



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 3
по курсу «Алгоритмы компьютерной графики»

Студент группы ИУ9-41Б Утебаева М. Б.

Преподаватель Цалкович П. А.

Москва 2024

1 Задача

1. Определить антипризму в качестве модели объекта сцены.
2. Реализовать изменение ориентации с помощью модельно-видовых преобразований (без gluLookAt). Управление производится интерактивно с помощью клавиатуры и/или мыши.
3. Предусмотреть возможность переключения между каркасным и твердотельным отображением модели (glFrontFace/ glPolygonMode).

2 Основная теория

Геометрические преобразования представляются матрицами, с помощью них можно осуществлять перенос, масштабирование, сдвиг и поворот объекта.

Есть три типа матриц преобразования, над которыми можно выполнять операции: 1) видовая матрица – определяет преобразования объекта в мировых координатах, такие как параллельный перенос, изменение масштаба и поворот; 2) матрица проекций – определяет, как будут проецироваться трехмерные объекты на плоскость экрана (в оконные координаты); 3) матрица текстуры определяет наложение текстуры на объект.

Антипризма – полуправильный многогранник, у которого две параллельные грани (основания) – равные между собой правильные n -угольники, а остальные $2n$ граней (боковые грани) – правильные треугольники.

```
glMatrixMode(GL_PROJECTION) #включение режима работы с матрицей проекций
glMatrixMode(GL_MODELVIEW) #включение режима работы с модельно-видовой матрицей
glLoadIdentity() #замена текущей матрицы на единичную
glEnable(GL_DEPTH_TEST) #отсечение заслонённых фигур
glDepthFunc(GL_LESS) #обработка глубины
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL) #перевод объекта в твёрдотельный
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE) #перевод объекта в каркасный
glBegin(GL_QUADS) #задание прямоугольников из 4-х последовательных вершин
glBegin(GL_TRIANGLES) #задание треугольников из 3-х последовательных вершин
glMultMatrixf(A) #умножение на матрицу A
```

3 Практическая реализация

```
import glfw
from OpenGL.GL import *
from math import cos, sin, sqrt, asin, pi

alpha = 0.0
beta = 0.0
fill = False

def prizma():
    glBegin(GL_POLYGON)
    #верхняя
    glColor3f(0, 0, 139/255)
    glVertex3f(1/2, 1/2, 8**(1/4)/4) #A
    glVertex3f(-1/2, 1/2, 8**(1/4)/4) #B
    glVertex3f(-1/2, -1/2, 8**(1/4)/4) #D
    glVertex3f(1/2, -1/2, 8**(1/4)/4) #C
```

```

glEnd()
glBegin(GL_TRIANGLES)
#1
glColor3f(139/255, 0, 0)
glVertex3f(1/2, 1/2, 8**(1/4)/4) #A
glVertex3f(0, sqrt(2)/2, -8**(1/4)/4) #E
glVertex3f(-1/2, 1/2, 8**(1/4)/4) #B
#2
glColor3f(255/255, 69/255, 0)
glVertex3f(0, sqrt(2)/2, -8**(1/4)/4) #E
glVertex3f(-1/2, 1/2, 8**(1/4)/4) #B
glVertex3f(-sqrt(2)/2, 0, -8**(1/4)/4) #H
#3
glColor3f(1.0, 1.0, 0)
glVertex3f(-1/2, 1/2, 8**(1/4)/4) #B
glVertex3f(-sqrt(2)/2, 0, -8**(1/4)/4) #H
glVertex3f(-1/2, -1/2, 8**(1/4)/4) #D
#4
glColor3f(138/255, 43/255, 226/255)
glVertex3f(-sqrt(2)/2, 0, -8**(1/4)/4) #H
glVertex3f(-1/2, -1/2, 8**(1/4)/4) #D
glVertex3f(0, -sqrt(2)/2, -8**(1/4)/4) #F
#5
glColor3f(255/255, 0, 255/255)
glVertex3f(-1/2, -1/2, 8**(1/4)/4) #D
glVertex3f(0, -sqrt(2)/2, -8**(1/4)/4) #F
glVertex3f(1/2, -1/2, 8**(1/4)/4) #C
#6
glColor3f(0, 255/255, 0)
glVertex3f(0, -sqrt(2)/2, -8**(1/4)/4) #F
glVertex3f(1/2, -1/2, 8**(1/4)/4) #C
glVertex3f(sqrt(2)/2, 0, -8**(1/4)/4) #G
#7
glColor3f(0, 100/255, 0)
glVertex3f(1/2, -1/2, 8**(1/4)/4) #C
glVertex3f(sqrt(2)/2, 0, -8**(1/4)/4) #G
glVertex3f(1/2, 1/2, 8**(1/4)/4) #A
#8
glColor3f(0, 255/255, 255/255)
glVertex3f(sqrt(2)/2, 0, -8**(1/4)/4) #G
glVertex3f(1/2, 1/2, 8**(1/4)/4) #A
glVertex3f(0, sqrt(2)/2, -8**(1/4)/4) #E
glEnd()
#нужня
glBegin(GL_POLYGON)
glColor3f(0, 0, 255/255)
glVertex3f(sqrt(2)/2, 0, -8**(1/4)/4) #G

