



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»

Рассчётно-пояснительная записка

к научно-исследовательской работе

на тему:

Трёхмерная среда для визуализации окружения робота-собеседника Ф-2

Студент ИУ7-55Б

(подпись, дата)

Бугаенко А.П.

(Фамилия И.О.)

Преподаватель

(подпись, дата)

Волкова Л.Л.

(Фамилия И.О.)

Москва, 2022

Содержание

	Введение	3
	Основная часть	4
1	Анализ предметной области	5
2	Классификация существующих решений	7
3	Построение прототипа метода	8
	3.1 Формализация трёхмерной виртуальной среды	8
	3.1.1 Анализ виртуальной сцены	8
	3.1.2 Анализ виртуального аватара робота Ф-2	8
	3.1.3 Анализ виртуальной камеры	9
	3.1.4 Анализ виртуального аватара пользователя	9
	3.2 Анализ методов, необходимых для формирования и отрисовки трёхмерного объекта	10
	3.2.1 Анализ способа представления поверхности	10
	3.2.2 Анализ алгоритмов удаления невидимых линий и поверхностей	11
	3.2.3 Анализ методов закрашивания	13
	3.3 Вывод	14
	Вывод	14
	Список использованных источников	15

Основная часть

1 Анализ предметной области

Ф-2 – это эмоциональный робот, разрабатываемый командой российских исследователей, обладающий умением общаться с людьми. С его помощью можно разрабатывать стратегии диалога, мимику и жесты, изменения направления взгляда и многое другое. Робот сделан максимально простым, чтобы его можно было легко собрать. Использование данной платформы позволяет исследовать эмоциональный контакт, возникающий за счёт поведения робота, а не за счёт его внешнего вида. Робот напоминает мультипликационных персонажей, которые совсем не похожи на человека, но при этом эмоциональны и симпатичны благодаря своим жестам и мимике, а не из-за внешнего сходства с человеком. Робот реагирует на слова: он принимает на вход текст на естественном языке, строит смысл этого текста, выбирает эмоцию и выполняет жесты, характерные для этой эмоции. Когнитивный компонент робота отвечает за мышление и за понимание текста. Для входящего высказывания когнитивный компонент должен построить некоторый набор умозаключений, выбрать эмоции, которые может вызывать текст, чтобы далее проявить эти эмоции в поведении робота. Робот может читать книги в виде текстовых файлов, новости и блоги через подписку RSS, воспринимать устную речь через сторонний сервис преобразования в письменную форму. По письменному тексту для каждого предложения когнитивный компонент строит синтаксическое дерево, а затем конструирует семантическое представление – смысл текста. Смысл текста вызывает у робота различные умозаключения, выводы и ответные реакции. Эти процессы моделируются системой отношений типа «если-то» – сценариев. Робот сравнивает смысл поступившего текста с посылками всех сценариев и активизирует ближайшие сценарии. Смысл поступившего текста может активизировать сценарии, ответственные за эмоциональную обработку – Меня никто не любит, Я никому не нужен или, наоборот, Приятно быть в центре внимания. Когнитивный компонент может работать отдельно от робота: он будет прочитывать множество текстов, конструировать их смысл, приписывать каждому смыслу эмоцию и сохранять результаты. При управлении роботом каждый сценарий может сформировать поведенческий пакет на языке VML и передать его роботу для выполнения. Таким образом, поведение робота составляется из VML-пакетов – реакций на окружающие события или просто движений в состоянии покоя: когда роботу нечего делать, он будет слегка двигаться [1].

Одной из проблем данного робота является невозможность проводить эксперименты без наличия физической копии робота. Решить данную проблему представляется возможным при помощи создания виртуальной среды, которая позволит эмулировать робота Ф-2 и набор из различных человеческих входов (положение головы, эмоции, которые данный вход показывает на лице и ограниченная мимика и жестикуляция).

Создание подобной виртуальной среды будет иметь следующие применения в проекте Ф-2.

