webgl 手臂文档

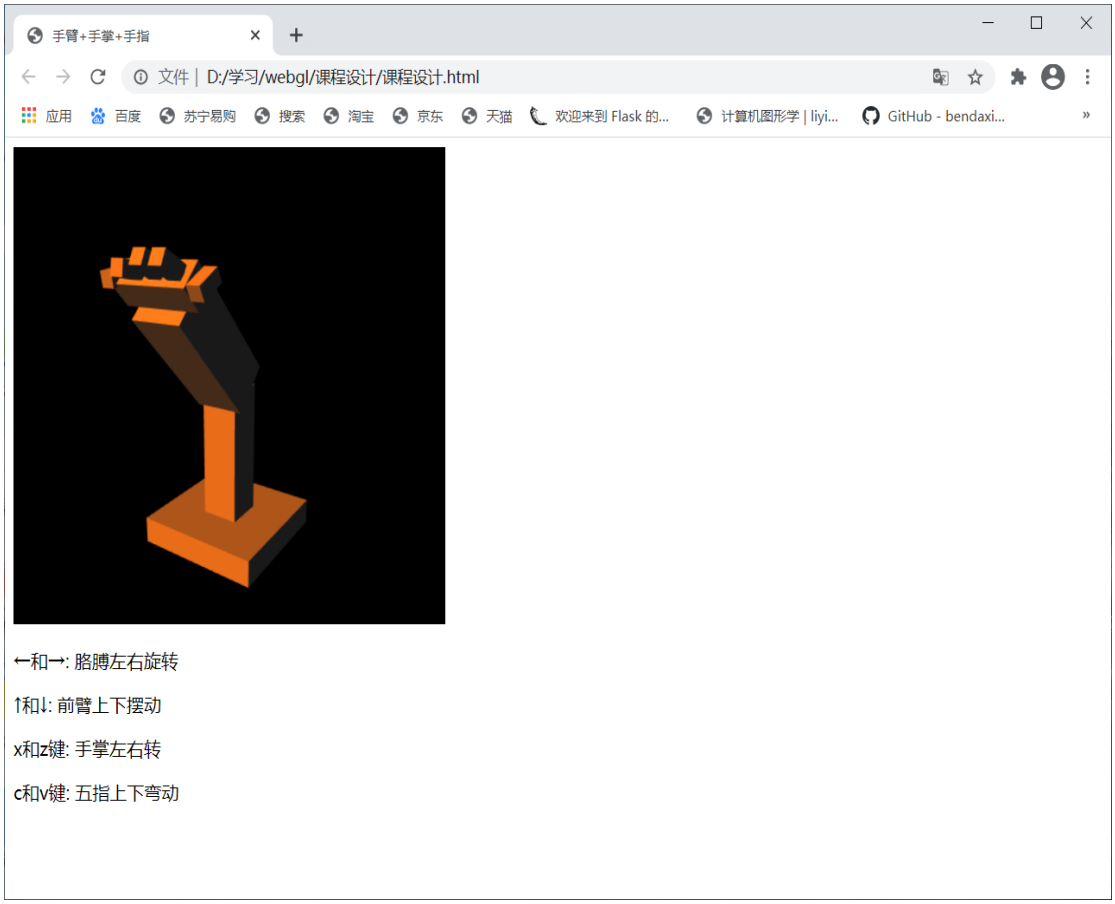
一. 需求分析

该模型属于层次模型的开发。建立模型的语言有 HTML5、JavaScrip 系统的平台是 Windows 10，开发工具：Hbuilder，测试的浏览器有 谷歌（最好）、360 极速浏览器以及 Hbuilder 自带的浏览器。通过建立模型矩阵和使用层次结构对各部分进行控制。

