**Лабораторная работа №6**

Задача: построить произвольный объект в первой четверти системы координат (например, флажок. кол-во вершин объекта должно быть на меньше 7) и организовать следующие преобразования объекта:

- общее преобразование (изменение масштаба, симметричное отражение относительно осей, сдвиг),

- вращение объекта относительно начала координат

- комбинированное преобразование (преобразование общего вида + вращение на произвольный угол относительно начала координат)

Матрица общего преобразования и угол поворота должны вводиться через пользовательский интерфейс, координаты вершин объекта можно указать явно в коде программы

Математическая модель:

-Вращение , где x, y – координаты, a – угол вращения

-Отражение относительно оси x , где x, y – координаты,

-Отражение относительно оси y , где x, y – координаты,

-Перемещение где x, y – координаты, n, m – координаты смещения для x, y

-Увеличение где x, y – координаты, a, b – коэффициенты увеличения

Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Значение | Тип |
| w | Ширина поля canvas | int |
| h | Высота поля canvas | int |
| xy1 | Координата первой точки | int |
| xy2 | Координата второй точки | int |
| xy3 | Координата третьей точки | int |
| xy4 | Координата четвертой точки | int |
| xy5 | Координата пятой точки | int |
| xy6 | Координата шестой точки | int |
| xy7 | Координата седьмой точки | int |
| flag | Список из всех координат флага в начальном положении | int |
| fl1 | Список из всех координат флага в текущем положении |  |
| i | Счетчик цикла, строка в матрице |  |
| temp1 | Подсчитанное значение координаты для x |  |
| temp2 | Подсчитанное значение координаты для y |  |
| otrMatrixX | Матрица для отражения относительно x |  |
| otrMatrixY | Матрица для отражения относительно y |  |

Код:

#возвращение флага в начальную точку

def resetFlag():

    xy1 = [0, 0]

    xy2 = [0, 100]

    xy3 = [50, 100]

    xy4 = [30, 75]

    xy5 = [50, 50]

    xy6 = [15, 50]

    xy7 = [15, 0]

    flag = [xy1, xy2, xy3, xy4, xy5, xy6, xy7]

    return flag

fl1 = resetFlag()

#отрисовка флага

def drawFlag(flag):

    xy1 = [w/2-flag[0][0], h/2-flag[0][1]]

    xy2 = [w/2-flag[1][0], h/2-flag[1][1]]

    xy3 = [w/2-flag[2][0], h/2-flag[2][1]]

    xy4 = [w/2-flag[3][0], h/2-flag[3][1]]

    xy5 = [w/2-flag[4][0], h/2-flag[4][1]]

    xy6 = [w/2-flag[5][0], h/2-flag[5][1]]

    xy7 = [w/2-flag[6][0], h/2-flag[6][1]]

    flag = [xy1, xy2, xy3, xy4, xy5, xy6, xy7]

    for i in range(6):

        canv.create\_line(flag[i][0], flag[i][1], flag[i+1][0], flag[i+1][1])

    canv.create\_line(flag[6][0], flag[6][1], flag[0][0], flag[0][1])

#отражения относительно оси X

def otrX():

    for i in range(7):

        k1 = fl1[i][0] \* otrMatrixX[0][0] + fl1[i][1] \* otrMatrixX[1][0]

        k2 = fl1[i][0] \* otrMatrixX[0][1] + fl1[i][1] \* otrMatrixX[1][1]

        fl1[i][0] = k1

        fl1[i][1] = k2

#отражения относительно оси Y

def otrY():

    for i in range(7):

        temp1 = fl1[i][0] \* otrMatrixY[0][0] + fl1[i][1] \* otrMatrixY[1][0]

        temp2 = fl1[i][0] \* otrMatrixY[0][1] + fl1[i][1] \* otrMatrixY[1][1]

        fl1[i][0] = temp1

        fl1[i][1] = temp2

#Вращение на выбраный угол

def clickOnButton1():

    canv.delete("all")

    grad = int(entryG.get())

    gradInRad = grad \* 3.141592/180

    for i in range(7):

        temp1 = fl1[i][0] \* math.cos(gradInRad) + fl1[i][1] \* (-math.sin(gradInRad))

        temp2 = fl1[i][0] \* math.sin(gradInRad) + fl1[i][1] \* math.cos(gradInRad)

        fl1[i][0] = temp1

        fl1[i][1] = temp2

    drawFlag(fl1)

#Отразить относительно X и отрисовка

def clickOnButton2():

    canv.delete("all")

    otrX()

    drawFlag(fl1)

#Отразить относительно Y и отрисовка

def clickOnButton3():

    canv.delete("all")

    otrY()

    drawFlag(fl1)

#Преобразование на введенную матрицу 3x3

def clickOnButton4():

    canv.delete("all")

    preobrM = entryM.get().split()

    for i in range(7):

        temp1 = fl1[i][0] \* int(preobrM[0]) + fl1[i][1] \* int(preobrM[3]) + int(preobrM[6])

        temp2 = fl1[i][0] \* int(preobrM[1]) + fl1[i][1] \* int(preobrM[4]) + int(preobrM[7])

        fl1[i][0] = temp1

        fl1[i][1] = temp2

    drawFlag(fl1)

#Сброс матрицы к начальному положению и отрисовка

def clickOnButton5():

    canv.delete("all")

    global fl1

    fl1 = resetFlag()

    drawFlag(fl1)

Результат работы программы:





