**МИНОБРНАУКИ РОССИИ**

**федеральное государс твенное бюджетное образовательное учреждение высшего образования**

**«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова» (БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова)**

|  |  |  |
| --- | --- | --- |
| Кафедра | O7 | Информационные системы и программная инженерия |
|  | шифр | наименование кафедры, по которой выполняется работа |
| Дисциплина | **Компьютерная графика и геометрия** | |
|  |  | наименование дисциплины |

|  |
| --- |
| УЧЕБНО-ПРАКТИЧЕСКАЯ РАБОТА |
| №4 |
| ТРЕХМЕРНЫЕ ПОСТРОЕНИЯ. БУФЕР ГЛУБИНЫ. |
| ВИДОВЫЕ ПАРАМЕТРЫ. ПАРАЛЛЕЛЬНАЯ И |
|  |

|  |  |  |
| --- | --- | --- |
| **ОБУЧАЮЩИЙСЯ** | | |
|  | группы | О727Б |
|  | Смагин Т.В. | |
| подпись | фамилия и инициалы | |
|  | 18.10.2024 |  |
|  | дата сдачи |  |
| **ПРОВЕРИЛ** | | |
| кандидат п.н, доцент, доцент | | |
| ученая степень, ученое звание, должность | | |
|  | Снижко Е.А. | |
| подпись | фамилия и инициалы | |
| Оценка / балльная оценка | | |
|  | 18.10.2024 | |
|  | дата проверки |  |

|  |  |  |
| --- | --- | --- |
| Санкт-Петербург | | |
| 20 | 24 | г. |

СОДЕРЖАНИЕ

1. [**Постановка задания** 3](#_bookmark0)
2. [**Реализация** 5](#_bookmark1)
3. **Постановка задания**

Требуется выполнить следующие задания:

Создать массив вершин, содержащий координаты вершин правильного n-угольника (для формирования массива требуется вычислять координаты вершин с помощью цикла). Используя массив индексов нарисовать:

а) все возможные непересекающиеся треугольники;

б) все возможные пересекающиеся линии;

в) контур n-угольника;

г) линию, соединяющую все чётные по индексу точки.

Все объекты должны быть изображены каркасно, каждый своим цветом.

Нарисовать треугольник, прямоугольник и линию.

Получить новое изображение, на котором:

− треугольник следует смасштабировать с коэффициентами (kx, ky) и перенести на вектор p;

− линию повернуть на угол α относительно начала координат

− прямоугольник повернуть относительно точки с координатами (x, y) на угол β.

На изображении должны быть отображены как старый, так и новый вариант каждой фигуры, либо должна быть предусмотрена возможность быстрого переключения между ними. Следует подобрать такие размеры фигур, чтобы все они поместились на одном изображении.

Используя примитив для рисования линий и операции геометрических преобразований, изобразить фигуру, показанную в вариативной части задания.

# Реализация

# Задание 1

Задание 1. Построить изображения трёх плоских фигур в пространстве (в соответствии с вариантом). Оценить полученный результат: правильно ли изображены фигуры, если нет – объяснить почему.

def draw\_shapes():  
 glDisable(GL\_DEPTH\_TEST)  
 glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT)  
  
 glBegin(GL\_TRIANGLES)  
 glColor3f(1, 0, 0)  
 for vertex in triangle1:  
 glVertex3fv(vertex)  
 glEnd()  
  
 glBegin(GL\_QUADS)  
 glColor3f(0, 1, 0)  
 for vertex in quad:  
 glVertex3fv(vertex)  
 glEnd()  
  
 glBegin(GL\_TRIANGLES)  
 glColor3f(0, 0, 1)  
 for vertex in triangle2:  
 glVertex3fv(vertex)  
 glEnd()

На рисунке 2.1 приведено выполнение задания 1а.

