Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ

УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра КСУП

Векторно-полигональная и аналитическая модели объекта. Преобразования в пространстве (3D-преобразования)

Отчет по лабораторной работе № 4

по дисциплине «Компьютерная графика»

Студенты гр. 582-1

_____ К. Н. Полушвайко

_____ А. Д. Рязанов

_____ А. А. Юрьев

«___» _____ 2024 г.

Проверил

канд. техн. наук,

доцент каф. КСУП

_____ Н. Ю. Хабибулина

«___» _____ 2024 г.

2024 г.

1 Цель работы и постановка задачи

Цель работы – изучение и реализация алгоритмов построения и преобразования трехмерного объекта с использованием векторнополигональной и аналитической моделей; изучение и реализация алгоритмов построения проекции фигуры, удаления невидимых линий и граней; изучение принципов применения библиотеки OpenGL при разработке приложений в C#.

Варианты задач:

1. Построение трехмерной функции (Вариант 6) – А. Юрьев;
2. Работа с библиотекой OpenGL – А. Рязанов;
3. Преобразование многогранника в пространстве с удалением невидимых частей (Вариант 1) – К. Полушвайко.

Роли:

- Руководитель – А. Юрьев;
- Технический писатель – А. Рязанов;
- Инженер-программист – К. Полушвайко.

2 Анализ задачи

Рассмотрим подробнее каждую из задач.

OpenGL (Open Graphics Library) – это некая спецификация, включающая в себя несколько сотен функций. Она определяет независимый от языка программирования кроссплатформенный программный интерфейс, с помощью которого программист может создавать приложения, использующие двухмерную и трехмерную компьютерную графику.

Так как прямой поддержки OpenGL в .NET Framework нет, поэтому мы будем использовать библиотеку Tao Framework.

Tao Framework – это свободно распространяемая библиотека с открытым исходным кодом, предназначенная для быстрой и удобной разработки кроссплатформенного мультимедийного программного обеспечения в среде .NET Framework и Mono. Tao Framework – это свободно

распространяемая библиотека с открытым исходным кодом, предназначенная для быстрой и удобной разработки кроссплатформенного мультимедийного программного обеспечения в среде .NET Framework и Mono.

Исходя из методички был воссоздан пример, в котором была реализована возможность изменения цвета объекта и перемещения точки освещения, а также непрерывный поворот объекта.

В качестве алгоритма удаления невидимых частей был реализован алгоритм Робертса, который представлен ниже:

1) Сформировать многоугольники (грани и ребра), исходя из списка вершин тела.
2) Вычислить уравнение плоскости для каждой полигональной грани тела.
3) Проверить знак уравнения плоскости:
    a. Взять любую точку внутри тела, например, усреднив координаты его вершин.
    b. Вычислить скалярное произведение уравнения плоскости и точки внутри тела.
    c. Если это скалярное произведение < 0, то изменить знак уравнения этой плоскости (чтобы отразить правильное направление нормали).
4) 4) Сформировать матрицу тела.
5) Умножить ее на матрицы видового преобразования и проецирования.
6) Определить нелицевые плоскости:
    a. Вычислить скалярное произведение пробной точки, лежащей в «минус» бесконечности, на преобразованную матрицу тела.
    b. Если это скалярное произведение < 0, то плоскость невидима.
    c. Удалить весь многоугольник, лежащий в этой плоскости.

Для вывода трехмерной функции был использовал обычных алгоритм вывода 3д объекта.

2 Описание структуры программы

Для удобства реализации и компоновки проекта был создан репозиторий на github с рабочим процессом Git-flow. Каждая лабораторная реализована в отдельной форме. А каждая задача была реализована в отдельном пользовательском интерфейсе для удобства соединения всех задач в одно целое. Также такой подход уменьшал количество конфликтов, при слиянии веток в репозитории.

3 Руководство пользователя

При запуске программы вас встретит главное меню с кнопками (рисунок 4.1). Нужная активная кнопка – это «лабораторная работа 4», нажав на нее откроется новое окно с вкладками, где находятся задачи на первую лабораторную работу (рисунки 4.2-4.5).

Каждая вкладка имеет PictureBox, в котором демонстрируется работа алгоритмов.

