*Федеральное государственное бюджетное образовательное учреждение*

*высшего профессионального образования*

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ\_\_\_\_\_\_\_\_Информатика и cистемы управления\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_Программное обеспечение ЭВМ и информационные технологии

**РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовому проекту на тему:**

Моделирование реалистичного изображения

влияния никотина на легкие человека

Студент  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Зиновьев Д.В.

(Подпись, дата) (И.О.Фамилия)

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_Деон А.Ф.

(Подпись, дата) (И.О.Фамилия)

Москва, 2016

Оглавление

[1.1. Введение 4](#_Toc469272368)

[1.2. Техническое задание 4](#_Toc469272369)

[2. Аналитический раздел 5](#_Toc469272370)

[2.1. Анализ предметной области 5](#_Toc469272371)

[2.2. Обзор существующих программных систем и обоснование необходимости разработки 5](#_Toc469272372)

[3. Конструкторский раздел 7](#_Toc469272373)

[3.1. Построение модели легких человека 7](#_Toc469272374)

[3.2. Растровая развертка многоугольников 7](#_Toc469272375)

[3.3. Удаление невидимых линий и поверхностей 8](#_Toc469272376)

[3.3.1. Алгоритм Робертса 8](#_Toc469272377)

[3.3.2. Алгоритм Варнока 8](#_Toc469272378)

[3.3.3. Алгоритм, использующий z-буфер 9](#_Toc469272379)

[3.3.4. Выбор метода 11](#_Toc469272380)

[3.4. Закрашивание поверхностей 11](#_Toc469272381)

[3.4.1. Простой метод 11](#_Toc469272382)

[3.4.2. Метод Гуро 11](#_Toc469272383)

[3.4.3. Метод Фонга 13](#_Toc469272384)

[3.4.4. Выбор метода 13](#_Toc469272385)

[3.5. Модели освещения 13](#_Toc469272386)

[3.5.1. Модель Ламберта 13](#_Toc469272387)

[3.6.2. Модель Фонга 14](#_Toc469272388)

[3.6.3. Фотонная модель 15](#_Toc469272389)

[3.6.4. Выбор модели 15](#_Toc469272390)

[3.7. Преобразования объектов 15](#_Toc469272391)

[3.8. Проецирование трехмерных объектов на плоскость 16](#_Toc469272392)

[3.8.1. Центральное и параллельное проецирование 16](#_Toc469272393)

[3.8.2. Аксонометрическая проекция 17](#_Toc469272394)

[3.8.3. Выбор способа проецирования 18](#_Toc469272395)

[4. Технологический раздел 19](#_Toc469272396)

[4.1. Выбор технологии программирования 19](#_Toc469272397)

[4.2. Выбор языка и среды разработки 19](#_Toc469272398)

[4.3. Использованные классы. 19](#_Toc469272399)

[4.4. Интерфейс программного продукта 22](#_Toc469272400)

[4.6. Преобразования объектов сцены 23](#_Toc469272401)

[4.7. Обработка исключительных ситуаций 24](#_Toc469272402)

[4.8. Исследование временных характеристик 24](#_Toc469272403)

[5. Заключение 26](#_Toc469272404)

[6. Литература 26](#_Toc469272405)

## 

## 1.1. Введение

Трехмерная графика уже давно прочно заняла свою нишу в современном мире. 3D технологии графики и технологии 3D печати проникли во многие сферы человеческой деятельности, и приносят колоссальную прибыль.

Сегодня на основе трехмерной графики можно создать высокоточную копию реального объекта, создать нечто новое, воплотить в жизнь самые нереальные дизайнерские задумки.

За короткое время своего существования 3D графика нашла свое применение в самых разных областях: кино и видео-индустрия (эпичные взрывы автомобилей, разрушения небоскребов, космические баталии и другие спецэффекты), реклама и дизайн, проектирование и разработка компьютерных игр, мультипликация, медицина и многие другие сферы. Удачные 3D проекты очень трудно отличить от фотографий. Кроме того существует множество художников трехмерной графики, которые создают замечательные 3D картины.

## 1.2. Техническое задание

Задачей разрабатываемого программного продукта является создание реалистичной модели легких человека и визуализация влияния никотина на легкие курильщика. Для достижения поставленной цели используется моделирование источников освещения.

# 2. Аналитический раздел

## 2.1. Анализ предметной области

Технологии трехмерного моделирования активно вошли и в область медицинских исследований – одним из самых ярких достижений является разработка точной трехмерной модели человеческого тела. Такой уровень детализации до появления 3D графики сложно было даже представить.

Одним из направлений использования 3D графики в медицине является создание трехмерных моделей органов и суставов. Трехмерное моделирование позволяет также создавать видеоролики на медицинскую тематику.

Трехмерную графику можно активно использовать в обучении студентов медиков.

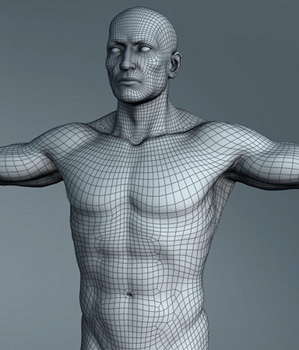


Рис. 2.1.1. Трехмерная модель тела человека

Трехмерная графика нашла применение еще в одной области медицины. В ведущих медицинских клиниках мира в последние годы появилось правило демонстрировать принцип проведения операций. Современные технологии 3D графики и анимации способны наглядно и условно (без изображения крови и других моментов, вызывающих у большинства людей неприязнь) показать пациенту процесс лечения. Подобные видеоролики используют для обучения стажеров и студентов медицинских вузов.

