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

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОМУ ПРОЕКТУ

НА ТЕМУ:

«Трёхмерная визуализация распространения Wi-Fi сигнала»

Студент _____ИУ7-55Б_____ _____ _____Е.Б. Гришин_____
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсового проекта _____ _____В.П. Степанов_____
(Подпись, дата) (И.О.Фамилия)

2020 г.

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

УТВЕРЖДАЮ
Заведующий кафедрой ___ИУ7____
(Индекс)
_____ _И.В. Рудаков_____
(И.О.Фамилия)
« _____ » _____ 20 ____ г.

З А Д А Н И Е
на выполнение курсового проекта

по дисциплине _____Компьютерная графика_____

Студент группы ____ИУ7-55Б____

_____Гришин Егор Борисович_____
(Фамилия, имя, отчество)

Тема курсового проекта _____
_____Трёхмерная визуализация распространения Wi-Fi сигнала_____
_____

Направленность КП (учебный, исследовательский, практический, производственный, др.)
_____учебный_____
Источник тематики (кафедра, предприятие, НИР) _____кафедра_____

График выполнения проекта: 25% к _4_ нед., 50% к _7_ нед., 75% к 11 нед., 100% к _14 нед.

Задание *Разработать программу для трёхмерной визуализации распространения Wi-Fi сигнала в помещении. Выбрать методы моделирования распространения сигнала и построения реалистичных изображенпий.*

Оформление курсового проекта:

Расчетно-пояснительная записка на 25-30 листах формата А4.
Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)
На защиту проекта должна быть предоставлена презентация, состоящая из 15-20 слайдов.
На слайдах должны быть отражены: постановка задачи, использованные методы и алгоритмы, расчетные соотношения, структура комплекса программ, интерфейс, результаты проведенных исследований.

Дата выдачи задания « ___ » _____ 20__ г.

Руководитель курсового проекта _____ _____В.П. Степанов___
(Подпись, дата) (И.О.Фамилия)

Студент _____ _____Е.Б. Гришин_____
(Подпись, дата) (И.О.Фамилия)

Оглавление

Введение 4
1. Аналитическая часть 5
1.1 Формализация объектов синтезируемой сцены 5
1.2 Анализ алгоритмов удаления невидимых линий и поверхностей 6
1.2.1 Алгоритм обратной трассировки лучей 6
1.2.2 Алгоритм, использующий Z буфер 7
1.2.3 Алгоритм Робертса 7
1.2.4 Вывод 8
1.3 Анализ методов закрашивания 8
1.4 Описание трехмерных преобразований сцены 9
1.5 Анализ моделей распространения Wi-Fi сигнала 9
1.6 Выводы из аналитического раздела 11
2. Конструкторская часть 12
2.1 Требования к программе 12
2.2 Общий алгоритм визуализации трехмерной сцены 12
2.3 Алгоритм Z-буфера 12
2.4 Простой метод освещения 13
2.5 Получение цвета точки полигона 13
2.6 Выбор используемых типов и структур данных 13
3. Технологическая часть 15
3.1 Выбор и обоснование языка программирования и среды разработки 15
3.2 Структура и состав классов 15
3.3 Сведения о модулях программы 19
4. Экспериментальная часть 21
4.1 Тестирование 21
4.3 Описание эксперимента 25
Заключение 27
Список использованной литературы 28

Введение

Современные компьютеры способны обрабатывать огромное количество информации. Компьютерная графика решает задачу визуализации результатов этой обработки. Она предоставляет широкий спектр алгоритмов для отображения трёхмерных сцен: от самых простых, не учитывающих оптические явления преломления, отражения и рассеивания света, сложную текстуру, до самых сложных, учитывающих большое количество факторов.

Зачастую эти алгоритмы ресурсозатратны: чем более качественное изображение требуется получить, тем больше времени и памяти тратится на его синтез. Это становится проблемой при создании динамической сцены, где на каждом временном интервале необходимо производить расчеты заново.

Цель данной работы – реализовать построение трехмерной сцены и визуализацию распространения Wi-Fi сигнала в пространстве.

Чтобы достигнуть поставленной цели, требуется решить следующие задачи:

1) описать структуру трехмерной сцены, включая объекты, из которых состоит сцена;
2) выбрать и/или модифицировать существующие алгоритмы трехмерной графики, которые позволят визуализировать трехмерную сцену;
3) выбрать подходящую модель распространения радиосигнала Wi-Fi;
4) реализовать данные алгоритмы для создания трехмерной сцены;
5) разработать программное обеспечение, которое позволит отобразить трехмерную сцену и визуализировать распространение Wi-Fi сигнала в пространстве.

1. Аналитическая часть

1.1 Формализация объектов синтезируемой сцены

Для сцены вводится понятие обрабатываемого пространства. Это область объектного координатного пространства, ограничиваемая правильным параллелепипедом, таким, чтобы при любом угле поворота сцены обрабатываемое пространство полностью попадало в экран.

Сцена состоит из следующих объектов:

- Источник света – представляют собой вектор направления света, предполагается, что источник расположен в бесконечности. Цвет свечения описывается через RGB параметры.
- Земля – ограничивающая обрабатываемое пространство снизу прямоугольная плоскость. Имеет ширину и длину отображаемой сцены.
- Препятствие – прямоугольный параллелепипед с гранями, параллельными граням обрабатываемого пространства, а одна из граней лежит в плоскости земли. Высотой препятствие от земли, до верхней границы обрабатываемого пространства. Препятствие задаётся координатами центра грани в плоскости земли (x, z), шириной и длиной.
- Антенна – куб, с гранями, параллельными граням обрабатываемого пространства, расположенный в любой точке обрабатываемого пространства.
- Индикаторный слой – плоскость, параллельная плоскости земли, расположенная на некоторой высоте в пределах обрабатываемого пространства. Текстура индикаторного слоя показывает потери распространения Wi-Fi сигнала на данной высоте.

Препятствия, антенна, индикаторный слой наилучшим образом описываются через поверхностные модели, т.к. каркасные модели не дадут реалистичное изображение, которого нужно достичь, а объемная модель будет излишеством, более затратным по памяти, чем поверхностная.

Поверхностную модель можно задать несколькими способами [1].

Параметрическим представлением – для получения поверхности нужно вычислять функцию, зависящую от параметра. Так как в сцене нет никаких поверхностей вращения, использование параметрического представления будет затруднительно.

Полигональной сеткой – совокупностью вершин, ребер и граней, которые определяют форму объекта [2].

- Вершинное представление (вершины указывают на другие вершины, с которыми они соединены). Для генерации списка граней для рендеринга нужно обойти все данные, что затрудняет работу с ним.
- Список граней представляет объект как множество граней и вершин.
- Таблица углов (веер треугольников) – хранит вершины в предопределенной таблице. Не подойдет для моей задачи, так как изменение данного представления затратно по времени.

Наиболее удобным способом хранения сцены в данном случае является список граней т.к. данные в нем можно эффективно преобразовывать.

1.2 Анализ алгоритмов удаления невидимых линий и поверхностей

При выборе алгоритма удаления невидимых линий нужно учесть особенности поставленной задачи. При фиксированном угле поворота сцены её объекты можно разбить на две группы: статические (земля, препятствия, антенна) и подвижные (индикаторный слой). Отображение динамических объектов можно накладывать на отображение статических, у последних оно не меняется.

Рассмотрим ключевые алгоритмы построения трехмерной сцены [5].

1.2.1 Алгоритм обратной трассировки лучей

В этом алгоритме за счет скорости работы достигается излишняя для данной сцены универсальность. Обратная трассировка позволяет работать с

несколькими источниками света, передавать множество разных оптических явлений [3].

Положительной стороной данного алгоритма является возможность использования в параллельных вычислительных системах (т.к. расчет отдельной точки выполняется независимо от других точек).

Серьезным недостатком этого алгоритма будет являться большое количество необходимых вычислений для синтеза изображения сцены. Алгоритм не подойдет для генерации динамических сцен [4].

1.2.2 Алгоритм, использующий Z буфер

Несомненным плюсом данного алгоритма может являться его простота, которая не мешает решению задачи удаления поверхностей и визуализации их пересечения. В этом алгоритме не тратится время на сортировку элементов сцены, что дает преимущество в скорости работы. Особенно полезным это может стать при большом количестве препятствий в сцене.