二. 文档介绍

使用 webgl 对整条手臂一下进行仿真，并实现一些简单的动作，例翻转，旋转等。通过文档的形式来介绍改系统的实现逻辑，便于理解。

1.1 使用界面



1.2 使用方法

进入网址：bendaxian.github.io。

通过键盘的 4 个方向键和 x，z，c，v 键进行相应的操作，用浏览器大可即可，无需密码登录。

左右键控制整个胳膊运动，上下键控制前臂的运动，x，z 控制手掌弯曲，c，v 控制手指弯曲。

1.3 运行环境：

操作系统：window，iOS，linux。

浏览器：谷歌（推荐），360 极速等主流浏览器。

1.4 参考文档

《webgl 编程指南》

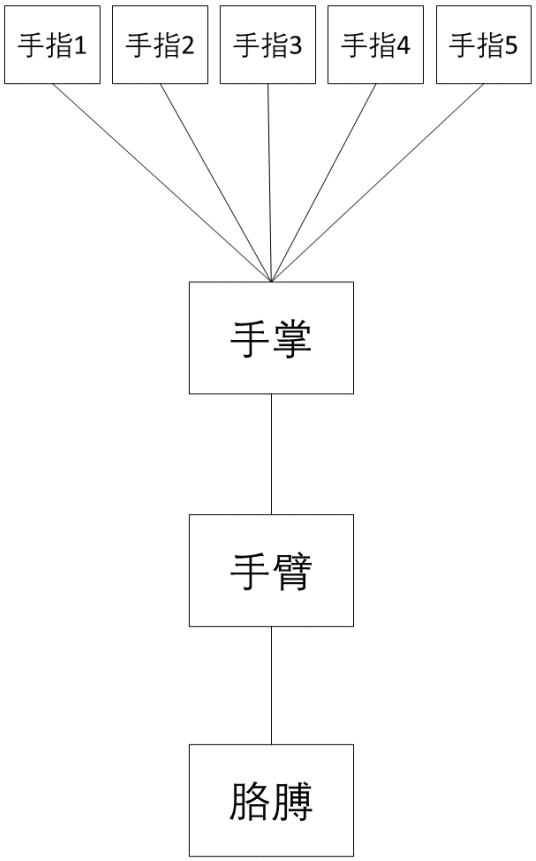
《交互式计算机图形学-基于 webgl 自顶向下的方法》

三. 系统结构

该系统仿真过程分为 4 个部分：手指部分，手掌部分，手臂和胳膊部分，将胳膊及其以下部分倒立于基座，该模型有 4 大功能模块：

1. 手指弯曲
2. 手掌弯曲
3. 前臂伸缩
4. 胳膊弯曲

具体模块结构示意图如下：



四. 系统设计

设计思路

采用层次结构顺序，在绘制手指之前，先将模型矩阵保存起来，绘制完手指后再将模型矩阵取出来作为当前模型矩阵，并绘制手掌。

该过程采用栈结构来完成操作，使用 pushMatrix 和 popMatrix 函数，将 g_modelMatrix 作为参数传入，将当时的模型矩阵的状态保存起来，绘制完后获取之前保存的矩阵，并赋给 g_modelMatrix，使用模型矩阵又回到绘制手指之前的状态，在此基础上绘制手掌，以此类推

直到手臂绘制完成。

用此方法可以绘制任意复杂的层次结构模型，按照层次顺序，从高到低绘制部件，并在绘制具有低部件的部件前将模型矩阵压入栈中，绘制完在弹出来。

部件设计

该系统分为手指，手掌，前臂，胳膊。用简单的模型按照层次顺序，从高到低逐一绘制，并在每个关节上应用模型矩阵。模型中每一个部件的数据都是提前定义在 initVertexBuffers 函数中，以下介绍每个部件的具体要求。

