

## ПОСТРОЕНИЕ 3D-МОДЕЛИ СРЕДСТВАМИ ЯЗЫКА ПРОГРАММИРОВАНИЯ

Сазонов А.В.

Белорусский государственный университет информатики и радиоэлектроники,  
г. Минск, Республика Беларусь

Научный руководитель: Омелькович Е.В. – магистр пед. наук, преподаватель кафедры ИКТ

**Аннотация.** Проанализированы возможности языка программирования Python для создания пространственных 3D-моделей и экспериментальным методом получено изображение простой 3D-модели с помощью библиотеки NumPy.

**Ключевые слова:** 3D-моделирование, 3D-модель, язык программирования Python, библиотека NumPy

**Введение.** 3D-моделирование – это процесс создания трехмерных объектов на компьютере с помощью специальных программных средств. Технологии 3D-моделирования используются в различных областях, например, в таких как проектирование, инженерия, медицина, игровая индустрия, архитектура и во многих других.

В данной статье представлены основные понятия полигонального моделирования для обеспечения процесса построения и манипулирования простой 3D-модели в среде языка программирования Python на базе библиотеки NumPy, которую можно использовать для работы с 3D-моделями.

**Основная часть.** Python – интерпретируемый, объектно-ориентированный, высокоуровневый язык программирования, который имеет достаточно простой синтаксис. Интерпретатор Python и большая стандартная библиотека распространяются свободно.

Для написания программного кода с помощью Python и библиотеки NumPy, необходимо представлять с какими типами элементов необходимо работать в процессе полигонального моделирования. К основным понятиям здесь относятся полигональная сетка и элементы моделирования этой сетки.

Полигональная сетка – это совокупность вершин, ребер, и граней, которые определяют форму многогранного объекта в трехмерной компьютерной графике и объемном моделировании. Объекты, созданные с помощью полигональных сеток, должны хранить разные типы элементов, такие как вершины, ребра, грани, полигоны и поверхности (Рисунок 1):

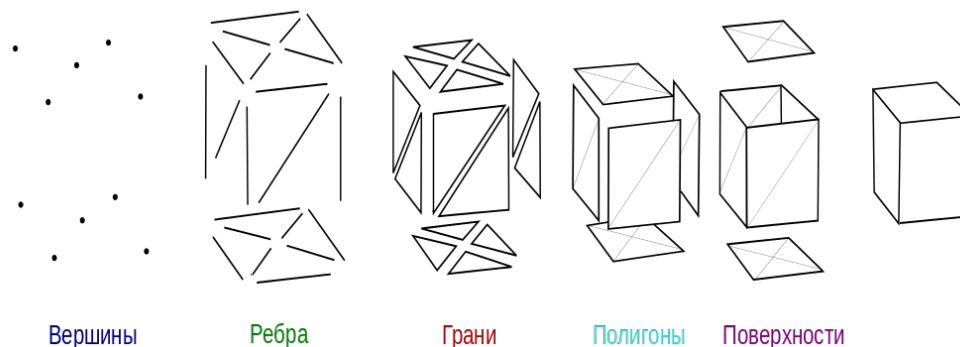


Рисунок 1 – Элементы полигональной сетки

Вершина – это позиция вместе с информацией о цвете, векторе нормали и координатах текстуры. Ребро – это соединение между двумя вершинами. Грань – это замкнутое множество ребер, в котором треугольная грань имеет три ребра, а четырехугольная – четыре. Полигон – это набор компланарных (лежащих в одной плоскости) граней. В системах, которые поддерживают многосторонние грани, полигоны и грани равнозначны.

На примере моделирования изображения усеченной пирамиды рассмотрим возможности одной из библиотек языка Python – библиотеки NumPy. Vertices (вершины) – набор точек, каждая точка имеет три координаты в трехмерном пространстве. Пример программного кода создания точек в пространстве с использованием библиотеки NumPy:

```
# Подключение библиотеки NumPy
import numpy as np
from TaskPlot import plot_create_vertices
# Координаты точек
vertices = np.array(
    [
        [-4, -4, 0],
        [+4, -4, 0],
        [-4, +4, 0],
        [+4, +4, 0],
        [-2, -2, 3],
        [+2, -2, 3],
        [-2, +2, 3],
        [+2, +2, 3],
    ]
)
# Вывод точек в трехмерное пространство
plot_create_vertices(vertices = vertices, isosurf = False)
```

Представление результата исполнения этого кода в пространстве (рисунок 2).