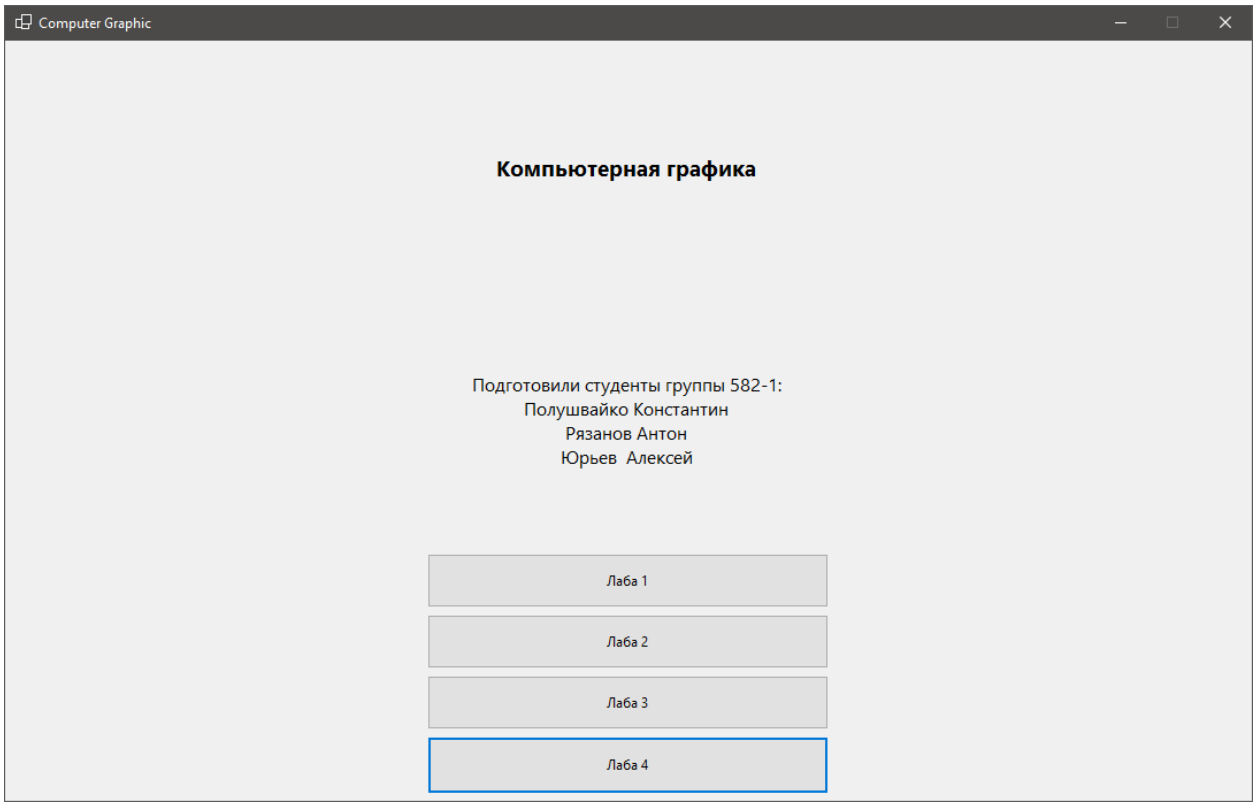

Рисунок 4.1 – Главное меню

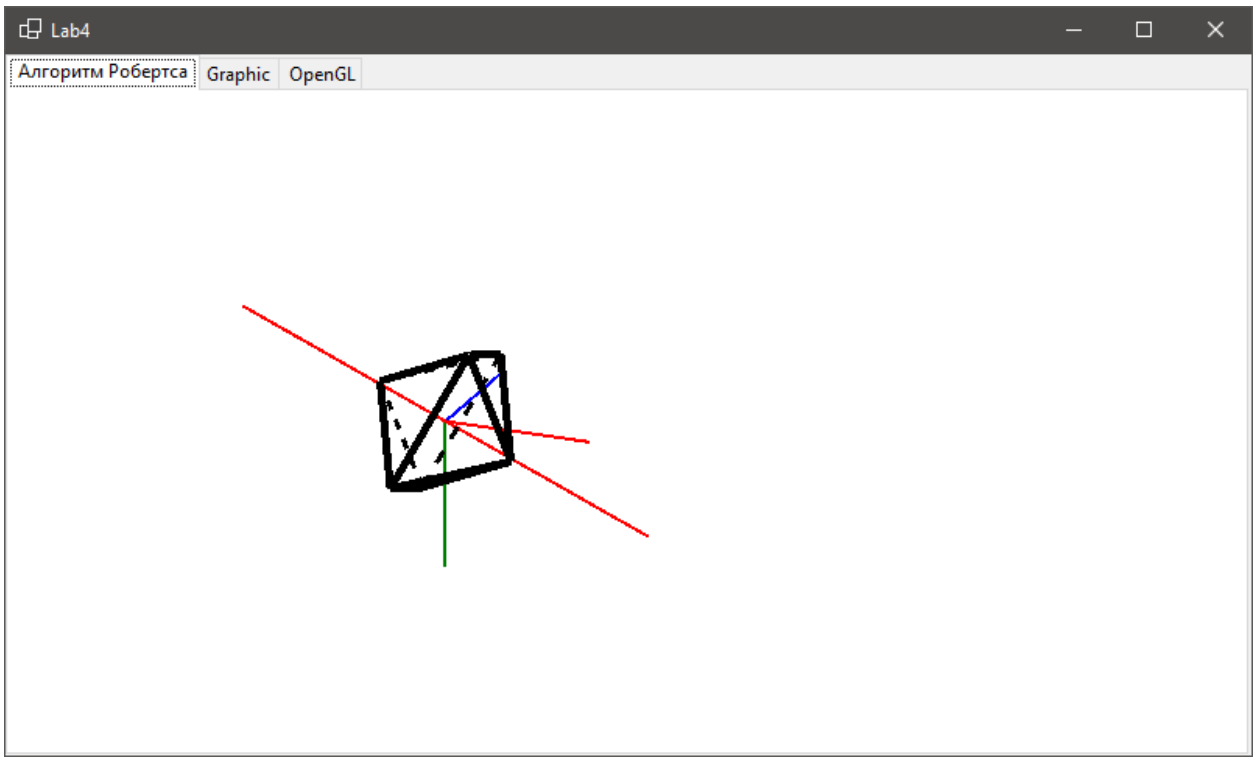

Рисунок 4.2 – Алгоритм Робертса

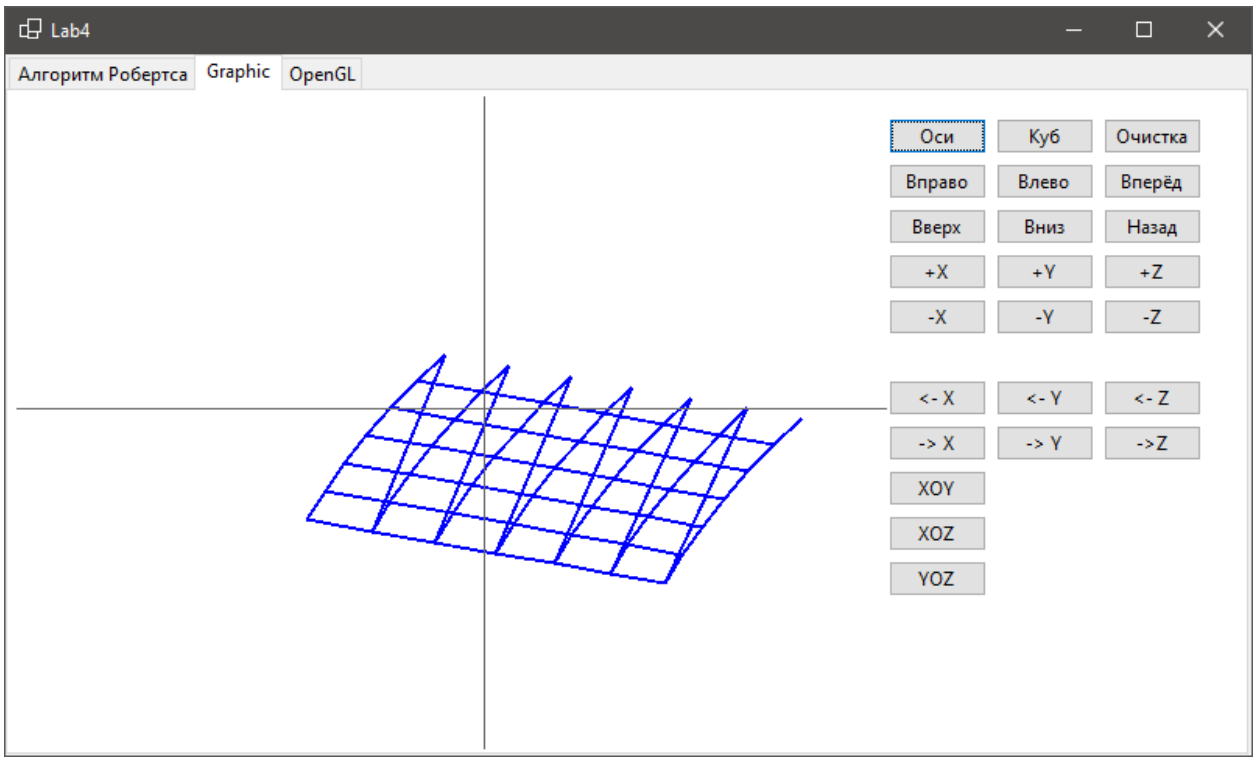

Рисунок 4.3 – Вывод трехмерной функции

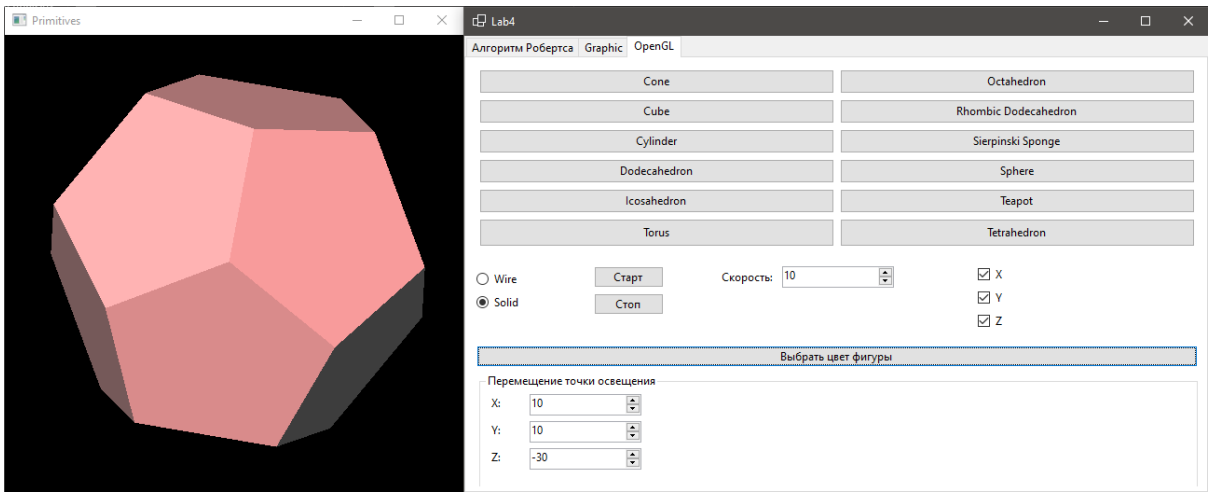

Рисунок 4.3 – Работа с объектами OpenGL

4 Контрольные вопросы по лабораторной работе № 4

1. Что такое векторно-полигональная модель объекта?

Векторно-полигональная модель объекта представляет собой геометрическое описание объекта, состоящее из множества полигонов, обычно треугольников или четырехугольников. Векторы определяют координаты вершин этих