



**Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

# **Разработка программного обеспечения для наложения текстур на трехмерные объекты**

**Студент: Фам Минь Хиеу ИУ7-52Б**

**Научный руководитель: Куров Андрей Владимирович**

2023 г.

# Цели и задачи

**Целью** данной курсовой работы является разработка программного обеспечения для наложения текстур на трёхмерные объекты.

## **Задачи:**

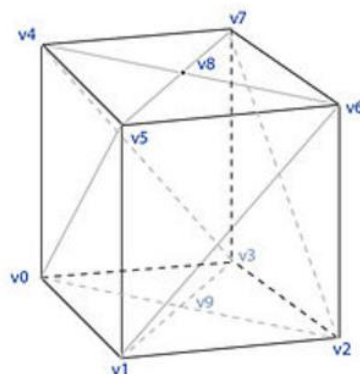
- провести анализ существующих алгоритмов компьютерной графики, используемых для создания трехмерных сцен;
- выбрать алгоритмы для решения поставленной задачи;
- выбрать язык программирования и среду разработки для реализации поставленной задачи;
- создать программное обеспечение, реализующее выбранные алгоритмы;
- провести замеры временных характеристик разработанного программного обеспечения.

# Методы представления полигональных моделей

Вершинное представление

Список вершин

v0	0,0,0	v1 v5 v4 v3 v9
v1	1,0,0	v2 v6 v5 v0 v9
v2	1,1,0	v3 v7 v6 v1 v9
v3	0,1,0	v2 v6 v7 v4 v9
v4	0,0,1	v5 v0 v3 v7 v8
v5	1,0,1	v6 v1 v0 v4 v8
v6	1,1,1	v7 v2 v1 v5 v8
v7	0,1,1	v4 v3 v2 v6 v8
v8	.5,.5,0	v5 v6 v7 v8
v9	.5,.5,1	v0 v1 v2 v3



✓ Список граней

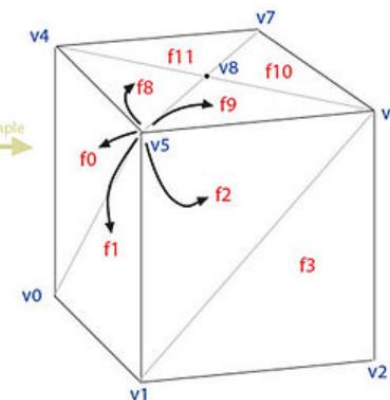
Список граней

f0	v0 v4 v5
f1	v0 v5 v1
f2	v1 v5 v6
f3	v1 v6 v2
f4	v2 v6 v7
f5	v2 v7 v3
f6	v3 v7 v4
f7	v3 v4 v0
f8	v8 v5 v4
f9	v8 v6 v5
f10	v8 v7 v6
f11	v8 v4 v7
f12	v9 v5 v4
f13	v9 v6 v5
f14	v9 v7 v6
f15	v9 v4 v7

Список вершин

v0	0,0,0	f0 f1 f12 f15 f7
v1	1,0,0	f2 f3 f13 f12 f1
v2	1,1,0	f4 f5 f14 f13 f3
v3	0,1,0	f6 f7 f15 f14 f5
v4	0,0,1	f6 f7 f0 f8 f11
v5	1,0,1	f0 f1 f2 f9 f8
v6	1,1,1	f2 f3 f4 f10 f9
v7	0,1,1	f4 f5 f6 f11 f10
v8	.5,.5,0	f8 f9 f10 f11
v9	.5,.5,1	f12 f13 f14 f15


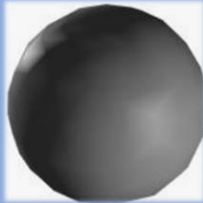
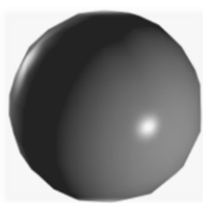
example