Так же данный алгоритм в полной мере позволяет использовать особенности поставленной задачи. При фиксированном угле поворота сцены можно обработать данным алгоритмом все статические объекты сцены, сохранить полученные изображение и z-буфер, а при каждом отображении индикаторного слоя накладывать его на них.

Так как размер синтезируемого изображения сравнительно мал, затраты по памяти, при хранении информации о каждом пикселе, в данном алгоритме незначительны для современных компьютеров.

1.2.3 Алгоритм Робертса

Серьезным недостатком является вычислительная трудоемкость алгоритма. В теории она растет как квадрат количества объектов. Поэтому при большом количестве препятствий в сцене, этот алгоритм будет показывать себя, как недостаточно быстрый. Можно использовать разные оптимизации для повышения эффективности, например сортировку по z.

Преимуществом данного алгоритма является точность вычислений. Она достигается за счет работы в объектном пространстве, в отличии от большинства других алгоритмов.

Некоторые из оптимизаций крайне сложны, что затрудняет реализацию этого алгоритма.

1.2.4 Вывод

Предпочтительнее использовать Z буфер для динамической сцены, т.к. важна скорость работы алгоритма. Остальные алгоритмы дадут изображение более высокого качества, но оно избыточно, при этом оно гораздо сложнее. Также его просто оптимизировать с учётом особенностей поставленной задачи.

1.3 Анализ методов закрашивания

Простая закраска

Вся грань закрашивается одним уровнем интенсивности, который высчитывается по закону Ламберта.

Этот метод крайне прост в реализации и совершенно не требователен к ресурсам. Однако плохо подходит для тел вращения, плохо учитывает отраженный свет. Для данной задачи этот метод очень хорошо подходит, так как вся работа ведется с гранями параллелепипедов и прямоугольными плоскостями, тел вращения нет.

Закраска по Гуро

Основа закраски по Гуро – билинейная интерполяция интенсивностей, за счет которой устраняется дискретность изменения интенсивности и создается иллюзия гладкой криволинейной поверхности. Хорошо сочетается с диффузным отражением.

Закраска по Фонгу

Основа закраски по Фонгу – билинейная интерполяция векторов нормалей. Достигается лучшая локальная аппроксимация кривизны поверхности. Изображение выходит более реалистичным, зеркальные блики выглядят правдоподобнее, чем в методе закраски по Гуро.

Однако по сравнению с методом Гуро, закраска по Фонгу требует больших вычислительных затрат, так как интерполируются значения векторов нормалей, на основе которых потом вычисляется интенсивность.

Вывод: так как фигуры сцены состоят из плоскостей закраска по Фонгу и Гуро будет скорее мешать: ребра препятствий и антенны будут сглажены. Они будут хуже восприниматься. Поэтому лучше всего использовать простую закраску.

1.4 Описание трехмерных преобразований сцены

Сдвиг точки:

$$\begin{cases} X = x + dx \\ Y = y + dy \\ Z = z + dz \end{cases}$$

Масштабирование относительно начала координат:

$$\begin{cases} X = x \cdot k_x \\ Y = y \cdot k_y \\ Z = z \cdot k_z \end{cases}$$

Поворот относительно осей x, y, z на угол φ:

$$\begin{cases} X = x \\ Y = y \cdot \cos\phi + z \cdot \sin\phi \\ Z = -y \cdot \sin\phi + z \cdot \cos\phi \end{cases} \quad \begin{cases} X = x \cdot \cos\phi - z \cdot \sin\phi \\ Y = y \\ Z = x\sin\phi + z\cos\phi \end{cases} \quad \begin{cases} X = x \cdot \cos\phi + y \cdot \sin\phi \\ Y = -x \cdot \sin\phi + y \cdot \cos\phi \\ Z = z \end{cases}$$

1.5 Анализ моделей распространения Wi-Fi сигнала

В идеальных условиях (отсутствуют препятствия на пути прохождения сигнала, и нет многократных переотражений сигнала) потеря мощности сигнала в любой точке свободного пространства (free space - FS), выраженная в децибелах, равна:

$$L_{FS} = 20\log\left(\frac{4\pi d}{\lambda}\right) \qquad (1)$$

Где:

- d – расстояние между антеннами в метрах;
- λ – длина волны в метрах.

Статистическая модель One-slope описывает зависимость увеличения потери мощности сигнала с расстоянием, с усредненным учетом препятствий:

$$L(d) = L_{FS} + 10n\log(\frac{d}{d_0}) \qquad (2)$$

Где:

- d_0 = 1 м;
- L_{FS} – потери в свободном пространстве на расстоянии d_0;
- n – коэффициент, зависящий от типа помещения, количества препятствий и их материала.

Статистическая модель Log-distance аналогична модели One-slope, но добавляет к потере мощности сигнала случайную величину X_σ:

$$L(d) = L_{FS} + 10n\log\left(\frac{d}{d_0}\right) + X_\sigma \qquad (3)$$

Где:

- X_σ – нормальная случайная величина в дБ, имеющая стандартное отклонение.

Статистическая модель, рекомендованная Международным союзом электросвязи, ITU-R 1238 [6] была разработана для расчетов внутри зданий и помещений:

$$L(d) = 20\log(f) + 10N\log(d) + L_f(n) \qquad (4)$$

Где:

- d в метрах и d > 1;
- f в МГц;
- N – коэффициент потери мощности сигнала с расстоянием, табличная величина;
- n – количество препятствий между приемником и передатчиком;
- $P_f(n)$ – параметр потери мощности сигнала при прохождении через препятствия, табличная формула.

Для расчёта потерь сигнала на расстоянии больше метра следует использовать модель ITU-R 1238. Из рассмотренных моделей она наиболее точно может подстраиваться под конкретные типы препятствий из-за изменяемых коэффициентов. При расчёте потерь сигнала на расстоянии меньше метра следует считать, что сигнал распространяется в свободном пространстве, т.к. модели с учётом препятствий применимы только для больших расстояний.

1.6 Выводы из аналитического раздела

В данном разделе были рассмотрены алгоритмы удаления невидимых линий и поверхностей, методы закрашивания поверхностей, алгоритмы построения теней и моделирования осадков. В качестве алгоритма удаления невидимых линий был выбран Zбуфер, методом закраски – простой, построение теней будет выполняться с помощью теневых карт, построенных алгоритмом Zбуфера, для расчёта потерь сигнал были выбраны модели ITU-R 1238 и FS.

2. Конструкторская часть

В данном разделе будут рассмотрены требования к программе и алгоритмы визуализации сцены.

2.1 Требования к программе

Программа должна предоставлять следующие возможности:

- визуальное отображение сцены;
- вращение сцены;
- добавление нового объекта в сцену;
- удаление объекта из сцены.

2.2 Общий алгоритм визуализации трехмерной сцены

Визуализации трехмерной сцены происходит поэтапно. Рассмотрим алгоритм визуализации.

1. Задать объекты сцены
2. Задать источник света и положение наблюдателя
3. Для каждого полигона высчитать нормаль и интенсивность цвета, найти внутренние пиксели
4. Используя алгоритм Z буфера получить изображение сцены, и буфер глубины
5. Используя алгоритм Z буфера получить буфер глубины для источника света
6. Найти затененные участки при помощи наложения двух буферов
7. Отобразить изображение.

2.3 Алгоритм Z-буфера

1. Всем элементам буфера кадра присвоить фоновое значение
2. Инициализировать Z буфер минимальными значениями глубины
3. Выполнить растровую развертку каждого полигона сцены:
    a. Для каждого пикселя, связанного с полигоном вычислить его глубину z(x, y)

b. Получить цвет точки полигона с координатами (x, y, z) с учётом освещения

c. Сравнить глубину пикселя со значением, хранимым в Z буфере.

Если z(x, y) > $z_{буф}$(x, y), то $z_{буф}$(x,y)=z(x,y), цвет(x, y) = цветТочки.

