# Задание:

Разработать программу, позволяющую изображать точку и ее проекции на пространственном и комплексном чертежах с возможностью изменять координаты точки.

# Содержание экрана:

* Пространственный чертеж с изображением точки A, ее проекций A1, A2, A3 линий проекционной связи (в виде параллелепипеда);
* комплексный чертеж с изображением проекций A1, A2, A3) и линий связи;
* ползунковые переключатели для интерактивного изменения координат (x, y, z) точки A.

# Динамика:

При изменении координат точки A должны изменяться соответствующие проекции и сама точка A на пространственном и комплексном чертежах.

# Структура решения.

При изменении пользователем координат (x, y, z) точки с помощью ползунковых переключателей в программу передаются управляющие параметры, исходя из которых, изменяются чертежи. Схема изменения выглядит так:

* Отображение точки на комплексном и пространственном чертежах
* Прорисовка нового состояния пространственного чертежа
* Прорисовка осей
* Прорисовка точки
* Прорисовка проекций точки A1, A2, A3 и линий проекционной связи в виде параллелепипеда
* Прорисовка нового состояния комплексного чертежа
* Прорисовка осей
* Прорисовка проекций A1, A2, A3 и линий связи

# Обзор и анализ методов решения

Пусть имеется точка с координатам (x, y, z).

## 1. Изображение проекций на комплексном чертеже

Картинка

Построение проекции точки по заданным координатам осуществляется в следующей последовательности:

* на оси абсцисс от начала координат откладывают отрезок ОАх=х
* через точку Аx проводят перпендикуляр к оси Ох, на котором с учетом знаков откладывают отрезки АхА1=у (получаем A1-горизонтальную проекцию) AxA2=z (получаем А2-фронтальную проекцию)
* через А2 проводят перпендикуляр к оси Oz
* с учетом знака откладывают отрезок АzА3=у (получаем А3-профильную проекцию)

## 2. Изображение точки, ее проекций и проекционных линий связи в виде параллелепипеда

Картинка

На координатной плоскости XZ изобразим пространственное изображение. Ось Y образовывает угол с осью X 45 градусов. Точка A(x, y, z) на координатной плоскости XZ будет иметь координаты (x1, y1), где

x1=x-y\*cos(45);

y1=z-y\*sin(45);

Горизонтальная проекция имеет координаты A1 (x1, y1-z).

Фронтальная проекция имеет координаты A2 (х1+y\*cos(45), y1 + y\*cos(45)). Профильная проекция имеет координаты A3 (x1-x, y1).

Построение параллелепипеда осуществляется следующим образом:

-соединить точку А с полученными точками A1, A2, Тз.

-из A1 провести прямую к оси ОХ параллельную OX параллельную к оси OY

-из A2 опустить перпендикуляры на оси OX, OZ

-из A3 провести прямую к оси OZ параллельную OY и к оси OY параллельную OZ

# Описание реализации применяемых методов.

Переменные, которые используются в реализации: x3D, y3D, z3D - координаты точки, вводимые пользователем

х, у - координаты точки на пространственном.

X0, Y0 - величина является началом координат в пространственном и комплексном чертежах (X0, Y0) (B координатах устройства).

## 1. Преобразование координат точки в координаты устройства для построение пространственного чертежа.

Система координат устройства имеет вид:

Картинка

Исходя из этого координаты точки в координатах устройства будут вычисляться таким образом:

x = X0 - Math.round((x3D - y3D \* Math.cos(ALPHA)) \* SCALE);

y = Y0 - Math.round((z3D - y3D \* Math.sin(ALPHA)) \* SCALE);

А проекции будут иметь координаты:

A1x= x-y\*cosα

A1y= -y\*sinα

A2x=-y\*cosα

A2y= z-y\*sinα

A3x= x

A3y= z

## 2. Преобразование координат точки в координаты устройства для построение комплексного чертежа

Аналогично предыдущему пункту, вследствие того что устройство имеет свою систему координат, нужно преобразовать координаты. Проекции будут иметь координаты:

A1(X0-A1x, Y0-A1y);

A2(X0-A2x, Y0-A2y);

A3(X0-A3x, Y0-A3y);

# Описание программного обеспечения

Программа на Javascript:

'use strict'**;**

/\*\*

\* Класс: Точка

\*/

***function*** Point**(**drawing**,** x**,** y**)** **{**

***this*.**x **=** x**;**

***this*.**y **=** y**;**

/\*\* Рисование точки \*/

***this*.**drawPoint **=** ***function*(**pointName**,** color**)** **{**

drawing**.**drawCircle**(*this*.**x**,** ***this*.**y**,** 5**,** color**);**

***if*** **(**pointName**)** ***this*.**drawText**(**pointName**,** color**);**

***return*** ***this*;**

**};**

/\*\* Рисование текста \*/

***this*.**drawText **=** ***function*** **(**text**,** color**)** **{**

drawing**.**drawTextXY**(**text**,** ***this*.**x**,** ***this*.**y**,** color**);**

