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 (ИУ)

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии (ИУ7)

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ

Студент Иванов Всеволод Алексеевич
фамилия, имя, отчество

Группа ИУ7-42Б

Тип практики Программа моделирования игры бильярд

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

Студент _____ Иванов В.А.
подпись, дата *фамилия, и.о.*

Руководитель практики _____ Куров А.В.
подпись, дата *фамилия, и.о.*

Оценка _____

2020 г.

Индивидуальное задание

Разработка программы для моделирования и трёхмерной визуализации настольной игры бильярд. Проанализировать методы построения реалистичных изображений и обосновать их выбор для поставленных задач.

Оглавление

Введение

Компьютерная графика занимает важное место в современных информационных технологиях. На эту область приходится решение таких задач как визуализация построек при конструировании, создание симуляций управления самолётов, поездов и т.п., и в компьютерных играх. Спектр проблем в данных областях достаточно широк, и поэтому существует множество алгоритмов для визуализации трёхмерных изображений. Их разнообразие объясняется тем, что не существует методов, одновременно создающих высоко реалистичное изображение и показывающих высокое быстродействие, поэтому различные алгоритмы позволяют сделать упор на наиболее важную характеристику.

Целью практической работы является создания ПО для моделирования игры бильярд. Актуальность данной темы объясняется тем, что для получения многих теоретических знаний о бильярде (понимание физики игры, знание стандартных приёмов и игровых ситуаций), компьютерная симуляция может оказаться даже более эффективной, чем реальная игра.

Программа будет предоставлять пользователю трёхмерную модель бильярдного стола и шаров с возможностью осуществления ударов и дальнейшей визуализацией поведения шаров. Для удобства пользователя также будет предоставлена возможность изменения положения камеры и источников света вокруг стола. В данной задаче реалистичность изображения не имеет столь значительную роль, как плавность изображения и правдоподобность движения шаров. Поэтому при дальнейшем анализе предпочтение будет отдано алгоритмам, обеспечивающим большую частоту вывода кадров на экран.

Основная часть

1 Аналитическая часть

1.1 Детализация задачи

Целью практической работы является создание программы для моделирования настольной игры бильярд. В программе должна иметься возможность взаимодействия с шарами, задания их расстановки, рассмотрения стола с различных ракурсов и освещённостью.

Для реализации всех перечисленных требований необходимо решить следующие задачи:

- Выделить объекты сцены и выбрать модель их представления
- Проанализировать, выбрать и реализовать алгоритмы визуализации объектов. Обязательным требованием является реалистичность создаваемого трёхмерного изображения
- Разработать физическую модель поведения объектов, обеспечивающую правдоподобное поведение объектов сцены
- Предоставить возможность задания начальных конфигурационных параметров игры
- Реализовать графический интерфейс для предоставления пользователю вышеописанных возможностей взаимодействия с игрой

1.2 Описание моделей визуализируемых объектов

В качестве объектов визуализируемой сцены можно выделить следующие сущности:

- Камера. Представляет собой наблюдателя, поэтому задаётся при помощи трёхмерной точки и трёхмерного вектора направления обзора.
- Источник света. Задаётся при помощи трёхмерной точки и значения интенсивности света.

- Ограничивающая плоскость. Используется как поверхность, на которой располагаются все остальные объекты сцены. Является параллельной плоскости 0XZ, задаётся значением y и цветом (или текстурой).
- Бильярдный стол. Составной объект, задаваемый при помощи каркасной модели. Для удобства, центр объекта расположен в точке (0, y, 0), а стороны ориентированы параллельно осям координат. В свою очередь, стол состоит из:
    - Плита, покрытая сукном. Является параллелепипедом, задаётся при помощи ширины, длинны и высоты.
    - Бортики с лузами. Являются параллелепипедами, имеющими кругообразные вырезы под лузы. Задаются при помощи ширины, длинны и высоты, а также радиусом и положением луз.
    - Ножки. Также являются параллелепипедами, задаются шириной, длиной и высотой.
- Бильярдные шары. Являются шарами, заданными с помощью аппроксимирующей каркасной модели. Параметрами являются трёхмерная точка положения шара и цвет.

1.3 Анализ алгоритмов удаления невидимых линий и поверхностей

Основным фактором, на который стоит сделать акцент в данной задаче является частота вывода изображения на экран. Пользователь должен получить достаточно плавную для восприятия анимацию, в то время как реалистичностью и детализацией изображения можно в разумных пределах пренебречь. Это обусловлено тем, что в данной программе более значительная роль отводится под саму природу поведения объектов, в то время как реалистичность отображаемого играет уже второстепенную роль. Поэтому, при выборе алгоритма следует ориентироваться в первую очередь на быстродействие.

Опираясь на вышеизложенное, требуется рассмотреть существующие алгоритмы, оценить их качества, и на основании этого выбрать наиболее подходящий вариант.

Алгоритм Робертса

Данный алгоритм работает в пространстве объектов. В первую очередь, для каждого из объектов формируются матрица тела, описывающих с помощью 4х коэффициентов уравнение плоскости каждой из граней тела. После чего, производится ориентация плоскостей так, чтобы нормаль каждой из них была внутренней (для этого используется заведомо внутренняя точка тела). Следующим этапом происходит удаление граней тела, экранируемых им самим. Далее, каждое ребро тела проверяется на факт экранирования частями других объектов, удаляются невидимые части.