Преподаватели медицинских учебных заведений могут использовать трехмерную графику в наглядных методах обучения для демонстрации приборов, технических установок, а также всевозможных физиологических процессов, таких как движение крови по сосудам, формирование холестериновых бляшек, размножение [вирусов](http://pandia.ru/text/category/virus/), процесс рождения ребенка и т. д. Все это позволяет создавать красочные, интересные и наиболее запоминающиеся занятия. Таким образом, обучающиеся могут в динамике увидеть многие процессы, которые раньше могли усваивать по статическим изображениям в учебниках. Очевидно, для осуществления подобных методов обучения необходимо наличие технических средств обучения – персональных компьютеров, мультимедийных проекторов.

## 2.2. Обзор существующих программных систем и обоснование необходимости разработки

В настоящее время существует множество программных средств, позволяющих проводить моделирование трехмерных объектов высокой сложности. Одними из самых популярных программ являются AutoCAD, КОМПАС, Corel Draw. В большинстве случаев такие программы являются универсальными, позволяющими пользователю обрабатывать объекты, относящиеся к разным предметным областям, ввиду чего такие средства, как правило, занимают большие объемы памяти и крайне сложны в освоении, особенно для пользователя, уровень владения вычислительной техникой которого низок.

Данный программный продукт разработан исключительно для применения в области медицины. Созданы индивидуальные, характерные только для данной области, предельно простые интерфейс и функциональные возможности, что позволяет неподготовленному пользователю быстро обучиться и приступить к работе с предлагаемым программным продуктом. И в этом бесспорное преимущество данного программного обеспечения.

# 3. Конструкторский раздел

## 3.1. Построение модели легких человека

В данной работе мы исходим из предположения, что данные поступают на основе показаний некоторого аппарата, с помощью которого проводится обследование пациента. Таким образом, на входе имеется файл с набором полигонов, представляющих модель легких человека.

Наибольшей простоты можно достичь в случае треугольных полигонов. Таким образом, будем аппроксимировать поверхность легких треугольными полигонами. Для отображения треугольников на экране необходимо выполнить растровую развертку этих треугольников, учитывая их взаимное расположение в пространстве относительно друг друга, то есть, удаляя невидимые линии и поверхности. Для этого можно использовать различные методы, которые будут подробнее рассмотрены далее. Закраска треугольника при растровой развертке выполняется в соответствии с выбранной моделью освещения.

## 3.2. Растровая развертка многоугольников

Существует несколько методов растровой развертки многоугольников. Самым простым является тест на принадлежность каждой точки поля внутренней области данного многоугольника.

Можно разработать более эффективный метод, если воспользоваться тем фактом, что соседние пикселы, вероятно, имеют одинаковые характеристики. Характеристики пикселов на данной строке изменяются только там, где ребро многоугольника пересекает строку. В условиях данной задачи можем воспользоваться тем фактом, что каждый многоугольник является треугольником. Таким образом, на каждой строке может быть не более двух точек пересечения сканирующей строки с ребром треугольника.

Отдельно необходимо обрабатывать точки пересечения сканирующей строки с горизонтальными ребрами треугольника и с его вершинами. В последнем случае точку пересечения необходимо учитывать два раза.

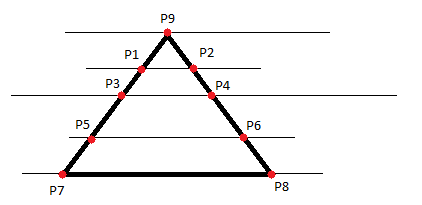


Рис. 3.3.1. Растровая развертка треугольника.

Таким образом, для каждой строки мы получаем ровно две точки – начало и конец отрезка, точки которого принадлежат внутренней области треугольника. Далее можем выполнить отрисовку отрезка, используя базовые алгоритмы растеризации отрезка (например, алгоритм ЦДА или алгоритм Брезенхема).

Существуют также и другие методы растровой развертки сплошных областей. Один из них – метод заполнения многоугольника по ребрам является простым в реализации, однако требует больших затрат памяти и времени. Другой алгоритм – алгоритм с упорядоченным списком ребер и флагом – неприменим в условиях данной задачи, так как использует графический способ определения принадлежности точки внутренней области многоугольника.

## 3.3. Удаление невидимых линий и поверхностей

### 3.3.1. Алгоритм Робертса

Этот алгоритм является первой разработкой такого рода и предназначен для удаления невидимых линий при штриховом изображении объектов, составленных из выпуклых многогранников. Он относится к алгоритмам, работающим в объектном пространстве, и очень элегантен с математической точки зрения.

Выпуклый многогранник однозначно определяется набором плоскостей, образующих его грани, поэтому исходными данными для алгоритма являются многогранники, заданные списком своих граней. Грани задаются в виде плоскостей, заданных в канонической форме.