4. Отобразить результат

2.4 Простой метод освещения

В простом методе освещения интенсивность рассчитывается по закону Ламберта:

$$I = I_0 * cos(\alpha) \qquad (5)$$

Где:

- I – результирующая интенсивность света в точке
- I_0 – интенсивность источника
- α – угол между нормалью к поверхности и вектором направления света

2.5 Получение цвета точки полигона

1. Если полигон закрашивается одним цветом:

   a. Вернуть базовый цвет полигона

   Иначе:

   a. Провести над точкой преобразования, обратные проведённым над полигоном

   b. Вернуть цвет из карты цветов полигона по полученным координатам

2.6 Выбор используемых типов и структур данных

Для разрабатываемого ПО нужно будет реализовать следующие типы и структуры данных.

- Источник света – задается направленностью света;
- Сцена – задается объектами сцены;
- Объект сцены – задаются вершинами и гранями;

- Полигон – прямоугольник, задающийся вершинами и картой цветов или базовым цветом
- Математические абстракции:
    - Точка – хранит координаты x, y, z
    - Вектор – хранит направление по x, y, z
- Wi-Fi модель – рассчитывает потери сигнала во всех точках обрабатываемого пространства;
- Симуляция – владеет сценой и Wi-Fi моделью, координирует физическую и графическую части программы, предоставляет весь функционал программы;
- Интерфейс – обеспечивает передачу данных от GUI к симуляции.

3. Технологическая часть

3.1 Выбор и обоснование языка программирования и среды разработки

В качестве языка программирования был выбран C# т.к. данный язык программирования объектно-ориентирован, что даст в полной мере реализовать следующие задачи:

- использовать наследование, абстрактные классы и т.д.;
- представлять трехмерные объекты сцены в виде объектов классов, что позволит легко организовать взаимодействие между ними, положительно влияя на читабельность кода.

В качестве среды разработки была выбрана «Visual Studio 2019» по следующим причинам:

- она бесплатна в пользовании студентами;
- она имеет множество удобств, которые облегчают процесс написания и отладки кода;
- она обеспечивает работу с Windows Forms – интерфейсом, который упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обертки для существующего Win32 API в управляемом коде.

3.2 Структура и состав классов

В этом разделе будут рассмотрена структура и состав реализованных классов, см. рис. 3.1, рис. 3.2, рис 3.3, рис 3.4.

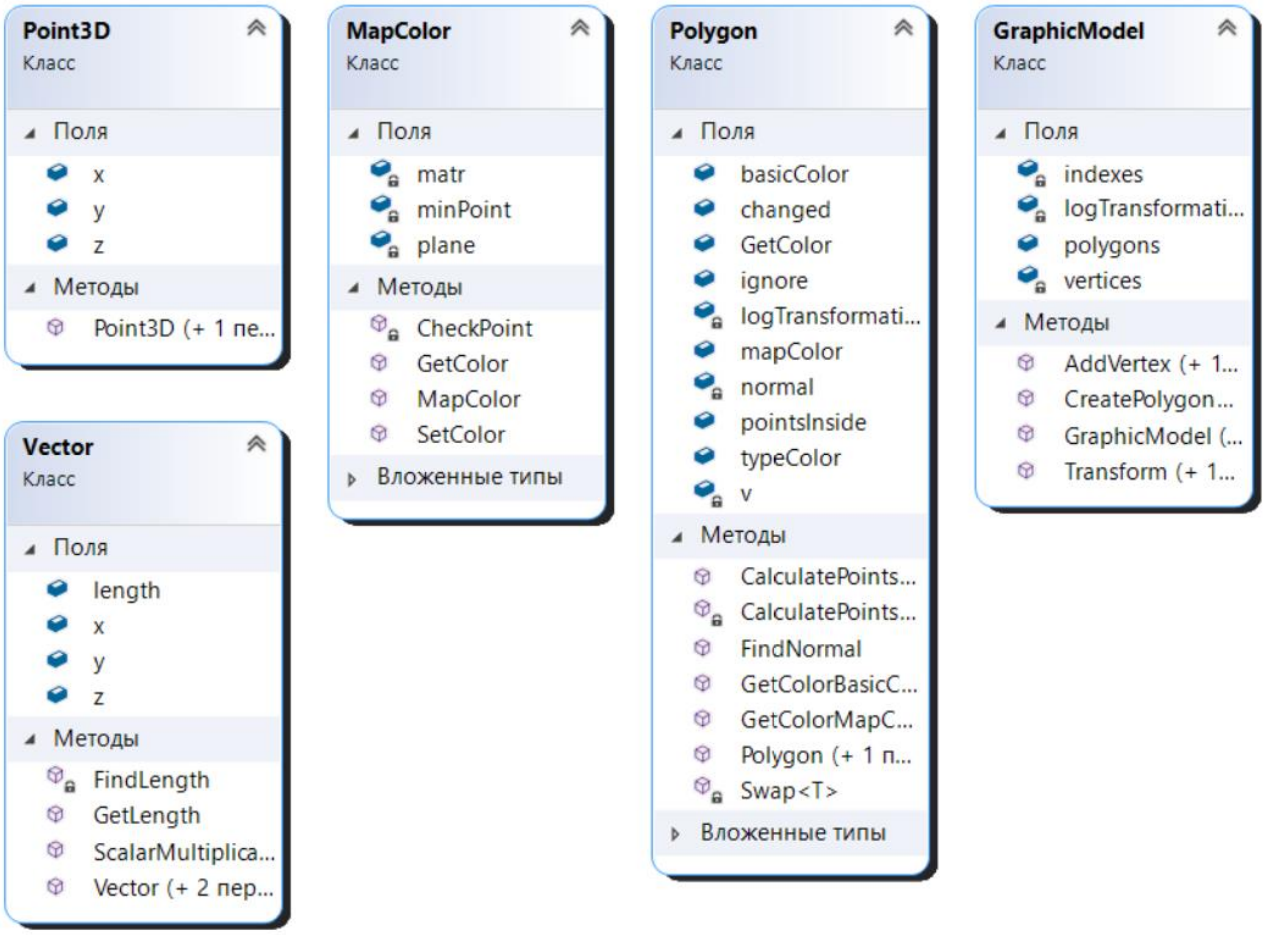

Рис. 3.1 структура классов Point3D, Vector, MapColor, Polygon, GraphicModel

- Point3D – класс точки, задаётся координатами x, y, z. Нужен для того, чтобы сделать их ссылочным типом данных, чтобы несколько объектов могли владеть одной точкой;
- Vector – класс вектора, задаётся координатами x, y, z;
- MapColor – класс карты цветов. Хранит карту цветов для прямоугольника, параллельного одной из координатных плоскостей, и информацию, необходимую для доступа к цвету конкретной точки этого прямоугольника;
- Polygon – класс прямоугольного полигона, хранит цвет (базовый для всего полигона или карту цветов), вершины, нормаль, точки внутри, лог трансформаций. Имеет методы вычисления нормали, получения точек внутри полигона, получения цветов этих точек;

- GraphicModel – хранит информацию о модели: ее вершины, индексы вершин, принадлежащих каждому полигону, полигоны, лог трансформаций, общий для всех полигонов. Имеет методы для трансформации модели.