1) будет предоставлена возможности воспроизведения реакций робота в тестовом режиме на этапе разработки элементов реакции или набора реакций, воспроизводимых совместно;

2) станет возможной постановка экспериментов по взаимодействию робота с респондентами в виртуальном режиме. Это позволит проводить больше экспериментов по коммуникации и, следовательно, получать больше обратной связи от респондентов, в том числе позволит респондентам взаимодействовать с роботом удалённо вместо того или вместе с тем, чтобы приезжать к роботу;

3) станет возможным проводить параллельные эксперименты по коммуникации человека и робота, что позволит объединить усилия множества респондентов, что приведет к более интенсивному развитию проекта на материале собранной обратной связи.

Таким образом целью данной работы является проектирование описанной выше виртуальной среды. Для того, чтобы достичь поставленной цели, необходимо выполнить следующие задачи:

1) провести анализ предметной области и сформировать основные требования к алгоритмам, которые будут использоваться в реализации виртуальной среды для робота Ф-2;

2) выбрать алгоритмы на основе проведённого выше анализа и описать принципы их работы;

3) оценить существующие решения данной задачи;

4) описать используемые в данных алгоритмах структуры данных;

5) привести схемы рассмотренных выше алгоритмов;

6) описать структуру разрабатываемого ПО;

2 Классификация существующих решений

3 Построение прототипа метода

В данном разделе рассматриваются трёхмерная виртуальная среда и объекты, которые в ней содержатся.

3.1 Формализация трёхмерной виртуальной среды

В данном подразделе рассматриваются объекты трёхмерной виртуальной среды, описываются их характеристики и описываются отношения между объектами.

3.1.1 Анализ виртуальной сцены

Основным объектом трёхмерной виртуальной среды является сцена - виртуальное трёхмерное пространство, являющееся базовым объектом трёхмерной виртуальной среды.

Сцена может содержать в себе следующие объекты:

- Виртуальный аватар робота Ф-2;
- Виртуальная камера;
- Аватар пользователя.

3.1.2 Анализ виртуального аватара робота Ф-2

Виртуальный аватар робота Ф-2 - объект сцены трёхмерной виртуальной среды, являющийся представлением реального прототипа робота Ф-2. Для того, чтобы получить возможность проводить эксперименты с виртуальным аватаром робота Ф-2, необходимо эмулировать данные, подающиеся на вход роботу и реакцию робота на полученные данные. При обработке информации робот Ф-2 использует два входных параметра - фразу на естественном языке и положение лица собеседника. [1]

Естественный язык - в лингвистике и философии языка язык, используемый для общения людей (в отличие от формальных языков и других типов знаковых систем, также называемых языками в семиотике) и не созданный целенаправленно (в отличие от искусственных языков). [2]

Необходимые данные виртуальный аватар робота получает от виртуального аватара пользователя. Положение аватара пользователя является положением лица собеседника. Фраза на естественном языке передаётся роботу в виде строки, помеченной как ввод от данного пользователя.

По причине того, что реакция робота на поданные на вход данные представляет из себя изменения положения тела робота и выражения "лица то на виртуальной сцене робот Ф-2 представляется в виде серии трёхмерных примитивов, образующих основные элементы геомет-

рии робота Ф-2, которые включают в себя основание робота, левую и правую конечности, шею и голову с дисплеем. Помимо эмуляции положения сочленений робота в пространстве также необходимо эмулировать различные эмоции на лице робота, что достигается использованием двумерной текстуры, которая затем накладывается на одну из плоскостей трёхмерного примитива, являющегося представлением экрана робота Ф-2.

3.1.3 Анализ виртуальной камеры

Виртуальная камера - объект сцены трёхмерной виртуальной среды, эмулирующий камеру в трёхмерном пространстве сцены и позволяющий получать двумерный снимок (проекцию) сцены.

3D-проекция (или графическая проекция)-это метод проектирования, используемый для отображения трехмерного (3D) объекта на двумерной (2D) поверхности. Эти проекции основаны на визуальной перспективе и анализе аспектов, чтобы спроецировать сложный объект для возможности просмотра на более простой плоскости. [3]

Существует два основных вида проецирования:

- - параллельная проекция;
- - перспективная проекция.

Параллельная проекция - это проекция объекта в трехмерном пространстве на неподвижную плоскость, известную как плоскость проекции или плоскость изображения, где лучи, известные как линии обзора или проекционные линии, параллельны друг другу.