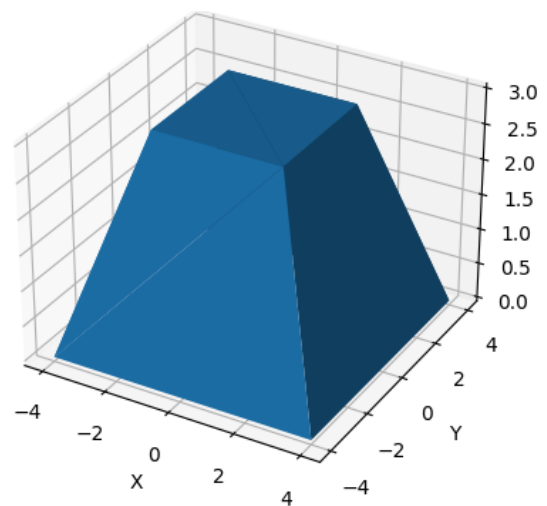
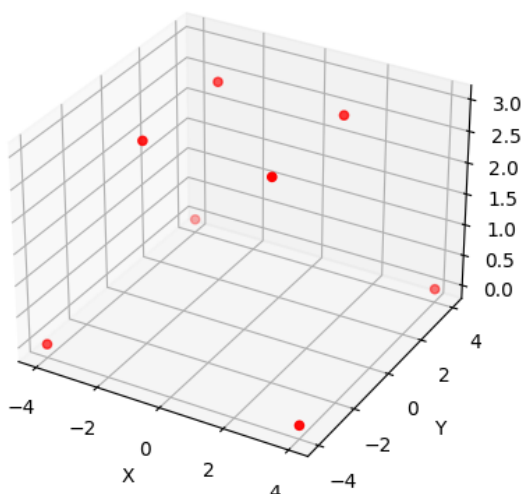


Рисунок 3 – Исоповерхность пирамиды

Рисунок 2 – Представление вершин пирамиды в пространстве

Хотя описаны только вершины усеченной пирамиды, уже можно посмотреть как будет выглядеть этот объект, если соединить их треугольниками с помощью изменения параметра функции – `plot_vertices(vertices = vertices, isosurf = True)`:

На рисунке 3 отчетливо видно, что точки соединены с помощью треугольников.

Кажется, грани уже есть, но пока существуют только вершины. Для того чтобы создать файл в формате STL, нам надо описать грани этой пирамиды, что можно сделать с помощью функции «`spatial.ConvexHull`» из библиотеки SciPy. Пример создания граней с помощью этой функции:

```
# Подключение библиотеки NumPy и scipy
from scipy import spatial
import numpy as np
from TaskPlot import plot_create_mesh
```

```

from stl import
# Координаты точек
vertices = np.array(
    [
        [-4, -4, 0],
        [+4, -4, 0],
        [-4, +4, 0],
        [+4, +4, 0],
        [-2, -2, 3],
        [+2, -2, 3],
        [-2, +2, 3],
        [+2, +2, 3],
    ]
)
# Формирования списка граней
hull = spatial.ConvexGull(vertices)
faces = hull.simplices

```

В результате массив `faces` содержит описание граней:

```

array([[3, 2, 0], [7, 3, 1],
       [3, 1, 0], [7, 5, 1],
       [5, 1, 0], [7, 3, 2],
       [5, 4, 0], [7, 6, 1],
       [6, 2, 0], [7, 5, 4],
       [6, 4, 0], [7, 6, 4]], dtype=int32)

```

Faces (грани) – список граней. Каждая треугольная грань описана тремя точками из набора точек `vertices`. Например, первая грань содержит точки 3, 2, 0.

Mesh (сетка) – набор вершин и граней, которые определяют форму многогранного объекта. Пример кода, который собирает эту сетку:

```

# Собираем сетку 3D модели
TruncPyram_mesh = mesh.Mesh(np.zeros(faces.shape[0], dtype=mesh.Mesh.dtype))
for i, f in enumerate(faces):
    for j in range(3):
        TruncPyram_mesh.vectors[i][j] = vertices[f[j],:]

plot_create_mesh(Truncpyram_mesh)

# Запись результата в файл с расширением .stl
TruncPyram_mesh.save('TruncPyramid.stl')

```

Представлением результата этого кода является такая же усеченная пирамида (рисунок 3), но уже с существующими гранями. Результатом исполнения предложенного кода и будет создание файла «TruncPyramid.stl» в котором хранится информация о этой пирамиде. Для просмотра файлов в формате STL существует довольно много программ. Одной из таких программ является «Blender».

