Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №2 по курсу**

**«Компьютерная графика»**

Выполнил: Лисин Р.С.

Группа: М8О-306Б-20

Преподаватель: Филиппов Г.С.

Дата: \_\_\_\_\_\_\_\_\_\_\_

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

**Лабораторная работа №2**

**Каркасная визуализация выпуклого многогранника. Удаление невидимых линий.**

**Задание:** Разработать формат представления многогранника и процедуру его каркасной отрисовки в ортографической и изометрической проекциях. Обеспечить удаление невидимых линий и возможность пространственных поворотов и масштабирования многогранника. Обеспечить автоматическое центрирование и изменение размеров изображения при изменении размеров окна.

**Вариант 5:** Обелиск (усеченный клин)

**Описание**

Для выполнения данного задания я использовал библиотеку matplotlib для Python. Эта библиотека позволяет довольно просто работать с трехмерными фигурами. Также matplotlib имеет автоматическое масштабирование.

Для отрисовки обелиска я использовал Poly3DCollection. Сначала я задал все 6 сторон обелиска по координатам. Далее я передал эти стороны Poly3DCollection, задал заливку 3D-фигуры и цвет рёбер.

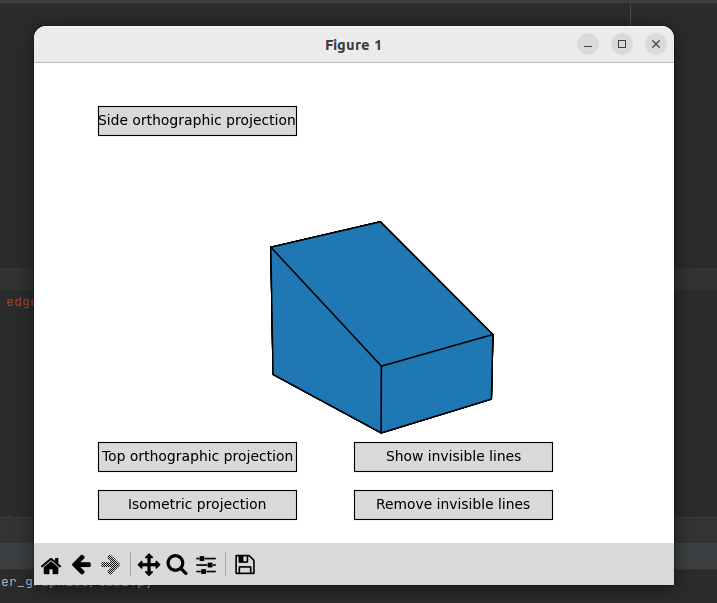
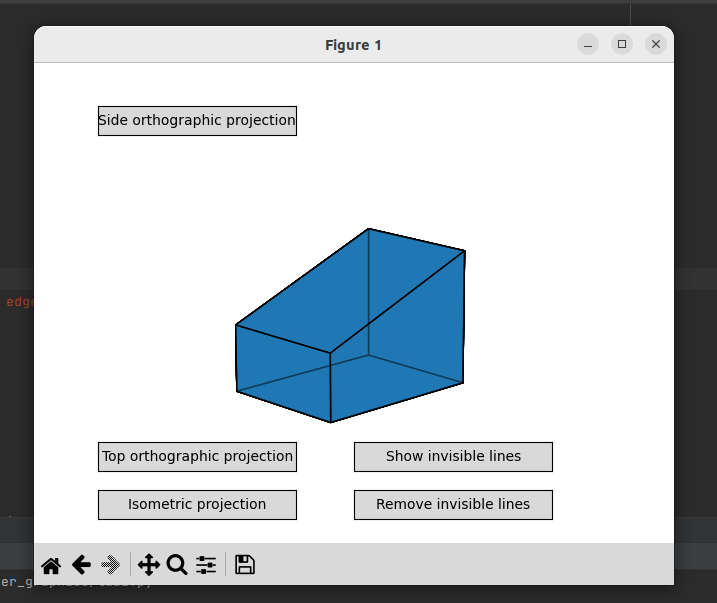
Для удаления невидимых линий я просто меняю заливку фигуры.

