 5px;

          position: absolute;

          top: 0px;

          display: none;

      }

右键单击时显示：

document.documentElement.onclick = function () {

    rm.style.display = "none";

  }

画三角形：

  var drawTriangle = function (color) {

    var triangleShape3 = new THREE.Shape();

    triangleShape3.moveTo(xposition, yposition);

    // 开始划线

    triangleShape3.lineTo(xposition - 20, yposition - 20);

    triangleShape3.lineTo(xposition + 20, yposition - 20);

    triangleShape3.lineTo(xposition, yposition);

    var geometry3 = new THREE.ShapeGeometry(triangleShape3);

    var material3 = new THREE.MeshBasicMaterial({

      color: color,

      side: THREE.DoubleSide

    });

    var cube3 = new THREE.Mesh(geometry3, material3);

    scene.add(cube3);

    function runder() {

      renderer.render(scene, camera);

      cube3.rotateZ(Math.PI / 200 \* (direction));

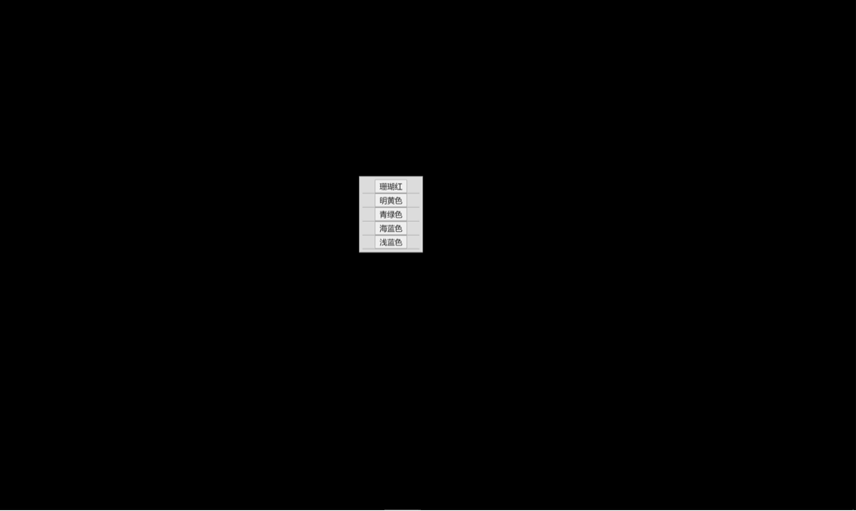
      requestAnimationFrame(runder);

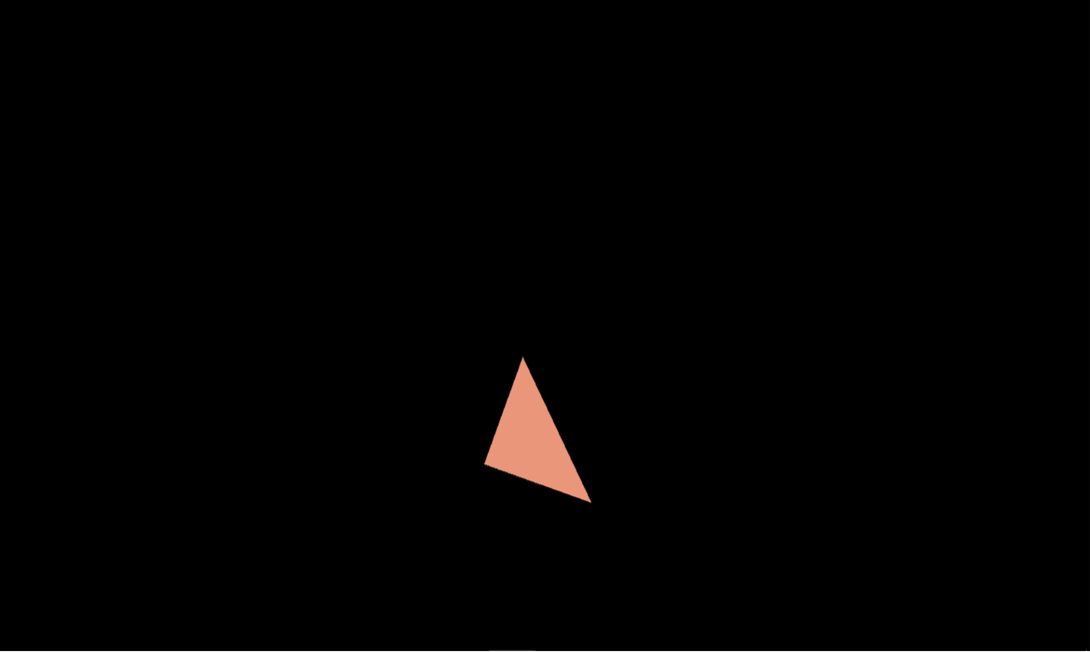
    }

    runder();