Преимуществом данного алгоритма является простота вычислений, и в то же время точность результата, которая главным образом достигается за счёт работы в объектном пространстве.

Недостатком является временная сложность, выражаемая как число объектов на сцене в квадрате. Существуют также более оптимизированные версии алгоритма, например использующие сортировку объектов по z, однако они заметно более сложные в реализации.

Учитывая тот факт, что из-за наличия множества шаров, на сцене будет большое количество объектов, алгоритм Робертса будет не самым оптимальным вариантом.

Алгоритм Варнока

Принцип алгоритма Варнока, работающий в пространстве изображения, можно описать как “Разделяй и властвуй”. Область рисунка разбивается на несколько частей, для каждой из которых производится анализ попавших в неё многоугольников. В случае, если цвет области не определяется одним

многоугольником, вышеописанная операция повторяется уже для данной области. Иначе, область полностью закрашивается цветом многоугольника.

Преимуществом алгоритма является то, что с помощью него можно эффективно работать с многоугольниками, занимающими большие области в пространстве изображений.

Недостатком является низкая эффективность, в случае, когда на сцене присутствует множество небольших по размеру многоугольников. В таком случае, для заливки объектов придётся делать достаточно много разбиений, а сами конечные заливаемые области по размеру будут стремиться к пикселю. Данный недостаток является существенным в данном случае, так как шары будут задаваться с помощью множества аппроксимирующих многоугольников.

Алгоритм, использующий Z-буфер

Данный алгоритм работает в пространстве изображения. Ключевым понятием в нём является буфер кадра, являющийся матрицей, содержащей некую информацию о каждом пикселе изображения. Основным буфером является Z-буфер, содержащий координату z точки на наиболее приближённом объекте в данном пикселе. Алгоритм сравнивает значения Z-буфера для каждого объекта с уже имеющимся. В случае, если для какого-любо пикселя значение z объекта больше, информация о данном пикселе обновляется в соответствии с данным объектом для каждого буфера, например, буфера интенсивности света.

Преимуществом алгоритма является линейная зависимость от числа визуализируемых объектов. Также не важен и порядок анализа объектов, что позволяет не осуществлять сортировку. Также, временные затраты практически не зависят от количества рассматриваемых многоугольников, что важно для шаров, имеющих большое количество граней.

Недостатком является объём занимаемой буферами памяти. Однако, данный недостаток не является столь существенным, ввиду достаточного количества памяти в современных условиях.

Алгоритм обратной трассировки лучей

Алгоритм работает в пространстве изображения. Для каждого пикселя от точки зрения наблюдателя исходит советующий луч, для которого ищутся пересечения со всеми объектами, после чего в соответствии с ближайшим из них происходит закраска пикселя.

Преимуществом алгоритма является линейная зависимость от количества объектов на сцене. Также данный алгоритм удобно использовать вместе с глобальной моделью освещения, что позволяет создавать реалистичные изображения.

Недостатком является большое количество вычислений для каждого из лучей, а также для его отражений, что является недопустимым в данной задаче, так как приведёт к низкой частоте кадров обновления изображения.

Вывод

Проанализировав данные алгоритмы, можно прийти к выводу, что наиболее подходящим решением для данной программы будет использование алгоритма z-буфера. Данный алгоритм позволит выполнять синтез изображения достаточно быстро, а также упростит задачу закраски и освещения объектов за счёт соответствующих буферов.

1.4 Анализ алгоритмов закрашивания

Существует несколько методов закраски, позволяющих по-разному передать цвет и освещённость граней объектов. Рассмотрим их, для того чтобы определиться с наиболее подходящим для данной задачи.

Простая модель закраски

В данном методе вся грань закрашивается с одинаковой интенсивностью, определяемой по нормали к данной поверхности. Алгоритм является самым быстродейственным, однако полученное изображение не будет отличаться реалистичностью отображения освещённости.

Закраска по Гуро

В данном методе используется интерполяция интенсивности. За счёт усреднения нормалей граней, сходящихся в одной точке, вычисляется нормаль к вершинам многогранника, по которой вычисляется интенсивность в вершине. Далее, полученные значения интерполируются сначала по граням, а потом и по плоскостям многогранника. В итоговом изображении происходит сглаживание ребристости объектов, что хорошо в случае аппроксимации гладких объектов с помощью каркасной модели.

Закраска по Фонгу

Данный метод схож с закраской по Гуро. Различием является то, что вместо интенсивности происходит интерполяция по значению самой нормали. Это позволяет получить улучшенную аппроксимацию кривизны поверхности по сравнению с Гуро и лучше передаёт блики, однако требует в 3 раза больше вычислений из-за интерполяции трёх значений вместо одного и требования вычислять интенсивность для каждой из нормалей.

Вывод

В поставленной задаче лучше всего подойдёт метод Гуро, так как он сможет обеспечить достаточно реалистичное изображение множества используемых в сцене закруглённых объектов, и при этом будет иметь приемлемую ресурсозатратность. Также данный метод будет достаточно удобно сочетать с выбранным алгоритмом z-буфера.

1.5 Анализ алгоритмов освещённости

1.6 Физическая модель поведения объектов

Ф

2 Конструкторская часть

Ф

3 Технологическая часть

Ф

Ф

Заключение

Ф

Список использованных источников

Ф

Приложения

Ф