

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

по дисциплине:

«Алгоритмические основы современной компьютерной графики»

на тему: «Реализация программ построения и преобразования двумерных объектов»

Выполнил: студент гр. ИТ-11,
Петров Н.Н.
Принял: доцент каф. «ИТ»,
к.ф.-м.н., доц.,
Иванов В.В.

Гомель 2021

Цель работы: изучить геометрические преобразования двумерных объектов (перенос, масштабирование, поворот).

Задание.

Для выбранного варианта необходимо:

1. Создать матрицу ключевых точек для каждой фигуры, образующей замкнутый контур. По координатам ключевых точек построить изображение заданной фигуры, используя 2D график.
2. Провести перенос фигуры на расстояние вдоль горизонтали и вертикали. Построить изображение преобразованной фигуры.
3. Провести неоднородное и однородное масштабирование заданной фигуры. Построить изображение преобразованной фигуры.
4. Провести поворот заданной фигуры. Построить изображение преобразованной фигуры.

Вариант 13

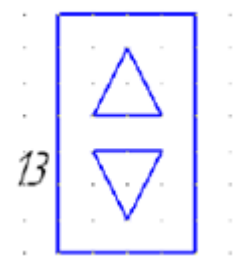


Рисунок 1 – Вариант задания

Теоретические сведения

Перенос

Параллельный перенос объекта сводится к перемещению всех его точек на одно и то же расстояние D в одном и том же направлении, заданном определённым вектором \vec{v} . Если вектор имеет длину D , то операция переноса может быть реализована путём сложения всех точек объекта с вектором \vec{v} . При такой операции сохраняются расстояния между точками и углы между отрезками, а также, отрезки прямых перейдут в отрезки прямых. Математическая модель переноса: для каждой точки $P(x,y)$, которая перемещается в новую точку $P'(x',y')$, сдвигаясь на Dx единиц по оси X и на Dy по оси Y :

$$x' = x + Dx, y' = y + Dy.$$

Определим векторы-строки: $P = (x \ y)$, $P' = (x' \ y')$, $T = (Dx \ Dy)$, тогда уравнение переноса можно записать в виде: $(x' \ y') = (x \ y) + (Dx \ Dy)$, или в матричной форме: $P' = P + T$.

Масштабирование

Масштабирование объекта можно реализовать путём умножения координат всех его точек на некоторое число. Пусть имеются точки с координатами (x_1, y_1) и (x_2, y_2) , над которыми выполняется такое преобразование. Результатом будут новые точки с координатами $(S_x x_1, S_y y_1)$ и $(S_x x_2, S_y y_2)$. Если $S_x = S_y = S_0$, то обе точки переместятся вдоль прямых, проходящих через саму точку и начало координат. При этом расстояние между новыми точками будет в S_0 раз отличаться от прежнего, но углы между отрезками сохранятся. Если коэффициент масштабирования $S_0 > 1$, соответствующий отрезок растягивается, а если меньше, то сжимается. Кроме того, при таком преобразовании объект смещается. В случае, когда $S_x \neq S_y$, расстояния между точками изменятся неравномерно, поскольку растяжения в горизонтальном и вертикальном направлениях будут различными. Углы между отрезками также не сохранятся. Математическая модель масштабирования при масштабировании в S_x раз по оси X и в S_y раз по оси Y , имеет вид: $x' = x S_x$, $y' = y S_y$.

Определив матрицу масштабирования S в виде:

$$S = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix}$$

можно записать в матричной форме:

$$(x' y') = (x y) \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix}$$

или $P' = P S$.