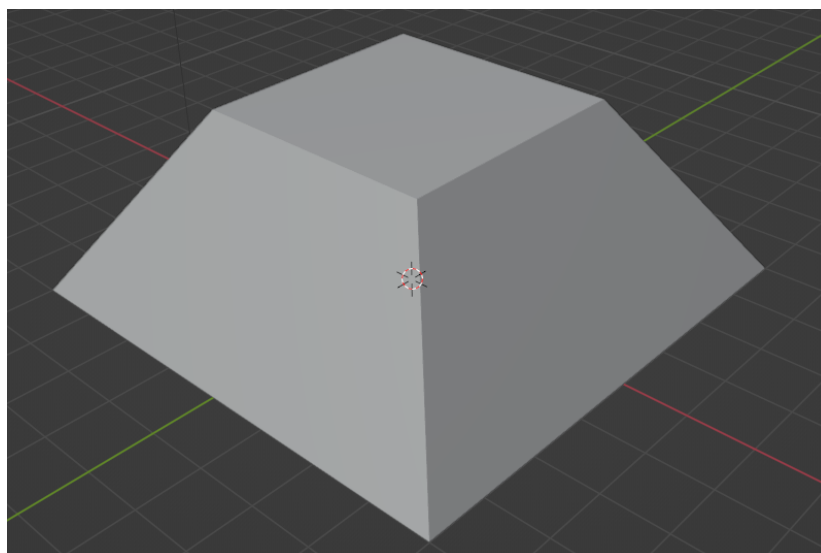


Рисунок 4 – Результат открытия файла «TruncPyramid.stl» с помощью программы Blender

**Заключение.** Python и библиотека NumPy предоставляют отличные возможности для создания 3D моделей и работы с ними. Библиотека NumPy позволяет удобно работать с данными в формате STL и взаимодействовать с 3D моделями в режиме реального времени.

Кроме того, в Python существуют и другие библиотеки для создания и обработки 3D моделей, такие как Blender Python API и Maya Python API, которые позволяют автоматизировать процесс создания моделей и взаимодействовать с ними с помощью языка программирования Python.

В целом, Python является мощным инструментом для 3D моделирования и визуализации, и его возможности в этой области постоянно расширяются. Он позволяет создавать сложные и интерактивные 3D приложения, что делает его полезным как для профессиональных 3D-художников и инженеров, так и для любителей.

### Список литературы

1. *OpenGL Programming Guide: The Official Guide to Learning OpenGL Ninth Edition* / Dave Sreiner, Graham Sellers, Jonh Kesenich, Bill Licea-Kane; Edited by PhD, Associate The Khronos OpenGL ARB Working Group. 2016 - 986 p. - ISBN 978-0134-495-51-4.
2. *Python Graphics: A Reference for Creating 2D and 3D Images First Edition* / B. J. Korities; Edited by PhD, Associate B.J. Korites. 2018. – 376 p. – ISBN 978-1484-233-77-1.
3. 3D моделирование в Python [Электронный ресурс] // telegraph.ph – Режим доступа: <https://telegra.ph/3D-modelirovanie-v-Python-08-15> - Дата доступа: 23.03.2023.
4. Полигональная сетка [Электронный ресурс] // ru.wikipedia.org – Режим доступа: [https://ru.wikipedia.org/wiki/Полигональная\\_сетка](https://ru.wikipedia.org/wiki/Полигональная_сетка) - Дата доступа: 23.03.2023.

UDC 004.94

## CREATING A 3D-MODEL BY MEANS PROGRAMMING LANGUAGE

Sazonov A. V.

*Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus*

*Omelkovich E. V. – master of pedagogical sciences, lecturer of the Department of ECG*

**Annotation.** The capabilities of the Python programming language for creating spatial 3D-models are analyzed and an image of a simple 3D-model is experimentally obtained using the NumPy library.

**Keywords:** 3D-modeling, 3D-model, Python programming language, NumPy library