Алгоритм, прежде всего, удаляет из каждого тела те ребра или грани, которые экранируются самим телом. Затем каждое из видимых ребер каждого тела сравнивается с каждым из оставшихся тел для определения того, какая его часть или части, если таковые есть, экранируются этими телами. Поэтому вычислительная трудоемкость алгоритма Робертса растет теоретически, как квадрат числа объектов. Однако математические методы, используемые в этом алгоритме, просты, мощны и точны. Более поздние реализации алгоритма, использующие предварительную приоритетную сортировку вдоль оси z и простые габаритные или минимаксные тесты, демонстрируют почти линейную зависимость от числа объектов.

Этапы работы алгоритма Робертса:

1. Определение нелицевых граней для каждого тела отдельно.

2. Определение и удаление ребер, экранируемых другими телами.

3. Определение видимости отрезков, образованных новыми ребрами при протыкании тел друг другом.

Преимущества алгоритма:

1. быстрое и простое построение изображений выпуклых фигур;

2. содержит в себе обработку пересечений объектов.

Недостатки алгоритма:

1. метод строго ориентирован только на выпуклые многогранники.

### 3.3.2. Алгоритм Варнока

Алгоритм Варнока является одним из примеров алгоритма, основанного на разбиении картинной плоскости на части, для каждой из которых исходная задача может быть решена достаточно просто. Алгоритм Варнока также работает в пространстве изображения.

В пространстве изображения рассматривается окно и решается вопрос о том, пусто ли оно, или его содержимое достаточно просто для визуализации. Если это не так, то окно разбивается на фрагменты до тех пор, пока содержимое фрагмента не станет достаточно простым для визуализации или его размер не достигнет требуемого предела разрешения. Используется то, что большие области изображения однородны. Такое свойство называют когерентностью, имея в виду, что смежные области (пиксели) вдоль обеих осей х и у имеют тенденцию к однородности.

При помощи изложенного алгоритма можно удалить либо невидимые линии, либо невидимые поверхности. Однако простота критерия разбиения, а также негибкость способа разбиения приводят к тому, что количество подразбиений оказывается велико.

Преимущества алгоритма:

1. использование свойства когерентности;

2. простота визуализации любого фрагмента.

Недостатки алгоритма:

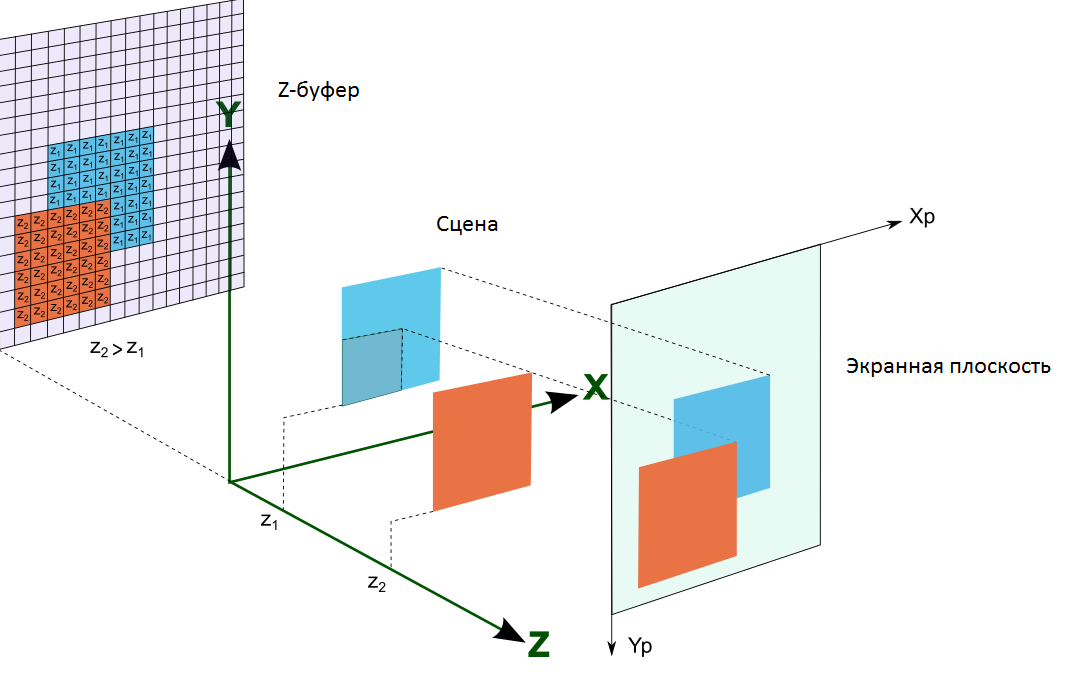
1. количество разбиений в случае сложного изображения (разбиения достигают размера в один пиксель) получается слишком большим, из-за чего увеличивается время работы алгоритма;

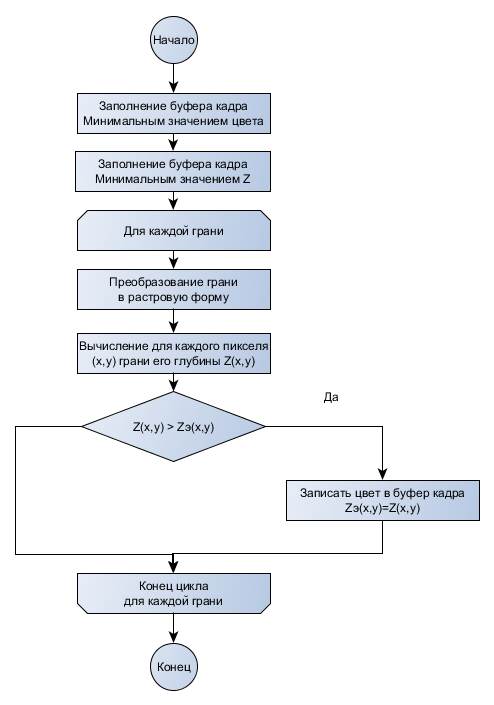
2. использование рекурсии.

