

手指模型设计说明

一、需求分析

1.1 技术需求

该模型属于 3D 模型的开发。建立模型的脚本语言有 HTML5、JavaScript、CSS，系统的平台是 Windows 10，开发工具是 Hbuilder，测试的浏览器有谷歌、IE、Hbuilder 自带的浏览器。通过建立模型矩阵和投影矩阵实现对各部分手指的关节控制。

1.2 功能需求

1.2.1 手指模型使用介绍：

该界面共有 15 个控制键，分别对应身体旋转角度（可理解为手掌心）和手指各个关节，可以通过鼠标控制，也可以通过点击之后用键盘的 Home 和 End 键进行控制，每个对应部位可以从-180 度旋转到 180 度，每个面都有不同颜色来进行区分，可以帮助观察者全方位看到并理解手部结构。

1.2.2 手指模型制作功能需求：

界面中需要将手指模型构建出来，并且整体及每个关节可以转动。

二、概要设计

2.1 设计思路

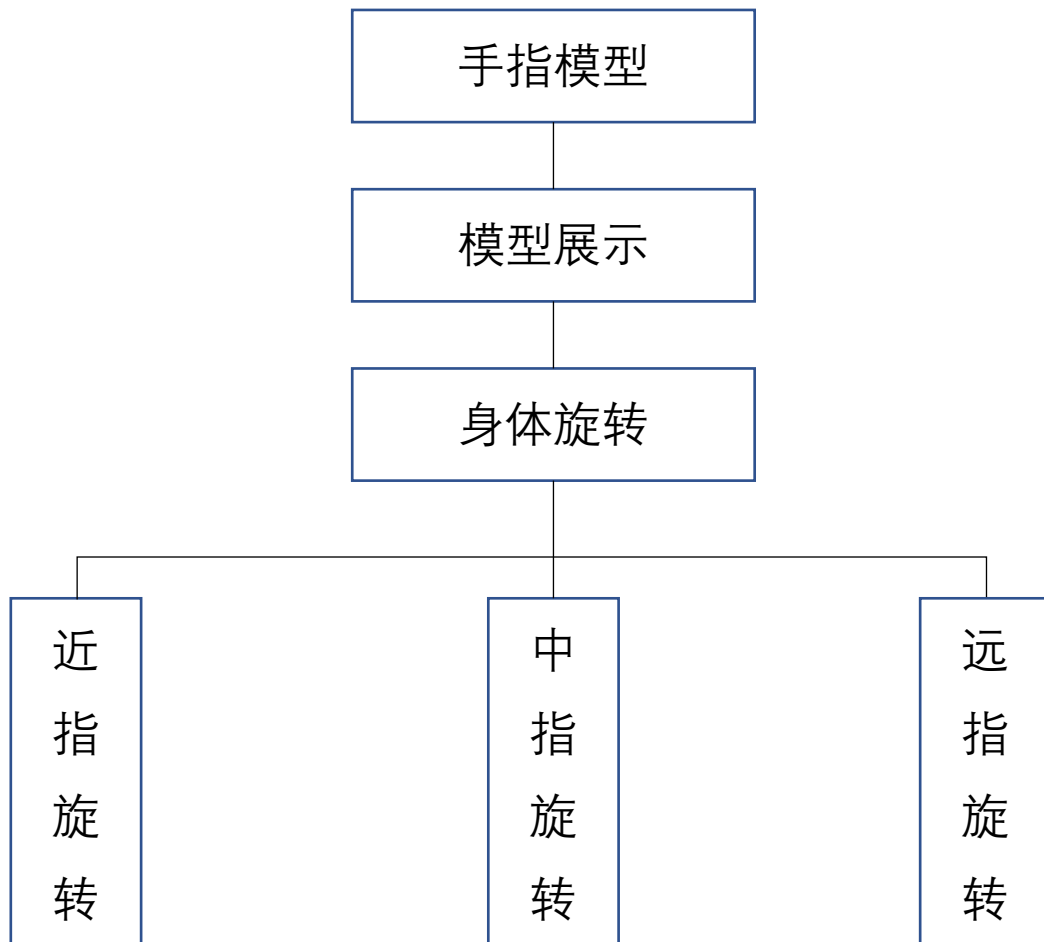
我们小组选择手部模型是因为比较好呈现且每根手指容易有区分度，由于能力不足，做出的模型简单且不够逼真。总体的设计思路是要保证模型能够 360 度旋转达到全方位观察的效果。

