

Московский авиационный институт  
(национальный исследовательский университет)

Факультет информационных технологий и прикладной  
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №2 по курсу «Компьютерная графика»

Студент: Ф. А. Иванов  
Преподаватель: Г. С. Филиппов  
Группа: М8О-308Б-19  
Дата: 21.12.2021  
Оценка:  
Подпись:

Москва, 2021

## Лабораторная работа №2

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

**Задача:** 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 3:** Параллелепипед

# 1 Описание

Требуется написать программу, строящую 3D-фигуру - параллелепипед. Размеры фигуры задает пользователь.

## Шаг 1.

Для начала определим какие библиотеки будем использовать. Для отрисовки графики удобно использовать библиотеку *matplotlib*, а именно инструмент *matplotlib.pyplot*. Также потребуется инструмент для создания множества значений. Эту роль возложим на библиотеку *numpy*.

## Шаг 2.

Отрисовку буду проводить с помощью вокселей *ax.voxels*. Это единичные кубообразные фигуры.

## Шаг 3.

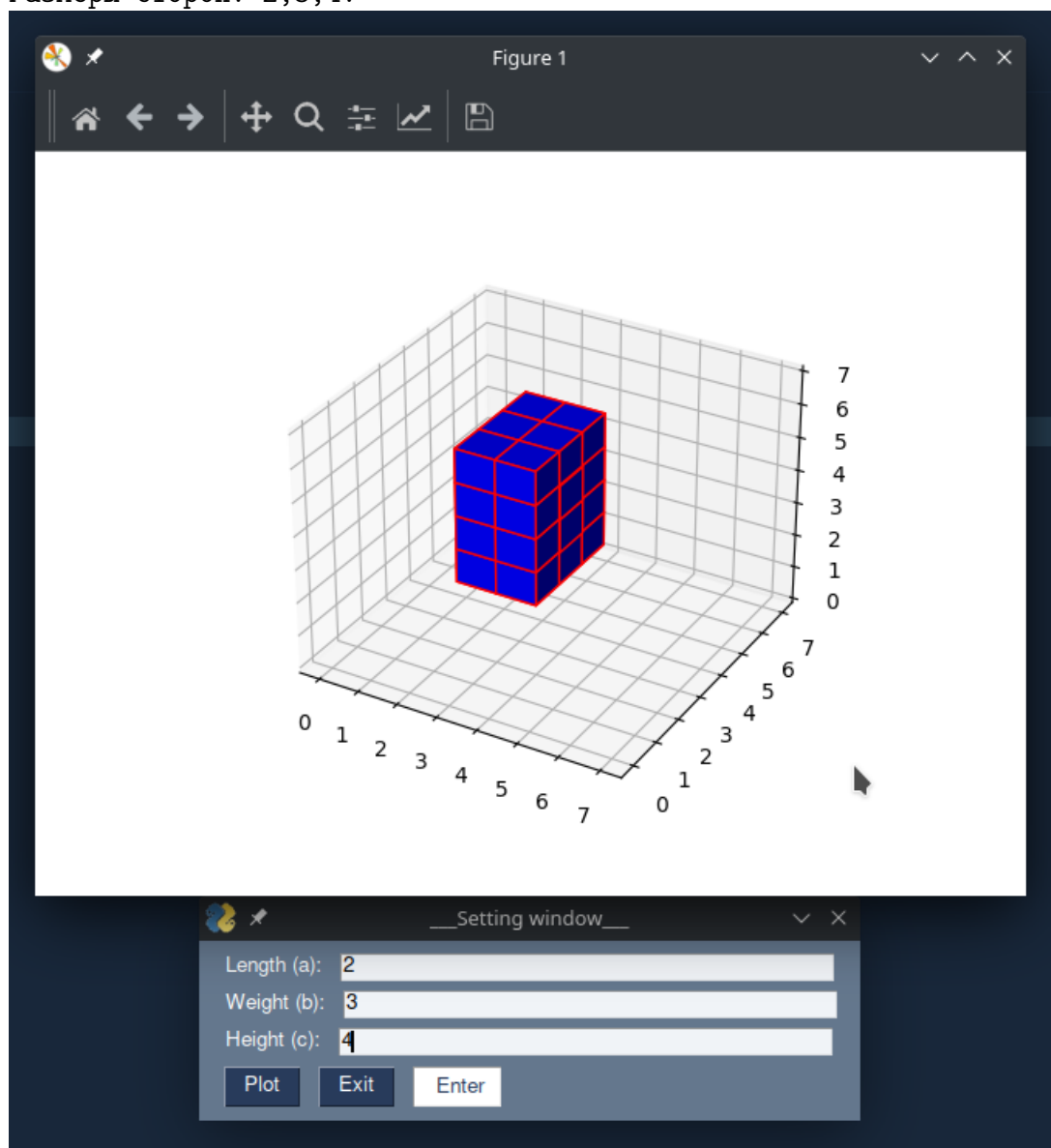
В качестве визуального взаимодействия с пользователем использую библиотеку *PySimpleGUI*.

