|  |  |
| --- | --- |
| **-** | **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”

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

Студент Царев Антон Андреевич

*фамилия, имя, отчество*

Группа ИУ7-53Б

Тип практики технологическая

Название предприятия МГТУ им. Н. Э. Баумана, кафедра ИУ7

Студент Царев А. А.

*подпись, дата фамилия, и.о.*

Руководитель практики Куров А. В

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2022 г.*

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

« \_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_ г.

**ЗАДАНИЕ**

**на прохождение производственной практики**

**\_\_\_\_\_\_\_\_\_\_\_\_технологическая\_\_\_\_\_\_\_\_\_\_\_**

Тип практики

Студент

Царев Антон Андреевич \_\_\_\_\_\_\_ курса группы \_\_\_\_\_\_\_\_\_\_\_\_\_

Фамилия Имя Отчество № курса индекс группы

в период с \_\_\_\_\_\_. \_\_\_\_\_\_.20\_\_\_\_\_ г. по \_\_\_\_\_\_. \_\_\_\_\_\_.20\_\_\_\_\_ г.

*Предприятие:* МГТУ им. Н. Э. Баумана

*Подразделение:* кафедра ИУ7

(отдел/сектор/цех)

*Руководитель практики от предприятия (наставник):*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия Имя Отчество полностью, должность)

*Руководитель практики от кафедры:*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия Имя Отчество полностью, должность)

*Задание:*

**1. Начать разработку программы моделирования детской игрушки “Лизун”**

**2. Определить способы представления объектов, проанализировать и выбрать алгоритмы для их обработки**

**3. Закрепить знания и навыки, полученные в ходе изучения курса компьютерной графики**

Дата выдачи задания « \_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

Руководитель практики от предприятия  **\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/**

Руководитель практики от кафедры **\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/**

Студент **\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/**

# Введение

В настоящее время компьютерная графика имеет широкое применение в различных сферах. В частности, визуализация объектов с помощью электронно-вычислительных машин используется в киноиндустрии, разработке компьютерных игр и моделировании физических процессов.

Особое внимание уделяется моделированию деформируемых тел, которые могут менять свою форму, внутреннюю структуру, объем и площадь поверхности под действием внешних сил. Одним из таких тел является детская игрушка “Лизун” (она же “Слайм”).

Целями данной работы являются:

- выбор и обоснование модели тела и алгоритмов, которые можно применить для реализации программы моделирования детской игрушки “Лизун”;

- разработка оконного интерфейса программы;

- закрепление знаний и навыков, приобретенных в ходе изучения курса компьютерной графики;

# 1. Аналитическая часть

## 1.1 Описание объектов сцены

Сцена состоит из следующих объектов: камера, источник света, пол, имеющий структуру, и слайм.

### 1.1.1 Камера

Камера представляет собой невидимый объект, содержащий в себе информацию о координатах положения камеры в пространстве и векторе направления взгляда и использующийся для получения изображения на дисплее.

### 1.1.2 Источник света

Источник света представляет собой материальную точку, испускающую лучи света во все стороны. Данный объект хранит в себе информацию о координатах положения в пространстве источника света, интенсивности и цвете в формате RGB.

### 1.1.3 Пол

Пол представляет собой бесконечную плоскость, заданную уравнением . Предполагается, что все объекты находятся над полом или на нем.

### 1.1.4 Слайм

В нашей программе слайм представляет собой упругое тело. Это означает, при деформации тела возникают силы, которые стремятся вернуть тело в прежнюю форму. При отсутствии деформирующих сил наше тело будет иметь форму шара.

#### 1.1.4.1 Выбор модели представления трехмерного объекта

Для задания трехмерных моделей выделяют три формы: каркасную, поверхностную и объемную.

Каркасная модель является простейшим видом. В ней задается информация о вершинах и ребрах объекта. Однако ввиду своей простоты данный вид обладает серьезным недостатком: не всегда корректно передается представление об объекте.

Поверхностная модель предполагает хранение информации не только о врешинах и ребрах объекта, но и о его поверхности. Поверхность может быть описана как аналитически, так и с помощью задания участков поверхности как поверхностей того или иного рода. Недостатком данной модели является невозможность определения, с какой стороны поверхности находится материал объекта.

Объемная модель отличается от поверхностной лишь тем, что в ней мы храним информацию о расположении материала объекта, указывая направление вектора внутренней нормали.