***return*** ***this*;**

**};**

/\*\* Рисование линии \*/

***this*.**drawLine **=** ***function*** **(**pointTo**,** color**)** **{**

drawing**.**drawLineXY**(*this*.**x**,** ***this*.**y**,** pointTo**.**x**,** pointTo**.**y**,** color**);**

***return*** ***this*;**

**};**

**}**

/\*\*

\* Класс: Drawing

\*/

***function*** Drawing**(**canvas**,** type**)** **{**

***var*** ALPHA **=** 45**;**

***var*** \_self **=** ***this*;**

/\*\* Создание 3D точки \*/

this.createPoint3D = function(x3D, y3D, z3D) {

var x = getX(x3D, y3D);

var y = getY(z3D, y3D);

return new Point(this, x, y);

};

/\*\* Создание 2D точки \*/

this.createPoint2D = function(x2D, y2D) {

var x = getX(x2D, 0);

var y = getY(y2D, 0);

return new Point(this, x, y);

};

/\*\* X - координата в 2D \*/

function getX(x3D, y3D) {

var v1 = X0 - (x3D - Math.round(y3D \* Math.cos(ALPHA))) \* SCALE; //Криво по Y

var v2 = X0 - Math.round((x3D - y3D \* Math.cos(ALPHA)) \* SCALE); //Нормально по Y

return v2;

}

/\*\* Y - координата в 2D \*/

function getY(z3D, y3D) {

var v1 = Y0 - (z3D - Math.round(y3D \* Math.sin(ALPHA))) \* SCALE;

var v2 = Y0 - Math.round((z3D - y3D \* Math.sin(ALPHA)) \* SCALE);

return v2;

}

/\*\* Рисование текста \*/

this.drawTextXY = function(text, x, y, color) {

canvas.fillStyle = color ? color : COLOR\_TEXT;

canvas.font = 'normal 15pt Arial';

canvas.fillText(text, x, y);

};

/\*\* Рисование линии \*/

this.drawLineXY = function(x1, y1, x2, y2, color) {

canvas.beginPath();

canvas.moveTo(x1, y1);

canvas.lineTo(x2, y2);

canvas.lineWidth = 2;

canvas.strokeStyle = color ? color : COLOR\_LINE;

canvas.stroke();

};

/\*\* Рисование окружности \*/

this.drawCircle = function(x, y, radius, color) {

canvas.fillStyle = color ? color : COLOR\_POINT;

canvas.beginPath();

canvas.arc(x, y, radius, 0, Math.PI\*2, false);

canvas.closePath();

canvas.fill();

};

/\*\* Очистка канвы \*/

this.clearCanvas = function() {

canvas.clearRect(MIN\_X, MIN\_Y, MAX\_X, MAX\_Y);

};

/\*\* Построение осей координат \*/

this.drawAxis = function() {

this.clearCanvas();

//Рисование осей

if (type == 'dim') {

var y1 = this.createPoint3D(0, -100, 0);

var y2 = this.createPoint3D(0, 100, 0);

y1.drawLine(y2, COLOR\_AXIS);

}

this.drawLineXY(X0, MIN\_Y, X0, MAX\_Y, COLOR\_AXIS);

this.drawLineXY(MIN\_X, Y0, MAX\_X, Y0, COLOR\_AXIS);

//Заголовки осей

var size = 30;

var titles = {

dim: [{title: 'X', x: MIN\_X, y: Y0}, {title: '-X', x: MAX\_X-size, y: Y0},

{title: 'Z', x: X0, y: MIN\_Y+size}, {title: '-Z', x: X0, y: MAX\_Y-size},

{title: 'Y', x: MAX\_X-size\*3, y: MAX\_Y-size}, {title: '-Y', x: MIN\_X+size\*2, y: MIN\_Y+size}],

cmplx: [{title: 'X', x: MIN\_X, y: Y0}, {title: 'Y1', x: MAX\_X-size, y: Y0},

{title: 'Z', x: X0, y: MIN\_Y+size}, {title: 'Y', x: X0, y: MAX\_Y-size}]

};

$.each(titles[type], function(index, value) {

\_self.drawTextXY(value.title, value.x, value.y, COLOR\_AXIS\_TITLE);

});

};

}

'use strict'**;**

/\*\* App Module \*/

***var*** app **=** angular**.**module**(**'myApp'**,** **[**'ngSanitize'**]);**

app**.**controller**(**'MyCtrl'**,** ***function*** MyCtrl**(**$scope**)** **{**

$scope**.**subject **=** 'КГ'**;**

$scope**.**lab **=** 'Лаба 1 (Точка в 3-D пространстве)'**;**

$scope**.**notSupport **=** 'Браузер не поддерживает Canvas'**;**

//Канва

***var*** cnvDim **=** document**.**getElementById**(**'canvasDimensional'**).**getContext**(**"2d"**);**

***var*** cnvCmplx **=** document**.**getElementById**(**'canvasComplex'**).**getContext**(**"2d"**);**

$**(**'#canvasDimensional'**).**attr**(**'width'**,** MAX\_X**).**attr**(**'height'**,** MAX\_Y**);**