### 3.3.3. Алгоритм, использующий z-буфер

Алгоритм, использующий z-буфер это один из простейших алгоритмов удаления невидимых поверхностей. Работает этот алгоритм в пространстве изображения.

Идея z-буфера является простым обобщением идеи о буфере кадра. Буфер кадра используется для запоминания атрибутов (интенсивности) каждого пиксела в пространстве изображения, z-буфер – это отдельный буфер глубины, используемый для запоминания координаты z или глубины каждого видимого пиксела в пространстве изображения. В процессе работы глубина или значение z каждого нового пиксела, который нужно занести в буфер кадра, сравнивается с глубиной того пиксела, который уже занесен в z-буфер. Если это сравнение показывает, что новый пиксел расположен впереди пиксела, находящегося в буфере кадра, то новый пиксел заносится в этот буфер и, кроме того, производится корректировка z-буфера новым значением z. Если же сравнение дает противоположный результат, то никаких действий не производится. По сути, алгоритм является поиском по х и у наибольшего значения функции z (х, у).



Рис. 3.4.1.1. Использование Z-буфера.

Преимущества алгоритма:

1. простота;

2. тривиальность визуализации пересечений сложных поверхностей;

3. произвольный уровень сложности сцены.

Недостатки алгоритма:

1. большой объем требуемой памяти;

2. трудоемкость и высокая стоимость устранения лестничного эффекта, а также реализации эффектов прозрачности и просвечивания.

### 3.3.4. Выбор метода

При решении поставленной задачи будет применен алгоритм, использующий Z-буфер, так как в общем случае на этапе разработки не известна форма объектов (легких) отображаемых на экране, однако известно, что объекты не обладают свойством прозрачности. При использовании современных ЭВМ затраты памяти не являются главной проблемой. В связи с этим, преимущество использования данного алгоритма в решаемой задаче вполне обоснованно.

## 3.4. Закрашивание поверхностей

### 3.4.1. Простой метод

Вычисляется один уровень интенсивности, который используется для закраски всего многоугольника, причем многоугольник представляет реальную моделируемую поверхность, а не является аппроксимацией криволинейной поверхности.

Преимущества алгоритма:

1. скорость работы позволяет выполнять его в реальном времени;

2. позволяет сформировать реалистичное с точки зрения освещения изображение (для плоских поверхностей, которые не являются аппроксимацией кривой).

Недостатки алгоритма:

1. эффект Маха (на ребрах разрыв интенсивности освещения);

2. некорректный расчет бликов;

3. при незначительных поворотах изображения интенсивность окраски может заметно изменяться.

### 3.4.2. Метод Гуро

Один из способов устранения дискретности интенсивностей закрашивания был предложен Гуро. Его метод заключается в том, что используются не нормали к плоским граням, а нормали к аппроксимируемой поверхности, построенные в вершинах многогранника. После этого вычисляются интенсивности в вершинах, а затем во всех внутренних точках многоугольника выполняется билинейная интерполяция интенсивности.

Метод сочетается с алгоритмом построчного сканирования. После того как грань отображена на плоскость изображения, для каждой сканирующей строки определяются ее точки пересечения с ребрами. В этих точках интенсивность вычисляется с помощью линейной интерполяции интенсивностей в вершинах ребра. Затем для всех внутренних точек многоугольника, лежащих на сканирующей строке, также вычисляется интенсивность методом линейной интерполяции двух полученных значений.

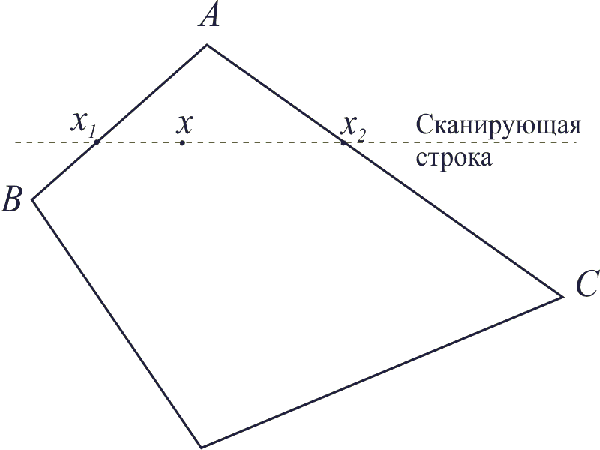


Рис. 3.5.2.1. Интерполяция интенсивности для сканирующей строки.

Пусть - интенсивности в вершинах , - горизонтальные координаты этих точек. Тогда в точках пересечения сканирующей строки с ребрами многоугольника интенсивности можно вычислить по формулам интерполяции:

После этого интенсивность в точке получаем путем интерполяции значений на концах отрезка:

Преимущества алгоритма:

1. позволяет создавать более реалистичное изображение (для поверхностей, которые являются аппроксимацией кривой);

2. вычисления почти в 3 раза проще, чем в методе Фонга.