Поворот

Вращения в плоскости перемещают точки по дуге окружности, центр которой находится в начале координат. Рассмотрим сначала движение одной точки при повороте на угол α (положительным является направление против часовой стрелки), т.е. поворот радиус-вектора на угол (рисунок 1.2). Пусть точка располагалась на расстоянии r от начала координат, а её радиус-вектор составлял угол β с осью абсцисс. Тогда координаты точки определяются формулами:

$$x = r \cos(\beta), \quad y = r \sin(\beta)$$

После поворота вектор будет составлять угол $(\beta + \alpha)$, а новые координаты точки будут определяться соотношениями:

$$x' = r \cos(\beta + \alpha) = r \cos(\beta) \cos(\alpha) - r \sin(\beta) \sin(\alpha) = x \cos(\alpha) - y \sin(\alpha),$$

$$y' = r \sin(\beta + \alpha) = r \sin(\beta) \cos(\alpha) + r \cos(\beta) \sin(\alpha) = x \sin(\alpha) + y \cos(\alpha).$$

При таком преобразовании сохраняются расстояния между точками и углы между отрезками.

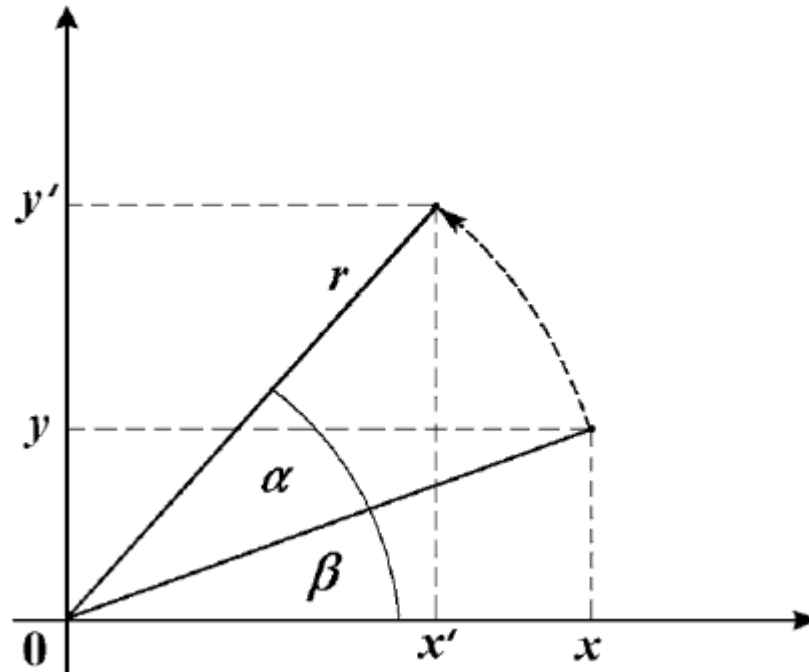


Рисунок 2 — Поворот на плоскости.

Преобразование поворота можно записать в матричной форме:

$$(x' y') = (x y) \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

или, определив матрицу поворота как:

$$R = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

в краткой форме соответствует записи:

$$P' = P R.$$

Листинг программы

```
//lab 1 AOSKG
//(c) Kaitanova A.N., IT-11, GSTU
//2016
clear// delete all variables
clf()//delete all objects
mx=[0,0,4,4,0,1,2,3,1,1,2,3,1]
my=[0,7,7,0,0,3,1,3,3,4,6,4,4]
plot2d(0,0,-1,"012", " ", [-4,-4,4,4])
```

```

xtitle("Лабораторная работа №1: Построение и преобразование двумерных объектов","Ось X","Ось Y")
// find x,y coordinates of the center of polygon
cX=(max(mx)-min(mx))./2
cY=(max(my)-min(my))./2
// move the center of polygon to [0,0]
cmx=mx-cX
cmy=my-cY
// plot moved poligon
pl=plot(cmx,cmy,"cyan")
//plot rotated polygon
h=uicontrol("style","pushbutton","string","Turn L","Callback",...
"rotplot(cmx,cmy,%pi/18)","Callback_Type",2,...
"position",[0, 30, 70, 20])
h=uicontrol("style","pushbutton","string","Turn R","Callback",...
"rotplot(cmx,cmy,-%pi/18)","Callback_Type",2,...
"position",[0, 10, 70, 20])
function [nmx, nmy]=mrot(lmx, lmy, f)
    nmx = ((lmx.*cos(f))-(lmy.*sin(f)))
    nmy = ((lmx.*sin(f))+(lmy.*cos(f)))
endfunction
function [mx, my]=rotplot(xx, yy, fi)
    [mx,my]=mrot(xx,yy,fi)
    delete(gcf());
    pl1=plot(mx,my,'cyan')
    [cmx,cmy]=return(mx,my)
endfunction
//povorot
h=uicontrol("style","pushbutton","string","Zoom(+)","Callback",...
"zoom(cmx,cmy,1.1)","Callback_Type",2,"position",[70, 30, 75, 20])
h=uicontrol("style","pushbutton","string","Zoom(-)","Callback",...
"zoom(cmx,cmy,0.9)","Callback_Type",2,"position",[70, 10, 75, 20])
function [mx, my]=zoom(xx, yy, c)
    mx = xx.*c
    my = yy.*c
    delete(gcf());
    pl1=plot(mx,my,'cyan')
    [cmx,cmy]=return(mx,my)
endfunction
//zoom
h=uicontrol("style","pushbutton","string","Left","Callback",...
"sdplot(cmx,-0.1,1)","Callback_Type",2,"position",[145,30,70,20])
h=uicontrol("style","pushbutton","string","Right","Callback",...
"sdplot(cmx,0.1,1)","Callback_Type",2,"position",[145,10,70, 20])
h=uicontrol("style","pushbutton","string","Up","Callback",...
"sdplot(cmy,0.1,2)","Callback_Type",2,"position",[215,30,70,20])
h=uicontrol("style","pushbutton","string","Down","Callback",...
"sdplot(cmy,-0.1,2)","Callback_Type",2,"position",[215,10,70,20])
function [c]=sdplot(cc, sd, tp)
    c=cc+sd;
    delete(gcf());
    if tp == 1 then
        pl = plot(c,cmy,'cyan');
        [cmx] = return(c);
    else
        pl = plot(cmx,c,'cyan');
        [cmy] = return(c);
    end
endfunction
//perenos
h=uicontrol("style","pushbutton","string","+ Sx>Sy","Callback",...

```

```

"nsplot(cmx,cmy,1,1,1.05)","Callback_Type",2,"position",[285, 30, 80, 20])
h=uicontrol("style","pushbutton","string","- Sx<Sy","Callback",...
"nsplot(cmx,cmy,0.9,0.95)","Callback_Type",2,"position",[285, 10, 80, 20])
function [mx, my]=nsplot(xx, yy, sx, sy)
    mx = xx.*sx;
    my = yy.*sy;
    delete(gcf());
    pl = plot(mx,my,'cyan');
    [cmx,cmy] = return(mx,my)
Endfunction