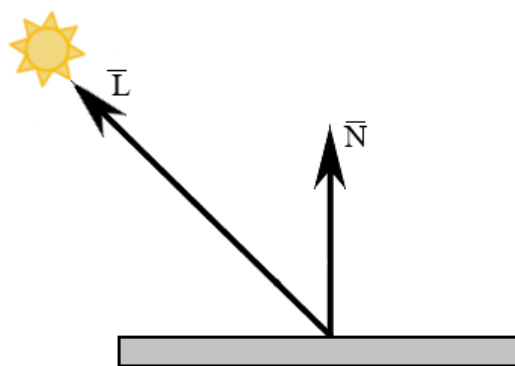
# Анализ алгоритмов удаления невидимых линий и поверхностей

Критерии сравнения \ Алгоритм	Алгоритм z-буфера	Алгоритм обратной трассировки лучей	Алгоритм Варнока
Пространство, в котором работает алгоритм	Пространство изображения	Пространство изображения	Пространство изображения
Сложность ( $N$ – количество граней, $C$ – количество пикселей)	$O(CN)$	$O(CN)$	$O(CN)$
Эффективность для сложный сцен	Высокая	Низкая	Средняя
Сложность реализации	Низкая	Средняя	Средняя

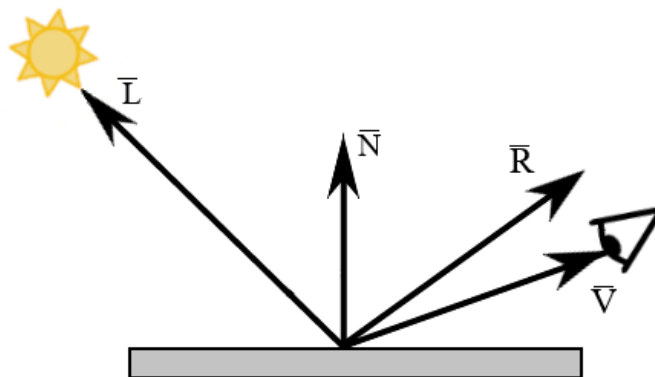
# Методы закрашки

Критерии сравнения \ Алгоритм	Простая закраска	Метод Гуро	Закраска по Фонгу
Сложность реализации	Низкая	Средняя	Высокая
Скорость работы	Высокая	Средняя	Низкая
Пример закрашки			