$**(**'#canvasComplex'**).**attr**(**'width'**,** MAX\_X**).**attr**(**'height'**,** MAX\_Y**);**

$**(**'#drawBlock'**).**css**(**'width'**,** BLOCK\_WIDTH**);**

$**(**'#sliderBlock'**).**css**(**'width'**,** SLIDER\_WIDTH**);**

/\*\* Инициализация слайдеров \*/

***function*** initSliders**()** **{**

$**(**'#sliderX'**).**slider**({**

range**:** 'max'**,** min**:** **-**10**,** max**:** 10**,** value**:** **-**7**,**

slide**:** ***function*(**event**,** ui**)** **{**

repaintDrawing**(**ui**.**value**,** null**,** null**);**

**}**

**});**

$**(**'#sliderY'**).**slider**({**

range**:** 'max'**,** min**:** **-**10**,** max**:** 10**,** value**:** **-**5**,**

slide**:** ***function*(**event**,** ui**)** **{**

repaintDrawing**(**null**,** ui**.**value**,** null**);**

**}**

**});**

$**(**'#sliderZ'**).**slider**({**

range**:** 'max'**,** min**:** **-**10**,** max**:** 10**,** value**:** **-**6**,**

slide**:** ***function*(**event**,** ui**)** **{**

repaintDrawing**(**null**,** null**,** ui**.**value**);**

**}**

});

}

/\*\* Перерисовка чертежей \*/

function repaintDrawing(valX, valY ,valZ) {

valX = valX != null ? valX : $('#sliderX').slider('value');

valY = valY != null ? valY : $('#sliderY').slider('value');

valZ = valZ != null ? valZ : $('#sliderZ').slider('value');

//Отображение значений ползунков

$scope.valX = valX;

$scope.valY = valY;

$scope.valZ = valZ;

angularApply($scope);

//Построение Пространственного чертежа

drawDimensional(valX, valY, valZ);

//Построение Комплексного чертежа

drawComplex(valX, valY, valZ);

}

/\*\* Построение Пространственного чертежа \*/

function drawDimensional(valX, valY ,valZ) {

var drwDim = new Drawing(cnvDim, 'dim');

//Очистка канвы и построение осей

drwDim.drawAxis();

//Проекция точки A

var point0 = drwDim.createPoint3D(0, 0, 0).drawPoint();

var pointAx = drwDim.createPoint3D(valX, 0, 0).drawPoint('Ax');

var pointAy = drwDim.createPoint3D(0, valY, 0).drawPoint('Ay');

var pointAz = drwDim.createPoint3D(0, 0, valZ).drawPoint('Az');

var pointA1 = drwDim.createPoint3D(valX, valY, 0).drawPoint('A1');

var pointA2 = drwDim.createPoint3D(valX, 0, valZ).drawPoint('A2');

var pointA3 = drwDim.createPoint3D(0, valY, valZ).drawPoint('A3');

point0.drawLine(pointAy).drawLine(pointAz).drawLine(pointAx);

pointAz.drawLine(pointA3).drawLine(pointA2);

pointAy.drawLine(pointA3).drawLine(pointA1);

pointAx.drawLine(pointA2).drawLine(pointA1);

//Точка A

var pointA = drwDim.createPoint3D(valX, valY, valZ).drawPoint('A', COLOR\_POINT\_A);

pointA.drawLine(pointA1).drawLine(pointA2).drawLine(pointA3);

}

/\*\* Построение Комплексного чертежа \*/

function drawComplex(valX, valY ,valZ) {

var drwCmplx = new Drawing(cnvCmplx, 'cmplx');

//Очистка канвы и построение осей

drwCmplx.drawAxis();

var pointA1 = drwCmplx.createPoint2D(valX, -valY).drawPoint('A1');

var pointA2 = drwCmplx.createPoint2D(valX, valZ).drawPoint('A2');

var pointA3 = drwCmplx.createPoint2D(-valY, valZ).drawPoint('A3');

var pointAy = drwCmplx.createPoint2D(0, -valY).drawPoint('Ay');

var pointAy1 = drwCmplx.createPoint2D(-valY, 0).drawPoint('Ay1');

pointAy.drawLine(pointAy1).drawLine(pointA1);

pointA3.drawLine(pointAy1).drawLine(pointA2);

pointA1.drawLine(pointA2);

}

//Start

initSliders();

repaintDrawing();

});

# Пример выполнения программы