Недостатки алгоритма:

1. он хорошо работает только с диффузной моделью отражения;

2. проблема построения нормалей к поверхности, так как она вычисляется путем усреднения нормалей к граням, примыкающим к этой вершине, что сильно зависит от характера разбиения;

3. блики не всегда соответствуют действительности.

### 

### 3.4.3. Метод Фонга

Фонг предложил вместо интерполяции интенсивностей произвести интерполяцию вектора нормали к поверхности на сканирующей строке. Этот метод требует больших вычислительных затрат, поскольку формулы интерполяции применяются к трем компонентам вектора нормали, но зато дает лучшую аппроксимацию кривизны поверхности. Поэтому зеркальные свойства поверхности воспроизводятся гораздо лучше.

Нормали к поверхности в вершинах многогранника вычисляются так же, как и в методе Гуро. А затем выполняется билинейная интерполяция в сочетании с построчным сканированием. После построения вектора нормали в очередной точке вычисляется интенсивность.

Преимущества алгоритма:

1. устраняет недостаток метода Гуро – зависимость освещённости грани от её положения относительно наблюдателя;

2. обеспечивает значительно лучшее качество изображения по сравнению с методом Гуро и простым методом.

Недостатки алгоритма:

1. данный метод в 3 раза длиннее метода Гуро.

### 3.4.4. Выбор метода

В условиях решаемой задачи невозможно проводить интерполяцию нормалей, так как на входе имеется лишь значение нормали, единое для каждой точки полигона. Таким образом, необходимо проводить интерполяцию интенсивности, в связи с чем выбран метод закраски Гуро.



Рис. 3.5.4.1. Сравнение методов закраски многоугольников.

## 3.5. Модели освещения

### 3.5.1. Модель Ламберта

Модель Ламберта моделирует идеальное диффузное освещение. Считается, что свет при попадании на поверхность рассеивается равномерно во все стороны. Таким образом, освещенность в точке определяется только плотностью света в точке поверхности, а она линейно зависит от косинуса угла падения.

Модель Ламберта является одной из самых простых моделей освещения. Данная модель очень часто используется в комбинации других моделей, практически в любой другой модели освещения можно выделить диффузную составляющую.

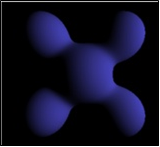


Рис. 3.6.1.1. Модель освещения Ламберта

Преимущества модели:

1. простая модель;

2. удобна для анализа свойств других моделей (за счет того, что ее легко выделить из любой модели и анализировать оставшиеся составляющие).

Недостатки модели:

1. моделирует только диффузное освещение.

### 3.6.2. Модель Фонга

Модель представляет собой комбинацию диффузной составляющей (модели Ламберта) и зеркальной составляющей и работает таким образом, что кроме равномерного освещения на материале может еще появляться блик, местонахождение которого определяется из закона равенства углов падения и отражения.

Расчёт освещения по Фонгу требует вычисления цветовой интенсивности трёх компонент освещения: фоновой, рассеянной и глянцевых бликов.

I=K_aI_a+K_d(\vec{n},\vec{l})+K_s(\vec{n},\vec{h})^p\,\!

\vec{n}\,\! — вектор нормали к поверхности в точке

\vec{l}\,\! — направление проецирования (направление на источник света)

\vec{h}\,\! — направление на наблюдателя

K_a\,\! — коэффициент фонового освещения

K_s\,\! — коэффициент зеркального освещения

 — коэффициент диффузного освещения



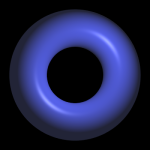


Рис. 3.6.2.1. Модель освещения Фонга

Преимущества модели:

1. дает довольно качественное изображение.

Недостатки модели:

1. многие оптические эффекты либо не учитываются, либо рассчитываются с сильным приближением.

3.6.3. Фотонная модель

В фотонной модели освещения источник представляет собой плоскость излучающую фотоны. По физическим законам определяется интенсивность и направление движения этих фотонов, которые определяют освещенность тех или иных граней.

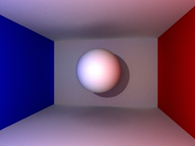


Рис. 3.6.3.1. Фотонная модель освещения

Преимущества модели:

1. реальное изображение, соответствующее физическим законам.

Недостатки модели:

1. высокая трудоемкость вычислений.

3.6.4. Выбор модели

В данной работе была выбрана модель освещения Ламберта, так как она довольно проста в реализации, требует меньших вычислительных затрат, по сравнению с двумя другими моделями, а также является наиболее подходящей в условиях решаемой задачи (поверхность легких можно считать абсолютно незеркальной) .

## 3.7. Преобразования объектов

В процессе работы программы может возникать необходимость преобразования объектов сцены – их сдвига, масштабирования, поворота.

Для трехмерного пространства любое аффинное преобразование может быть представлено последовательностью этих простейших операций.