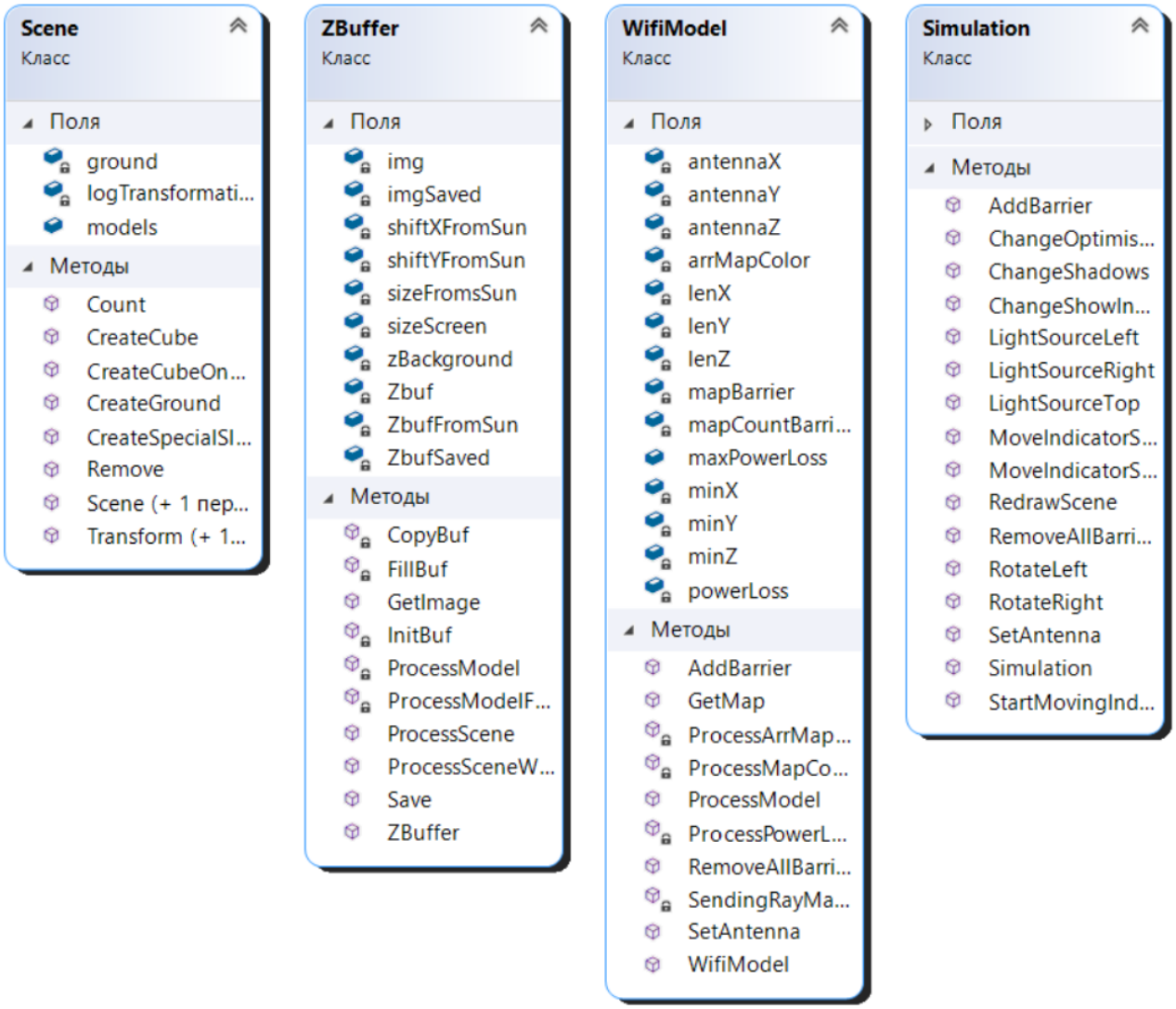

Рис. 3.2 структура классов Scene, ZBuffer, WifiModel, Simulation

- Scene – хранит информацию о сцене: размеры, модели внутри сцены. Имеет методы создания объектов сцены, её трансформации;
- ZBuffer – хранит заранее выделенные буфер кадра и z-буфер. Имеет методы создания изображения сцены;
- WifiModel – хранит данные физической модели: размеры обрабатываемого пространства, карту препятствий, координаты антенны,

массив матриц потерь мощности сигнала (по высотам). Имеет методы нахождения потерь мощности сигнала, получения карт цветов по высоте, добавления и удаления препятствий, изменения положения антенны;

- Simulation – хранит объекты Scene, ZBuffer и WifiModel. Обеспечивает обмен данными между ними. Имеет методы, соответствующие функционалу программы.

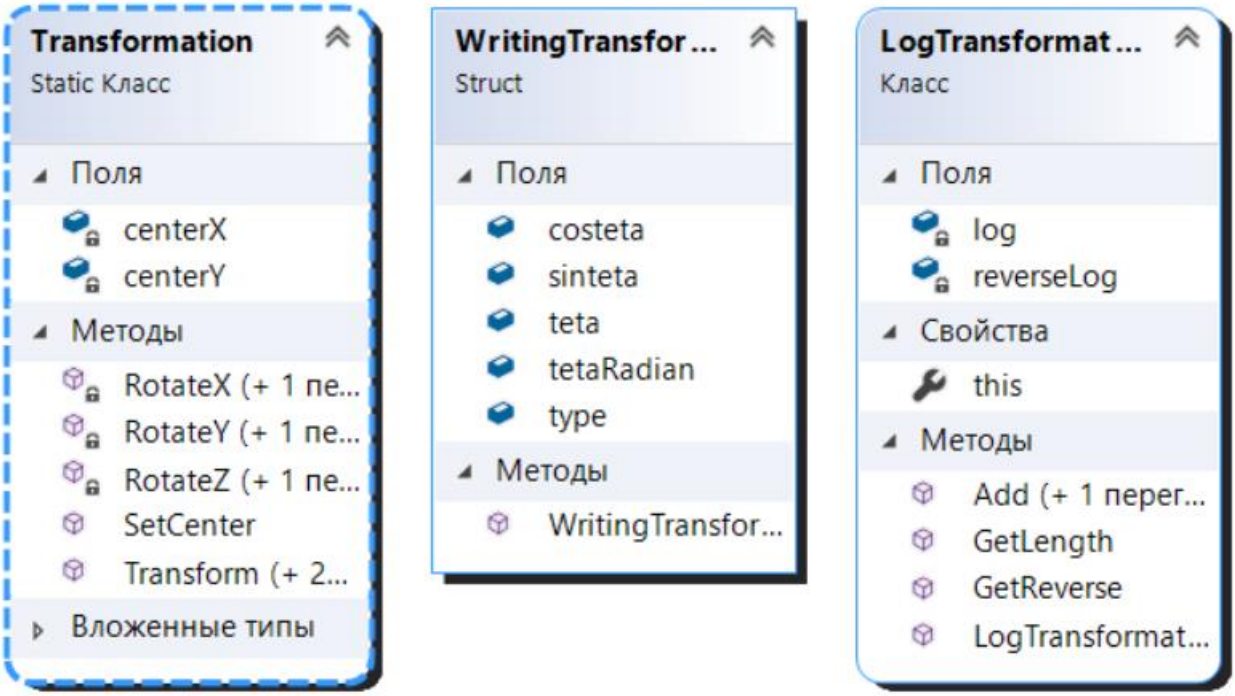

Рис. 3.3 структура классов Transformation, LogTransformation, структуры WritingTransformation

- Transformation – статический класс. Хранит координаты центра вращения. Имеет методы вращения точки вокруг одной из осей и серии вращений;
- WritingTransformation – структура, хранящая информацию о трансформаций: тип вращения, угол вращения, его косинус и синус;
- LogTransformation – класс хранящий лог трансформаций. Имеет методы добавления новой записи о трансформации, получения лога обратных трансформаций.

