

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

ОТЧЕТ

по лабораторной работе № 1

по дисциплине «Компьютерная графика»

Тема: Исследование математических методов представления и преобразования графических объектов на плоскости и в пространстве

Студентка гр. 1361

Галунина Е.С.

Студентка гр. 1361

Горбунова Д.А.

Студентка гр. 1361

Токарева У.В.

Преподаватель

Колев Г.Ю.

Санкт-Петербург

2023

Цель работы.

Реализовать поворот объемного тела относительно осей координат на заданный угол.

Основные теоретические положения.

В данной работе нашей задачей было осуществить поворот объемного тела (мы остановили свой выбор на пирамиде) на выбранный пользователем угол.

Теоретические сведения, использованные для данной работы представлены на рисунке 1.

Поворот фигуры вокруг оси 'z'

$$T_z = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ -\sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

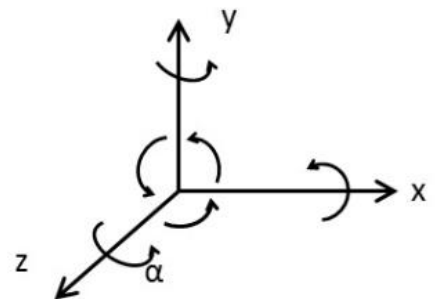
Поворот фигуры вокруг оси 'x'

$$T_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & \sin\beta & 0 \\ 0 & -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Поворот фигуры вокруг оси 'y'

$$T_y = \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma & 0 \\ 0 & 1 & 0 & 0 \\ \sin\gamma & 0 & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

При переходе от 'z' к 'x' ('x' к 'y') вся таблица смещается на один элемент вправо и вниз не выходя за границы матрицы 3x3



- Правосторонняя система координат. Поворот против часовой стрелки, если смотреть из конца оси в начало координат.

Рисунок 1 – Теоретические сведения

Формализация.

Работа выполнена на языке программирования Python. Для визуализации данных трехмерной графики используется библиотека matplotlib.

Также в файле описана реализация функций для работы с матрицами (массивы координат с вершинами треугольника), необходимыми для работы с пирамидой. Для перемещения используются вспомогательные матрицы,

при умножении на которые основной матрицы с координатами пирамиды получается новая матрица с новыми координатами пирамиды, соответствующая выполняемому действию.

Для поворота происходит вычисление середины фигуры и получение вспомогательной матрицы для последующего пересчета основной матрицы с координатами пирамиды.

Для закрашки видимых граней происходит расчет по глубине с последующей закрашкой по приоритету.

Управление треугольником осуществляется через ввод значение угла и оси с клавиатуры.

Экспериментальные результаты.

На рисунке 2 представлен поворот фигуры по оси X.

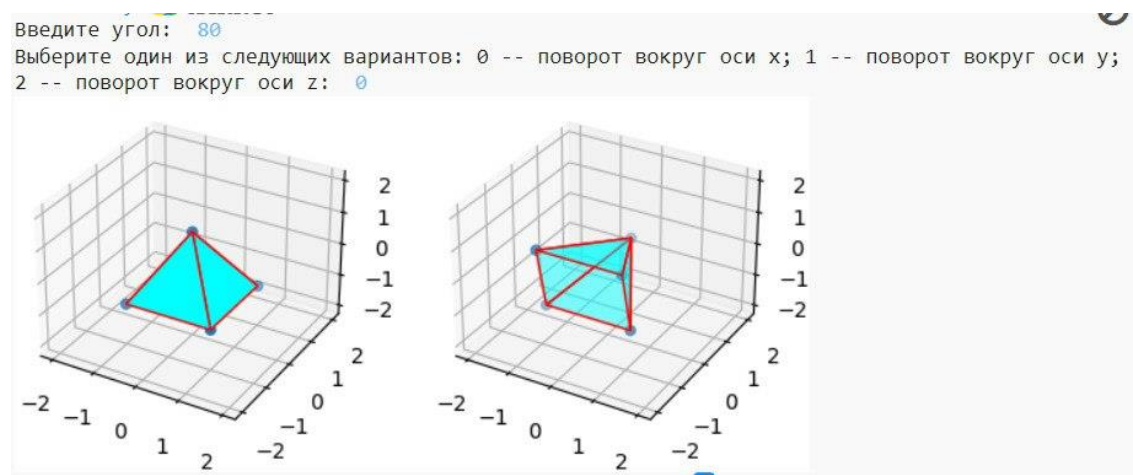


Рисунок 2 – Поворот фигуры по оси X.

На рисунке 3 представлен поворот фигуры по оси Y.

