

REPORT



소속학과: 소프트웨어학과
학 번 : 32191976
이 름 : 배건우

```
"use strict";
var canvas;
var gl;
var NumVertices =36;
var points =[];
var colors =[];
var rotationMatrix;
var rotationMatrixLoc;
var angle =0.0;
var axis =[0,0,1];
var trackingMouse =false;
var trackballMove =false;
var lastPos =[0,0,0];
var curx,cury;
var startX,startY;
function trackballView(x,y) {
    var d,a;
    var v =[];
    v[0]=x;
    v[1]=y;
    d =v[0]*v[0]+v[1]*v[1];
    if (d <1.0)v[2]=Math.sqrt(1.0 -d);
    else{
        v[2]=0.0;
        a =1.0 /Math.sqrt(d);
        v[0]*=a;
        v[1]*=a;
    }
    return v;
}
//마우스를 움직일때 함수
function mouseMotion(x,y) {
    var dx,dy,dz;
    var curPos =trackballView(x,y);
    if (trackingMouse){
        dx =curPos[0]-lastPos[0];
        dy =curPos[1]-lastPos[1];
        dz =curPos[2]-lastPos[2];
        if (dx ||dy ||dz){
            angle =-0.1 *Math.sqrt(dx *dx +dy *dy +dz *dz);
            axis[0]=lastPos[1]*curPos[2]-lastPos[2]*curPos[1];
            axis[1]=lastPos[2]*curPos[0]-lastPos[0]*curPos[2];
            axis[2]=lastPos[0]*curPos[1]-lastPos[1]*curPos[0];
            lastPos[0]=curPos[0];
            lastPos[1]=curPos[1];
        }
    }
}
```

```

        lastPos[2]=curPos[2];
    }
}

    render();
}
//마우스를 눌렀을때 실행되는 함수
function startMotion(x,y) {
    trackingMouse =true;
    startX =x;
    startY =y;
    curx =x;
    cury =y;
    lastPos =trackballView(x,y);
    trackballMove =true;
}
//마우스를 뺐을경우 실행되는 함수
function stopMotion(x,y) {
    trackingMouse =false;
    if (startX !=x ||startY !=y){
    else{
        angle =0.0;
        trackballMove =false;
    }
}
}
window.onload =function init() {
    canvas =document.getElementById("gl-canvas");
    //예외처리
    gl =WebGLUtils.setupWebGL(canvas);
    if (!gl){
        alert("WebGL isn't available");
    }

    colorCube();
    gl.viewport(0,0,canvas.width,canvas.height);
    gl.clearColor(1.0,1.0,1.0,1.0); //캠버스의 색을 흰색으로 지정
    gl.enable(gl.DEPTH_TEST);
    // shader 로드 및 속성 버퍼 초기화
    var program =initShaders(gl,"vertex-shader","fragment-shader");
    gl.useProgram(program);
    var cBuffer =gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER,cBuffer);
    gl.bufferData(gl.ARRAY_BUFFER,flatten(colors),gl.STATIC_DRAW);
    var vColor =gl.getAttribLocation(program,"vColor");
    gl.vertexAttribPointer(vColor,4,gl.FLOAT,false,0,0);
    gl.enableVertexAttribArray(vColor);
    var vBuffer =gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER,vBuffer);

```

```

gl.bufferData(gl.ARRAY_BUFFER,flatten(points),gl.STATIC_DRAW);
var vPosition =gl.getAttribLocation(program,"vPosition");
gl.vertexAttribPointer(vPosition,4,gl.FLOAT,false,0,0);
gl.enableVertexAttribArray(vPosition);
rotationMatrix =mat4();
rotationMatrixLoc =gl.getUniformLocation(program,"r");

gl.uniformMatrix4fv(rotationMatrixLoc,false,flatten(rotationMatrix));
//마우스를 눌렀을때
canvas.addEventListener("mousedown",function(event) {
    var x =(2 *event.clientX)/canvas.width -1; //x지정
    var y =(2 *(canvas.height -event.clientY))/canvas.height -1;
//y지정
    startMotion(x,y);
});
//마우스를 뗐을때
canvas.addEventListener("mouseup",function(event) {
    var x =(2 *event.clientX)/canvas.width -1; //x지정
    var y =(2 *(canvas.height -event.clientY))/canvas.height -1;
//y지정
    stopMotion(x,y);
});
//마우스를 움직일때
canvas.addEventListener("mousemove",function(event) {
    var x =(2 *event.clientX)/canvas.width -1; //x지정
    var y =(2 *(canvas.height -event.clientY))/canvas.height -1;
//y지정
    mouseMotion(x,y);
});
render();
};
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);
}
function quad(a,b,c,d) {
    var vertices =[
        vec4(-0.5,-0.5,0.5,2.0),
        vec4(-0.5,0.5,0.5,2.0),
        vec4(0.5,0.5,0.5,2.0),