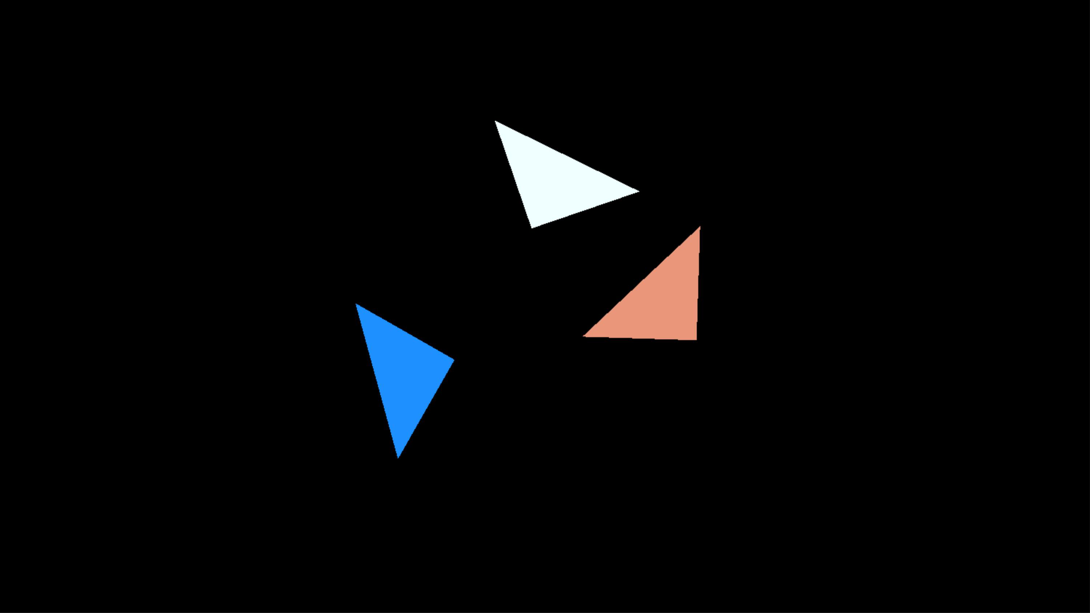
  }

鼠标右键选择颜色

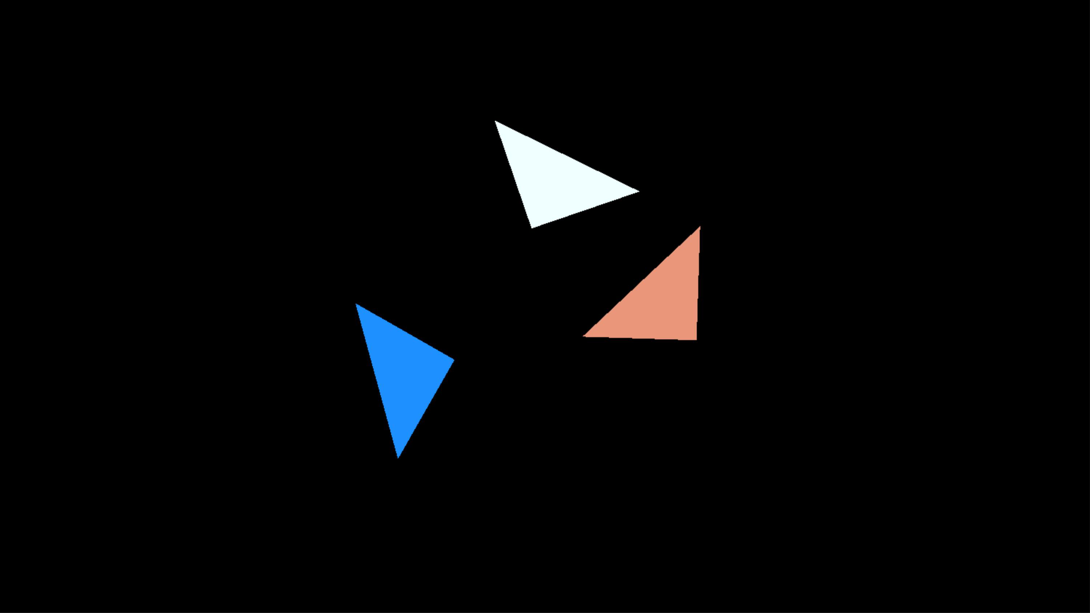
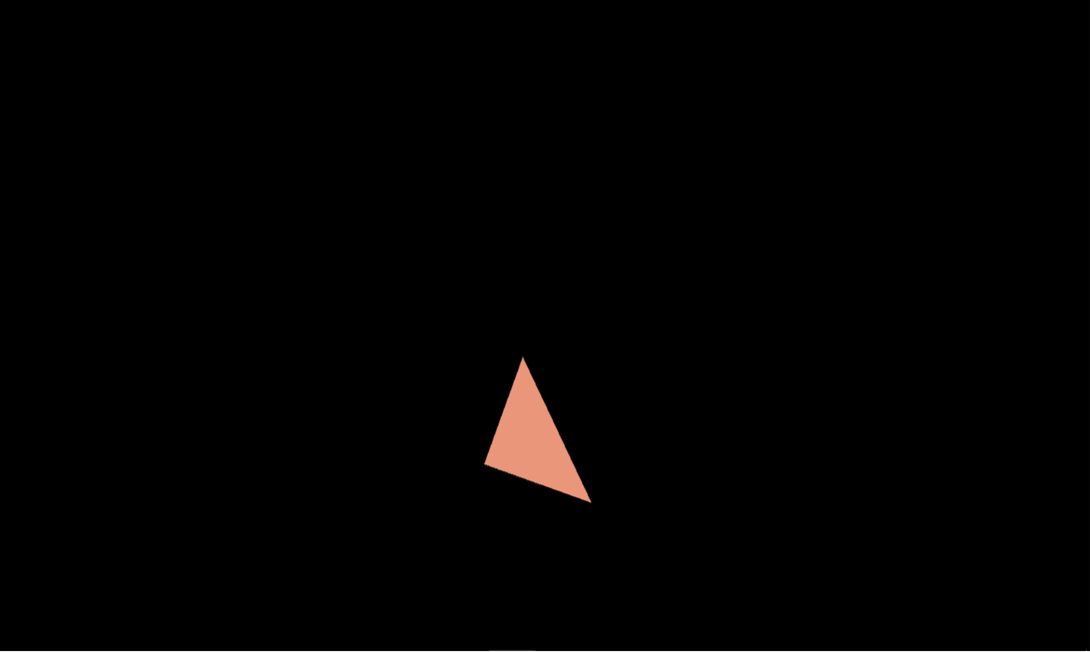




## 右键菜单、鼠标点击事件(包括左键右键)、画三角形



# 测试（详情，请见视频）



# 总结

## 分工

此次期中作业为小组二人一起完成，王宇航负责编写使用鼠标选择窗口内任意位置画N个等腰三角形，要画的三角形大小及颜色通过右键弹出菜单选定(任选M个颜色)；丁智颖负责设定三角形绕其中心旋转的开始、停止、旋转方向的功能键，并通过相应按键控制三角形旋转。部分代码的编写。报告及视频录制由两人共同完成。

## 小组工作总结

本次期中作业是做旋转的三角形。其中的.js文件是在网上找的框架，在此基础上稍作修改，html文件是由小组成员编写。总体来说，有一定的难度，毕竟在此之前没有接触过，但通过本次练习，小组内以及小组间互相学习，也学到了不少知识。

# 源码

<html>

<head>

  <meta charset="UTF-8">

  <title>三角形</title>

  <style>

      body {

          margin: 0;

      }

      canvas {

          width: 100%;

          height: 100%

      }

      #rightMenu {

          list-style: none;

          background: gainsboro;

          border: solid 1px darkgrey;

          margin: 0px;

          padding: 5px;

          position: absolute;

          top: 0px;

          display: none;

      }

      #rightMenu li {

          border-bottom: solid 1px darkgrey;

      }

  </style>

</head>

<body>

<ul id="rightMenu" style="width: 100px;">

  <li style="text-align: center"><input type="button" value="珊瑚红" onclick="drawTriangle('darksalmon')"></li>

