

文档一 确定主题

一、主题：嘴唇

二、实现形式：WebGL

三、成员及分工：

刘云鹤：功能分析，网页制作，代码实现

周晔：功能分析，文档书写，代码实现

四、主要功能及其描述

主要实现功能有旋转、平移、比例缩放的几何变换。

平移

平移变换是把点沿着给定而方向移动固定的距离，改变点的位置。

$$\text{平移矩阵: } T = \begin{pmatrix} 1 & 0 & 0 & -\alpha_x \\ 0 & 1 & 0 & -\alpha_y \\ 0 & 0 & 1 & -\alpha_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

旋转

旋转是图形的所有点在二维平面中旋转一定角度的过程。可分为逆时针旋转，顺时针旋转，同时必须指定旋转角度和旋转点，旋转角的正值表示旋转方向为顺时针，负值表示旋转方向为逆时针。

旋转点在原点的旋转

$$\text{旋转矩阵: } R_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

缩放

是一种仿射变换，通过缩放变换可以放大或缩小对象。可以分为均匀缩放和

非均匀缩放。缩放变换有缩放方向和缩放因子 s ，当 $s>1$ 时，对象沿着指定的方向伸长；当 $0<s<1$ 时，对象沿着指定方向缩短；当 $s<0$ 时，对象沿着缩放方向反射变换。

$$S = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

文档二 详细设计

一、系统完整功能描述

本文设计的嘴唇是由两个长方体组成，两个长方体分别代表最纯的上下两部分，可以实现嘴唇的开合，旋转，平移以及缩放的功能。

二、各模块流程、结构具体实现、关键函数、变量说明

平移模块：

利用屏幕上的控件获得平移的方向，`<button id="xtbutton">T X</button>`，`<button id="ytbutton">T Y</button>`，`<button id="ztbutton">T Z</button>` 分别代表向 x ， y ， z 轴平移，通过 `document.getElementById("xtbutton")` 函数获得方向信息，通过 `d[taxis] += 0.01`；获得平移大小的信息，得到平移矩阵 d ，与原矩阵相乘得到新的坐标点。

旋转模块：

利用键盘上的上下左右键控制旋转的方向，嘴唇的开合的利用的是旋转实现的，上下键可以控制嘴唇的上下移动方向，通过 `g_arm2Angle += ANGLE_STEP`；控制嘴唇开合的大小。左右键可以使嘴唇整个进行旋转，`g_arm1Angle = (g_arm1Angle + ANGLE_STEP) % 360`；调整旋转角度的大小。

缩放模块：

利用屏幕上的控件获得平移的方向，`id="xs"`，`id="ys"`，`id="zs"` 分别代表在 x ， y ， z 轴方向上进行缩放，通过 `document.getElementById("xs").value` 获得缩放的数值信息，，得到缩放矩阵 s ，与原矩阵相乘得到新的坐标点。

三、实现工具，开发环境，工具库

实现工具：HBuilder

开发环境：firefox 浏览器

工具库：webgl

文档三 最终设计

一、系统实现功能详细描述

1. 相机选取

使用相机让物体从三维空间投影到二维空间的方式有两种，即透视投影和正投影。透视投影遵循“远则小，近则大”的投影方式，与日常生活人们所看到的物体方式一致；正投影不管物体和视点的距离远近，对物体都按照统一的大小进行绘制，我们课程设计时使用的是透视投影。

2. 模型的平移、旋转和缩放

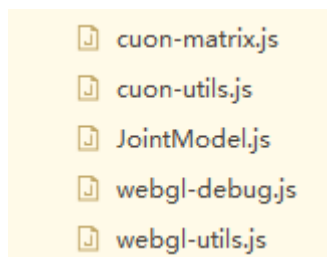
模型的平移、旋转和缩放都是要进行矩阵的相乘，原矩阵与平移矩阵、旋转矩阵、缩放矩阵的相乘，因此关键是找到平移矩阵和旋转矩阵。平移和旋转都需要方向和大小，三维空间中，方向可以有 x ， y ， z ，大小的正负值可以表示正负方向。

可以由屏幕上的控件进行控制。先把平移矩阵设置为单位矩阵，平移位置由 $d.x$ ， $d.y$ ， $d.z$ 代替，初始值为 0.0，屏幕上的 TX ， TY ， TZ ，控制平移方向，点击一次，以 $d.x$ 或 $d.y$ 或 $d.z$ 加上特定的步长得到新的平移矩阵。

模型的旋转过程与平移类似，不同的是加上的是旋转的角度，绕 x ， y ， z 旋转的矩阵不同。旋转是由键盘的上下左右键控制方向，按键一次，角度增加特定的数值，以此达到旋转的效果，缩放只有屏幕上的控件可以控制，在一定的范围内可以是选取的数字，同样通过函数得到方向和大小，更新缩放矩阵内的值，再与原矩阵相乘，得到模型新的位置矩阵，达到缩放变换的效果。

二、核心功能源文件清单，核心代码说明

文件清单：



核心代码说明：

1. 长方体的绘制

```

1 var vertices = new Float32Array([
    7, 3, 1.5, -7, 3, 1.5, -7, 0.0, 1.5, 7, 0.0, 1.5, // v0-v1-v2-v3 front
    7, 3, 1.5, 7, 0, 1.5, 7, 0.0, -1.5, 7, 3, -1.5, // v0-v3-v4-v5 right
    7, 3, 1.5, 7, 3, -1.5, -7, 3, -1.5, -7, 3, 1.5, // v0-v5-v6-v1 up
    -7, 3, 1.5, -7, 3, -1.5, -7, 0.0, -1.5, -7, 0.0, 1.5, // v1-v6-v7-v2 left
    -7, 0, -1.5, 7, 0, -1.5, 7, 0.0, 1.5, -7, 0.0, 1.5, // v7-v4-v3-v2 down
    7, 0, -1.5, -7, 0, -1.5, -7, 3, -1.5, 7, 3, -1.5 // v4-v7-v6-v5 back
]);

