**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

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

**(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Кафедра |  | О7 |  | Информационные системы и программная инженерия |
|  |  | шифр |  | наименование кафедры, по которой выполняется работа |
| Дисциплина |  | **Компьютерная геометрия и графика** | | |
|  |  | наименование дисциплины | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УЧЕБНО-ПРАКТИЧЕСКАЯ РАБОТА | | | | |
|  | 2 |  |
|  | номер задания (при наличии) |  |
| Примитивы OpenGL | | | |
| Основные приёмы построения двумерных объектов  Вариант 13 | | | |

при наличии указать тему учебно-практической работы и (или) номер варианта

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ОБУЧАЮЩИЙСЯ** | | | | | | |
| группы | | | |  | | О727Б |
|  |  | Смагин Т.В. | | | | |
| подпись |  | фамилия и инициалы | | | | |
|  | | | | |
| дата сдачи | | | | |
| **ПРОВЕРИЛ** | | | | | | |
|  |  | Снижко Е.А. | | | | |
| подпись |  | фамилия и инициалы | | | | |
| Оценка / балльная оценка | | |  | | | |
|  | | | | |
| дата проверки | | | | |

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

**СОДЕРЖАНИЕ**

1 Постановка задачи 3

2 Выполнение работы 4

2.1 Задание 1 4

2.2 Задание 2 4

2.3 Задание 3 5

2.4 Задание 4 6

2.5 Задание 5 6

2.6 Задание 6 9

2.7 Задание 7 10

2.8 Задание 8 10

**1 Постановка задачи**

Требуется выполнить следующие задания:

– Построить точки, расположенные в вершинах правильного n-угольника (координаты вершин рекомендуется высчитывать в цикле). Установить режим сглаживания для точек. Экспериментально определить максимальный размер точки, при котором возможно сглаживание;

– Используя примитив для вывода линий, нарисовать правильный n-угольник. Изменить толщину линий;

– Используя примитив для вывода ломаной линии, нарисовать первую фигуру, указанную в варианте;

– Используя примитив для вывода замкнутой ломаной, нарисовать вторую фигуру, указанную в варианте;

– Построить вторую фигуру, указанную в варианте, разбив её на треугольники (каждый треугольник окрашен случайным цветом). Выполните три варианта построений с использованием примитивов:

а) треугольник;

б) лента треугольников;

в) веер треугольников.

Чем отличаются результаты при изменении способа тонирования?

– Используя примитив для вывода веера треугольников, построить правильный n-угольник;

– Построить третью фигуру, указанную в варианте, представив её в виде совокупности отдельных треугольников, назначив каждому треугольнику свой цвет. Посмотреть результат работы программы для различных способов тонирования;

– Изменить программу предыдущей задачи таким образом, чтобы:

а) лицевые грани изображались только вершинами;

б) лицевые грани изображались закрашенными, а обратные – линиями;

в) лицевые и обратные грани изображались линиями.

**2 Выполнение работы**

**2.1 Задание 1**

def draw\_points(vertices, point\_size):  
 glPointSize(point\_size)  
 glBegin(GL\_POINTS)  
 glColor3f(1.0, 1.0, 1.0)  
 for vertex in vertices:  
 glVertex2f(vertex[0], vertex[1])  
 glEnd()

На рисунке 2.1 приведено выполнение задания 1.

Изображение выглядит как темнота, Астрономический объект, черный, пространство

Автоматически созданное описание

Рисунок 2.1 – Задание 1

**2.2 Задание 2**

def draw\_polygon(vertices):  
 glBegin(GL\_LINE\_LOOP)   
 glColor3f(1.0, 1.0, 1.0)   
 for vertex in vertices:  
 glVertex2f(vertex[0], vertex[1])  
 glEnd()

На рисунке 2.2 приведено выполнение задания 2.

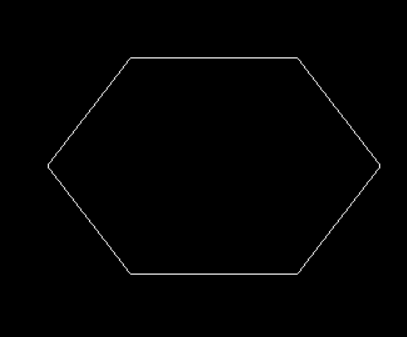


Рисунок 2.2 – Задание 2

**2.3 Задание 3**

def draw\_task3():  
 glBegin(GL\_LINE\_STRIP)   
 glVertex2f(-0.7, 1)  
 glVertex2f(-0.7, -1)  
 glVertex2f(-0.2, -1)  
 glVertex2f(-0.5, -0.6)  
 glVertex2f(0.1, 0.6)  
 glVertex2f(0.6, -0.2)  
 glVertex2f(0.1, -1)  
 glVertex2f(1, -1)  
 glEnd()

На рисунке 2.3 приведено выполнение задания 3.