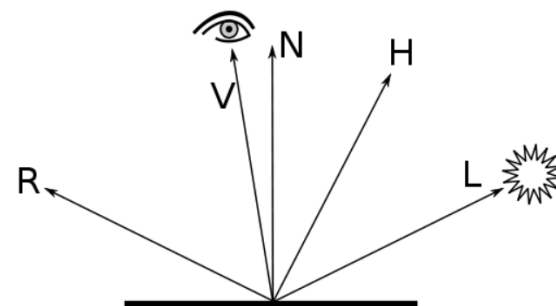
# Методы освещения



Метод Ламберта



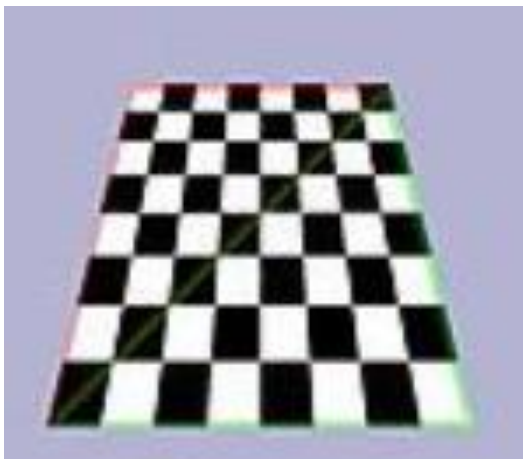
Метод Фонга



✓ Метод Фонга-Блинна