Для данного проекта была выбрана поверхностная модель, так как каркасная модель не всегда правильно передает представление об объекте, а объемная модель является избыточной в рамках решаемой задачи.

#### 1.1.4.2 Выбор модели представления поверхности объекта

Поверхность объекта может быть описана как аналитически, так и с помощью полигональной сетки. В аналитическом методе поверхность объекта рассматривается как множество точек, координаты которых удовлетворяют заданному уравнению, в то время как полигональная сетка представляет собой совокупность вершин, ребер и полигонов, соединенных таким образом, что каждое ребро принадлежит не более, чем двум многоугольникам.

Существует несколько способов описания полигональных сеток:

1) Список вершин

В этом случае информация о сетке хранится в виде перечня координат вершин, в котором каждая вершина записывается один раз в виде тройки координат. При этом каждый многоугольник задается множеством порядковых номеров вершин в списке.

2) Список ребер

Базовую информацию дает список вершин, но многоугольники описываются ссылками на список ребер, в котором каждое из них упоминается только один раз. В свою очередь, ребра представляются в виде пары граничных вершин и перечня смежных многоугольников, число которых не превшает двух.

3) Список граней

В данном способе хранятся список вершин и список граней. В каждой грани хранится информация о трех вершинах.

Для данного проекта была выбрана модель полигональной сетки, ввиду простоты реализации и избежания проблем при описании сложных объектов. Кроме того, хранение информации о сетке будет осуществляться с помощью списка граней, так как данный способ предоставляет полную информацию о гранях, которая может быть крайне полезна при реализации алгоритмов удаления невидимых ребер и поверхностей, и позволяет преобразовывать объект через список вершин.

#### 1.1.4.3 Метод физического моделирования

Для описания динамики движения упругого тела была выбрана модель масс с пружинами, так как она позволяет выполнять простые вычисления и дает приемлемую точность моделирования.

Система масс с пружинами подразумевает хранение объекта, как совокупности точечных масс, соединенных между собой сетью невесомых пружин. Пружины обычно моделируются как упругие, то есть сила, действующая на массу i со стороны пружины, вычисляется по закону Гука:

(1)

где – жесткость пружины, соединящий массы i и j; - разница радиус-векторов масс j и i соответственно; - длина покоя, то есть расстояние между массами i и j, при котором сила упругости равна нулю.

Физические тела не являются идеально упругими, так как они рассеивают энергию во время деформации. Чтобы учесть это, для затухания относительного движения используются вязкоупругие пружины. Обычно они моделируются следующим выражением:

(2)

где - коэффициент затухания; , - векторы скоростей масс i и j соответсвенно. Однако данная формула неудобна для использования, так как гасит вращение твердого тела. Поэтому предпочтительнее применять немного измененное выражение:

(3)

где . По сути, мы проецируем разность скоростей на вектор разности радиус-векторов масс и допускаем силу только в этом направлении.

<написать, что по итогу из себя представляет слайм, если хватит места>

## 1.2 Выбор алгоритма удаления невидимых ребер и поверхностей

<введение>

1) Алгоритм Робертса

Алгоритм Робертса работает в объектном пространстве, осуществляя удаление невидимых линий выпуклых тел. Выполнение данного алгоритма можно разделить на 4 этапа:

1. Подготовка исходных данных;

2. Удаление ребер, экранируемых самим телом;

3. Удаление ребер, экранируемых другими телами;

4. Удаление линий пересечения тел, экранируемых самими телами и другими телами, связанными отношением протыкания.

На первом этапе для каждого тела формируется матрица, каждый столбец которой содержит коэффициенты уравнения плоскости

(4)

проходящей через соответствующую грань:

(5)

Матрица тела должна быть сформированна корректно. Это значит, что любая точка, лежащая внутри тела, должна лежать по положительную сторону от каждой грани тела. Если для какой-либо грани это условие не выполняется, соответствующий столбец матрицы умножается на -1.

На втором этапе указывается расположение наблюдателя и направление его взгляда. Как правило, наблюдатель располагается в бесконечности на положительной полуоси z и смотрит в сторону начала координат. В однородных координатах вектор такого направления равен

(6)

Для определения невидимых граней нужно умножить вектор на матрицу тела . Отрицательные компоненты результирующего вектора соответствуют невидимым ребрам тела.