## 2 Исходный код

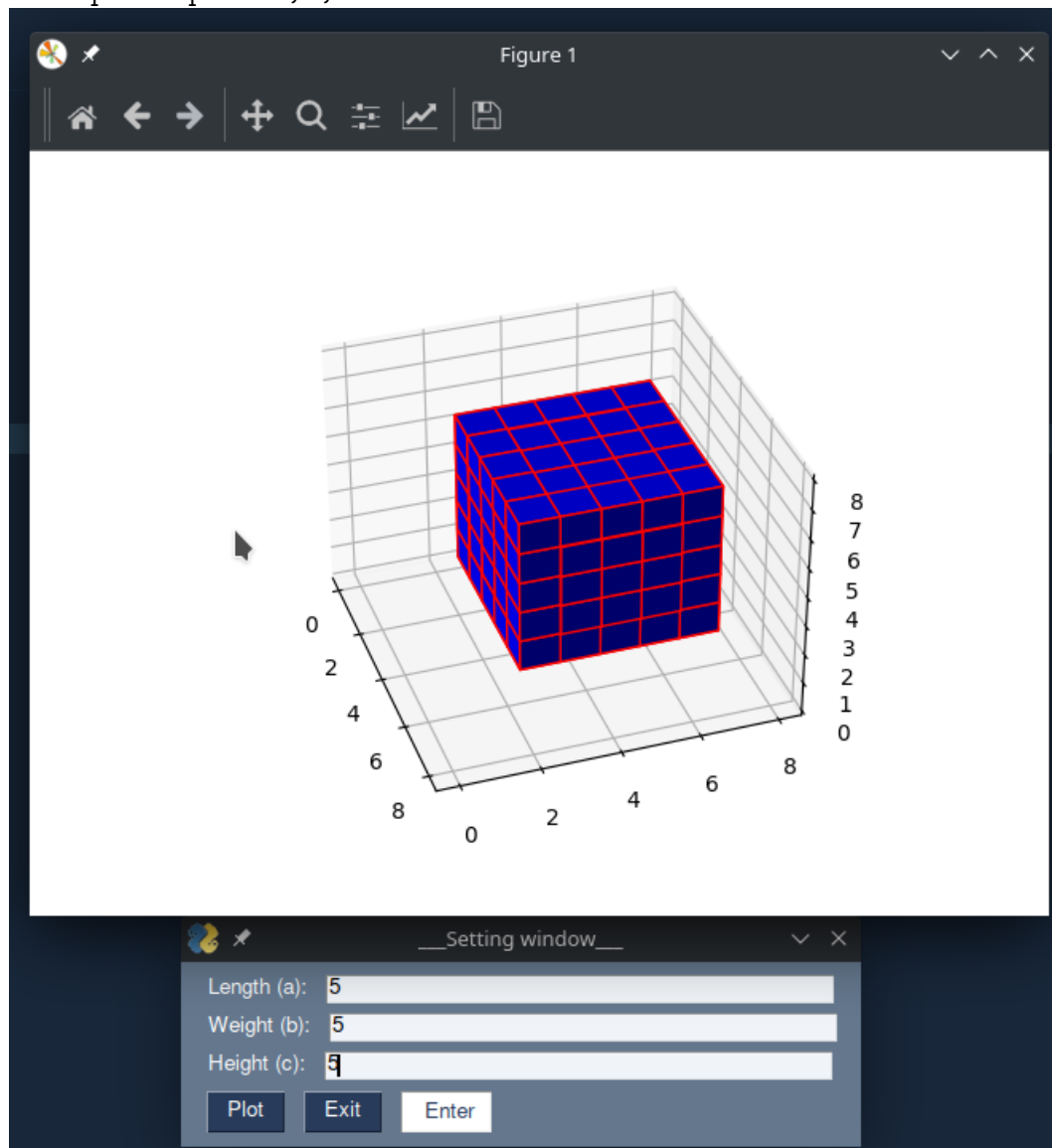
```
1 import PySimpleGUI as sg
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5
6 def draw_plot(a=2, b=3, c=4):
7     list_tmp = [a, b, c]
8     list_tmp.sort()
9     x, y, z = np.indices((list_tmp[2] + 3, list_tmp[2] + 3, list_tmp[2] + 3))
10
11     cube1 = (x < a + 2) & (y < b + 2) & (z < c + 2) & (1 < x) & (1 < y) & (1 < z)
12
13     colors = np.empty(cube1.shape, dtype=object)
14     colors[cube1] = 'blue'
15
16     ax = plt.figure().add_subplot(projection='3d')
17     ax.voxels(cube1, facecolors=colors, edgecolor='r')
18
19     plt.show()
20
21 layout = [
22     [sg.Text('Length (a):'), sg.InputText()],
23     [sg.Text('Weight (b):'), sg.InputText()],
24     [sg.Text('Height (c):'), sg.InputText()],
25     [sg.Button('Plot'), sg.Button('Exit'), sg.Button('Enter')]]
26
27 window = sg.Window('___Setting window___', layout)
28
29 while True:
30     event, values = window.read()
31     if event in (sg.WIN_CLOSED, 'Exit'):
32         break
33     elif event == 'Plot':
34         draw_plot()
35     elif event == 'Enter':
36         draw_plot(int(values[0]), int(values[1]), int(values[2]))
37 window.close()
```