# Методы текстуризации

Перспективно-корректное текстурирование

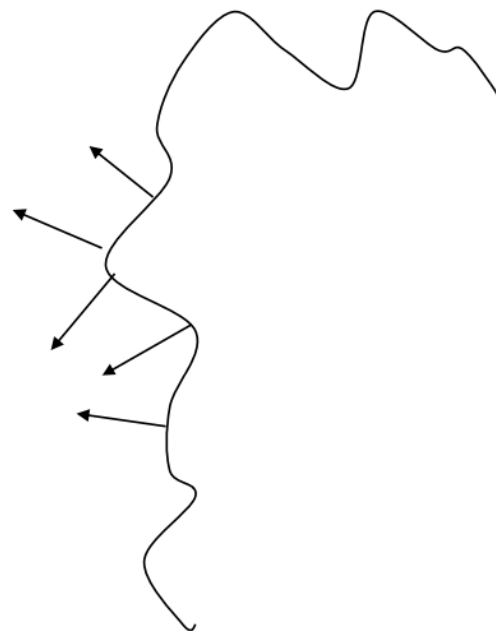


$$u = (u/Z) / (1/Z),$$

$$v = (v/Z) / (1/Z).$$



Моделирование неровностей



# Схема алгоритмов



Схема алгоритма синтеза изображения

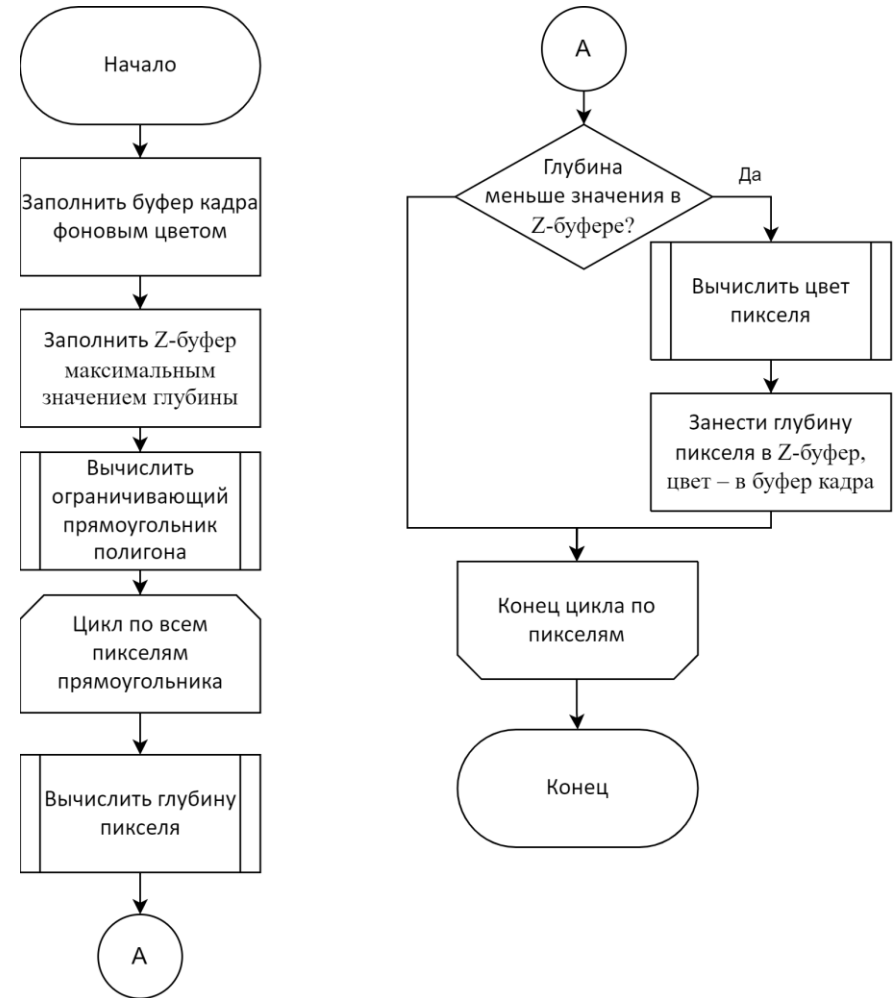
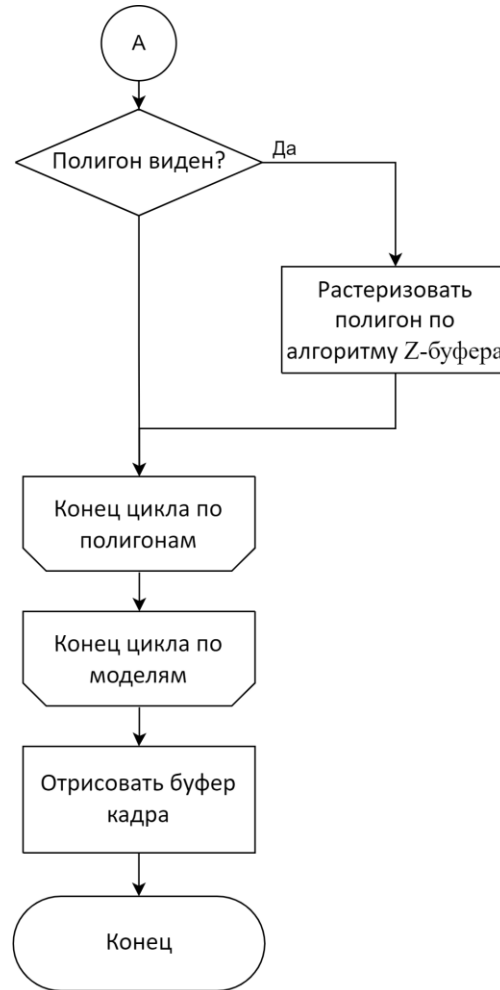
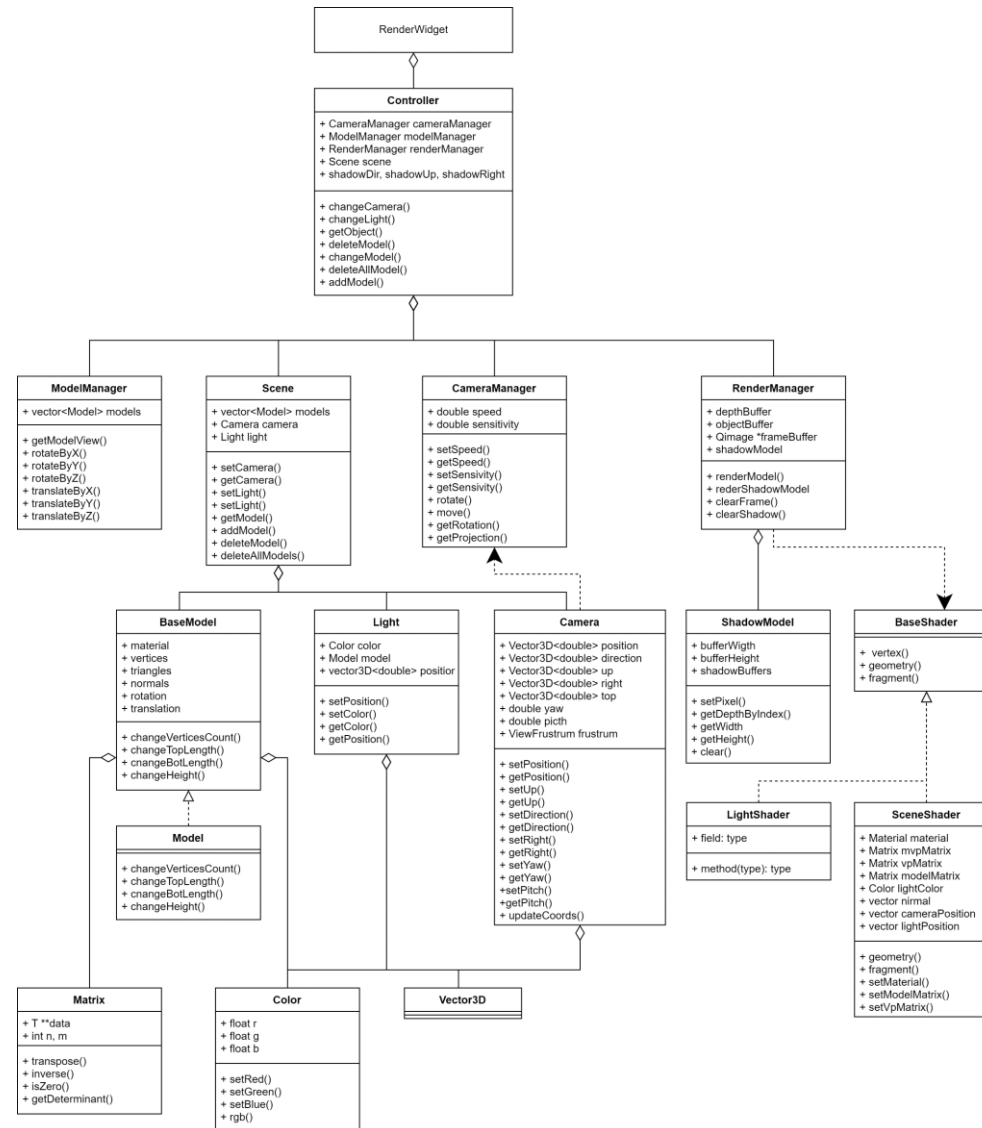