```

```

glVertex3f(0, sqrt(2)/2, -8**(1/4)/4) #E
glVertex3f(-sqrt(2)/2, 0, -8**(1/4)/4) #H
glVertex3f(0, -sqrt(2)/2, -8**(1/4)/4) #F
glEnd()

def display(window):
    glEnable(GL_DEPTH_TEST) #отсечение заслонённых фигур
    glDepthFunc(GL_LESS) #обработка глубины(?)
    glClear(GL_COLOR_BUFFER_BIT)
    glClear(GL_DEPTH_BUFFER_BIT)
    glMatrixMode(GL_PROJECTION) #задаём видимое пространство
    glMatrixMode(GL_MODELVIEW) #матрица представления модели
    glLoadIdentity() #единичная матрица
    Z = [cos(beta), 0, -sin(beta), 0,
         0, 1, 0, 0,
         sin(beta), 0, cos(beta), 0,
         0, 0, 0, 1] #вращение вокруг OZ
    Y = [1, 0, 0, 0,
         0, cos(alpha), -sin(alpha), 0,
         0, sin(alpha), cos(alpha), 0,
         0, 0, 0, 1] #вращение вокруг OY
    glMultMatrixf(Z)
    glMultMatrixf(Y)
    prizma()
    glfw.swap_buffers(window)
    glfw.poll_events()

def key_callback(window, key, scancode, action, mods):
    global alpha
    global beta
    if action == glfw.PRESS or action == glfw.REPEAT:
        if key == glfw.KEY_RIGHT:
            beta += 0.1
        elif key == glfw.KEY_LEFT:
            beta -= 0.1
        elif key == glfw.KEY_UP:
            alpha += 0.1
        elif key == glfw.KEY_DOWN:
            alpha -= 0.1
        elif key == glfw.KEY_F:
            global fill
            if fill :
                glPolygonMode(GL_FRONT_AND_BACK, GL_FILL) #твёрдотельное
            else:
                glPolygonMode(GL_FRONT_AND_BACK, GL_LINE) #каркасное
            fill = not fill

```

```
def main():
    if not glfw.init():
        return
    window = glfw.create_window(640, 640, "laba3", None, None)
    if not window:
        glfw.terminate()
        return
    glfw.make_context_current(window)
    glfw.set_key_callback(window, key_callback)
    while not glfw.window_should_close(window):
        display(window)
    glfw.destroy_window(window)
    glfw.terminate()

main()
```

