



Факультет компьютерных наук

Образовательная программа
«Программная инженерия»

Москва
2024

3D Renderer с нуля

3D Renderer from scratch

Программный курсовой проект

Выполнил студент группы БПИ223
Гетманов Михаил Михайлович

Научный руководитель
Доцент департамента больших данных и
информационного поиска факультета
компьютерных наук
Кантонистова Елена Олеговна



О чём проект?

Ожидаемые компоненты:

- Библиотека отрисовки
- Демонстрационное приложение

Ожидаемые итоги:

- Процесс отрисовки 3D объектов
- Интерактивное наблюдение за объектами
- Добавление объектов в реальном времени



Цель и задачи

Цели:

1. Реализация библиотеки отрисовки 3D объектов
2. Демонстрационное приложение.

Задачи:

1. Выбор технологий для реализации
2. Изучение выбранных технологий
3. Изучение математической теории
4. Разработка архитектуры и продумывание технических решений.
5. Разработка библиотеки и приложения
6. Разработка технической документации



Функциональные требования

Библиотека отрисовки:

- Поддержка треугольных примитивов
- Поддержка нормалей
- Поддержка направленного и фонового света
- Камера
- Процесс отрисовки

Демонстрационное приложение:

- Добавление объектов на сцену в реальном времени
- Парсинг 3D объектов формата .obj в объекты на сцене
- Интерактивное управление камерой



Выбор используемых в работе методов, алгоритмов, моделей

Библиотека отрисовки:
Eigen – линейная алгебра

Приложение:
Message Driven Экосистема – SFML

Стэк технологий:

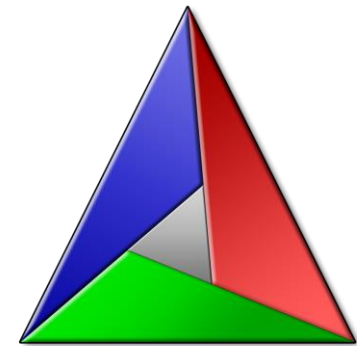
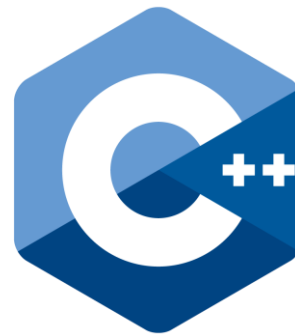
C++

CMake

Git

Clang-format

Clang-tidy





Описание библиотеки

Компонент World:

- Камера
- Мир (Сцена)

Компонент Primitives:

- Треугольник
- Триангулируемый объект

Компонент Light:

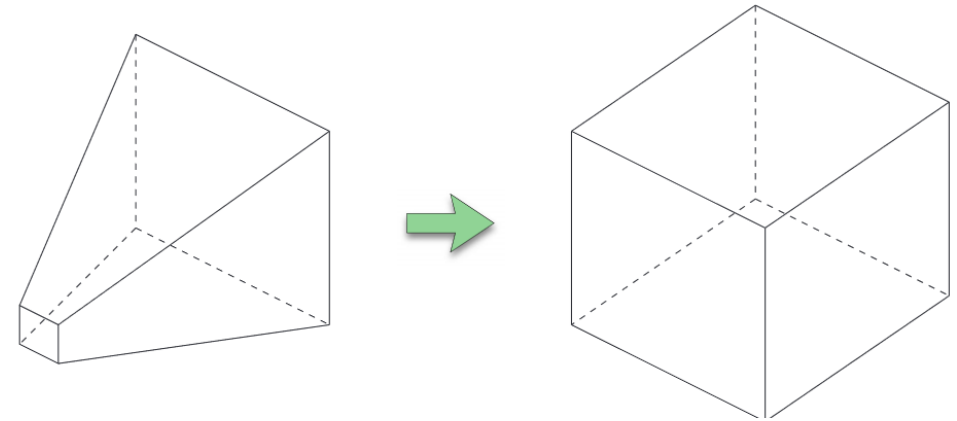
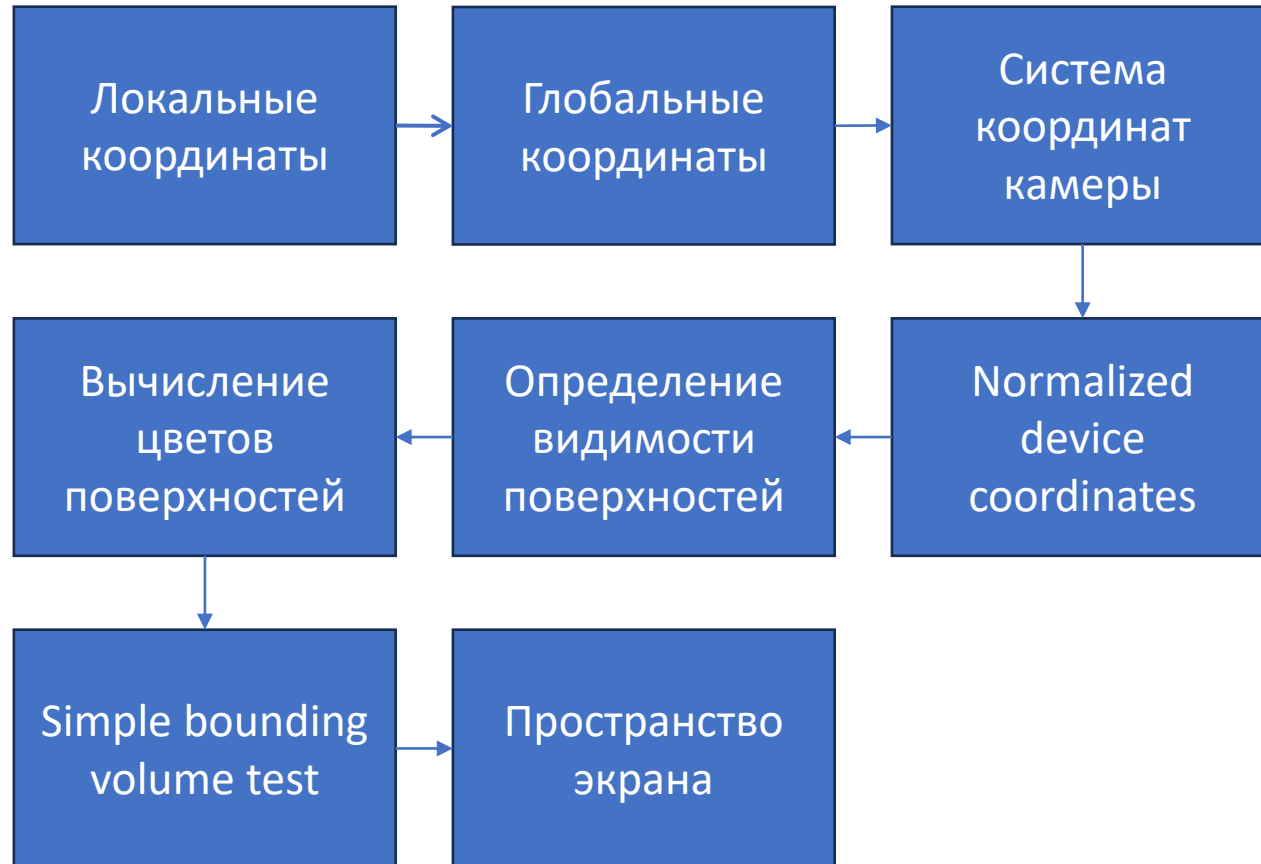
- Цвет
- Арифметика цветов
- Фоновый свет
- Направленный свет

Логика и результат отрисовки:

- Отрисовщик
- Пиксельный экран

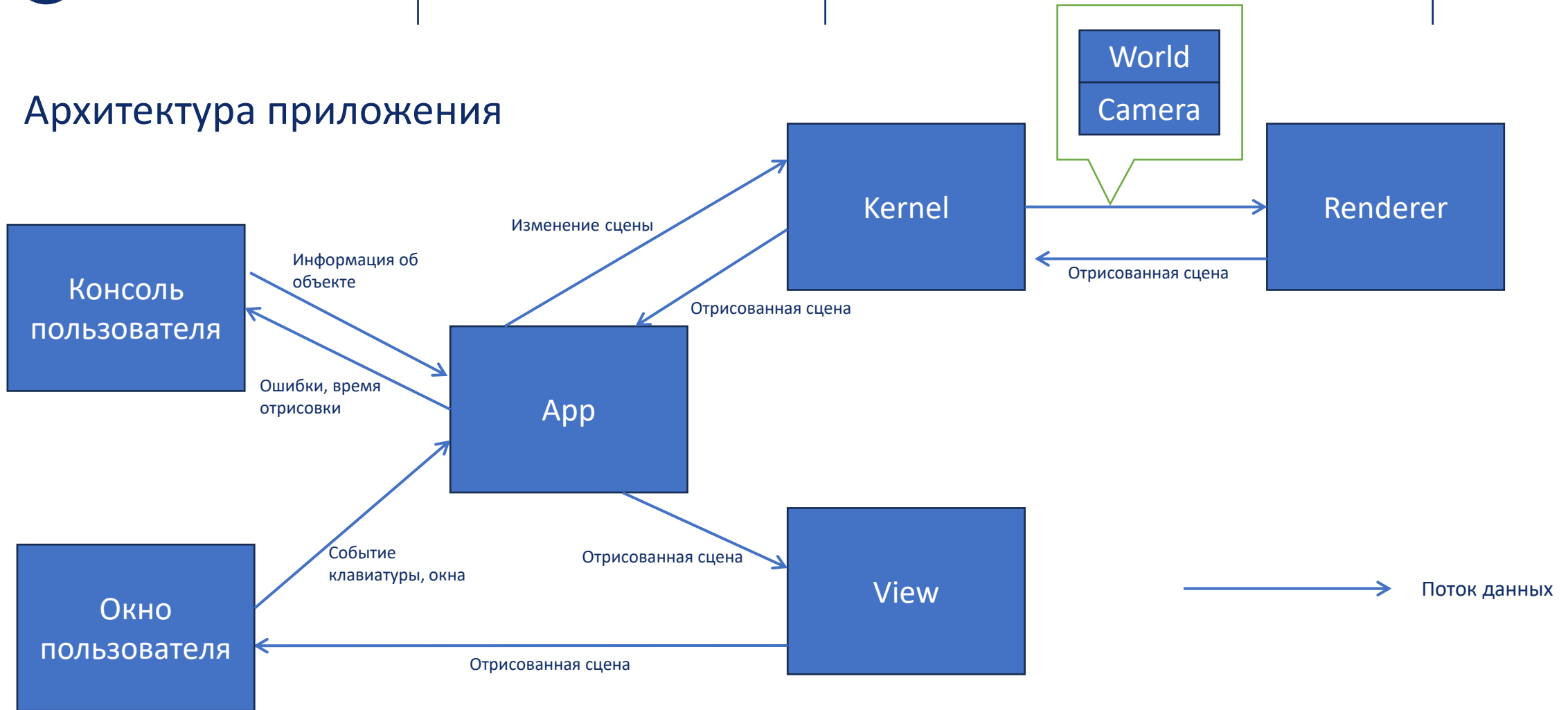


Описание разработанного алгоритма



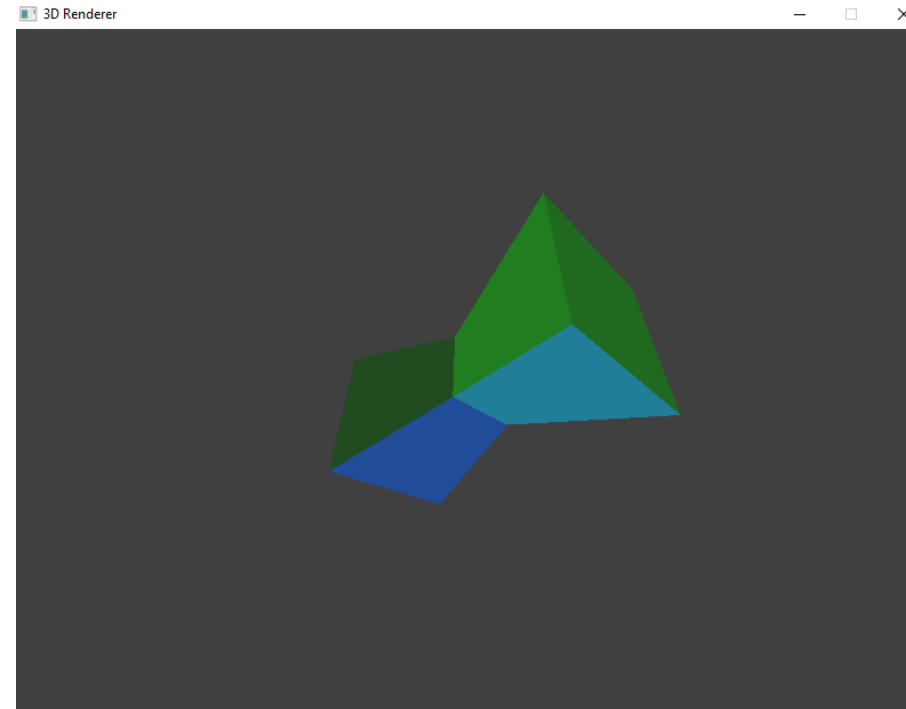


Архитектура приложения





Демонстрация работы



```
D:\programming\3D-Renderer\build\3d_renderer.exe
[INFO] [2024/03/02 09:39:54] New frame was rendered in 49 millisecond.
[INFO] [2024/03/02 09:39:56] New frame was rendered in 17 millisecond.
[INFO] [2024/03/02 09:40:00] New frame was rendered in 7 millisecond.
[INFO] [2024/03/02 09:40:05] New frame was rendered in 8 millisecond.
[INFO] [2024/03/02 09:41:09] New frame was rendered in 7 millisecond.
[INFO] Enter path to .obj file with new object:
D:\programming\3D-Renderer\object_samples\cube.obj
[INFO] Enter color of new object: (format: <R> <G> <B>, where R, G, B are numbers, each number is integer and >=0 and <255)
[INFO] Example: 128 0 255
128 128 128
[INFO] Enter position of new object: (format: <x> <y> <z>, where x, y, z are real numbers)
[INFO] Example: 1.5 -0.5 0
0 0 -3
[INFO] New object was added successfully!
[INFO] [2024/03/02 09:41:21] New frame was rendered in 90 millisecond.
```



Результаты работы

Библиотека:

- Рабочий процесс отрисовки статических триангулируемых объектов с динамической камерой
- Поддержка фонового и направленного света
- Камера
- Базовое определение видимости поверхностей

Демонстрационное приложение:

- Показ пользователю 3D сцены
- Добавление объектов на сцену, заданных в формате .obj
- Перемещение по сцене
- Логгирование информации

```
SetAmbientLight(AmbientLight({64, 64, 64}));  
AddDirectionalLight(DirectionalLight({0, 0, -1}, {255, 0, 0}));  
AddDirectionalLight(DirectionalLight({1, 0, 0}, {0, 0, 255}));  
AddDirectionalLight(DirectionalLight({1, -1, 1}, {0, 255, 0}));
```

```
const PixelScreen& Kernel::MakeScene() {  
    renderer_.Render(world_, cam_, screen_buffer_);  
    return screen_buffer_;  
}
```