Схема алгоритма с Z-буфером



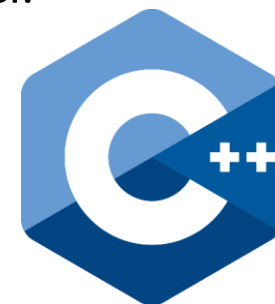
# UML диаграмма классов



# Выбор языка программирования и среды разработки

В качестве языка программирования был выбран Си++.

- С++ — объектно-ориентированный язык, а именно такая методология программирования была выбрана для разработки программы.
- В данном языке имеется большое количество библиотек и шаблонов, позволяющих не тратить время на изобретение готовых конструкций.;
- Обладает высокими показателями вычислительной производительности.

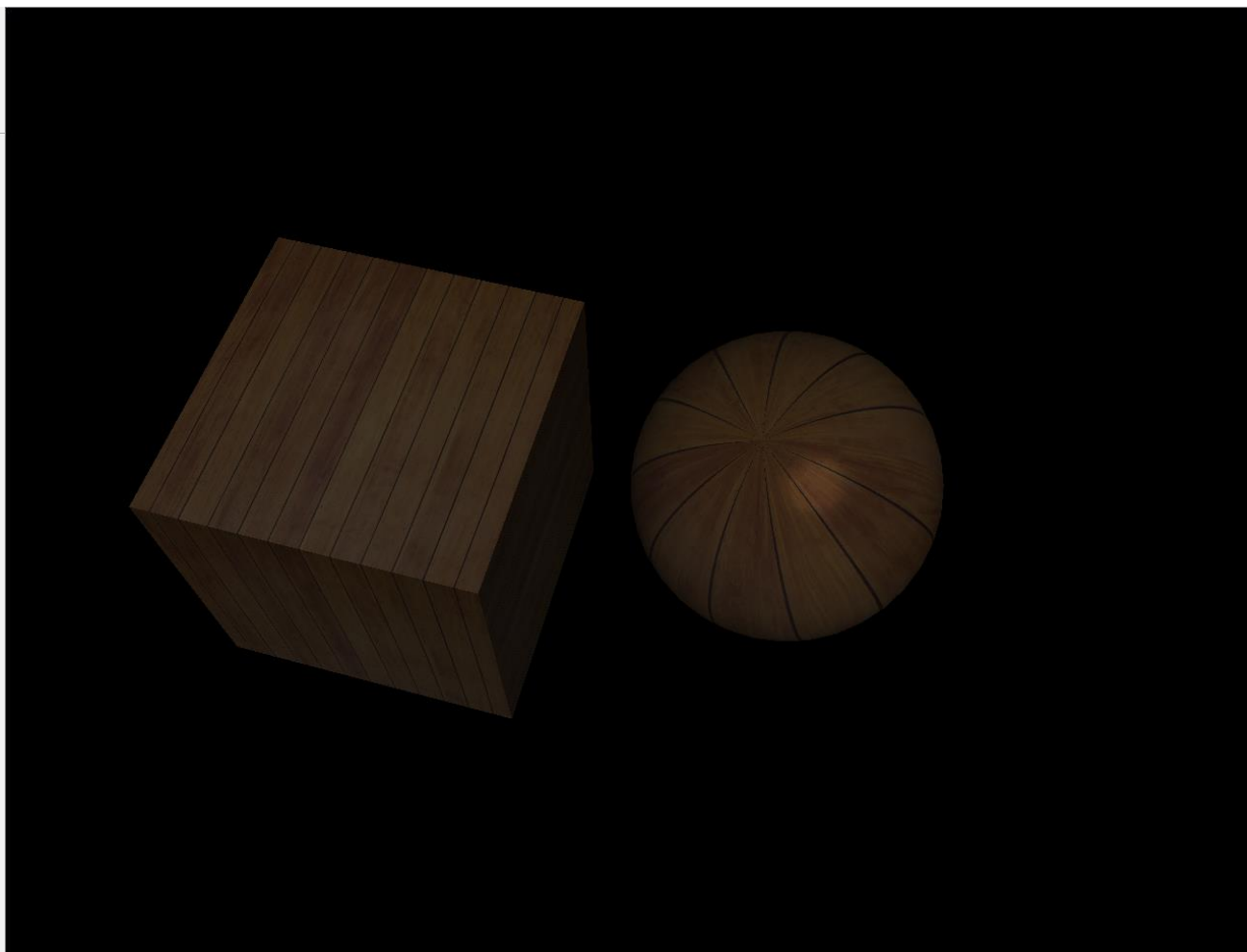
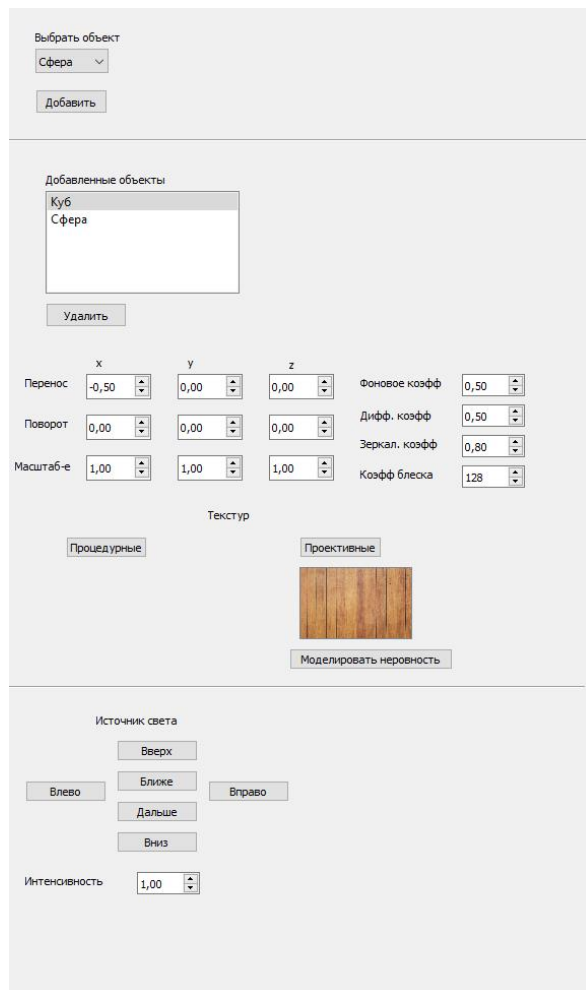


Выбранная среда разработки – QtCreator.

- в QT Creator есть возможность быстрого создания интерфейса с помощью расширения QT Design.
- QT Creator обладает всем необходимым функционалом для написания, профилирования и отладки программ.