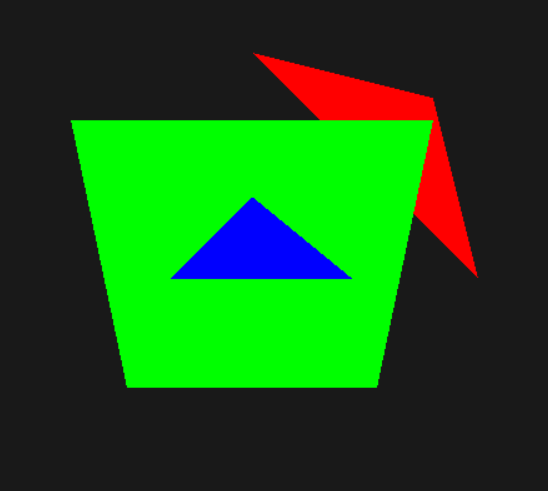


Рисунок 2.1 – Задание 1

# Задание 2

Установить режим проверки буфера глубины в программе к заданию 1. Сравнить полученный результат с предыдущим

def draw\_shapes2():  
 glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT)  
 glEnable(GL\_DEPTH\_TEST)   
  
 glBegin(GL\_TRIANGLES)  
 glColor3f(1, 0, 0)  
 for vertex in triangle1:  
 glVertex3fv(vertex)  
 glEnd()  
  
 glBegin(GL\_QUADS)  
 glColor3f(0, 1, 0)  
 for vertex in quad:  
 glVertex3fv(vertex)  
 glEnd()  
  
 glBegin(GL\_TRIANGLES)  
 glColor3f(0, 0, 1)  
 for vertex in triangle2:  
 glVertex3fv(vertex)  
 glEnd()

На рисунке 2.2 приведено выполнение задания 2.

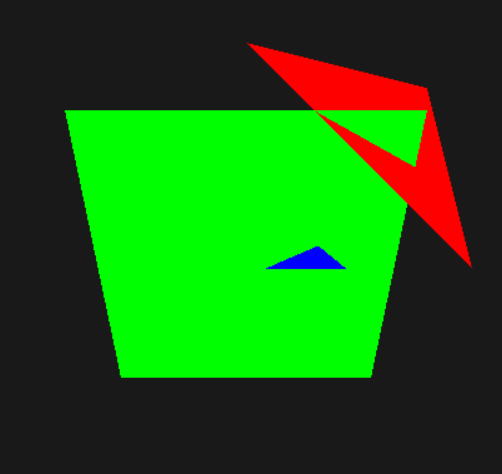


Рисунок 2.2 – Задание 2

# Задание 3

Установить видовые параметры в соответствии с вариантом и параллельную проекцию; построить изображение куба с координатами 56 вершин, равными по модулю 1 (куб изображается с помощью 12 треугольников). Оценить результат.

def draw\_tr\_cube():  
 glBegin(GL\_TRIANGLES)  
 glVertex3fv(vertices[0])  
 glVertex3fv(vertices[1])  
 glVertex3fv(vertices[2])  
  
 glVertex3fv(vertices[0])  
 glVertex3fv(vertices[2])  
 glVertex3fv(vertices[3])  
  
 glVertex3fv(vertices[4])  
 glVertex3fv(vertices[5])  
 glVertex3fv(vertices[6])  
  
 glVertex3fv(vertices[4])  
 glVertex3fv(vertices[6])  
 glVertex3fv(vertices[7])  
  
 glVertex3fv(vertices[3])  
 glVertex3fv(vertices[7])  
 glVertex3fv(vertices[0])  
  
 glVertex3fv(vertices[0])  
 glVertex3fv(vertices[7])  
 glVertex3fv(vertices[4])  
  
 glVertex3fv(vertices[1])  
 glVertex3fv(vertices[5])  
 glVertex3fv(vertices[6])  
  
 glVertex3fv(vertices[1])  
 glVertex3fv(vertices[6])  
 glVertex3fv(vertices[2])  
  
 glVertex3fv(vertices[3])  
 glVertex3fv(vertices[7])  
 glVertex3fv(vertices[6])  
  
 glVertex3fv(vertices[3])  
 glVertex3fv(vertices[6])  
 glVertex3fv(vertices[2])  
  
 glVertex3fv(vertices[0])  
 glVertex3fv(vertices[1])  
 glVertex3fv(vertices[5])  
  
 glVertex3fv(vertices[0])  
 glVertex3fv(vertices[5])  
 glVertex3fv(vertices[4])  
  
 glEnd()

На рисунке 2.3 приведено выполнение задания 3.

Изображение выглядит как снимок экрана, Цвет электрик, Красочность, синий

Автоматически созданное описание

Рисунок 2.3 – Задание 3

# Задание 4

Установить перспективную проекцию и построить каркасное изображение куба

def draw\_wireframe\_cube():  
 glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT)  
 glColor3f(1, 1, 1)  
 glBegin(GL\_LINES)  
  
 for edge in cube\_edges:  
 for vertex in edge:  
 glVertex3fv(cube\_vertices[vertex])  
  
 glEnd()

На рисунке 2.4 приведено выполнение задания 4.