所有部件的弯曲角度统一定义为 45°—135°。

手指部分

手指采用立方体设计，由于设计者技术水平有限，故五指的运动采用统一运动，即：五指同时弯曲。

手掌部分

手掌设定为长方体，旋转采用上下旋转，旋转角设定于手指大小一致，但是手指和手掌属于从属关系，要随着手指同步变换位置。

前臂部分

前臂绕胳膊前后摆动，在摆动时手指和手掌要跟随同步摆动，设立模型矩阵，在前臂摆动时，其他部件的信息也都含在里面，同步进行。

胳膊部分

胳膊在整个系统中属于最高级别的部件，所以在旋转中要包含前面所有部件的信息，即：胳膊带动所有部件转动。

数据定义部分：该模型采用提前设定数据和参数。

```
//立方体顶点坐标
    var vertices_base = new Float32Array([ //底座(10x2x10)
        5.0, 2.0, 5.0, -5.0, 2.0, 5.0, -5.0, 0.0, 5.0,   5.0, 0.0, 5.0, // v0-v1-v2-v3 front
        5.0, 2.0, 5.0,   5.0, 0.0, 5.0,   5.0, 0.0,-5.0,   5.0, 2.0,-5.0, // v0-v3-v4-v5 right
        5.0, 2.0, 5.0,   5.0, 2.0,-5.0, -5.0, 2.0,-5.0, -5.0, 2.0, 5.0, // v0-v5-v6-v1 up
        -5.0, 2.0, 5.0, -5.0, 2.0,-5.0, -5.0, 0.0,-5.0, -5.0, 0.0, 5.0, // v1-v6-v7-v2 left
        -5.0, 0.0,-5.0,   5.0, 0.0,-5.0,   5.0, 0.0, 5.0, -5.0, 0.0, 5.0, // v7-v4-v3-v2 down
        5.0, 0.0,-5.0, -5.0, 0.0,-5.0, -5.0, 2.0,-5.0,   5.0, 2.0,-5.0   // v4-v7-v6-v5 back
    ]);
    var vertices_arm1 = new Float32Array([   //  胳膊(3x10x3)
        1.5, 10.0, 1.5, -1.5, 10.0, 1.5, -1.5,   0.0, 1.5,   1.5,   0.0, 1.5, // v0-v1-v2-v3 front
        1.5, 10.0, 1.5,   1.5,   0.0, 1.5,   1.5,   0.0,-1.5,   1.5, 10.0,-1.5, // v0-v3-v4-v5 right
        1.5, 10.0, 1.5,   1.5, 10.0,-1.5, -1.5, 10.0,-1.5, -1.5, 10.0, 1.5, // v0-v5-v6-v1 up
        -1.5, 10.0, 1.5, -1.5, 10.0,-1.5, -1.5,   0.0,-1.5, -1.5,   0.0, 1.5, // v1-v6-v7-v2 left
        -1.5,   0.0,-1.5,   1.5,   0.0,-1.5,   1.5,   0.0, 1.5, -1.5,   0.0, 1.5, // v7-v4-v3-v2 down
        1.5,   0.0,-1.5, -1.5,   0.0,-1.5, -1.5, 10.0,-1.5,   1.5, 10.0,-1.5   // v4-v7-v6-v5 back
    ]);
    var vertices_arm2 = new Float32Array([   //  前臂(4x10x4)
        2.0, 10.0, 2.0, -2.0, 10.0, 2.0, -2.0,   0.0, 2.0,   2.0,   0.0, 2.0, // v0-v1-v2-v3 front
        2.0, 10.0, 2.0,   2.0,   0.0, 2.0,   2.0,   0.0,-2.0,   2.0, 10.0,-2.0, // v0-v3-v4-v5 right
        2.0, 10.0, 2.0,   2.0, 10.0,-2.0, -2.0, 10.0,-2.0, -2.0, 10.0, 2.0, // v0-v5-v6-v1 up
        -2.0, 10.0, 2.0, -2.0, 10.0,-2.0, -2.0,   0.0,-2.0, -2.0,   0.0, 2.0, // v1-v6-v7-v2 left
        -2.0,   0.0,-2.0,   2.0,   0.0,-2.0,   2.0,   0.0, 2.0, -2.0,   0.0, 2.0, // v7-v4-v3-v2 down
```

```
        2.0,    0.0,-2.0, -2.0,    0.0,-2.0, -2.0, 10.0,-2.0,    2.0, 10.0,-2.0    // v4-v7-v6-v5 back
    ]);
    var vertices_palm = new Float32Array([    //  手掌(2x2x6)
        1.0, 2.0, 3.0, -1.0, 2.0, 3.0, -1.0, 0.0, 3.0,    1.0, 0.0, 3.0, // v0-v1-v2-v3 front
        1.0, 2.0, 3.0,    1.0, 0.0, 3.0,    1.0, 0.0,-3.0,    1.0, 2.0,-3.0, // v0-v3-v4-v5 right
        1.0, 2.0, 3.0,    1.0, 2.0,-3.0, -1.0, 2.0,-3.0, -1.0, 2.0, 3.0, // v0-v5-v6-v1 up
        -1.0, 2.0, 3.0, -1.0, 2.0,-3.0, -1.0, 0.0,-3.0, -1.0, 0.0, 3.0, // v1-v6-v7-v2 left
        -1.0, 0.0,-3.0,    1.0, 0.0,-3.0,    1.0, 0.0, 3.0, -1.0, 0.0, 3.0, // v7-v4-v3-v2 down
        1.0, 0.0,-3.0, -1.0, 0.0,-3.0, -1.0, 2.0,-3.0,    1.0, 2.0,-3.0    // v4-v7-v6-v5 back
    ]);
    var vertices_finger = new Float32Array([    //  五指(1x2x1)
        0.5, 2.0, 0.5, -0.5, 2.0, 0.5, -0.5, 0.0, 0.5,    0.5, 0.0, 0.5, // v0-v1-v2-v3 front
        0.5, 2.0, 0.5,    0.5, 0.0, 0.5,    0.5, 0.0,-0.5,    0.5, 2.0,-0.5, // v0-v3-v4-v5 right
        0.5, 2.0, 0.5,    0.5, 2.0,-0.5, -0.5, 2.0,-0.5, -0.5, 2.0, 0.5, // v0-v5-v6-v1 up
        -0.5, 2.0, 0.5, -0.5, 2.0,-0.5, -0.5, 0.0,-0.5, -0.5, 0.0, 0.5, // v1-v6-v7-v2 left
        -0.5, 0.0,-0.5,    0.5, 0.0,-0.5,    0.5, 0.0, 0.5, -0.5, 0.0, 0.5, // v7-v4-v3-v2 down
        0.5, 0.0,-0.5, -0.5, 0.0,-0.5, -0.5, 2.0,-0.5,    0.5, 2.0,-0.5    // v4-v7-v6-v5 back
    ]);

    //法线
    var normals = new Float32Array([
        0.0, 0.0, 1.0,    0.0, 0.0, 1.0,    0.0, 0.0, 1.0,    0.0, 0.0, 1.0, // v0-v1-v2-v3 front
        1.0, 0.0, 0.0,    1.0, 0.0, 0.0,    1.0, 0.0, 0.0,    1.0, 0.0, 0.0, // v0-v3-v4-v5 right
        0.0, 1.0, 0.0,    0.0, 1.0, 0.0,    0.0, 1.0, 0.0,    0.0, 1.0, 0.0, // v0-v5-v6-v1 up
        -1.0, 0.0, 0.0, -1.0, 0.0, 0.0, -1.0, 0.0, 0.0, -1.0, 0.0, 0.0, // v1-v6-v7-v2 left
        0.0,-1.0, 0.0,    0.0,-1.0, 0.0,    0.0,-1.0, 0.0,    0.0,-1.0, 0.0, // v7-v4-v3-v2 down
        0.0, 0.0,-1.0,    0.0, 0.0,-1.0,    0.0, 0.0,-1.0,    0.0, 0.0,-1.0    // v4-v7-v6-v5 back
    ]);
    //顶点索引
    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
    ]);
```