  <li style="text-align: center"><input type="button" value="明黄色" onclick="drawTriangle('yellow')"></li>

  <li style="text-align: center"><input type="button" value="青绿色" onclick="drawTriangle('aquamarine')"></li>

  <li style="text-align: center"><input type="button" value="海蓝色" onclick="drawTriangle('dodgerblue')"></li>

  <li style="text-align: center"><input type="button" value="浅蓝色" onclick="drawTriangle('azure')"></li>

</ul>

<script src="three.js"></script>

<script>

  var rm = document.getElementById("rightMenu");

  document.documentElement.oncontextmenu = function (e) {

    e = e || window.event;

    var mx = e.clientX;

    var my = e.clientY;

    var rmWidth = parseInt(rm.style.width);

    var pageWidth = document.documentElement.clientWidth;

    //console.log(pageWidth);

    if ((mx + rmWidth) < pageWidth) {

      rm.style.left = mx + "px";

      rm.style.top = my + "px";

    } else {

      rm.style.right = mx + "px";

      rm.style.top = my + "px";

    }

    rm.style.display = "block";

    return false;

  };

  //不需要时隐藏右键菜单

  document.documentElement.onclick = function () {

    rm.style.display = "none";

  }

</script>

<script>

  var scene = new THREE.Scene();

  var camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight, 0.1, 1000);

  var renderer = new THREE.WebGLRenderer();

  renderer.setSize(window.innerWidth, window.innerHeight);

  document.body.appendChild(renderer.domElement);

  var triangleShape = new THREE.Shape();

  triangleShape.moveTo(0, 0);

  triangleShape.lineTo(0, 0);

  triangleShape.lineTo(0, 0);

  triangleShape.lineTo(0, 0);

  var geometry = new THREE.ShapeGeometry(triangleShape);

  var material = new THREE.MeshBasicMaterial({

    color: 'black',

    side: THREE.DoubleSide

  });

  var cube = new THREE.Mesh(geometry, material);

  scene.add(cube);

  camera.position.z = 105;

  var direction = 1;

  var animate = function () {

    requestAnimationFrame(animate);

    cube.rotateZ(Math.PI / 200 \* (direction));

    renderer.render(scene, camera);

  };

  animate();

  var xposition = 0;

  var yposition = 0;

  var drawTriangle = function (color) {

    var triangleShape3 = new THREE.Shape();

    triangleShape3.moveTo(xposition, yposition);

    // 开始划线

    triangleShape3.lineTo(xposition - 20, yposition - 20);

    triangleShape3.lineTo(xposition + 20, yposition - 20);

    triangleShape3.lineTo(xposition, yposition);

    var geometry3 = new THREE.ShapeGeometry(triangleShape3);

    var material3 = new THREE.MeshBasicMaterial({

      color: color,

      side: THREE.DoubleSide

    });

    var cube3 = new THREE.Mesh(geometry3, material3);

    scene.add(cube3);

    function runder() {

      renderer.render(scene, camera);

      cube3.rotateZ(Math.PI / 200 \* (direction));

      requestAnimationFrame(runder);

    }

    runder();

  }

  function setEventsMouse() {

    //点击了鼠标左键

    window.addEventListener('click', function (e) {

      if (e.button === 0) {

        console.log("点击了鼠标左键");

      }

    });

    //点击了鼠标右键

    window.addEventListener('contextmenu', function (e) {

      function convertTo3DCoordinate(clientX, clientY) {

        var mv = new THREE.Vector3(

          (clientX / window.innerWidth) \* 2 - 1,

          -(clientY / window.innerHeight) \* 2 + 1,

          0.5);

        mv.unproject(this.camera);   //这句将一个向量转成threejs坐标

        return mv;

      }

      if (e.button === 2) {

        xy = convertTo3DCoordinate(e.x, e.y);

        xposition = xy.x \* 300;

        yposition = xy.y \* 300;

        console.log(xposition, yposition);

      }

    });

  }

  function setKeyEvents() {

    window.addEventListener('keydown', function (e) {

      if (e.code === 'KeyW') {

        if (direction === 1) {

          direction = -1;

        } else {

          direction = 1;

        }

      }

      if (e.code === 'KeyS') {

        direction = 0;

      }

    });

  }

  setEventsMouse();

  setKeyEvents();

</script>

</body>

</html>

<!-- w方向 -->

<!-- s停止 -->