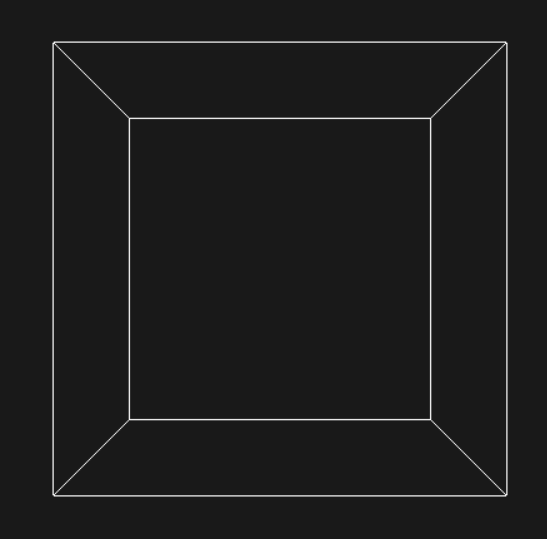


Рисунок 2.4 – Задание 4

# Задание 5

Задание 2. Повернуть оси координат в соответствии с вариантом и постройте изображение куба. Оценить результат. Почему части куба отсечены? Изменить видовые параметры таким образом, чтобы куб изображался без отсечений. Изменить код программы таким образом, чтобы каждая грань куба изображалась своим цветом.

def draw\_colored\_cube():  
 glClearColor(0.1, 0.1, 0.1, 1)  
 glMatrixMode(GL\_PROJECTION)  
 glLoadIdentity()  
 glMatrixMode(GL\_MODELVIEW)  
 glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT)  
  
 glBegin(GL\_QUADS)  
 for i, face in enumerate(cube\_faces):  
 glColor3fv(face\_colors[i])   
 for vertex in face:  
 glVertex3fv(cube\_vertices[vertex])  
 glEnd()

На рисунке 2.5 приведено выполнение задания 2.

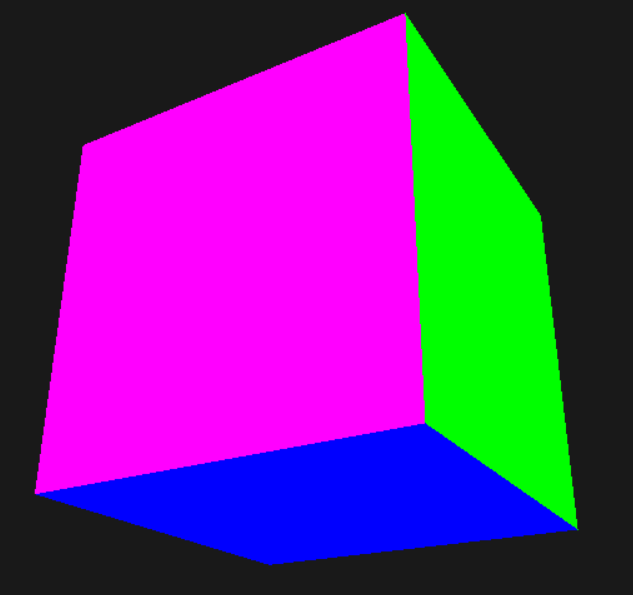


Рисунок 2.5 – Задание 5

# Задание 3

Задание 3. Используя примитив для рисования линий и операции геометрических преобразований, изобразить фигуру, показанную в вариативной части задания.

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

def draw\_line(x1, y1, x2, y2):  
 glBegin(GL\_LINES)  
 glVertex2f(x1, y1)  
 glVertex2f(x2, y2)  
 glEnd()  
  
  
def draw\_right\_side():  
 glColor3f(1, 1, 1)   
  
 draw\_line(0, 0, 1, 0.5)  
 draw\_line(1, 0.5, 1, 0)  
 draw\_line(1, 0, 0, 0)  
  
 draw\_line(0, 0, -0.7, -0.7)  
 draw\_line(-0.7, -0.7, -0.4, -1)  
 draw\_line(-0.4, -1, 0,0)  
  
  
def draw\_shape():  
 draw\_right\_side()  
 glPushMatrix()  
 glScalef(-1, 1, 1)  
 draw\_right\_side()  
 glPopMatrix()

На рисунке 2.9 приведено выполнение задания 3.

Изображение выглядит как линия, дизайн

Автоматически созданное описание

Рисунок 2.9 – Задание 3