### 3 Пример работы программы

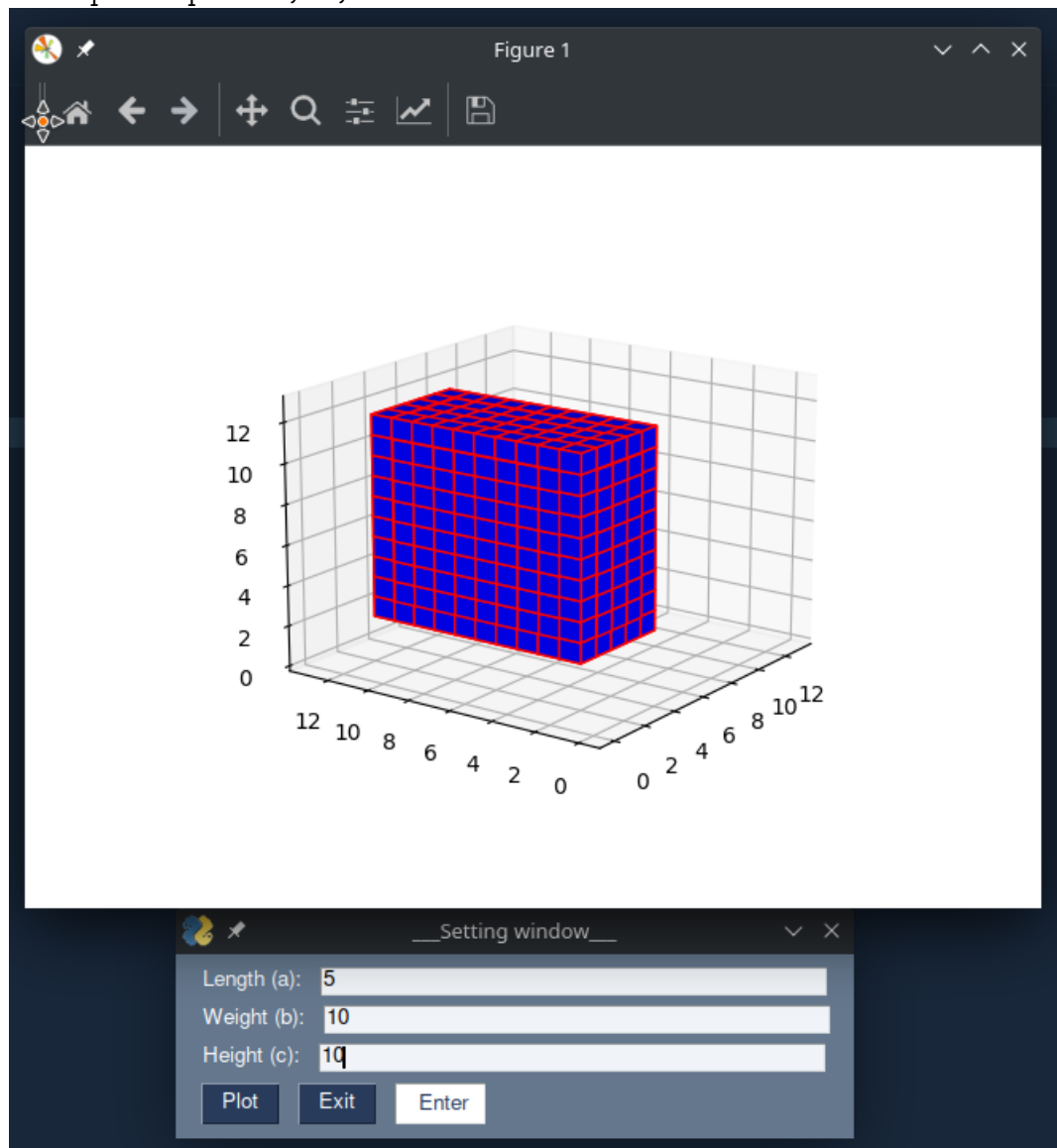
Размеры сторон: 2,3,4:



Размеры сторон: 5,5,5:



Размеры сторон: 5,10,10:



## 4 Выводы

Выполнив 2-ую лабораторную работу по курсу Компьютерной графики, а познакомился с базовыми особенностями построения 3D-фигур. Научился работать с инструментами библиотеки *matplotlib* для построения объемных объектов.



## Список литературы

- [1] *Документация по библиотеке PySimpleGUI*  
URL: <https://pysimplegui.readthedocs.io/en/latest/> (дата обращения: 18.12.2021).
- [2] *Краткое руководство по Matplotlib*  
URL: [https://pyprog.pro/mpl/mpl\\_short\\_guide.html](https://pyprog.pro/mpl/mpl_short_guide.html) (дата обращения: 18.12.2021).
- [3] *Руководство по Numpy*  
URL: <https://numpy.org/doc/stable/reference/> (дата обращения: 18.12.2021).