**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 |
| №3 |
| ИСПОЛЬЗОВАНИЕ МАССИВОВ ИНДЕКСОВ. |
| ПРЕОБРАЗОВАНИЯ КООРДИНАТ ВАРИАНТ №9 |
|  |

|  |  |  |
| --- | --- | --- |
| **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 1а

Задание 1. 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;

def draw\_triangles(vertices):  
 glBegin(GL\_TRIANGLES)  
 n = len(vertices)  
 for i in range(n):  
 glColor3f(\*colors[i])  
 glVertex2f(vertices[i][0], vertices[i][1])  
 glVertex2f(vertices[(i + 1) % n][0], vertices[(i + 1) % n][1])  
 glVertex2f(0, 0)  
 glEnd()

На рисунке 2.1 приведено выполнение задания 1а.

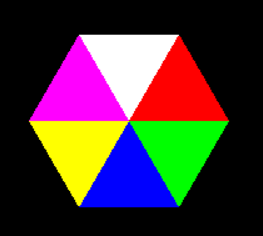


Рисунок 2.1 – Задание 1а

# Задание 1б

б) все возможные пересекающиеся линии;

def draw\_intersecting\_lines(vertices):  
 glBegin(GL\_LINES)  
 n = len(vertices)  
 for i in range(n):  
 for j in range(i + 1, n):  
 glVertex2f(vertices[i][0], vertices[i][1])  
 glVertex2f(vertices[j][0], vertices[j][1])  
 glEnd()

На рисунке 2.2 приведено выполнение задания 1б.

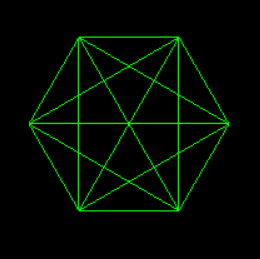


Рисунок 2.2 – Задание 1б

# Задание 1в

в) контур n-угольника;

def draw\_polygon(vertices):  
 glBegin(GL\_LINE\_LOOP)  
 for v in vertices:  
 glVertex2f(v[0], v[1])  
 glEnd()

На рисунке 2.3 приведено выполнение задания 1в.

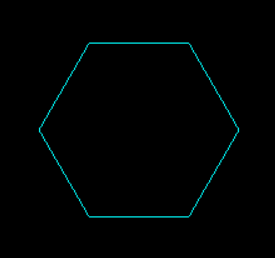


Рисунок 2.3 – Задание 1в

# Задание 1г

г) линию, соединяющую все чётные по индексу точки.

def draw\_even\_lines(vertices):  
 glBegin(GL\_LINES)  
 for i in range(0, len(vertices), 2):  
 glVertex2f(vertices[i][0], vertices[i][1])  
 glVertex2f(vertices[(i + 2) % len(vertices)][0], vertices[(i + 2) % len(vertices)][1])  
 glEnd()

На рисунке 2.4 приведено выполнение задания 4. Изображение выглядит как линия, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 2.4 – Задание 1г

# Задание 2

Задание 2. Нарисовать треугольник, прямоугольник и линию. Получить новое изображение, на котором:

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

def draw\_task\_2():  
 # Оригинальный треугольник  
 glColor3f(1, 0, 0) # Красный  
 glBegin(GL\_TRIANGLES)  
 glVertex2f(-0.5, -0.5)  
 glVertex2f(0.5, -0.5)  
 glVertex2f(0, 0.5)  
 glEnd()  
  
 # Масштабированный и сдвинутый треугольник  
 glColor3f(0, 1, 0) # Зеленый  
 glPushMatrix()  
 glTranslatef(p[0], p[1], 0)  
 glScalef(kx, ky, 1)  
 glBegin(GL\_TRIANGLES)  
 glVertex2f(-0.5, -0.5)  
 glVertex2f(0.5, -0.5)  
 glVertex2f(0, 0.5)  
 glEnd()  
 glPopMatrix()  
  
 # Оригинальный прямоугольник  
 glColor3f(0, 0, 1) # Синий  
 glBegin(GL\_QUADS)  
 glVertex2f(-0.5, -0.3)  
 glVertex2f(0.5, -0.3)  
 glVertex2f(0.5, 0.3)  
 glVertex2f(-0.5, 0.3)  
 glEnd()  
  
   
 glPushMatrix()  
 glTranslatef(x, y, 0)  
 glRotatef(beta, 0, 0, 1)  
 glTranslatef(-x, -y, 0)  
 glBegin(GL\_QUADS)  
 glVertex2f(-0.5, -0.3)  
 glVertex2f(0.5, -0.3)  
 glVertex2f(0.5, 0.3)  
 glVertex2f(-0.5, 0.3)  
 glEnd()  
 glPopMatrix()  
  
 # Оригинальная линия  
 glColor3f(1, 1, 0) # Желтая  
 glBegin(GL\_LINES)  
 glVertex2f(-1, -1)  
 glVertex2f(1, 1)  
 glEnd()  
  
 # Повернутая линия  
 glPushMatrix()  
 glRotatef(alpha, 0, 0, 1)  
 glBegin(GL\_LINES)  
 glVertex2f(-1, -1)  
 glVertex2f(1, 1)  
 glEnd()  
 glPopMatrix()

На рисунке 2.5 приведено выполнение задания 2.

Изображение выглядит как снимок экрана, Красочность, Графика, дизайн

Автоматически созданное описание

Рисунок 2.5 – Задание 2 до пребразований

# 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