```

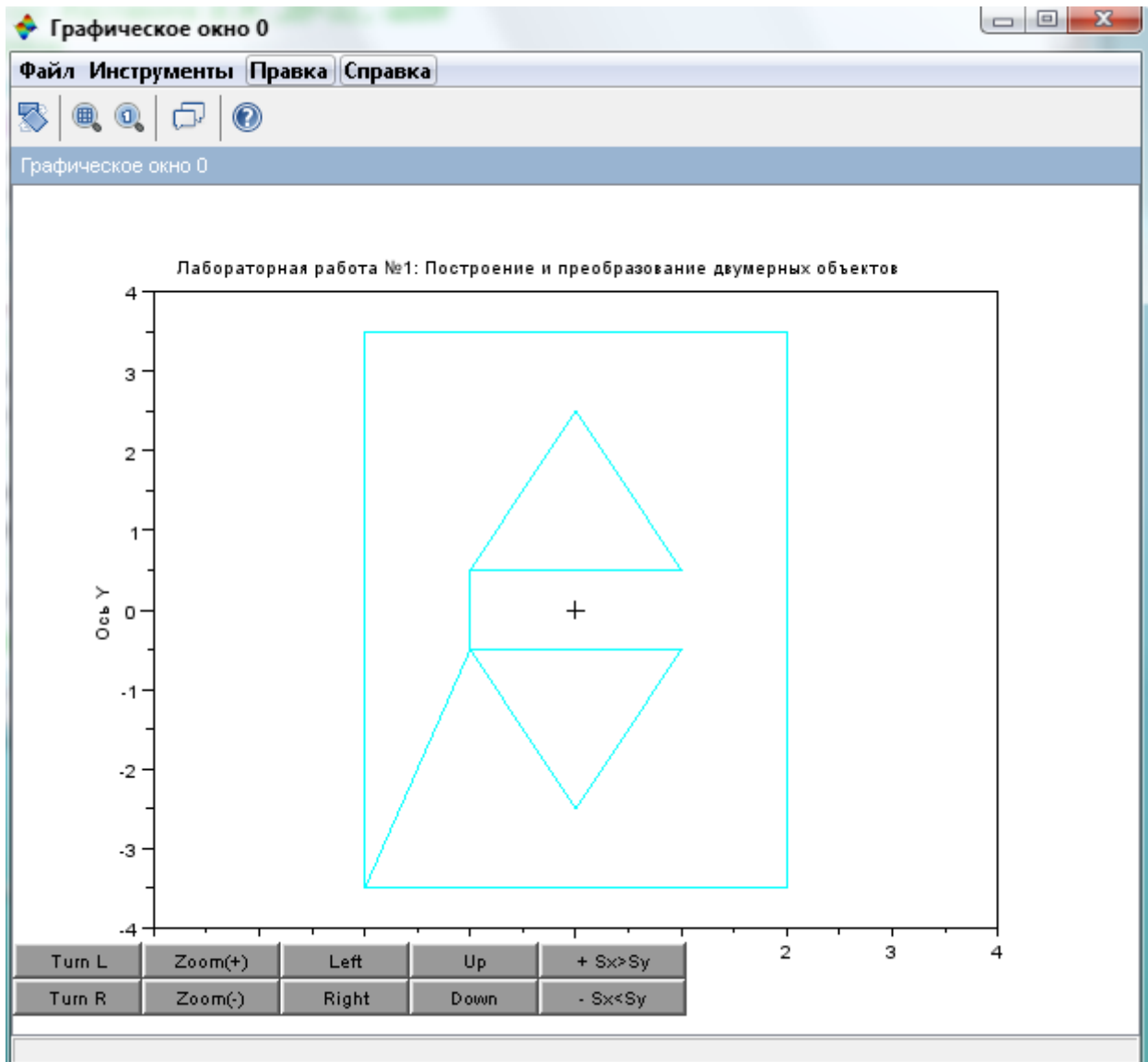


Рисунок 3 – Выполнение программы

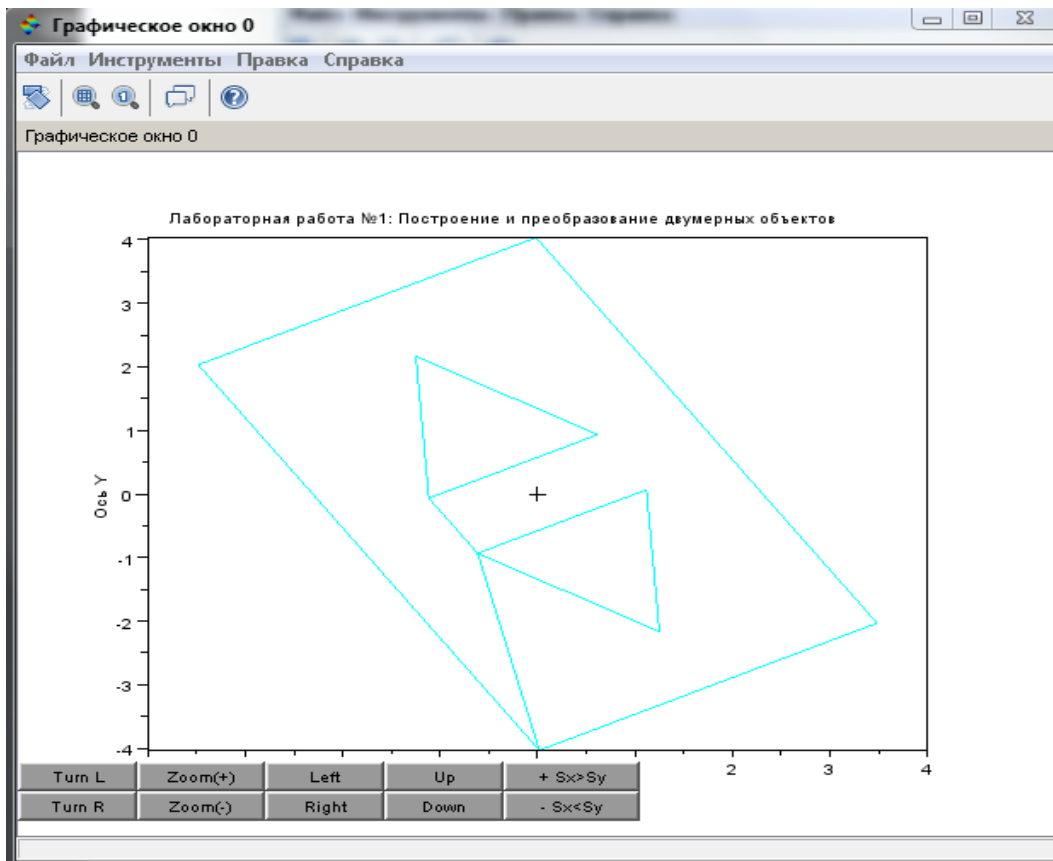


Рисунок 4 - Поворот

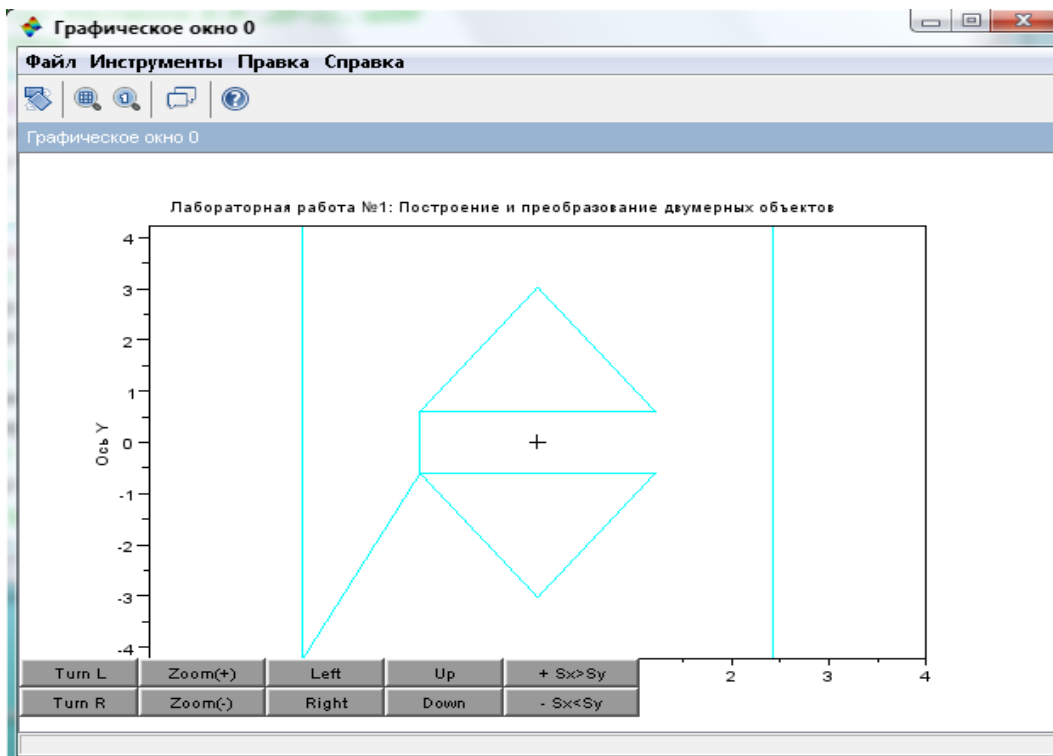


Рисунок 5 - Увеличение