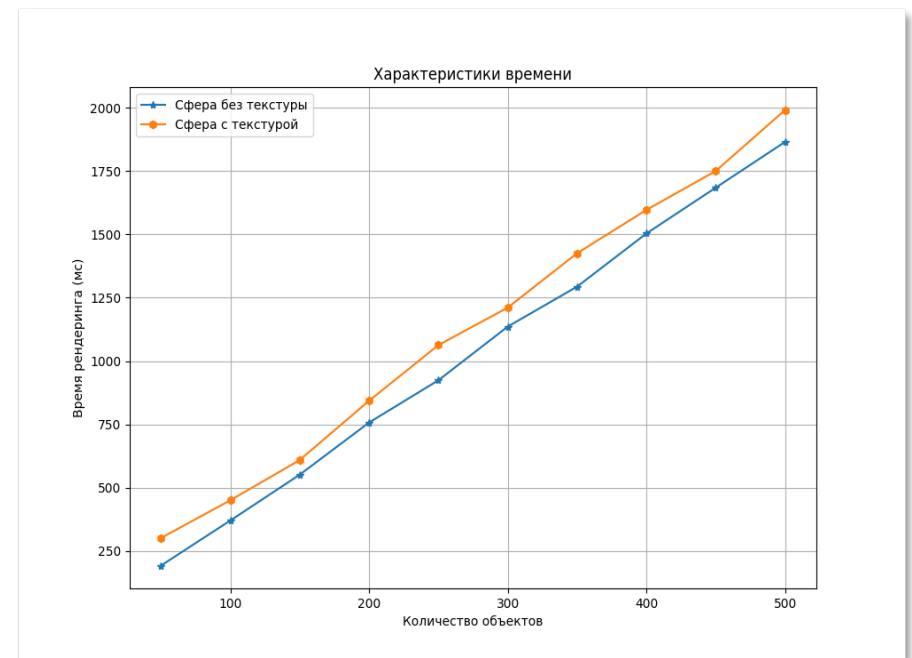
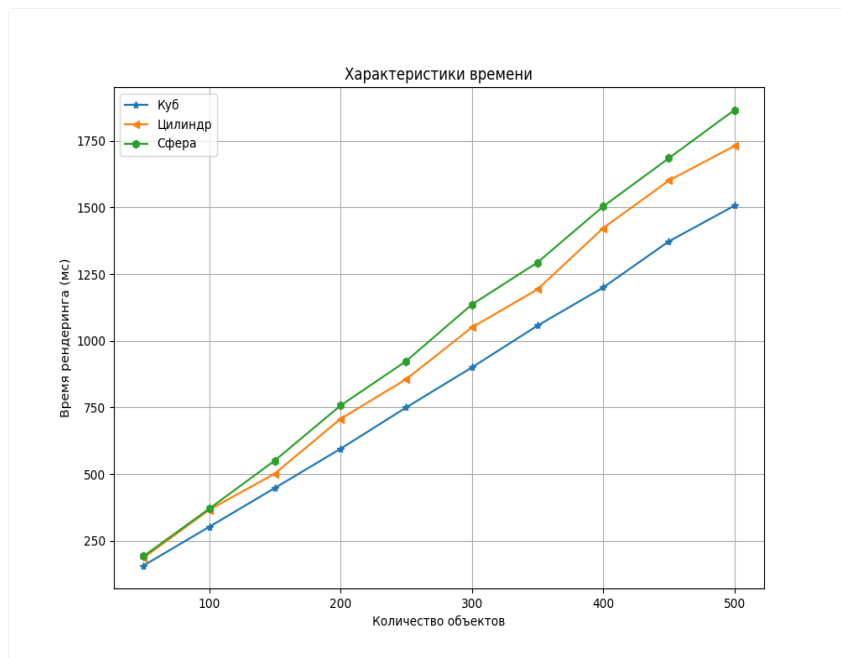
# Интерфейс программы. Пример работы.



# Анализ производительности.

## Зависимость времени рендеринга:

- От количества объектов на сцене
- От учета текстуры



# Заключение

**Цель** курсовой работы была достигнута, то есть был разработано программное обеспечение для наложения текстур на трёхмерные объекты.

Для достижение цели были решены все задачи:

- проведен анализ существующих алгоритмов компьютерной графики, используемых для создание трехмерной сцены;
- выбраны алгоритмы для решения поставленной задачи;
- выбраны язык программирования и среда разработки для реализации поставленной задачи;
- создано программное обеспечение, реализующее выбранные алгоритмы;
- проведены замеры временных характеристик разработанного программного обеспечения;