Ниже приводятся уравнения и матрицы преобразований:

* сдвиг точки вдоль координатных осей на dx, dy, dz:

 ;

* масштабирование относительно начала координат с коэффициентами kx, ky, kz:

 ;

* поворот относительно осей x, y, z на угол :
  + ось x:

* ось y:

* ось z:

## 3.8. Проецирование трехмерных объектов на плоскость

### 3.8.1. Центральное и параллельное проецирование

При отображении пространственных объектов на экране необходимо знать координаты объектов. Рассмотрим две системы координат. Первая – мировые координаты, которые описывают истинное положение объектов в пространстве с заданной точностью. Другая – экранная система координат – система координат монитора, на экран которого осуществляется вывод изображения.

Пусть мировые координаты будут трехмерными декартовыми координатами. Для синтеза изображения на плоскости достаточно, двумерной экранной системы координат, однако для использования алгоритма Z-буфера, они должны быть трехмерными.

В компьютерной графике распространены параллельная и центральная (перспективная) проекции.

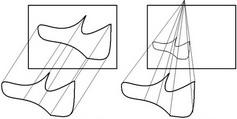


Рис. 3.8.1. Параллельное и центральное проецирование

Для центральной проекции лучи проецирования исходят из одной точки, размещенной на конечном расстоянии от объектов и плоскости проецирования. Для параллельной проекции лучи проецирования параллельны.

### 3.8.2. Аксонометрическая проекция

Аксонометрической называется проекция, при которой лучи проецирования располагаются под прямым углом к плоскости проецирования. Как правило, для аксонометрической проекции расположение плоскости проецирования задается с помощью двух углов – и.

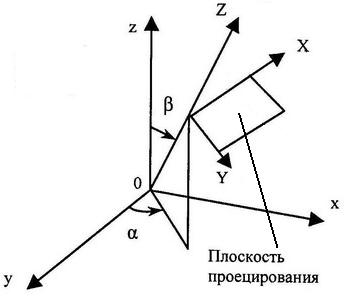


Рис. 3.8.2.1. Аксонометрическая проекция

Для того, чтобы найти соотношения между координатами (x, y, z) и (X, Y, Z) для любой точки в трехмерном пространстве, рассмотрим преобразование системы координат (x, y, z) в систему (X, Y, Z). Такое преобразование может быть задано двумя шагами:

1. поворот системы координат относительно оси z на угол . Такой поворот описывается матрицей

;

2. поворот новой системы координат (x, y, z) относительно оси x на угол  - получение координаты (X, Y, Z). Матрица поворота:

.

Преобразование координат выражается произведением матриц 



Это же преобразование можно записать в формульном виде:



При формировании изображения на экране монитора, как уже было сказано, необходимы дополнительные преобразования, поскольку система координат в плоскости проецирования может не совпадать с системой координат устройства отображения.

Пусть (Xэ, Yэ, Zэ) – это экранные координаты объектов в графическом устройстве отображения. Координаты проецирования обозначим как (X, Y, Z).

Преобразование координат проекции в экранные координаты можно представить как растяжение/сжатие и сдвиг:

.

Такое преобразование сохраняет пропорции объектов благодаря одинаковому коэффициенту растяжения/сжатия k для всех координат.

### 3.8.3. Выбор способа проецирования

В данной задаче удобно использовать аксонометрическую проекцию, так как она позволяет передать форму объекта без искажений (параллельность линий при проецировании сохраняется), что является важным при построении модели в данной задаче.

# 4. Технологический раздел

## 4.1. Выбор технологии программирования

При разработке программного продукта предполагается использовать объектно-ориентированный подход, так как он позволяет использовать в программах понятия, более близкие к предметной области, а также упрощает внесение изменений в код.

## 4.2. Выбор языка и среды разработки

В качестве языка программирования выбран C#, так как:

1. является современным объектно-ориентированным языком;

2. имеет встроенный сборщик мусора (избавляет программиста от необходимости жесткого контроля работы с памятью);

3. предоставляет преимущества в скорости разработки на начальном этапе;

4. имеет обширную коллекцию библиотечных классов;

5. поддерживает компиляцию «на лету».

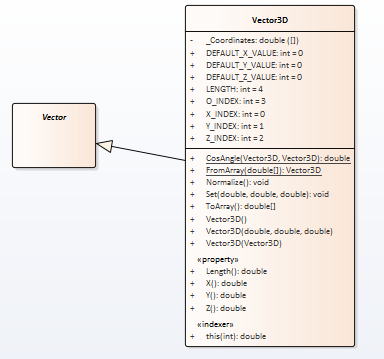
В качестве среды разработки выбрана Visual Studio, так как:

1. имеется бесплатная Express версия для студентов;

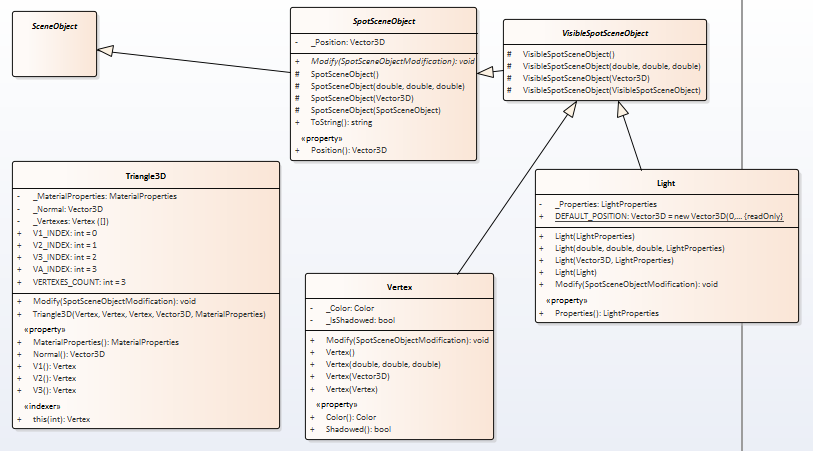
2. предоставляет множество мощных средств для быстрой и легкой отладки кода.