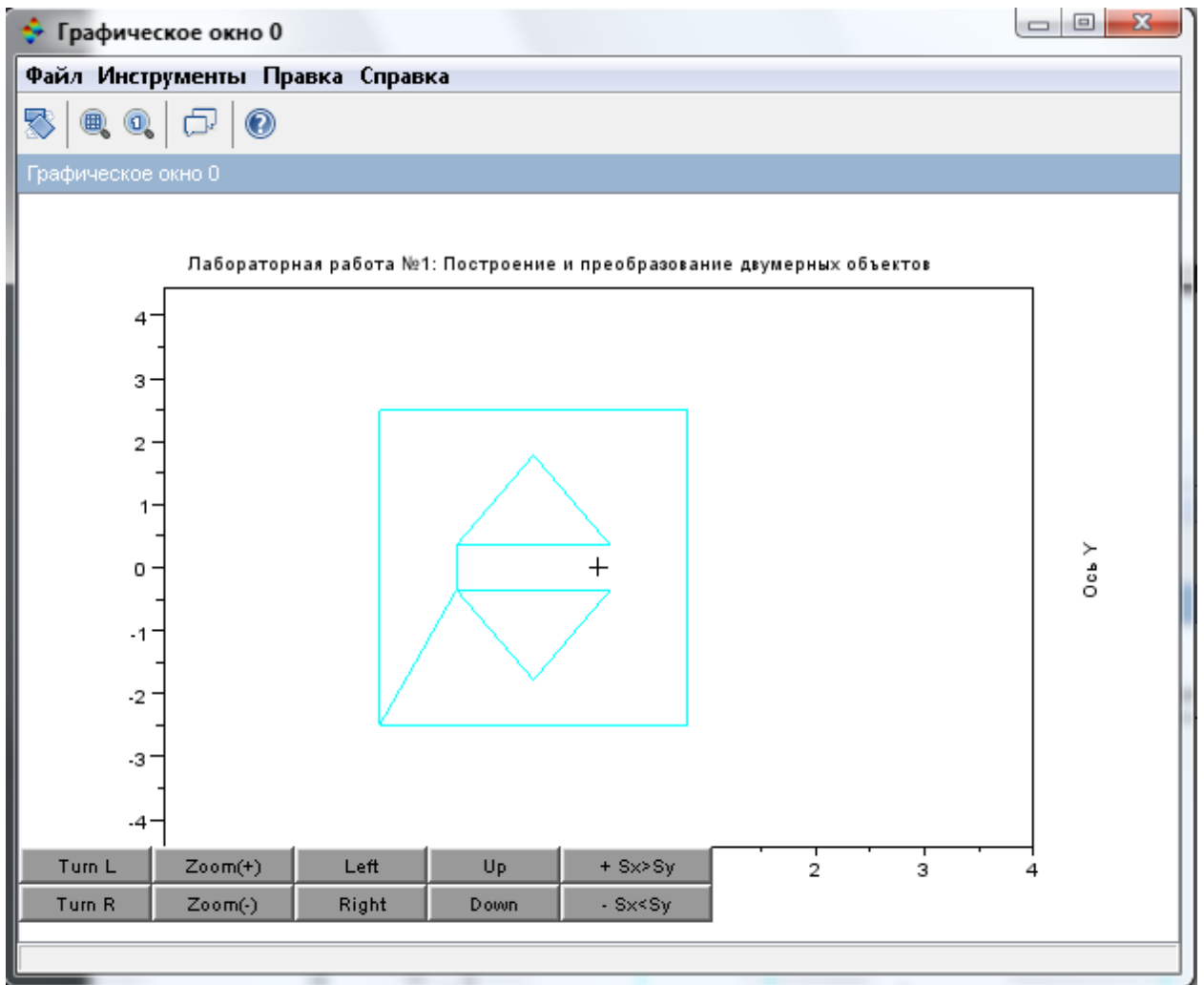


Рисунок 6 – Смещение в сторону

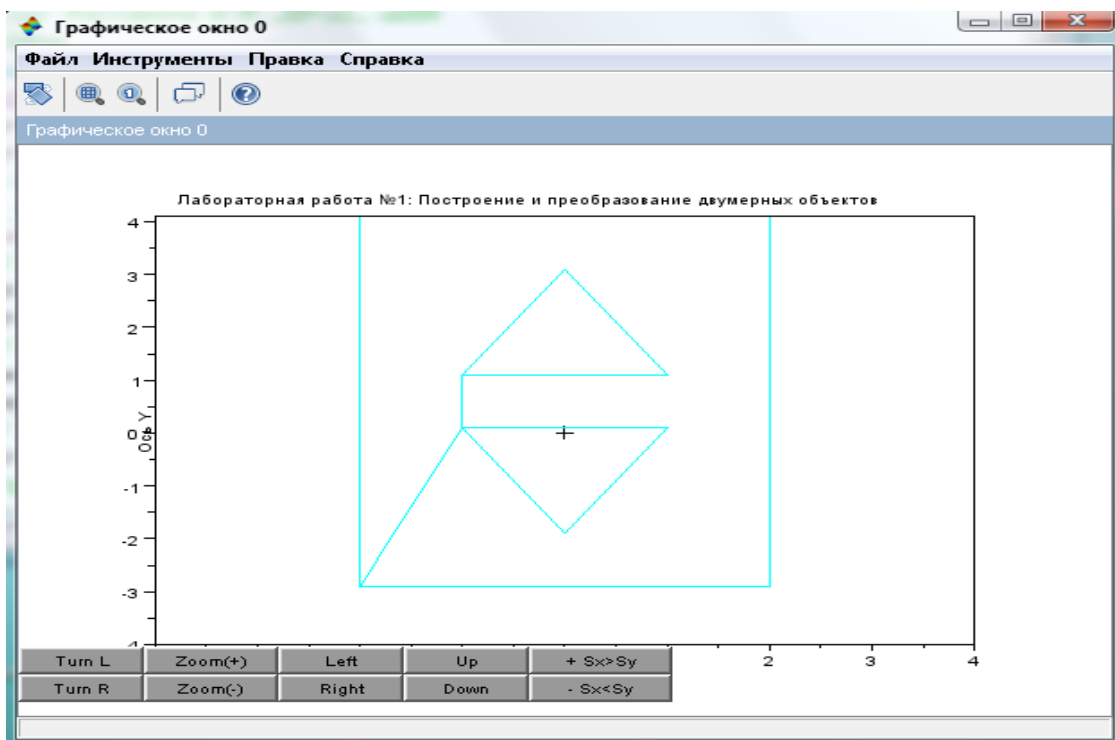


Рисунок 7 – Смещение вверх

Вывод: в ходе лабораторной работы были изучены и освоены на практике алгоритмы преобразования двумерных объектов в среде программирования Scilab, такие как перенос, поворот и масштабирование.

Список использованных источников:

1. Цитринов А.В. Лекция 2. Возможности современных математических пакетов при решении задач математического моделирования физических систем. /2016,89 с.

2. Е.Р. Алексеев, О.В. Чеснокова, Е.А. Рудченко. Scilab. Решение инженерных и математических задач./ Москва ALT Linux; БИНОМ. Лаборатория знаний 2008,260 с.