2.2 功能模块设计

该模型有五大功能模块，分别为：

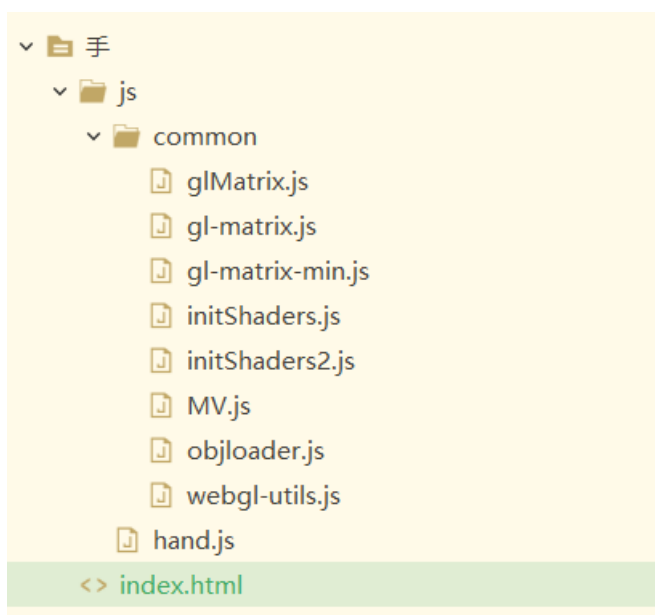
1. 模型展示
2. 身体旋转
3. 近指旋转
4. 中指旋转
5. 远指旋转

具体功能模块示意图如下：



三、详细设计

3.1 模型工程视图

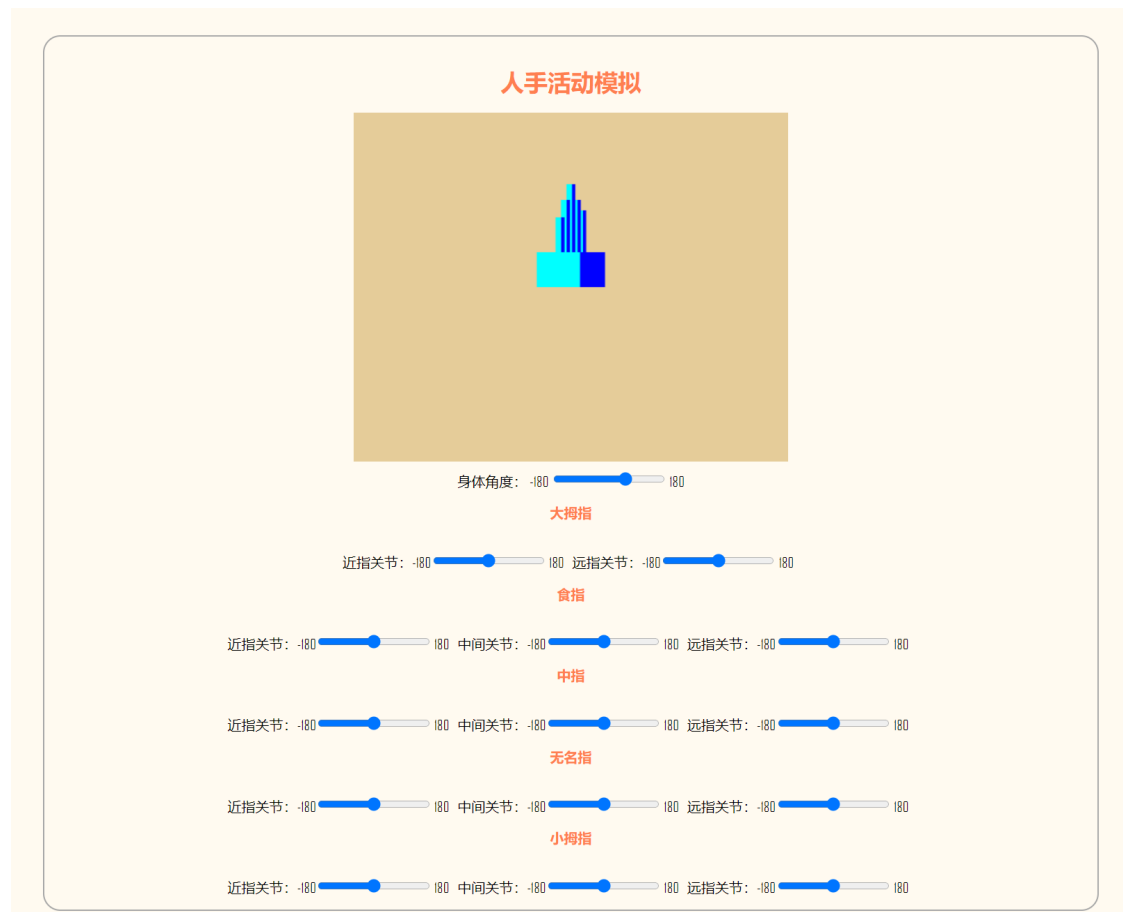


3.2 模型视图



3.3 模型界面布局

有模型展示和模型控制两部分



四、模型基本代码简介

主体代码都写在 `hand.js` 中，同时还需引用 `webgl-utils.js`、`initShaders.js`、`MV.js`

4.1 初始设置: 设置一下初始化常数, 包括底座坐标及各面颜色、底座、底部手指、中部手指、上部手指的宽高

```
var vertices = [  
    vec4( -0.5, -0.5,  0.5, 1.0 ),//0  
    vec4( -0.5,  0.5,  0.5, 1.0 ),//1  
    vec4(  0.5,  0.5,  0.5, 1.0 ),//2  
    vec4(  0.5, -0.5,  0.5, 1.0 ),//3  
    vec4( -0.5, -0.5, -0.5, 1.0 ),//4  
    vec4( -0.5,  0.5, -0.5, 1.0 ),//5  
    vec4(  0.5,  0.5, -0.5, 1.0 ),//6  
    vec4(  0.5, -0.5, -0.5, 1.0 )//7  
];  
  
// RGBA colors  
var vertexColors = [  
    vec4( 0.0, 0.0, 0.0, 1.0 ), // black  
    vec4( 0.0, 0.0, 1.0, 1.0 ), // blue  
    vec4( 1.0, 0.0, 0.0, 1.0 ), // red  
    vec4( 1.0, 0.0, 1.0, 1.0 ), // magenta  
    vec4( 0.0, 1.0, 0.0, 1.0 ), // green  
    vec4( 0.0, 1.0, 1.0, 1.0 ), // cyan  
    vec4( 1.0, 1.0, 0.0, 1.0 ), // yellow  