полигонов, что позволяет точно представить форму и структуру объекта в трехмерном пространстве.

2. Опишите способы представления векторной полигональной модели

Способы представления векторной полигональной модели включают:

- Списки вершин: Массив координат, где каждая вершина описывается своими координатами (x, y, z).
- Списки полигонов: Массив, где каждый элемент указывает на вершины, образующие полигоны.
- Нормали: Векторы нормалей для полигонов или вершин для освещения и теней.
- Текстурные координаты: Координаты для наложения текстур на полигоны.
- Материалы и цвета: Свойства материалов, такие как цвет, отражение, прозрачность.

3. Опишите матрицу преобразований в пространстве общего вида. Раскройте назначение каждого элемента

Матрица преобразований в трехмерном пространстве обычно имеет размер 4x4 и выглядит так:

$$\begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix}$$

Где:

- m_{11}, m_{12}, m_{13}, m_{21}, m_{22}, m_{23}, m_{31}, m_{32}, m_{33}: Компоненты, определяющие поворот и масштабирование.
- m_{14}, m_{24}, m_{34}: Компоненты, определяющие перемещение (translation).
- m_{41}, m_{42}, m_{43}: Компоненты, связанные с перспективным проецированием.
- m_{44}: Обычно равен 1 для стандартных преобразований.