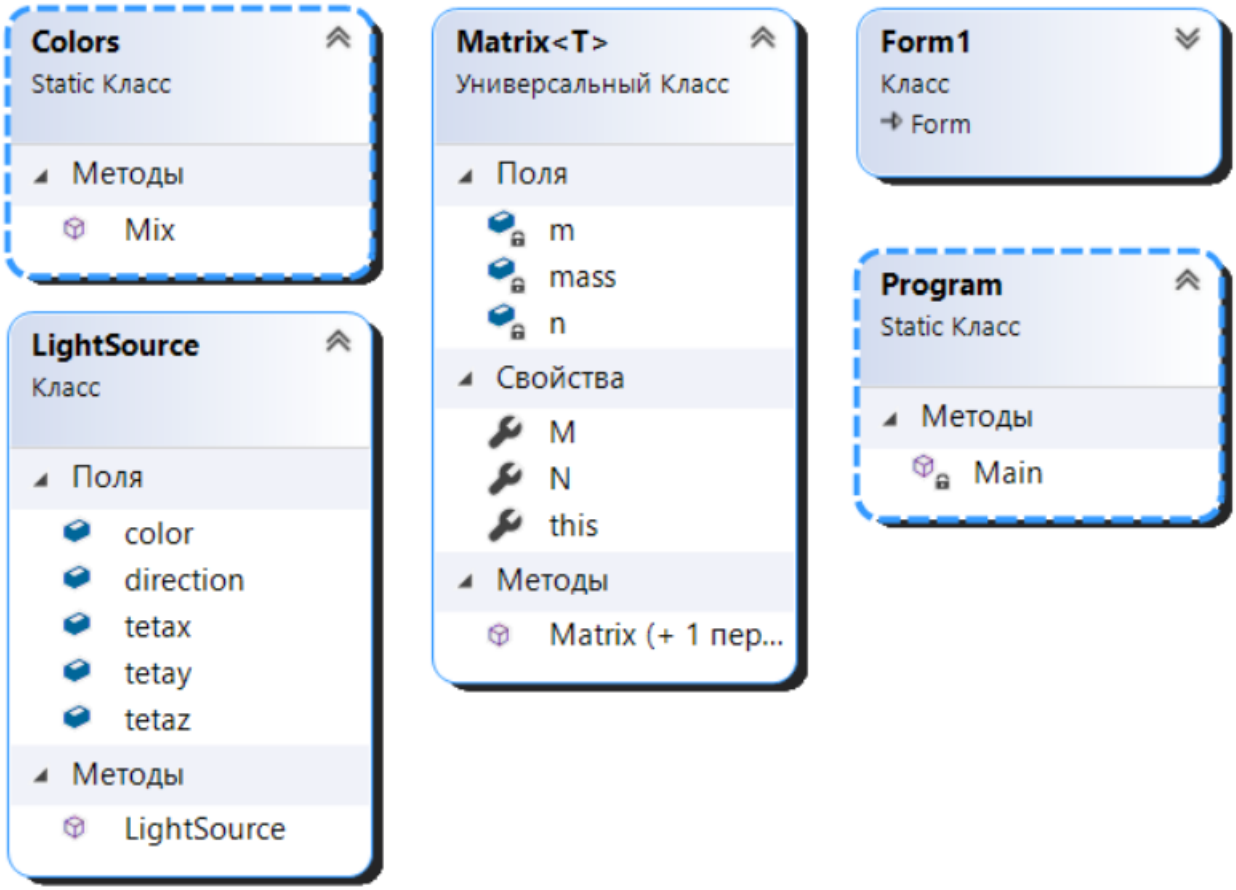

Рис. 3.4 структура классов Colors, LightSource, Matrix<T>, Form1, Program

- Colors – статический класс имеющий метод смешивания цветов;
- LightSource – класс хранящий в себе цвет света и направление света;
- Matrix<T> – универсальный класс, организующий матрицу, хранящую тип T;
- Form1 – класс пользовательского интерфейса;
- Program – точка входа в программу.

3.3 Сведения о модулях программы

- Program.cs – точка входа в приложение;
- Form1.cs – интерфейс;
- Light.cs – описание источников света;
- Scene.cs – описание сцены, методы взаимодействия с ней;
- GraphicModel.cs – описание объектов сцены, методы взаимодействия с ними и их частями;

- Zbuffer.cs – алгоритм Z буфера;
- Colors.cs – взаимодействие цветов;
- Transformation.cs – функции преобразования координат;
- WifiModel.cs – физическая модель потерь мощности сигнала;
- Simulation.cs – координация взаимодействия всех объектов программы по сигналу от интерфейса;
- Matrix.cs – реализация матрицы.

4. Экспериментальная часть

В данном разделе будет проведено тестирование реализованной программы для проверки корректности ее работы и поставлен эксперимент по оценки эффективности работы программы.

4.1 Тестирование

На рисунках 4.1 и 4.2 показана одна и та же сцена при разном положении наблюдателя.

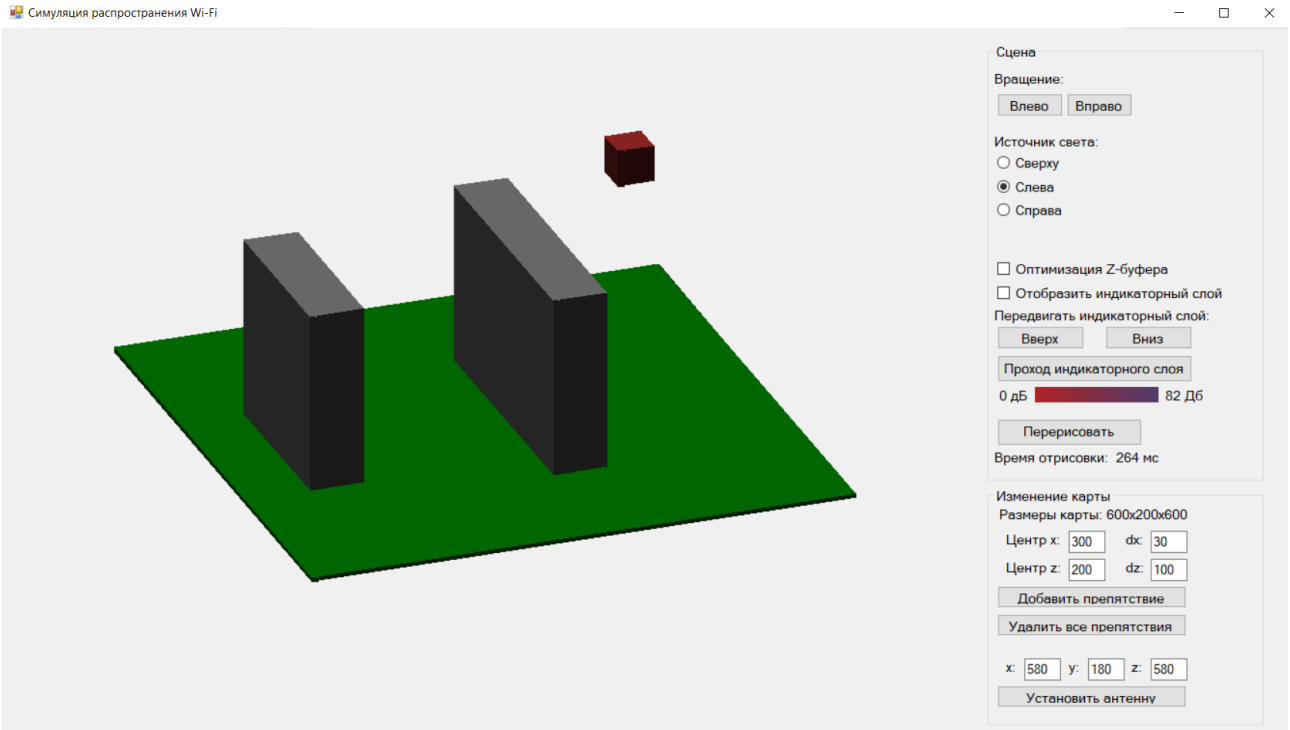

Рис. 4.1 Первое положение наблюдателя

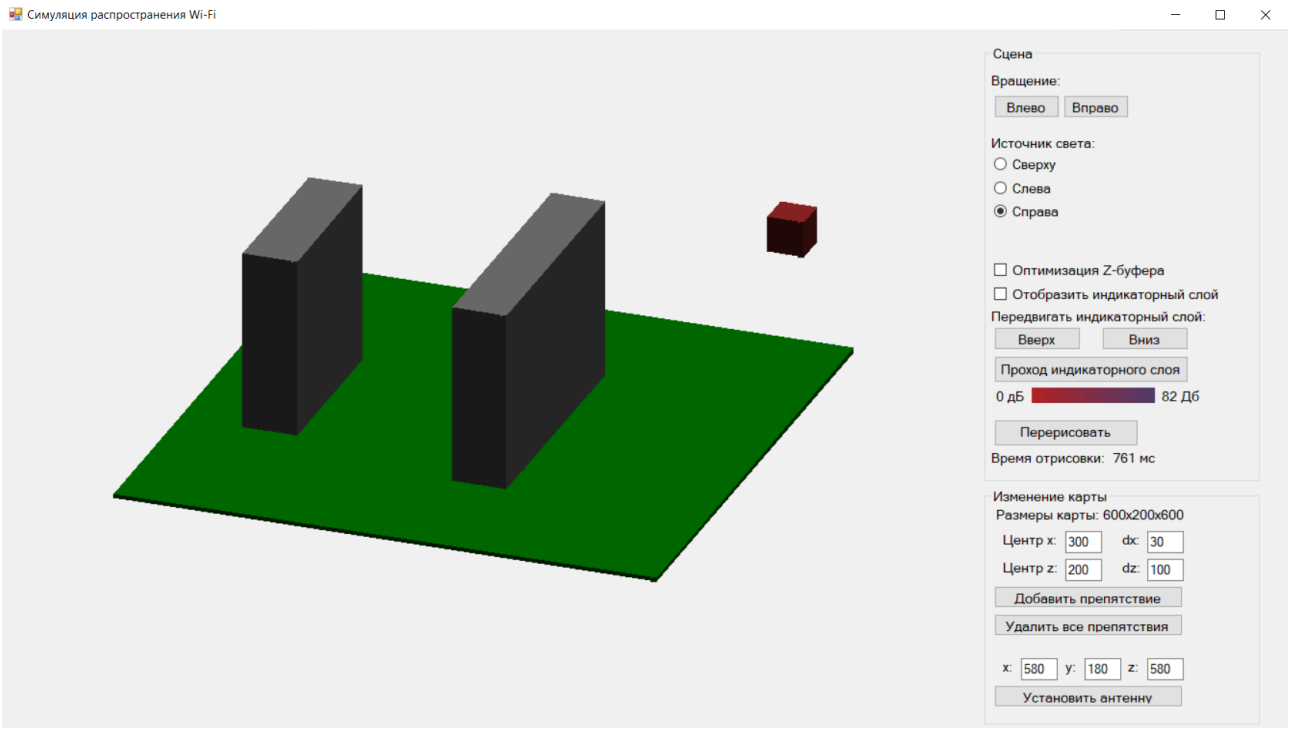

Рис. 4.2 Второе положение наблюдателя

На рисунках 4.3 и 4.4 показана одна и та же сцена при разном положении источника света. Смена положения источника освещения работает корректно. При положении источника справа, грани препятствий становятся невидимыми из источника света, закрашиваются одним цветом и сливаются.

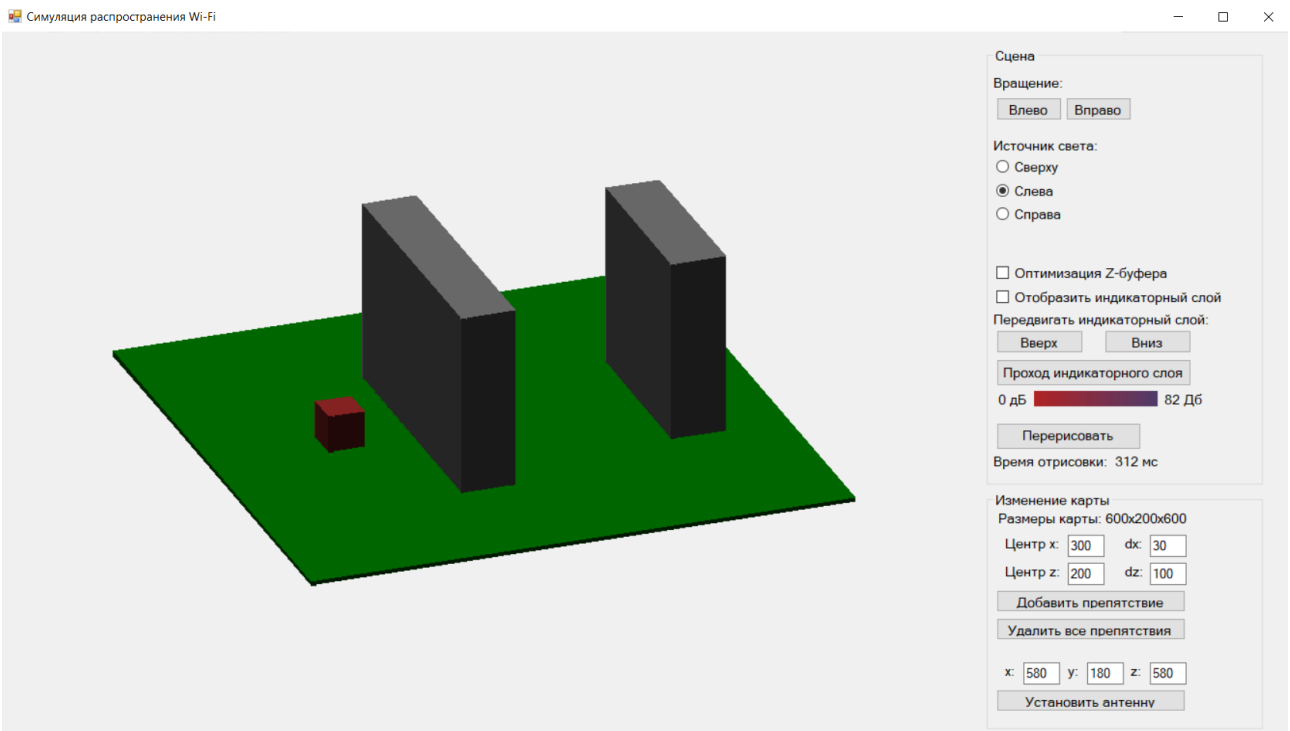

Рис. 4.3 Положение источника света слева

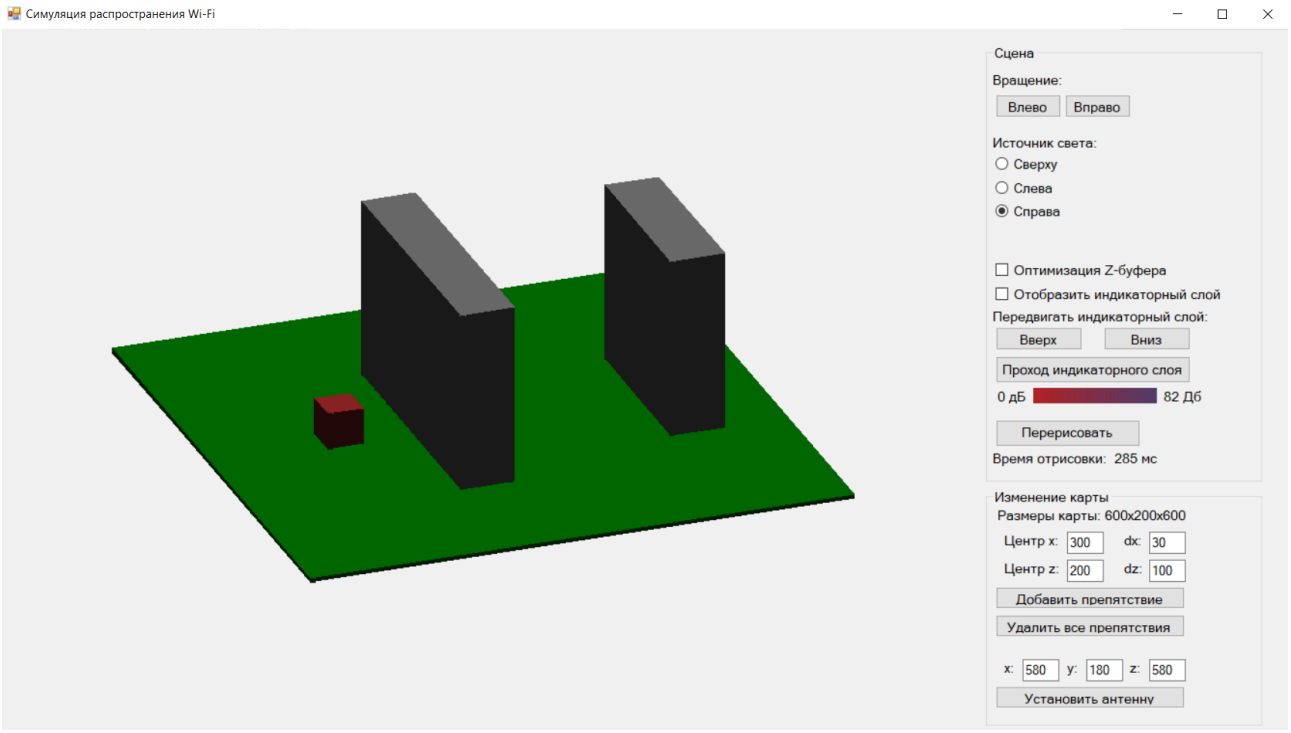

Рис. 4.4 Положение источника света справа

На рисунке 4.5 показан индикаторный слой. Из рисунка видно, что наложение сложных текстур происходит корректно.