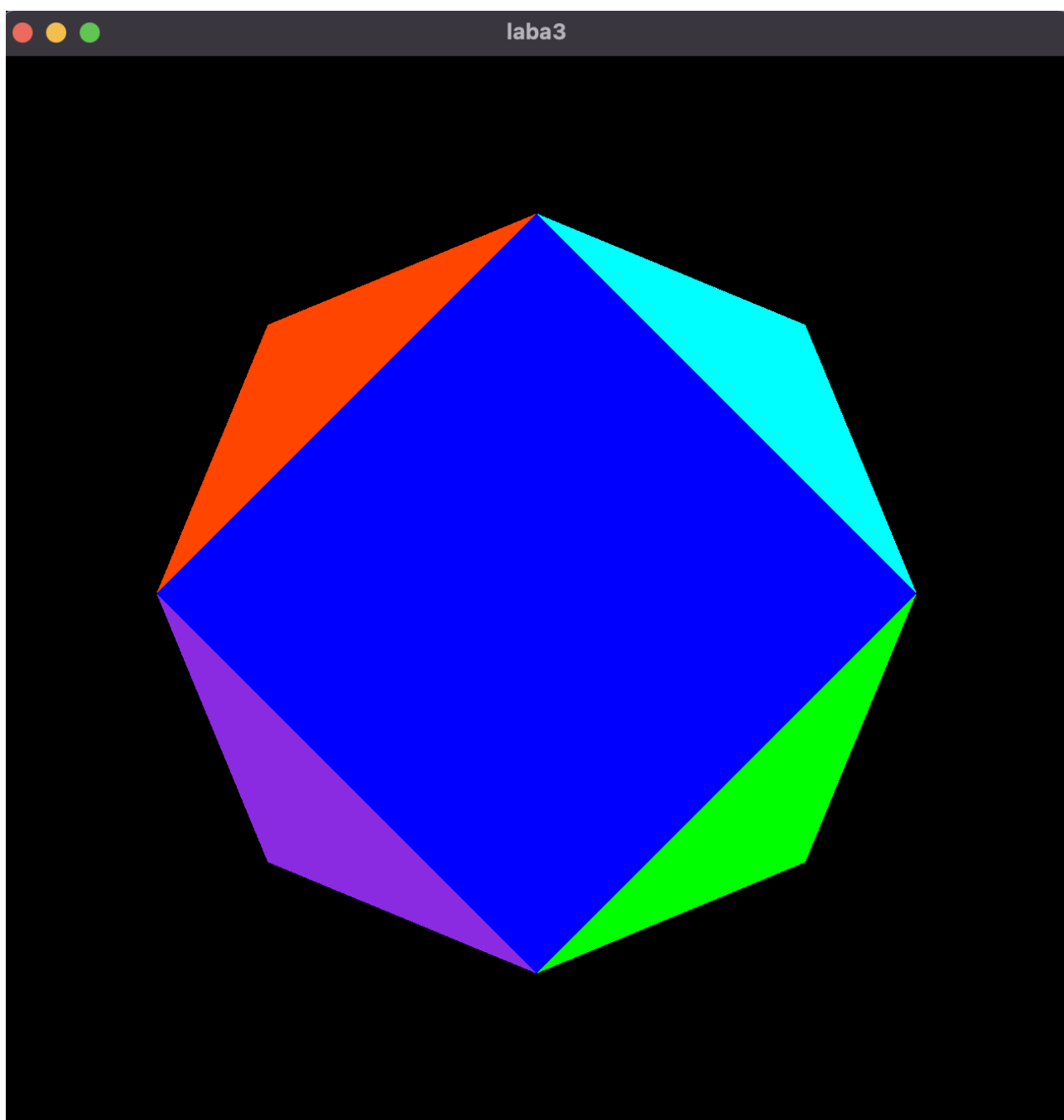


Рис. 1 — Начальное состояние

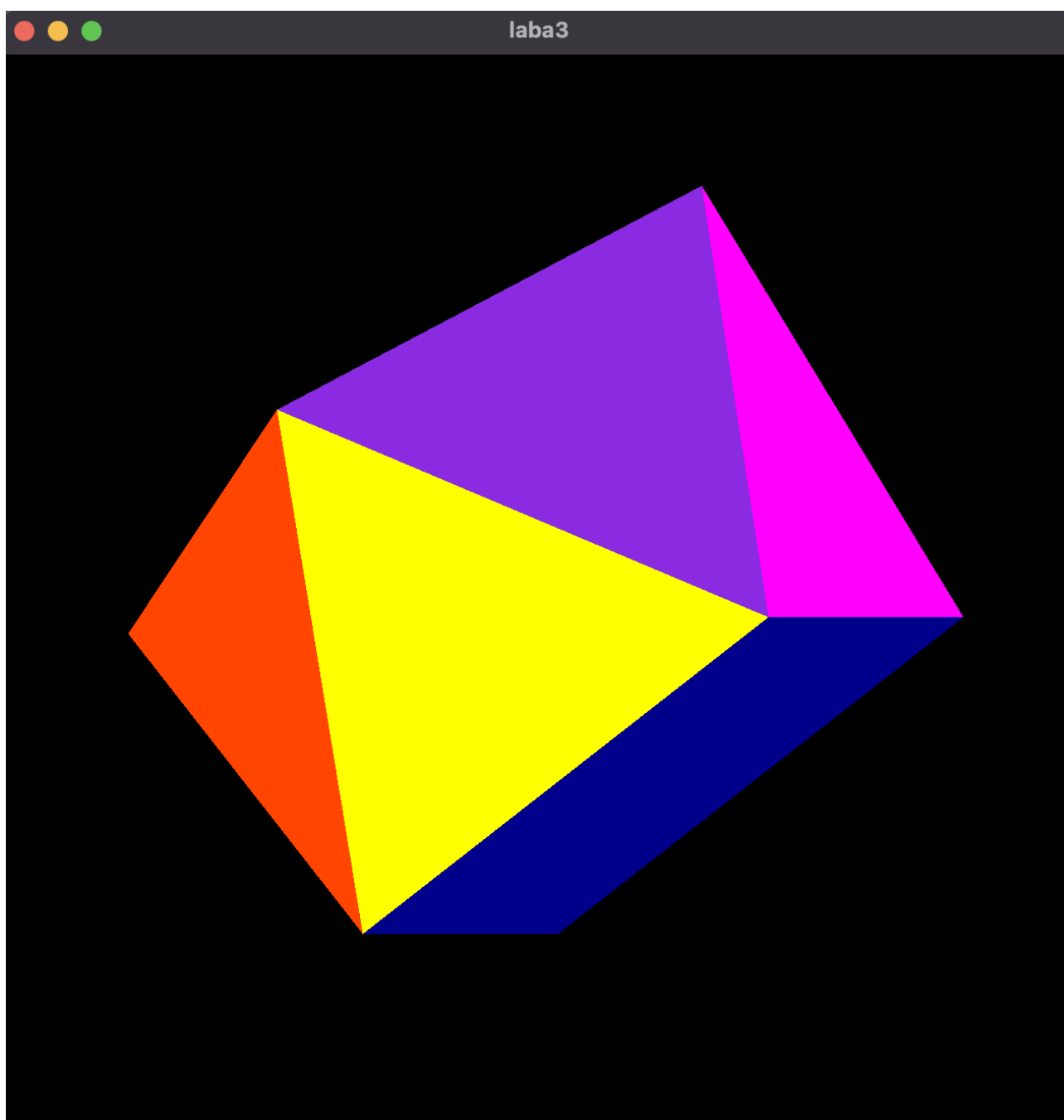


Рис. 2 — Антипризма после вращения

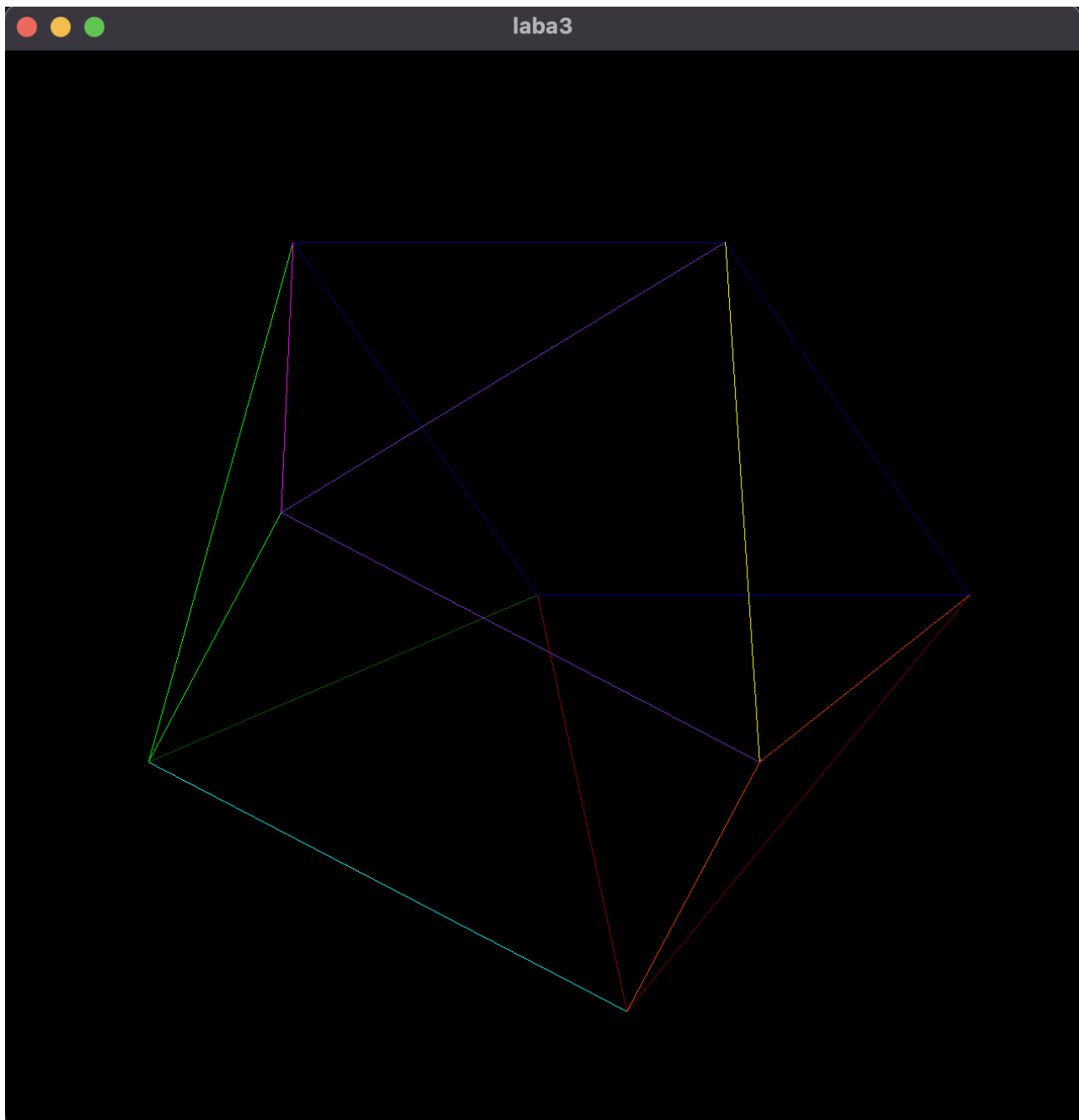


Рис. 3 — Каркасное отображение антипризмы после поворотов

4 Заключение

Я узнала, что такое и из чего состоят модельно-видовые преобразования: из композиции переносов, поворотов, масштабирований. Познакомилась с функциями операций над матрицами преобразований и видами матриц преобразований. При помощи задания специальных матриц, я реализовала изменение ориентации объекта с помощью модельно-видовых преобразований с возможностью управления при помощи клавиш и переключение между каркасным и твёрдотельным отображением модели. Также я познакомилась с новой фигурой – антипризмой, которая представляет из себя полуправильный многогранник, у которого две параллельные грани равные между собой правильные n -угольники, а остальные $2n$ граней – правильные треугольники.