4. Опишите матрицы, используемые для 3D-преобразования

Основные матрицы для 3D-преобразований включают:

- Матрица трансляции:

$$\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Матрица масштабирования:

$$\begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Матрицы вращения:
  - Вокруг оси X:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

  - Вокруг оси Y:

$$\begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Вокруг оси Z:

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

5. Опишите процесс получения 3D-преобразований

Процесс получения 3D-преобразований включает:

- Определение необходимого преобразования (трансляция, масштабирование, вращение).
- Создание соответствующей матрицы преобразования.
- Применение матрицы к координатам точек объекта с помощью матричного умножения.
- Комбинирование нескольких преобразований путем умножения соответствующих матриц.
-

6. Что такое «аналитическая модель»?

Аналитическая модель – это математическое описание объекта или системы с помощью уравнений и формул. В контексте графики, аналитическая модель может описывать поверхность объекта через функции или уравнения.

7. Опишите процедуру отображения графика трехмерной функции на экране

- Выбор функции: Определение трехмерной функции f(x, y, z).

- Дискретизация: Разделение области определения функции на сетку.
- Вычисление значений функции в узлах сетки.
- Создание полигональной сетки: Использование триангуляции для соединения узлов сетки.
- Преобразования координат: Применение матриц преобразований (масштабирование, поворот, перемещение).
- Проецирование: Перевод 3D-координат в 2D для отображения на экране.
- Растеризация: Отображение полученных полигонов на экране.

8. Опишите процедуры получения вращения, масштабирования и перемещения полученной поверхности

- Вращение: Умножение координат точек на матрицы вращения вокруг соответствующих осей.
- Масштабирование: Умножение координат точек на матрицу масштабирования.
- Перемещение (трансляция): Добавление вектора трансляции к координатам точек.

9. Что такое проецирование? Для чего оно используется

Проецирование – это процесс перевода 3D-координат в 2D-координаты для отображения на экране. Используется для визуализации трехмерных объектов на двухмерных дисплеях.

10. Опишите существующую классификацию видов проекций

- Параллельные проекции:
    - Ортографическая: Проецирование перпендикулярно плоскости проекции.

- Косоугольная (обликовая): Проецирование под углом к плоскости проекции.

- Перспективные проекции:
  - Одноточечная перспектива: Проецирование с одной точкой схода.
  - Двухточечная перспектива: Проецирование с двумя точками схода.
  - Трехточечная перспектива: Проецирование с тремя точками схода.

11. В чем основное различие параллельной и перспективной проекции

- Параллельная проекция: Прямые, параллельные в пространстве, остаются параллельными на проекции.
- Перспективная проекция: Прямые, параллельные в пространстве, сходятся в одной или нескольких точках схода, создавая иллюзию глубины.

12. Опишите процедуру получения вращения трехмерной фигуры на экране

- Выбор оси и угла вращения.
- Построение матрицы вращения вокруг выбранной оси.
- Умножение координат точек фигуры на матрицу вращения.
- Перевод преобразованных координат в 2D с помощью матрицы проекции.
- Отображение проецированных точек на экране.

13. Сущность и назначение OpenGL

OpenGL (Open Graphics Library) – это стандарт, определяющий кроссплатформенный интерфейс для программирования графики. Предназначен для рендеринга 2D и 3D-графики.

14. Назначение Tao Framework

Tao Framework – это набор оберток для использования OpenGL и других графических библиотек в C#. Позволяет разработчикам использовать мощные возможности OpenGL из среды .NET.

15. Инициализация OpenGL в C#

- Создание окна с контекстом OpenGL.
- Установка параметров отображения (буферов, глубины и т.д.).
- Инициализация библиотеки OpenGL через обертки (например, Tao Framework).
- Загрузка и компиляция шейдеров.
- Установка начальных состояний OpenGL (настройка вида, проекции и т.д.).

16. Опишите основные методы преобразования объектов, реализованные в OpenGL, использованные в лабораторной работе

- glTranslate: Перемещение объекта.
- glScale: Масштабирование объекта.
- glRotate: Вращение объекта.
- glLoadMatrix: Загрузка пользовательской матрицы.
- glMultMatrix: Умножение текущей матрицы на пользовательскую.
- glPushMatrix и glPopMatrix: Сохранение и восстановление состояния матриц.

Эти методы позволяют выполнять различные геометрические преобразования для отображения и манипуляции 3D-объектов в пространстве.

5 Заключение

В данной лабораторной работе был изучен и реализован алгоритм Робертса (удаления невидимых линий), изучена библиотека Tao Framework для работы с OpenGL и реализован вывод трехмерной функции.