К entities и points добавим Polygon.

// Полигоны будут четырехугольниками

class Polygon{

constructor(points=[],color=’#ff0000’){

this.points=points;

this.color=color;

this.distance = 0;

}

}

class Surface{

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

this.points = points;

this.edges = edges;

this.polygons = polygons;

}

}

// Сложность в отрисовке полигонов заключается в необходимости соблюдения правильного порядка их рисования (сначала задние, потом передние)

// Проблема оптимизации: в среднем примерно половина полигонов в сцене перекрыта другими полигонами, грубо говоря пользователь их не видит

//Алгоритм рисования полигонов(Алгоритм художника): Для каждого полигона сцены расчитывается расстояние от полигона до камеры из которой мы смотрим, потом все полигоны сортируются от этого расстояния, от меньшего к большему и полученный массив полигонов мы рисуем

Graph3D.js

renderScene(){

//дописывается к уже существующему методу

this.math3D.calcDistance(this.scene, this.WIN.CAMERA, ‘distance’);

this.math3D.sortByArtistAlgorithm(this.scene);

this.scene.polygons.forEach(polygon => {

const points = polygon.points.map(index => new Point(this.math3D.xs(this.scene.points[index], this.math3D.ys(this.scene.points[index]));

this.graph.polygon(points, color);

});

}

Math3D.js

calcDistance(surface, endPoint, name){

surface.polygons.forEach(polygon => {

let x = 0, y = 0, z = 0;

polygon.points.forEach(index => {

x += surface.points[index].x;

y += surface.points[index].y;

z += surface.points[index].z;

});

x /= polygon.points.length;

y /= polygon.points.length;

z /= polygon.points.length;

polygon[name]= Math.sqrt((endPoint.x-x)\*\*2+(endPoint.y-y)\*\*2+(endPoint.z-z)\*\*2;

});

}

sortByArtistAlgorithm(surface){

surface.polygons.sort((a,b) =>

(a.distance < b.distance) ? 1 : -1);

}

// Домашка: для каждой существующей поверхности добавить полигоны и нарисовать их, сделать чекбоксы, которые позволят рисовать либо точки, либо ребра, либо полигоны, добавить селектбокс в котором будет отрисовываемая поверхность. Из поверхностей должны быть: сфера, куб, тор, бутылка Клеина. У кубика его поверхности должны быть закрашены разными цветами

// Задание со звездочкой: не рисовать невидимые полигоны

// Задание с 2 звездами: одновременно на сцене рисовать несколько фигур