五． 必要函数和功能说明

矩阵求逆

Matrix4.setInverse0f(m)：使自身成为 m 的逆矩阵。

Matrix4.transpose()：对自身进行转置操作

绘制函数：

keydown()：

在按键按下时做出相应，并更新相应变量。最后调用 draw 函数把整个模型画出来。

drawBox()：接受参数，绘制手臂的一个部件。

width，height，depth 分别表示部件的宽度，高度和深度。viewMatrix 表示视图矩阵，u_MvpMatrix 表示模型视图的投影矩阵，u_NormalMatrix 表示计算变换后的法向量矩阵。

drawSegments()：

对每一个部件，都定义一组顶点数据，并存储一个单独的缓冲区对象中，各个部件共享法向量和索引值。

处理模型矩阵：

```
var g_matrixStack = []; ：存储矩阵的数组。
function pushMatrix(m) //将矩阵压入栈
{
    var m2 = new Matrix4(m);
    g_matrixStack.push(m2);
}
function popMatrix() //将矩阵弹出栈
{
    return g_matrixStack.pop();
}
```

五．设计不足

1. 系统选择的对象简单，并将一些功能简化例如五指的动作统一为弯曲，没有体现出差异，所以实现过程也不是很难。

2. 模型是以简单的几个图形代替，表现上说得过去，但是与仿真相差太大。

六．小组分工及自评

小组成员	自评分数	分工情况
王磊	98	模型制作，文档撰写
赵婷	97	PPT 制作，文档修改