Изображение выглядит как снимок экрана, черный, черно-белый, линия

Автоматически созданное описание

Рисунок 2.3 – Задание 3

**2.4 Задание 4**

def draw\_task4():  
 glBegin(GL\_LINE\_LOOP)   
 glVertex2f(0, 0)   
 glVertex2f(1, 0)   
 glVertex2f(1, 0.7)   
 glVertex2f(0.3, 0.3)   
 glVertex2f(-0.4, 0.7)  
 glVertex2f(-1, 0.2)  
 glVertex2f(-0.7, -0.7)  
 glVertex2f(0.2, -1)  
 glEnd()

На рисунке 2.4 приведено выполнение задания 4.

Изображение выглядит как диаграмма, дизайн

Автоматически созданное описание

Рисунок 2.4 – Задание 4

**2.5 Задание 5**

def draw\_triangle\_simple():  
 vertices = [  
 (1.0, 0.0), (1.0, 0.7), (0.0, 0.0),  
 (1.0, 0.7), (0.0, 0.0), (0.3, 0.3),  
 (0.0, 0.0), (0.3, 0.3), (-0.4, 0.7),  
 (-0.4, 0.7), (0.0, 0.0), (-1.0, 0.2),  
 (0.0, 0.0), (-1.0, 0.2), (-0.7, -0.7),  
 (-0.7, -0.7), (0.0, 0.0), (0.2, -1.0)  
 ]  
 glBegin(GL\_TRIANGLES)  
 for i in range(len(vertices)):  
 glColor3f(\*colors[i])  
 glVertex2f(\*vertices[i])  
  
 glEnd()

На рисунке 2.5 приведено выполнение задания 5-1.

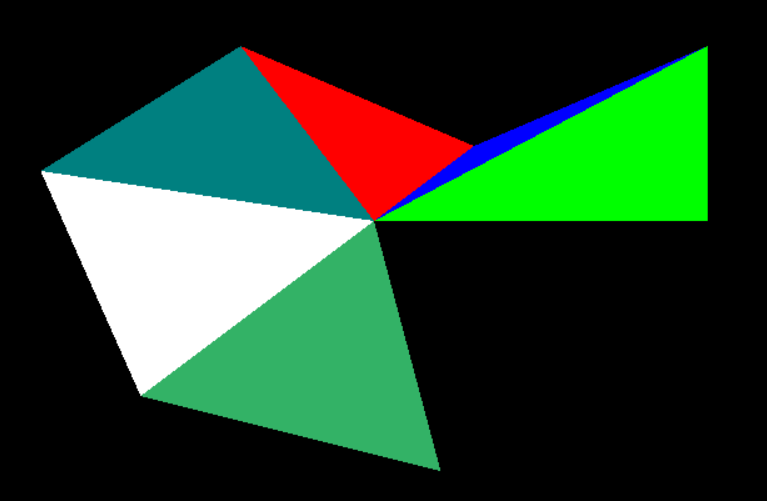


Рисунок 2.5 – Задание 5-1

def draw\_triangle\_strip():  
 vertices = [  
 (1.0, 0.0),  
 (1.0, 0.7),  
 (0.0, 0.0),  
 (0.3, 0.3),  
 (-0.4, 0.7),  
 (0.0, 0.0),  
 (-1.0, 0.2),  
 (-0.7, -0.7),  
 (0.0, 0.0),  
 (0.2, -1.0)  
 ]  
 glBegin(GL\_TRIANGLE\_STRIP)  
  
 for i in range(len(vertices)):  
 glColor3f(\*colors[i])  
 glVertex2f(\*vertices[i])  
  
 glEnd()

На рисунке 2.6 приведено выполнение задания 5-2.

Изображение выглядит как Красочность, Графика, треугольник, дизайн

Автоматически созданное описание

Рисунок 2.6 – Задание 5-2

def draw\_triangle\_fan():  
 vertices = [  
 (0.0, 0.0),   
 (1.0, 0.0),  
 (1.0, 0.7),  
 (0.3, 0.3),  
 (-0.4, 0.7),  
 (-1.0, 0.2),  
 (-0.7, -0.7),  
 (0.2, -1.0)  
 ]  
 glBegin(GL\_TRIANGLE\_FAN)  
 for i in range(len(vertices)):  
 glColor3f(\*colors[i])  
 glVertex2f(\*vertices[i])  
 glEnd()

На рисунке 2.7 приведено выполнение задания 5-3.

Изображение выглядит как Красочность, Графика, дизайн

Автоматически созданное описание

Рисунок 2.7 – Задание 5-3

**2.6 Задание 6**

def draw\_task6(n, radius=0.5):  
 angle\_step = 2 \* pi / n  
 glBegin(GL\_TRIANGLE\_FAN)  
  
 glColor3fv(random\_color())  
 glVertex2f(0.0, 0.0)   
  
 for i in range(n + 1):  
 angle = i \* angle\_step  
 x = radius \* cos(angle)  
 y = radius \* sin(angle)  
 glColor3fv(random\_color())  
 glVertex2f(x, y)  
 glEnd()

На рисунке 2.8 приведено выполнение задания 6.