```

```

var indices = new Uint8Array([
    0, 1, 2, 0, 2, 3, // front
    4, 5, 6, 4, 6, 7, // right
    8, 9, 10, 8, 10, 11, // up
    12, 13, 14, 12, 14, 15, // left
    16, 17, 18, 16, 18, 19, // down
    20, 21, 22, 20, 22, 23 // back
]);

```

2. 控制相机位置

透视投影通过 lookAt () 函数实现

```

var viewProjMatrix = new Matrix4();
viewProjMatrix.setPerspective(50.0, canvas.width / canvas.height, 1.0, 100.0);
viewProjMatrix.lookAt(20.0, 10.0, 30, 0.0, 0.0, 0.0, 1.0, 0.0);

```

3. 屏幕上控件获得物体平移旋转的方向

```

//平移
document.getElementById("xtbutton").onclick = function () {
    taxis = 0;
}

document.getElementById("ytbutton").onclick = function () { 
}

document.getElementById("ztbutton").onclick = function () { 
}

```

4. 按键次数控制大小

```

// Draw
d[taxis] += 0.01;
if(d[taxis]>0.5) d[taxis]=-d[taxis];
gl.uniform3fv(dLoc,d);

```

5. 缩放

```
function x_suofang() {
    xs = document.getElementById("xs").value;
    saxis = 0;
    s[saxis] = xs;
    gl.uniform3fv(sLoc, s);
}

function y_suofang() { ... }

function z_suofang() { ... }
```

4. 键盘控制嘴唇的开合

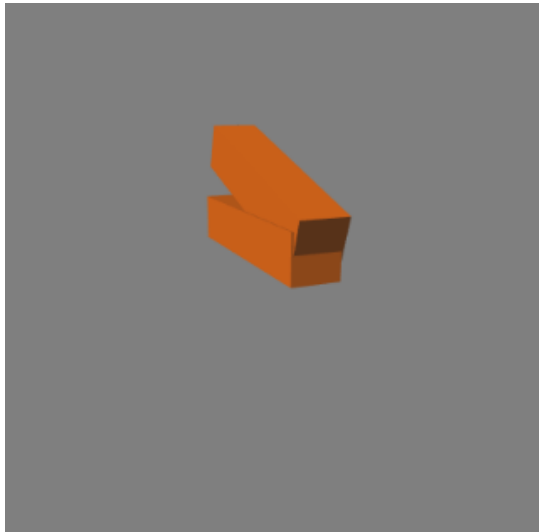
```
function keydown(ev, gl, n, viewProjMatrix, u_MvpMatrix, u_NormalMatrix) {
    switch (ev.keyCode) { // ev.keyCode: 获取按键
        case 38: // key -> 绕z轴正方向旋转
            if (g_arm2Angle < 30) g_arm2Angle += ANGLE_STEP;
            break;
        case 40: // key -> 绕z轴负方向旋转
            if (g_arm2Angle > 0) g_arm2Angle -= ANGLE_STEP;
            break;
        case 39: // Right arrow key -> 绕y轴正方向
            g_arm1Angle = (g_arm1Angle + ANGLE_STEP) % 360;
            break;
        case 37: // Left arrow key -> 绕y轴负方向
            g_arm1Angle = (g_arm1Angle - ANGLE_STEP) % 360;
            break;
        default: return; // Skip drawing at no effective action
    }
}
```

三、实现功能展示

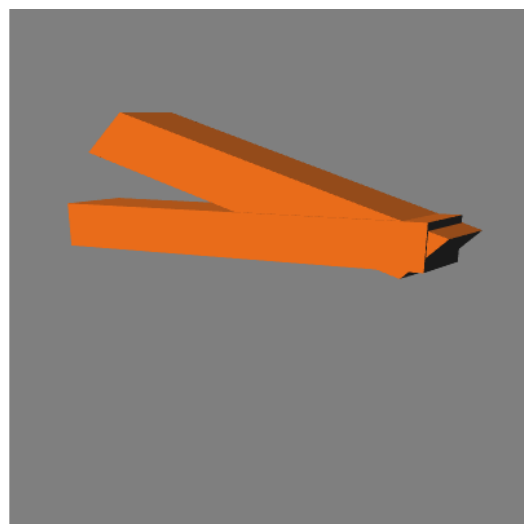
平移:



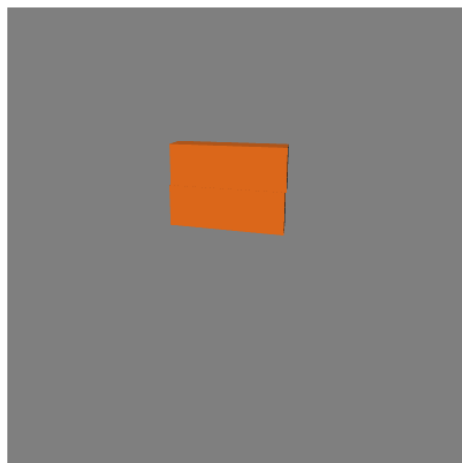
旋转



缩放



嘴巴闭和



四、小组成员实际完成工作情况及自评

姓名	实际完成情况	自评
刘云鹤	代码书写完成 网页制作完成	95
周晔	文档书写完成 代码书写完成	95