    vec4( 1.0, 1.0, 1.0, 1.0 ) // white  
];  
  
//底座宽高  
var BASE_HEIGHT      = 2.0;  
var BASE_WIDTH       = 2.3;  
  
//底部手指宽高  
var LOWER_ARM_HEIGHT = 1.0;  
var LOWER_ARM_WIDTH  = 0.3;  
var LOWER_ARM_HEIGHT_SEC = 1.0;  
var LOWER_ARM_WIDTH_SEC  = 0.3;  
var LOWER_ARM_HEIGHT_THI = 1.3;  
var LOWER_ARM_WIDTH_THI  = 0.3;  
var LOWER_ARM_HEIGHT_FOR = 1.0;  
var LOWER_ARM_WIDTH_FOR  = 0.3;  
var LOWER_ARM_HEIGHT_FIF = 0.8;  
var LOWER_ARM_WIDTH_FIF  = 0.3;  
  
//中部手指宽高  
var MID_ARM_HEIGHT_SEC=1.0  
var MID_ARM_WIDTH_SEC=0.3
```

```

var MID_ARM_HEIGHT_THI=1.3
var MID_ARM_WIDTH_THI=0.3
var MID_ARM_HEIGHT_FOR=1.0
var MID_ARM_WIDTH_FOR=0.3
var MID_ARM_HEIGHT_FIF=0.8
var MID_ARM_WIDTH_FIF=0.3
//上面手指宽高
var UPPER_ARM_HEIGHT = 1.0;
var UPPER_ARM_WIDTH  = 0.3;
var UPPER_ARM_HEIGHT_SEC = 1.0;
var UPPER_ARM_WIDTH_SEC  = 0.3;
var UPPER_ARM_HEIGHT_THI = 1.3;
var UPPER_ARM_WIDTH_THI  = 0.3;
var UPPER_ARM_HEIGHT_FOR = 1.0;
var UPPER_ARM_WIDTH_FOR  = 0.3;
var UPPER_ARM_HEIGHT_FIF = 0.8;
var UPPER_ARM_WIDTH_FIF  = 0.3;