Для обработки пользовательского ввода я использовал виджет matplotlib Button, который позволяет легко добавить кнопки в окно.

**Исходный код**

import matplotlib.pyplot as plt  
from mpl\_toolkits.mplot3d.art3d import Poly3DCollection  
from matplotlib.widgets import Button  
  
fig = plt.figure()  
ax = fig.add\_subplot(111, projection='3d')  
  
x = [0, 0, 0, 0]  
y = [2, 8, 8, 2]  
z = [0, 0, 4, 4]  
sides = [list(zip(x, y, z))]  
  
x = [0, 8, 8, 0]  
y = [2, 2, 8, 8]  
z = [0, 0, 0, 0]  
sides.append(list(zip(x, y, z)))  
  
x = [8, 8, 8, 8]  
y = [2, 2, 8, 8]  
z = [8, 0, 0, 8]  
sides.append(list(zip(x, y, z)))  
  
x = [0, 8, 8, 0]  
y = [2, 2, 2, 2]  
z = [0, 0, 8, 4]  
sides.append(list(zip(x, y, z)))  
  
x = [0, 8, 8, 0]  
y = [8, 8, 8, 8]  
z = [0, 0, 8, 4]  
sides.append(list(zip(x, y, z)))  
  
x = [0, 0, 8, 8]  
y = [8, 2, 2, 8]  
z = [4, 4, 8, 8]  
sides.append(list(zip(x, y, z)))  
  
poly = Poly3DCollection(sides, alpha=0.5, edgecolors='black')  
  
ax.add\_collection3d(poly)  
  
ax.set\_xlim(0, 10)  
ax.set\_ylim(0, 10)  
ax.set\_zlim(0, 10)  
  
  
def button\_callback\_remove(event):  
 ax.add\_collection3d(Poly3DCollection(sides, alpha=1, edgecolors='black'))  
 plt.draw()  
  
  
button\_ax\_remove = fig.add\_axes([0.5, 0.05, 0.31, 0.06])  
button\_remove = Button(button\_ax\_remove, "Remove invisible lines")  
button\_remove.on\_clicked(button\_callback\_remove)  
  
  
def button\_callback\_show(event):  
 ax.add\_collection3d(Poly3DCollection(sides, alpha=0.5, edgecolors='black'))  
 plt.draw()  
  
  
button\_ax\_show = fig.add\_axes([0.5, 0.15, 0.31, 0.06])  
button\_show = Button(button\_ax\_show, "Show invisible lines")  
button\_show.on\_clicked(button\_callback\_show)  
  
  
def button\_callback\_isometric(event):  
 ax.view\_init(20, 145)  
 plt.draw()  
  
  
button\_ax\_isometric = fig.add\_axes([0.1, 0.05, 0.31, 0.06])  
button\_isometric = Button(button\_ax\_isometric, "Isometric projection")  
button\_isometric.on\_clicked(button\_callback\_isometric)  
  
  
def button\_callback\_orthographic\_top(event):  
 ax.view\_init(90)  
 plt.draw()  
  
  
button\_ax\_orthographic\_top = fig.add\_axes([0.1, 0.15, 0.31, 0.06])  
button\_orthographic\_top = Button(button\_ax\_orthographic\_top, "Top orthographic projection")  
button\_orthographic\_top.on\_clicked(button\_callback\_orthographic\_top)  
  
  
def button\_callback\_orthographic\_hip(event):  
 ax.view\_init(0, 90)  
 plt.draw()  
  
  
button\_ax\_orthographic\_front = fig.add\_axes([0.1, 0.85, 0.31, 0.06])  
button\_orthographic\_front = Button(button\_ax\_orthographic\_front, "Side orthographic projection")  
button\_orthographic\_front.on\_clicked(button\_callback\_orthographic\_hip)  
  
ax.grid(None)  
ax.axis('off')  
plt.show()

**Результат работы**

****

**Выводы**

Выполнив эту лабораторную работу, я научился рисовать трёхмерные фигуры по координатам сторон на компьютере при помощи библиотеки matplotlib.