```

```

    vec4(0.5,-0.5,0.5,2.0),
    vec4(-0.5,-0.5,-0.5,1.0),
    vec4(-0.5,0.5,-0.5,1.0),
    vec4(0.5,0.5,-0.5,1.0),
    vec4(0.5,-0.5,-0.5,1.0),
  ];
  var vertexColors =[
    [0.0,0.0,0.0,1.0],// black
    [1.0,0.0,0.0,1.0],// red
    [1.0,1.0,0.0,1.0],// yellow
    [0.0,1.0,0.0,1.0],// green
    [0.0,0.0,1.0,1.0],// blue
    [1.0,0.0,1.0,1.0],// magenta
    [0.0,1.0,1.0,1.0],// cyan
    [1.0,1.0,1.0,1.0],// white
  ];
  // quad 두 개의 삼각형으로 분할해야 한다.
  // quad 인덱스의 삼각형
  //정점의 인덱스에 의해 할당된 정점 색상
  var indices =[a,b,c,a,c,d];
  for (var i =0; i <indices.length; ++i){
    points.push(vertices[indices[i]]);
    //보간된 색상사용
    //단색면 사용
    colors.push(vertexColors[a]);
  }
}
//reder()함수 정의
function render() {
  gl.clear(gl.COLOR_BUFFER_BIT |gl.DEPTH_BUFFER_BIT);
  if (trackballMove){
    axis =normalize(axis);
    rotationMatrix =mult(rotationMatrix,rotate(angle,axis));

  gl.uniformMatrix4fv(rotationMatrixLoc,false,flatten(rotationMatrix));
}
  gl.drawArrays(gl.TRIANGLES,0,NumVertices);
  requestAnimationFrame(render);
}

```

<소스코드 분석>

마우스의 시작위치와 현재 움직이는 위치를 나타내는 curx와 cury변수, startx, starty변수를 선언해주고 마우스가 움직일 때인지를 나타내는 변수 trackballMove를 기본값 false로 설정해줬습니다.

canvas를 document객체의 getElementById메소드로 gl-canvas라는 id를 받아온후 예외처리를해주고 배경을 흰색으로 지정하고 볼수있는영역을 캔버스의 전체면적으로 설정을 해줍니다. 그리고 shader로드와 버퍼 초기화코드를 작성해준후 마우스를 클릭하거나 떼거나, 움직일 때 발생하는 이벤트리스너를 작성해줍니다. 마우스를 움직일 경우에는 canvas.addEventListener("mousemove")을, 마우스를 클릭했을 경우에는 canvas.addEventListener("mousedown")을, 마우스를 떼을 경우에는 canvas.addEventListener("mouseup")를 통해 각각 이벤트가 발생했을 경우에 차례대로 mouseMotion(), startMotion(), stopMotion()을 실행합니다. 마우스가 클릭되었을 때는 위에서 설정해준 변수 trackballmove를 true로 바꿔주고 startx,y와 curx,y를 설정해줍니다. 마우스를 떼을 경우에는 trackballmove를 false로, 움직일 경우에는 true로 바꿔주는 형식으로 코드를 작성해주었습니다.

함수 colorCube같은 경우에는 총 6면체의 6개의면을 quad함수를 통해 작성해줍니다. quad함수에는 총 8개의 정점과 8개의 color를 설정을 해줍니다. 보간된색상과 단색면 사용을 위해 points.push()와 color.push()메소드를 이용해줍니다. 마지막으로 render()함수를 통해 캔버스에 설정해준 육면체를 그려줍니다.