```

4.2 为手部的关节做标记，建立 theta 角度数组

```

var Base = 0;
var LowerArm = 1;
var LowerArm1 = 2;
var LowerArm2 = 3;
var LowerArm3 = 4;
var LowerArm4 = 5;
var UpperArm = 6;
var UpperArm1 = 7;
var UpperArm2 = 8;
var UpperArm3 = 9;
var UpperArm4 = 10;
var MidArm1=11;
var MidArm2=12;
var MidArm3=13;
var MidArm4=14;

var theta= [0,0,0,0,0,0,0,0,0,0,0,0,0,0];

```

4.3 绘制一个立方体单位，每当增加一个点位置同时增加一个颜色索引，以该单位进行编写立方体参数更改函数(长宽高、位置)

```

function quad( a, b, c, d ) {
    colors.push(vertexColors[a]);
    points.push(vertices[a]);
    colors.push(vertexColors[a]);

```

```

        points.push(vertices[b]);
        colors.push(vertexColors[a]);
        points.push(vertices[c]);
        colors.push(vertexColors[a]);
        points.push(vertices[a]);
        colors.push(vertexColors[a]);
        points.push(vertices[c]);
        colors.push(vertexColors[a]);
        points.push(vertices[d]);
    }

```

```

function colorCube() {
    quad( 1, 0, 3, 2 );
    quad( 2, 3, 7, 6 );
    quad( 3, 0, 4, 7 );
    quad( 6, 5, 1, 2 );
    quad( 4, 5, 6, 7 );
    quad( 5, 4, 0, 1 );
}
//长宽高
function scale4(a, b, c) {
    var result = mat4();
    result[0][0] = a;
    result[1][1] = b;
    result[2][2] = c;
    return result;
}

```

4.4 定义交互代码。在 HTML 中设置滑动条，产生一个依赖于滑动条位置的值，是关节旋转的角度，展示部分代码如下：

```

document.getElementById("slider2").onchange = function(event) {
    theta[1] = event.target.value;
};
document.getElementById("slider3").onchange = function(event) {
    theta[2] = event.target.value;
};
document.getElementById("slider4").onchange = function(event) {
    theta[3] = event.target.value;
};
document.getElementById("slider5").onchange = function(event) {
    theta[4] = event.target.value;
};

```

```

        document.getElementById("slider6").onchange = function(event)
        {
            theta[5] = event.target.value;
        };

```

4.5 设置各个关节的函数并调用函数，部分代码如下：

```

//第一部分代码确定每个关节模型在画布中的位置
function upperArm1() {
    var s = scale4
    (UPPER_ARM_WIDTH_SEC, UPPER_ARM_HEIGHT_SEC, UPPER_ARM_WIDTH_SEC);
    var instanceMatrix =
    mult(translate( 0.0, 0.5 * UPPER_ARM_HEIGHT_SEC, 0.0 ),s);
    var t = mult(modelViewMatrix, instanceMatrix);
    gl.uniformMatrix4fv( modelViewMatrixLoc, false, flatten(t) );
    gl.drawArrays( gl.TRIANGLES, 0, NumVertices );
}
//确定每个模型进行旋转的定点后调用运行
modelViewMatrix =
mult(modelViewMatrix2, translate(-1.0, BASE_HEIGHT, 0.0));
modelViewMatrix =
mult(modelViewMatrix, rotate(theta[LowerArm], 0, 0, 1 ));
lowerArm();

//整个运行函数调用
requestAnimationFrame(render);

```

五、小组分工情况及自评

小组成员	自评分数	分工情况	
盛洋	98	实现模型旋转、PPT制作	共同纠正代码、文档、PPT中的错误
铁鑫	97	实现交互、建立模型、文档制作	