## 4.3. Использованные классы.

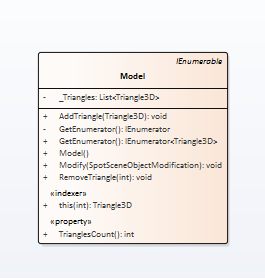
1. Класс Vector3D используется для представления координат точки в пространстве.



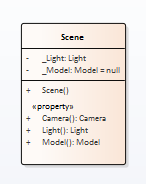
2. Класс Vertex используется для представления точки в пространстве. Triangle – для представления треугольника. Light – источник света.



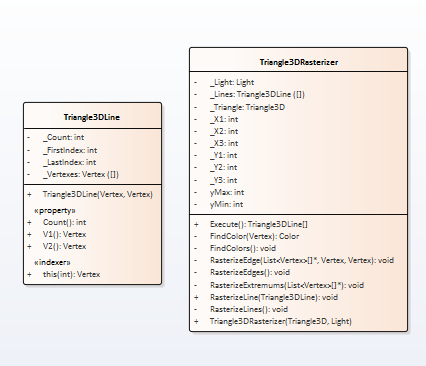
3. Класс Model – модель, состоящая из треугольников.



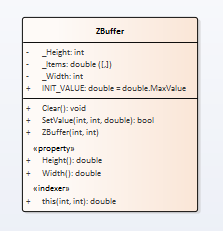
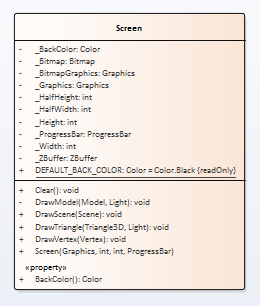
4. Класс Scene используется для представления сцены, содержащей модель и источник света.



5. Triangle3DRasterizer выполняет растровую развертку треугольника. Triangle3DLine представляет горизонтальную строку треугольника.



6. Для вывода изображения на экран используется класс Screen. Для удаления невидимых граней используется ZBuffer.



7. Для выполнения преобразований объектов сцены используются классы модификаций, содержащие матрицы соответствующих преобразований.

## 4.4. Интерфейс программного продукта

Программный продукт имеет интуитивно понятный пользовательский интерфейс.

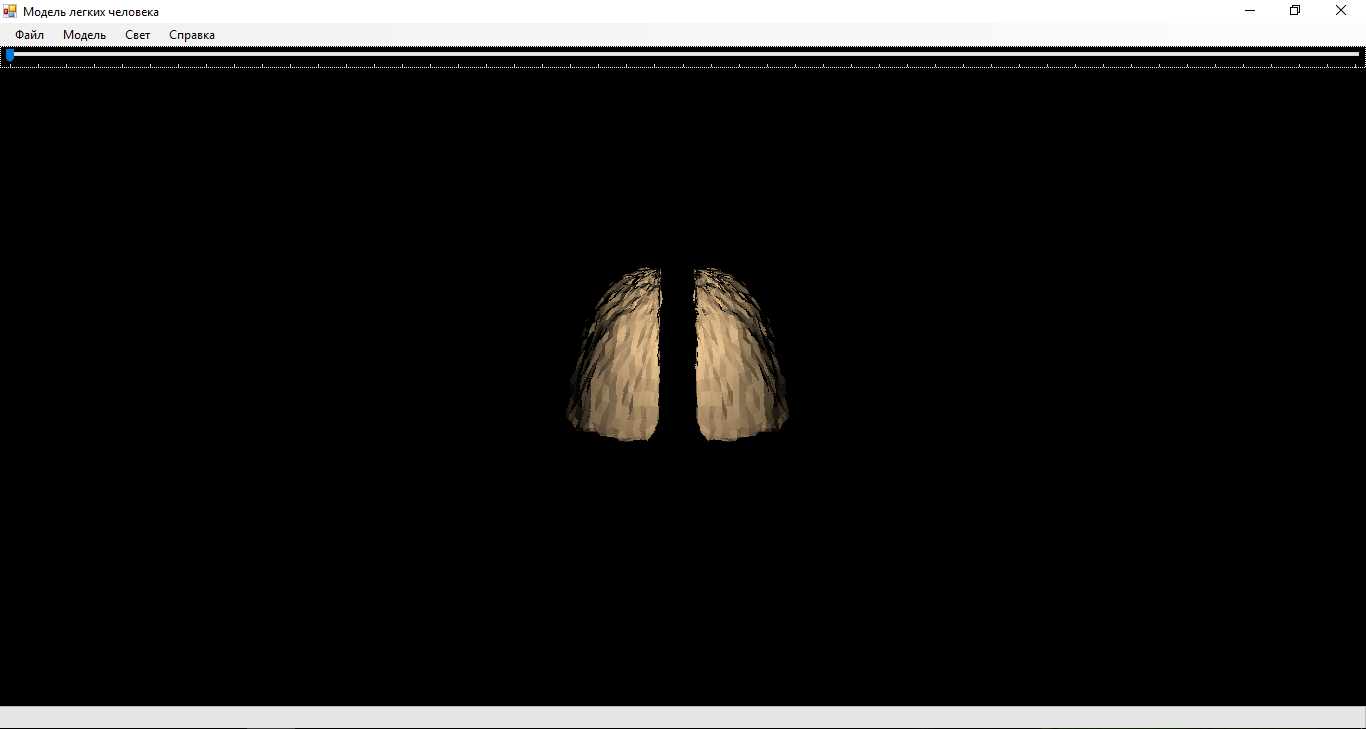


Рис. 4.4.1. Интерфейс программного продукта.