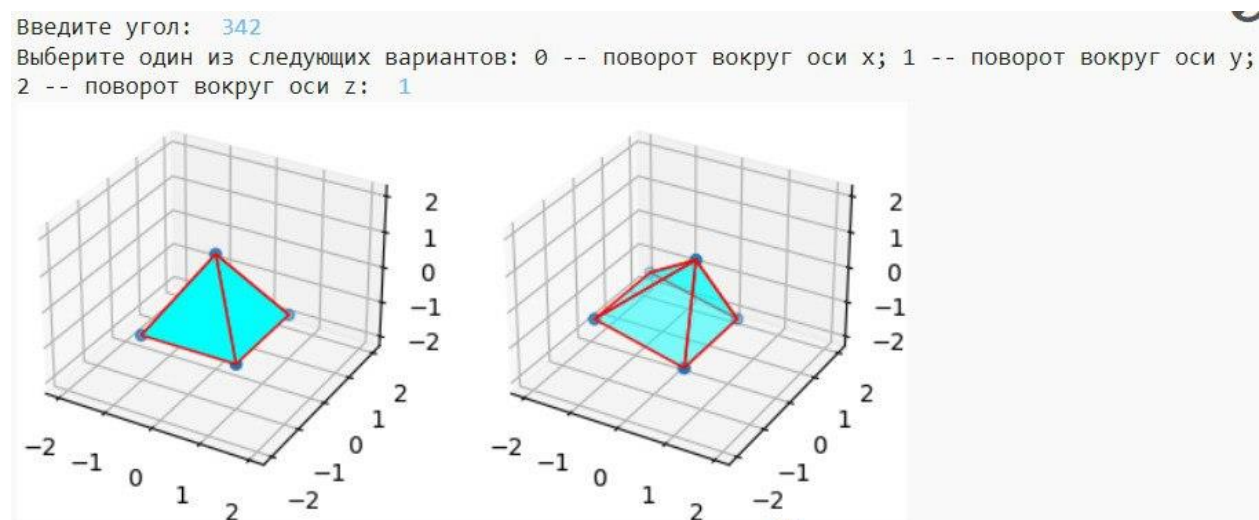


Рисунок 3 – Поворот фигуры по оси Y.

На рисунке 4 представлен поворот фигуры по оси Z.

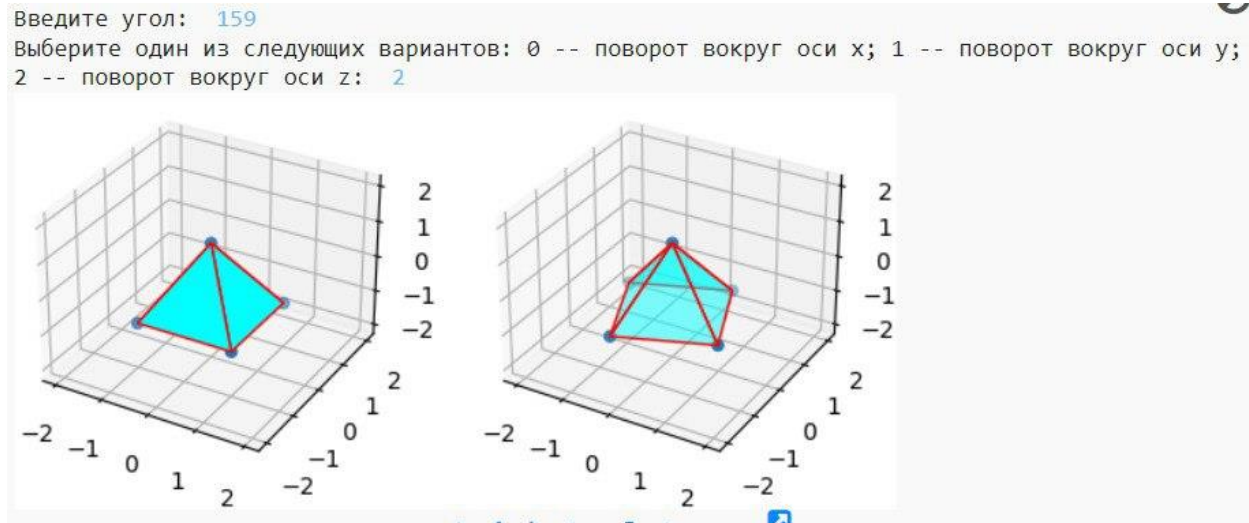


Рисунок 4 – Поворот фигуры по оси Z.

Исходный код программы

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d.art3d import Poly3DCollection
import numpy as np

def rotation_matrix(axis, theta):
    axis = np.asarray(axis)
    axis = axis / np.sqrt(np.dot(axis, axis))
    a = np.cos(theta / 2.0)
    b, c, d = -axis * np.sin(theta / 2.0)
    return np.array([[a*a+b*b-c*c-d*d,      2*(b*c-a*d),
2*(b*d+a*c)],
[2*(b*c+a*d),      a*a+c*c-b*b-d*d,
2*(c*d-a*b)],
[2*(b*d-a*c),      2*(c*d+a*b),
a*a+d*d-b*b-c*c]])

v = np.array([[1, 1, -1], [1, -1, -1], [-1, -1, -1], [-1, 1, -1], [0, 0, 1]])
```

```

# Создаем два подграфика
fig,      (ax1,      ax2)      =      plt.subplots(1,      2,
subplot_kw=dict(projection='3d'))

ax1.scatter(v[:, 0], v[:, 1], v[:, 2])

faces = [[v[j] for j in face] for face in [[0, 1, 4],
[1, 2, 4], [2, 3, 4], [3, 0, 4]]]
ax1.add_collection3d(Poly3DCollection(faces,
facecolors='cyan', linewidths=1, edgecolors='r'))

theta = np.radians(int(input("Введите угол: ")))
select = int(input("Выберите один из следующих
вариантов: 0 -- поворот вокруг оси x; 1 -- поворот
вокруг оси y; 2 -- поворот вокруг оси z: "))

rot = rotation_matrix([select == i for i in range(3)],
theta)
v_rot = np.dot(v, rot.T)

ax2.scatter(v_rot[:, 0], v_rot[:, 1], v_rot[:, 2])

faces_rot = [[v_rot[j] for j in face] for face in [[0,
1, 4], [1, 2, 4], [2, 3, 4], [3, 0, 4]]]
ax2.add_collection3d(Poly3DCollection(faces_rot,
facecolors='cyan',      linewidths=1,      edgecolors='r',
alpha=0.5))

ax1.auto_scale_xyz([-2, 2], [-2, 2], [-2, 2])

```

```
ax2.auto_scale_xyz([-2, 2], [-2, 2], [-2, 2])
```

```
plt.show()
```

Выводы.

В результате выполнения работы нами был реализован код для поворота объемного тела на заданный градус по оси на языке Python.