Задача отображение нескольких фигур на сцене. Проблема заключается, в том, что отрисовка полигонов не связана с расположением фигур. Способ решения: Все полигоны всех фигур добавляются в один массив, который сортируется по расстоянию до камеры и полигоны рисуются соответственно массиву.

Освещение и тени это две разные задачи. Задача: освещенность в зависимости от расстояния до источника света => нужно поменять цвет полигона в нужную сторону. Рассматриваем задачу точечных источников света.

Новый класс в entities под названием Light

class Light extends Point {

constructor(x,y,z, lumen = 1000){

super(x,y,z);

this.lumen = lumen;

}

}

class Polygon{

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

this.points = points;

this.color = this.hexToRgb(color);

this.distance = 0;

this.lumen = 1;

}

hexToRgb(hex){

// Регулярные выражения это написанные по специальным правилам строки определяющие правила сравнения входящих значений(тяжелая вычислительная операция). Их применяют в сравнении вводимых пользователем данных на соответствие маски.

const result = /^#?([a-f\d]{2})([a-f\d]){2}([a-f\d]){2}$/i.exec(hex);

return result ? {

r:parseInt(result[1],16),

g:parseInt(result[2],16),

b:parseInt(result[3],16)

} : {r:0, g:0, b:0};

}

rgbToHex(r,g,b){

return `rgb(${r},${g},${b})`;

}

}

Math3D.js

calcIllumination(distance, lumen){

const illum = distance ? lumen / distance\*\*2: 1;

return illum>1 ? 1: illum;

}

Graph3D.js

constructor(){

this.LIGHT = new Light(-40,15,0,15000);

}

renderScene(){

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

this.math3D.calcDistance(this.scene, this.LIGHT, ‘lumen’);

this.math3D.sortByArtistAlgorithm(this.scene);

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

const points = … ;

const lumen = this.math3D.calcIllumination(polygon.lumen, this.LIGHT.lumen);

let {r,g,b} = polygon.color;

r = Math.round(r\*lumen);

g = Math.round(g\*lumen);

b = Math.round(b\*lumen);

this.graph.polygon(points, polygon.rgbToHex(r,g,b));

});

}

// Написать весь код, сделать чекбокс для вращения источника света вместе со сценой

// \*\* Добавить несколько источников света

// \*\*\* Уметь добавлять не белый источник света

// На зачет нужны следующие поверхности 2 порядка: гиперболический цилиндр, параболический цилиндр, элиптический цилиндр, однополосной гиперболоид, двуполосной гиперболоид, эллипсоид, конус, сфера, эллиптический параболоид, гиперболический параболоид, тор, бутылка Клеина.