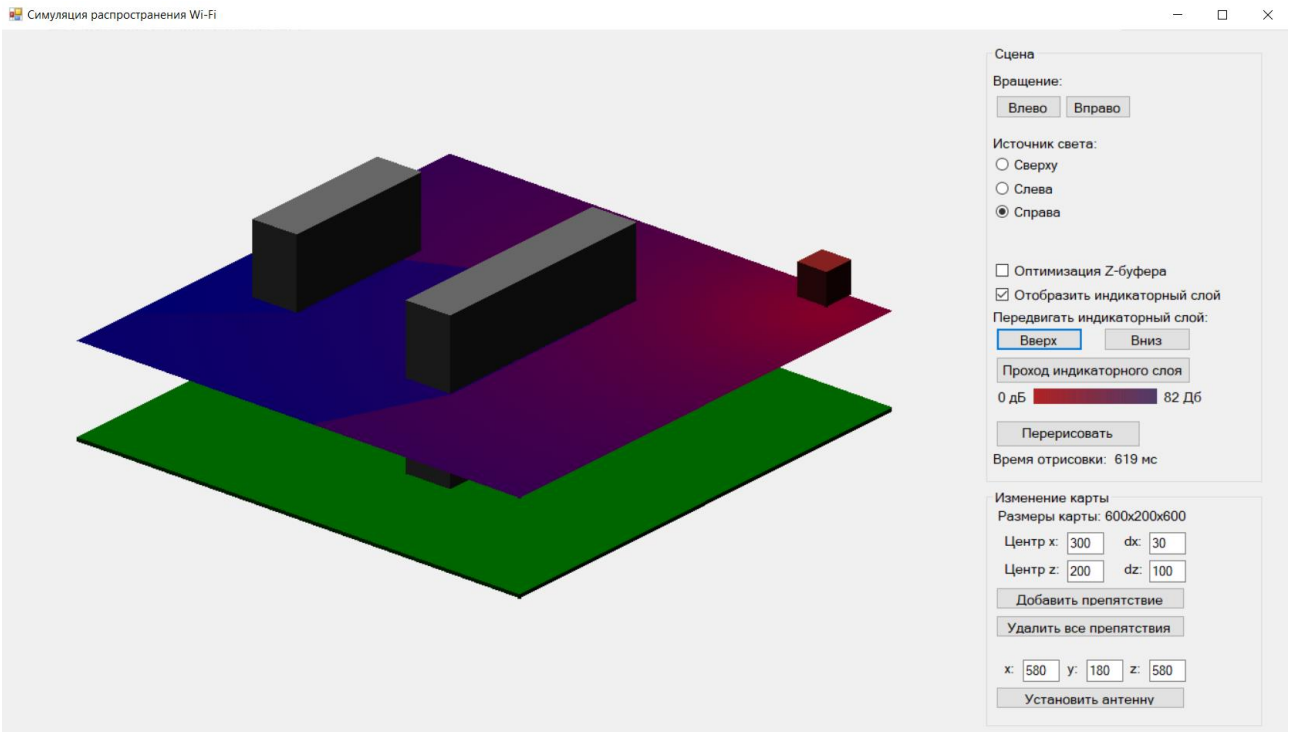

Рис. 4.5 Индикаторный слой

На рисунке 4.6 показано удаление всех препятствий.

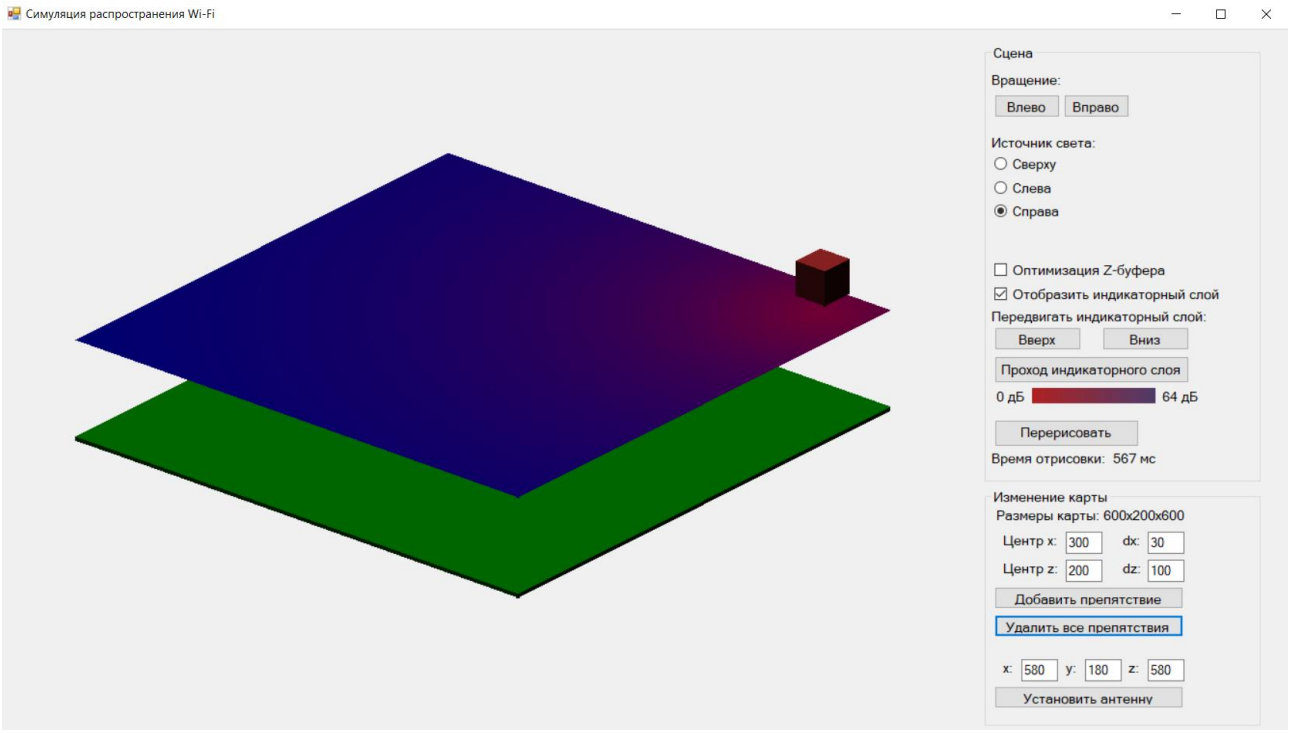

Рис. 4.6 Удаление всех препятствий

На рисунке 4.7 показано добавление препятствия.