Изображение выглядит как фиолетовый, Красочность, Сирень

Автоматически созданное описание

Рисунок 2.8 – Задание 6

**2.7 Задание 7**

def draw\_task7():  
 vertices = [  
 (1, -1), (0.3, -1), (0.8, 1),  
 (0.3, -1), (0.8, 1), (0.3, 0),  
 (0.3, 0), (-0.1, 0.3), (0.8, 1),  
 (0.3, 0), (-0.1, 0.3), (0.0, 0),  
 (-0.1, 0.3), (0.0, 0), (-1, 0.7),  
 (-1, 0.7), (-0.6, -0.5), (0.0, 0),  
 (-1, 0.7), (-0.6, -0.5), (-1, -1),  
 (0, -1), (-0.6, -0.5), (-1, -1)  
 ]  
 glBegin(GL\_TRIANGLES)  
  
 for i in range(len(vertices)):  
 glColor3f(\*colors[i])  
 glVertex2f(\*vertices[i])  
  
 glEnd()

На рисунке 2.9 приведено выполнение задания 7.

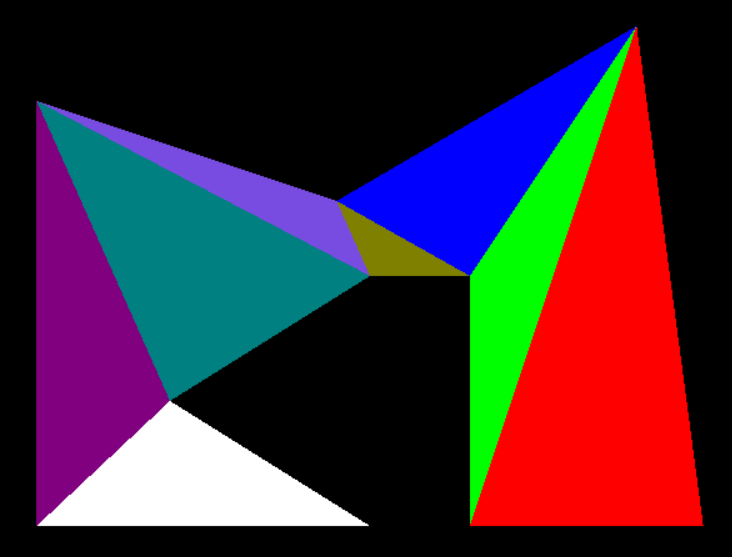


Рисунок 2.9 – Задание 7

**2.8 Задание 8**

Все координаты и значения для фигуры в задании 8 остались прежними, но перед непосредственной отрисовкой была добавлены строки:

glPolygonMode(GL\_FRONT, GL\_POINT)  
glPolygonMode(GL\_BACK, GL\_FILL)

На рисунке 2.10 приведено выполнение задания 8-1

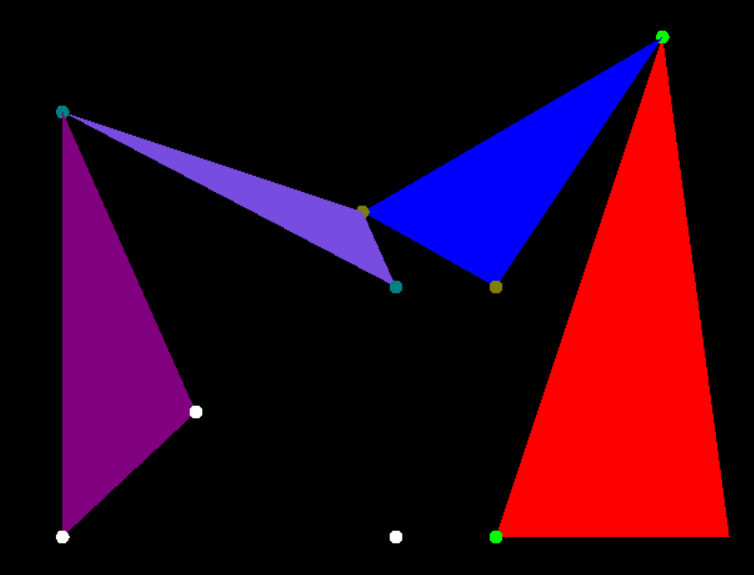


Рисунок 2.10 – Задание 8-1

glPolygonMode(GL\_FRONT, GL\_FILL)  
glPolygonMode(GL\_BACK, GL\_LINE)

На рисунке 2.11 приведено выполнение задания 8-2.

Изображение выглядит как Красочность, треугольник, Графика, линия

Автоматически созданное описание

Рисунок 2.11 – Задание 8-2

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE)

На рисунке 2.12 приведено выполнение задания 8-3.

Изображение выглядит как линия, Красочность, искусство, дизайн

Автоматически созданное описание

Рисунок 3.12 – Задание 8в