Трехмерная графика

x

y

z

В любой трехмерной сфере нужно знать угол обзора, точку и направление. В нашем проекте угол обзора совпадает с осью Oz к центру координат.

Point.js

class Point {

constructor(x=0,y=0,z=0){

this.x = x;

this.y = y;

this.z = z;

}

}

Edge.js

class Edge {

constructor(p1=0, p2=0){

this.p1 = p1;

this.p2 = p2;

}

}

Surface.js

class Surface{

constructor(points=[], edges=[], polygons=[]){

this.points = points;

this.edges = edges;

this.polygons = polygons;

}

}

Math3D.js

class Math3D{

constructor({WIN}){

this.WIN = WIN;

}

xs(point){

const zs = this.WIN.CENTER.z;

const z0 = this.WIN.CAMERA.z;

const x0 = this.WIN.CAMERA.x;

return ((point.x-x0)/(point.z-z0)\*(z’-z0)+x0);

// ys написать аналогично

ys(point){

const zs = this.WIN.CENTER.z;

const z0 = this.WIN.CAMERA.z;

const y0 = this.WIN.CAMERA.y;

return ((point.y-y0)/(point.z-z0)\*(z’-z0)+y0);

}

graph3DTemplate.js

Template.prototype.graph3DTemplate = () => `<canvas id = ‘graph3D’></canvas>`;

Graph3D.js

class Graph3D extends Component {

constructor(options){

super(options);

const WIN = {

LEFT:-5,

BOTTOM:-5,

WIDTH:10,

HEIGHT:10,

CENTER: new Point(0,0,-30),

CAMERA: new Point(0,0,-50);

}

this.graph = new Graph({id:’graph3D’, width:600, height: 600, WIN});

this.math3D = new Math3D({WIN});

this.scene = this.cube();

this.renderScene();

}

cube(){

return new Surface([

new Point(10,10,10),

new Point(10,-10,10),

new Point(-10,-10,10),

new Point(-10,10,10),

new Point(10,10,-10),

new Point(10,-10,-10),

new Point(-10,-10,-10),

new Point(-10,10,-10)],

[new Edge(0,1), //написать все 12 ребер]);

}

renderScene(){

this.scene.points.forEach(point => this.graph.point(this.math3D.xs(point), this.math3D.ys(point));

this.scene.edges.forEach(edge => {

const point1= this.scene.points[edge.p1];

const point2 = this.scene.points[edge.p2];

this.graph.line(this.math3D.xs(point1), this.math3D.ys(point1)),

this.math3D.xs(point2), this.math3D.ys(point2));

});

}

}

//Со звездочкой нарисовать поверхность тор (бублик) или сферу

graph3DTemplate.js

Graph3D.js

Graph3D

Surface.js

Edge.js

Points.js

entities

Math3D.js

Math3D

components

modules

js