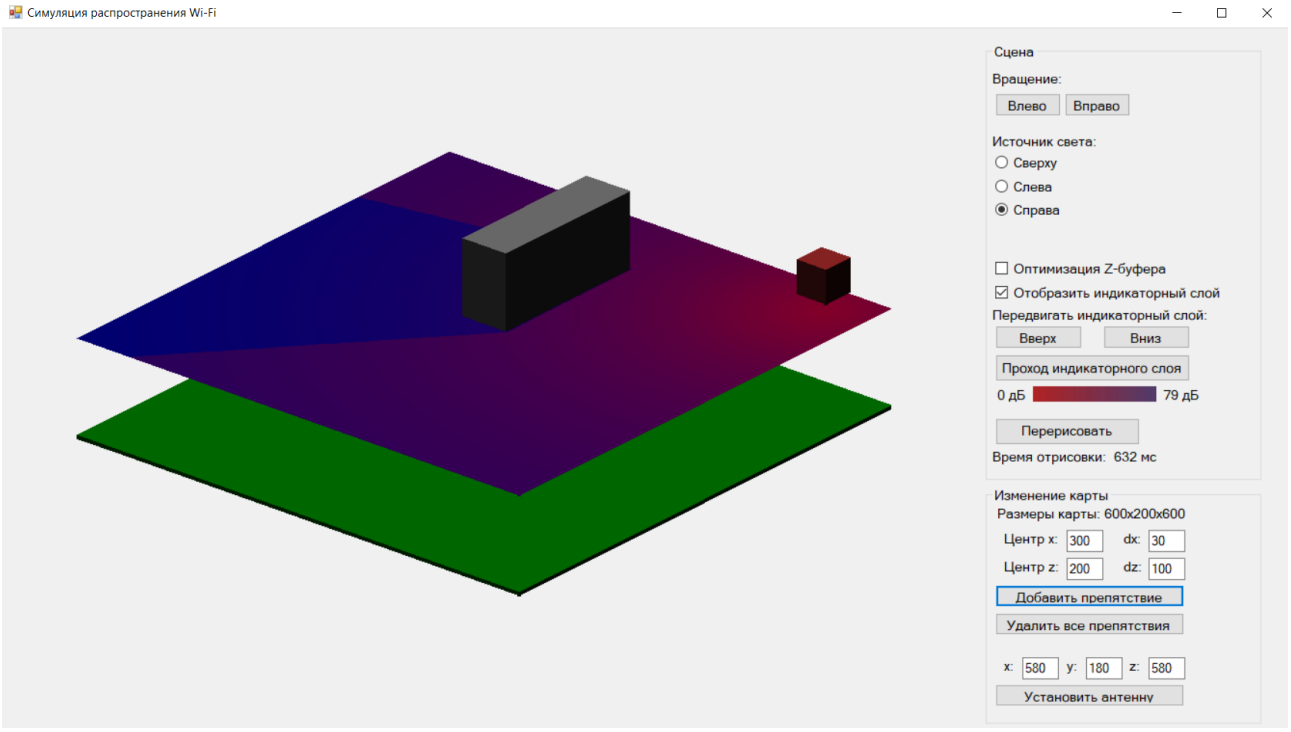

Рис. 4.6 Добавление препятствия

На рисунке 4.8 показано изменение положения антенны.

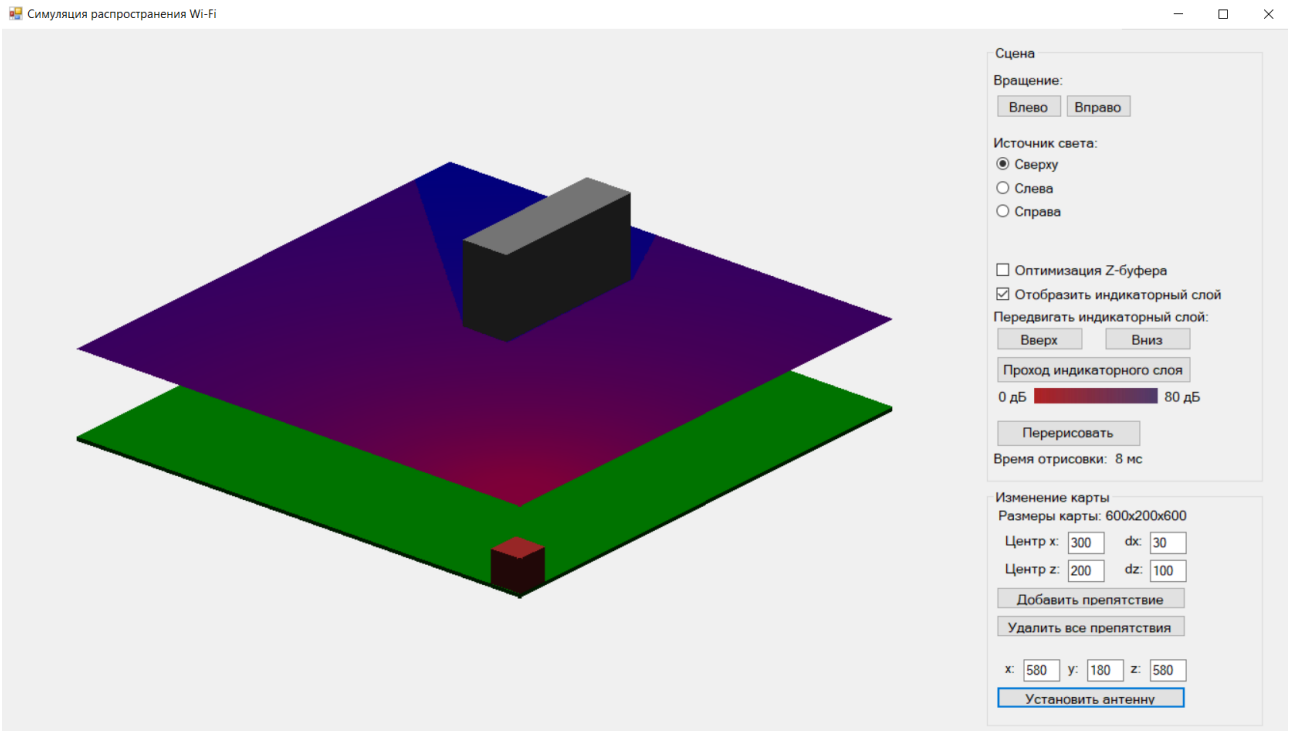

Рис. 4.6 Изменение положения антенны

4.3 Описание эксперимента

Был реализован оптимизированный с учётом особенностей задачи алгоритм Z-буфера. В нём предварительно формируется буфер кадра и z-буфер для статических объектов, а динамические объекты накладываются на них.

Эксперимент проводился на компьютере со следующими характеристиками:

- Операционная система Windows 10 64-bit;
- Процессор AMD Ryzen 5 3500U CPU @ 2.1GHz;
- 8 ядер;
- 20 Гб оперативной памяти.

В данном эксперименте замерялось время отображения сцены обычным и оптимизированным алгоритмами при разном количестве статических объектов сцены:

- 554 мс – обычный алгоритм, 1 препятствие;
- 248 мс – оптимизированный алгоритм, 1 препятствие;

- 618 мс – обычный алгоритм, 2 препятствия;
- 239 мс – оптимизированный алгоритм, 2 препятствия;
- 646 мс – обычный алгоритм, 3 препятствия;
- 242 мс – оптимизированный алгоритм, 3 препятствия.

Как и ожидалось, эксперимент показал уменьшение времени отображения для оптимизированного алгоритма. Время его работы не зависит от количества статических объектов.

Заключение

Во время выполнения курсового проекты была описана структура трехмерной сцены, были рассмотрены основные алгоритмы удаления невидимых линий, методы закрашивания, модели распространения радиосигнала. Были проанализированы их достоинства и недостатки, выбраны и реализованы наиболее подходящие для решения поставленной задачи. Было разработано программное обеспечение для визуализации распространения Wi-Fi сигнала.

Программа реализована таким образом, что пользователь может удалять препятствия, добавлять новые, изменять положение антенны, источника света.

В ходе выполнения поставленной задачи были изучены возможности Windows Forms, получены знания в области компьютерной графики.

Список использованной литературы

1. Методы представления дискретных данных [Электронный ресурс] – URL: https://www.graphicon.ru/oldgr/ru/library/multires_rep/index.html (дата обращения 15.12.20)
2. Bruce Baumgart, Winged-Edge Polyhedron Representation for Computer Vision. National Computer Conference, May 1975
3. Е. А. Снижко. Компьютерная геометрия и графика [Текст], 2005. - 17 с.
4. Проблемы трассировки лучей – из будущего в реальное время. [Электронный ресурс]. – URL: https://nvworld.ru/articles/ray-tracing/3/ (дата обращения 16.12.20)
5. D. F. Rogers. Procedural Elements for Computer Graphics. 2nd ed., 1998 – p.457-517
6. Данные о распространении радиоволн и методы прогнозирования для планирования систем радиосвязи внутри помещений и локальных зоновых радиосетей в диапазоне частот 300 МГц – 450 ГГц [Электронный ресурс]. – URL: https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.1238-10-201908-I!!PDF-R.pdf (дата обращения 20.12.20)
7. Модели распространения радиосигнала WI-FI [Электронный ресурс]. – URL: http://conf.nsc.ru/files/conferences/MIT-2013/fulltext/146127/151267/Startsev.pdf (дата обращения 20.12.20)