



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 № 1
по курсу «Алгоритмы компьютерной графики»

Студент группы ИУ9-41Б Утебаева М. Б.

Преподаватель Цалкович П. А.

Москва 2024

1 Задача

Реализовать любой графический примитив, добавить любое геометрическое преобразование (сдвиг, поворот и т.д.), добавить обработку события (нажатия на кнопку и т.д.)

2 Основная теория

```
import glfw
from OpenGL.GL import * #импорт библиотеки
glClear(GL_COLOR_BUFFER_BIT) #очистка буфера цвета
glPushMatrix() #проталкивает текущую матрицу в стек матриц
glBegin(GL_POLYGON) #задает многоугольник с количеством углов равным количеству заданных вершин
glColor3f(...) #устанавливаем цвет
glVertex2f(..., ...) #задаём позицию
window = glfw.create_window(640, 640, "laba1", None, None) #задаём окно
glfw.set_key_callback(window, key_callback) #обработчик нажатия клавиш
glfw.destroy_window(window) #разрушение окна
```

3 Практическая реализация

```
import glfw
from OpenGL.GL import *

delta = 0.05
angle = 0.0
posx = -0.2
posy = -0.2
koef = 1.5

def display(window):
    global angle
    glClear(GL_COLOR_BUFFER_BIT)
    glLoadIdentity()
    glClearColor(0.0, 0.0, 0.0, 1.0)
    glPushMatrix()
    glRotatef(angle, 0.45, 1, 0.90)
    glBegin(GL_POLYGON)

    glColor3f(1.0, 0.0, 0.0)
    glVertex2f(posx + 0*koef, posy + 0*koef)

    glColor3f(0.0, 1.0, 0.0)
    glVertex2f(posx + 0.2*koef, posy + 0.2*koef)

    glColor3f(0.0, 0.0, 0.1)
    glVertex2f(posx + 0.4*koef, posy + 0.2*koef)
```

```

glColor3f(1.0, 0.0, 0.0)
glVertex2f(posx + 0.6*koef, posy + 0*koef)

glColor3f(0.0, 1.0, 0.0)
glVertex2f(posx + 0.4*koef, posy - 0.2*koef)

glColor3f(0.0, 0.0, 1.0)
glVertex2f(posx + 0.2*koef, posy - 0.2*koef)

glColor3f(0.5, 0.5, 0.5)
glVertex2f(posx + 0*koef, posy + 0*koef)

glEnd()
glPopMatrix()
angle += delta
glfw.swap_buffers(window)
glfw.poll_events()

def key_callback(window, key, scancode, action, mods):
    global delta
    global angle
    if action == glfw.PRESS:
        if key == glfw.KEY_RIGHT:
            delta = -0.5
        if key == glfw.KEY_LEFT:
            delta = 0.5
        if key == glfw.KEY_ENTER:
            delta = 0

def main():
    if not glfw.init():
        return
    window = glfw.create_window(640, 640, "lab1", 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()

```

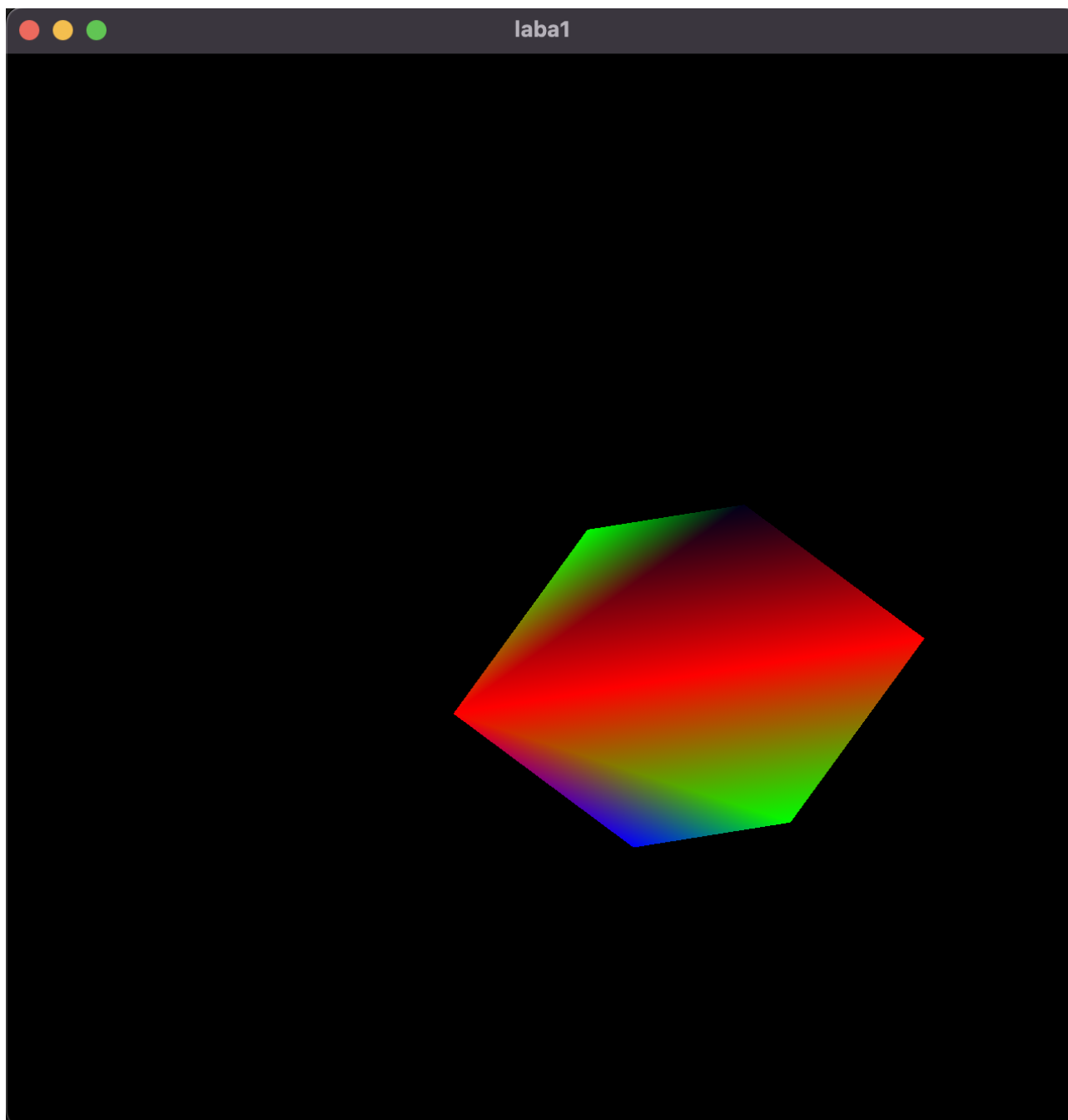


Рис. 1 — Результат

4 Заключение

Я познакомилась с библиотекой OpenGL и её основными методами, реализовала и "раскрасила" свой первый графический примитив – шестиугольник, задав его поточечно. Также освоила добавление такого геометрического преобразования как вращение, и метод обработки событий, а именно – обработку нажатия на клавиши (Left, Right, Enter).