На третьем этапе работы алгоритма для определения невидимых линий строится луч, соединяющий точку наблюдения с точкой на ребре. Точка невидима, если луч проходит через тело.

На четвертом этапе ведется поиск ребер, образовавшихся в результате взаимного протыкания тел. Вновь полученные ребра проверяются на экранирование телами.

Данный алгоритм дает высокую точность вычислений, однако сложен в реализации.

2) Алгоритм Варнока

Данный алгоритм работает в пространстве изображений. 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 подокна не станет достаточно простым для визуализации или его размер не достигнет требуемого предела разрешения.

Конкретная реализация алгоритма Варнока заисит от метода разбиения окна и от критериев, используемых для того, чтобы решить, является ли содержимое окна достаточно простым. В оригинальной версии алгоритма Варнока каждое окно разбивается на четыре одинаковых подокна. Окно является пустым, когда все многоугольники сцены являются внешними по отношению к этому окну. Пределом разбиения является размер окна в 1 пиксель. Тогда определяем глубину каждого из рассматриваемых многоугольников в этой точке и изображаем точку многоугольника, наиболее близкого к пользователю.

Данный алгоритм прост в реализации, однако возможны большие затраты по времени, если рассматривается область с большим количеством информации.

3) Алгоритм, использующий z-буфер

Данный алгоритм работает в пространстве изборажений. В нем используются буфер кадра и z-буфер. Буфер кадра используется для запоминания интенсивности каждого пиксела в пространстве изображения. z-буфер — это отдельный буфер глубины, используемый для запоминания глубины (координаты z) каждого видимого пиксела в пространстве изображения.

В процессе работы значение z каждого нового пикселя, 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 производится корректировка z-буфера новым значением z. Если же сравнение дает противоположный результат, то никаких действий не производится.

Главное преимущество алгоритма — простота. Кроме того, этот алгоритм решает задачу об удалении невидимых поверхностей и делает тривиальной визуализацию пересечений сложных поверхностей. Однако, несмотря на свою простоту, алгоритм требует использование большого объема памяти. Кроме того, недостаток алгоритма заключается еще в трудоемкости реализации эффектов прозрачности.

4) Алгоритм обратной трассировки лучей

Основная идея, лежащая в основе трассировки, заключается в том, что наблюдатель видит любой объект посредством испускаемого неким источником света, который падает на объект и затем каким-то путем доходит до наблюдателя согласно законам оптики. Однако не все лучи света доходят до наблюдателя, что делает процесс отслеживания этих лучей вычислительно неэффективным. Поэтому лучше отслеживать лучи в обратном направлении, то есть от наблюдателся к объекту, причем испуская лучи от точки наблюдателя через каждый пиксел экрана.

Для данной лабораторной работы был выбран алгоритм обратной трассировки лучей, так как он, несмотря на малую производительность из-за большого количества вычислений, позволяет получить реалистичное изображение с учетом оптических явлений, таких как отражение и преломление. При этом скорость работы данного алгоритма слабо зависит от сложности сцены.

## 1.3 Выбор модели освещения

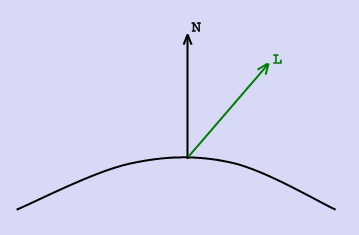
Построение реалистических изображений включает в себя как физические, так и психологические процессы. В частности, при визуализации сцены учитываются законы оптики: отражение, преломление, поглощение света и т. д. Поэтому необходимо проанализировать и выбрать модель освещения для построения изображения.

1) Модель Ламберта

Модель Ламберта моделирует диффузное освещение, то есть свет точеного источника диффузно отражается от идеальной поверхности по закону косинусов Ламберта: интенсивность отраженного света пропорциональна косинусу угла между направлением света L и нормалью к поверхности N (рисунок 1), то есть

(7)

где - интенсивность отраженного света; - интенсивность точечного источника в направлении L; - коэффициент диффузного отражения; - угол между направлением света L и нормалью к поверхности n, (рисунок 1).

Рисунок 1: Модель Ламберта

2) Модель Фонга

<описание>

3) Глобальная модель освещения

<описание>

<обоснование выбора глобальной модели освещения и ее модификация с применением закона Бугера-Ламберта-Бера>

# 2. Конструкторская часть

## 2.1 Общий алгоритм программы

# 3. Технологическая часть