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

|  |  |  |
| --- | --- | --- |
| ПРАКТИЧЕСКАЯ РАБОТА | № 5 |  |
|  |  |  |
| Сложные трёхмерные объекты, камера | | |
|  | | |
| Вариант №3 | | |
|  | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ОБУЧАЮЩИЙСЯ** | | | | | | |
| группы | | | |  | | О727Б |
|  |  | Cмагин Т.В. | | | | |
| подпись |  | фамилия и инициалы | | | | |
| 21.11.24 | | | | |
| дата сдачи | | | | |
| **ПРОВЕРИЛ** | | | | | | |
| к.пед.н., доцент, доцент | | | | | | | |
| ученая степень, ученое звание, должность | | | | | | | |
|  |  | Снижко Е.А. | | | | | |
| подпись |  | фамилия и инициалы | | | | | |
| Оценка / балльная оценка | | |  | | | |
| 21.11.24 | | | | |
| дата проверки | | | | |

**СОДЕРЖАНИЕ**

[1 Цель работы и постановка задачи 3](#_Toc182491109)

[1.1 Цель работы 3](#_Toc182491110)

[1.2 Вариативная часть задания 3](#_Toc182491111)

[2 Демонстрация реализации и работы программы 4](#_Toc182491112)

[2.1 Первое задание 4](#_Toc182491113)

[2.2 Второе задание 4](#_Toc182491114)

[2.3 Третье задание 4](#_Toc182491115)

[2.3 Четвертое задание 5](#_Toc182491116)

[2.3 Пятое задание 6](#_Toc182491117)

# **1 Цель работы и постановка задачи**

## **1.1 Цель работы**

Изучение возможностей моделирования сложных графических объектов использованием средств библиотеки OpenGL; установка и настройка параметров камеры; сравнение результатов, полученных при перемещении камеры и при перемещении объекта.

## **1.2 Вариативная часть задания**

3 вариант.

Угол вертикального обзора 35 градусов.

Расстояние до передней плоскости отсечения: 6.

Расстояние до задней плоскости отсечения: 12.

Объект «Тележка».

# **2 Демонстрация реализации и работы программы**

## **2.1 Первое задание**

Установить перспективу в соответствии с вариантом.

Ниже представлен фрагмент кода, устанавливающий перспективу:

gluPerspective(FOV, (display[0] / display[1]), NEAR\_CLIP, FAR\_CLIP)  
glTranslatef(0.0, 0.0, -camera\_distance)

## **2.2 Второе задание**

С помощью загрузки моделей или готовых объектов, изобразить сцену в соответствии с вариантом.

Ниже представлен фрагмент кода, включающий загрузку модели.

model = pywavefront.Wavefront('model.obj', create\_materials=True, collect\_faces=True)  
model\_display\_list = glGenLists(1)  
  
glNewList(model\_display\_list, GL\_COMPILE)  
glBegin(GL\_TRIANGLES)  
for mesh in model.mesh\_list:  
 for face in mesh.faces:  
 for vertex\_i in face:  
 glVertex3fv(model.vertices[vertex\_i])  
glEnd()  
glEndList()

## **2.3 Третье задание**

Установить камеру таким образом, чтобы она была направлена в центр сцены.

Ниже представлены фрагменты, которые позволяют вывести модель на экран.

FOV = 35  
NEAR\_CLIP = 6  
FAR\_CLIP = 12  
  
camera\_distance = 10  
camera\_angle = 0  
object\_rotation = 0  
object\_position\_x = 0  
object\_position\_z = 0  
  
pygame.init()  
display = (800, 600)  
pygame.display.set\_mode(display, DOUBLEBUF | OPENGL)  
gluPerspective(FOV, (display[0] / display[1]), NEAR\_CLIP, FAR\_CLIP)  
glTranslatef(0.0, 0.0, -camera\_distance)

Cцена представлена на рисунке 1.



Рисунок 1 – сцена

## **2.3 Четвертое задание**

Реализовать перемещение камеры относительно объекта:

− объезд камеры вокруг объекта;

− приближение/удаление камеры относительно объекта.

Ниже представлен фрагмент кода, который отвечает за изменение положения камеры.

keys = pygame.key.get\_pressed()

if keys[K\_LEFT]:  
 camera\_angle += 5  
if keys[K\_RIGHT]:  
 camera\_angle -= 5  
if keys[K\_UP]:  
 camera\_distance -= 0.5  
if keys[K\_DOWN]:  
 camera\_distance += 0.5

Изображение повёрнутой модели приведено на рисунке 2.



Рисунок 2 – Повёрнутая модель

## **2.3 Пятое задание**

Реализовать движение объекта относительно камеры:

− движение объекта влево/вправо относительно камеры;

− вращение объекта вокруг своей оси.

Ниже представлен фрагмент кода, который отвечает за изменение положения объекта.

if keys[K\_a]:  
 object\_position\_x -= 0.1 \* cos(radians(camera\_angle))  
 object\_position\_z -= 0.1 \* sin(radians(camera\_angle))  
if keys[K\_d]:  
 object\_position\_x += 0.1 \* cos(radians(camera\_angle))  
 object\_position\_z += 0.1 \* sin(radians(camera\_angle))  
if keys[K\_w]:  
 object\_rotation += 5  
if keys[K\_s]:  
 object\_rotation -= 5

На рисунке 3 – представлено изменение положения объекта.



Рисунок 3 – Изменённое положение объекта