При успешном запуске появляется окно программы, имеющее графическую поверхность и меню. Пользователю предоставляется возможность выбрать папку, содержащую файл в формате XML, содержащий модель легких человека.

4.5. Изменение внешнего вида легких под воздействием никотина

В интернете можно найти немало пугающих фотографий, которые наглядно демонстрируют, как страшно, ужасающе выглядят лёгкие курильщика. Но из-за этого многим людям наоборот начинает казаться, что всё это – не более чем подделка. Но, к сожалению, это не так.

Когда человек появляется на свет, то его лёгкие имеют яркий цвет. Как известно, при курении легкие тускнеют и значительно изменяются визуально, на них начинают появляться складки, затрудняющие дыхание человека. Причина таких изменений – выделение различных отравляющий веществ, содержащихся в сигаретном дыме, некоторые из которых смертельно опасны и при продолжительном курении могут привести к смерти человека.



Рис. 4.5.1. Легкие здорового человека.

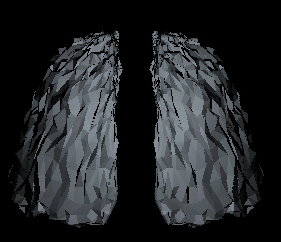


Рис. 4.5.2. Легкие курильщика со стажем.

## 4.6. Преобразования объектов сцены

В программе предусмотрена возможность проведения геометрических преобразования с моделью и источником света.

Модель можно масштабировать и вращать.

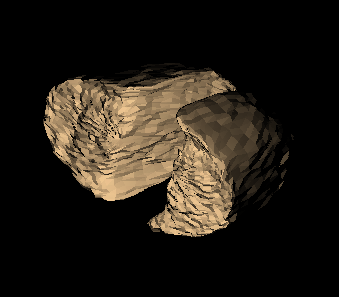


Рис. 4.6.1. Геометрические преобразования модели.

Источник света можно перемещать в любом направлении.

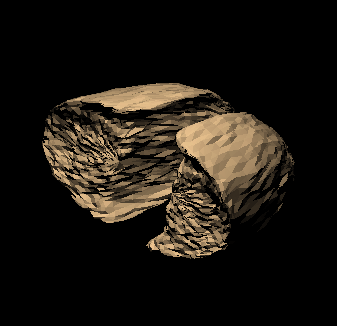


Рис. 4.6.2. Геометрические преобразования источника света.

## 4.7. Обработка исключительных ситуаций

В случае возникновения исключительных ситуации в результате нехватки системных ресурсов или неверных действий пользователя предусмотрен вывод сообщений о возникновении исключительной ситуации.

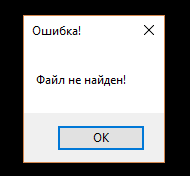


Рис. 4.7.1. Вывод сообщения об исключительной ситуации.

## 4.8. Исследование временных характеристик

|  |  |
| --- | --- |
| Коэффициент масштабирования | Время подготовки изображения, с |
| 1 | 0,1930 |
| 1,5 | 0,2109 |
| 2,25 | 0,2845 |
| 3,375 | 0,3994 |
| 5,0625 | 0,6307 |
| 7,5938 | 1,1044 |
| 11,3906 | 1,9993 |

Рис. 4.8.1. Зависимость времени подготовки изображения от масштаба.

# 

# 5. Заключение

В данной работе было реализовано построение реалистичной модели легких человека и визуализация влияния никотина на легкие человека.

В ходе работы были выявлены зависимости визуальных характеристик от положения источника света и временных характеристик от масштаба изображения.

В качестве путей развития программы можно выбрать два: оптимизацию скорости построения изображений и улучшение визуальной составляющей модели. Первое может быть достигнуто с использованием аппаратного ускорения и графических библиотек, а также с помощью реализации многопоточности. Второе достигается с помощью более сложной модели освещения: к таковым можно отнести модель модель Фонга, фотонная модель.

# 

# 6. Литература

1. Роджерс Д. Алгоритмические основы машинной графики: Пер. с англ. - М.: Мир, 1989

2. Иванов Д.В. , Карпов А.С., Кузьмин Е.П., Лемпицкий В.С., Хропов А.А. Алгоритмические основы растровой машинной графики – М.: Интернет-Университет Информационных технологий; БИНОМ. Лаборатория знаний, 2012

3. Никулин Е.А. Компьютерная геометрия и алгоритмы машинной графики.-СПб. :БХВ-Петербург. 2003

4. Иванов В.П., Батраков А.С. Трехмерная компьютерная графика /Под ред. М.Полищука.- М.: Радио и связь, 1995

5. Курсы лекций по компьютерной графике: Куров А.В.