Перспективная проекция - это приблизительное представление, обычно на плоской поверхности, изображения таким, каким оно видится глазу.

В случае виртуальной камеры для трёхмерной виртуальной среды будет использоваться перспективная проекция для того, чтобы позволить пользователям, использующим редактор для проведения экспериментов с виртуальным аватаром робота Ф-2, интуитивно понимать, каким образом объекты сцены расположены относительно друг друга.

3.1.4 Анализ виртуального аватара пользователя

Виртуальный аватар пользователя - объект сцены трёхмерной виртуальной среды, эмулирующий пользователя в трёхмерном пространстве сцены и позволяющий симулировать работу робота Ф-2.

Объект пользователя представляется на виртуальной сцене как трёхмерный примитив. Положение данного примитива в пространстве является положением лица пользователя, которое передаётся на вход виртуальному аватару робота Ф-2. Также объект пользователя позво-

ляет формировать и отправлять роботу строки, содержащие фразы на естественном языке и помеченные как сообщение от данного пользователя.

3.2 Анализ методов, необходимых для формирования и отрисовки трёхмерного объекта

В данном подразделе рассматриваются методы, используемые для формирования и отрисовки трёхмерных объектов, находящихся в пространстве виртуальной сцены.

3.2.1 Анализ способа представления поверхности

При рассмотрении того, каким образом можно задать некоторую поверхность в пространстве виртуальной сцены выделяется три основных способа:

- каркасный — модель представляется в виде множества вершин, связанных между собой рёбрами;
- поверхностный — модель представляется в виде множества поверхностей;
- объёмный — модель формируется из элементарных объектов (базисных тел) с использованием логических операций объединения, вычитания, пересечения.

Для того, чтобы представить трёхмерные примитивы, используется поверхностный способ описания модели, так как каркасные модели не обладают достаточным количеством деталей, что отрицательно скажется на читаемости сцены в процессе экспериментов. Объёмные модели не имеют смысла, поскольку работа ведётся не с объёмом модели, а только с её поверхностью.

Поверхностную модель мы можем задать следующими способами:

Параметрическое представление — поверхность определяется некой функцией, зависящей от одного или нескольких параметров. В основном данный метод используется для задания в трёхмерном пространстве объектов вращения.

Полигональная сетка — в этом случае поверхность задаётся совокупностью вершин, рёбер и граней, которые определяют внешний вид объекта.

Параметрическое представление не подходит, так как представление в сцене объектов, не являющихся объектами вращения будет затруднительно. В свою очередь полигональная сетка является универсальным форматом представления модели, позволяющим легко представить практически любые формы в трёхмерном пространстве и имеющим широкое применение в современном трёхмерном моделировании.

Полигональная сетка может быть реализована несколькими разными способами:

- Вершинное представление — в этом случае формируется список, в котором для каждой вершине ставятся в соответствие вершины, с которыми она соединена рёбрами.
- Таблица углов — в данном случае вершины хранятся в предопределённой таблице.
- Список граней — представляет объект как множество граней и вершин.
- Список полигонов — представляет объект как множество поверхностей, формирующих поверхность модели.

Так как виртуальная среда должна работать в режиме реального времени, то вершинное представление не используется, поскольку для того, чтобы сформировать грани необходимо пройти по данным несколько раз, что сказывается на производительности программы. По той же причине не используется таблица углов, поскольку изменение данных является затратным по времени.

Для решения данной задачи будет использоваться список треугольных полигонов, которые в совокупности будут составлять поверхность модели. Данный является наиболее подходящим, так как треугольник в пространстве однозначно задаёт плоскость, вследствие чего алгоритмы работы с ним реализуются легче и работают быстрее, чем с произвольными полигонами.

3.2.2 Анализ алгоритмов удаления невидимых линий и поверхностей

При выборе алгоритма удаления невидимых линий и поверхностей нужно учитывать особенности поставленной задачи. Одной из таких особенностей является то, что нам не нужно реалистическое изображение, поскольку для обозначения положения в виртуальном пространстве сцены аватара робота и аватаров пользователя достаточно нескольких трёхмерных примитивов, позволяющих схематически отображать состояние объектов на сцене. При этом не использование сложных алгоритмов не предпочтительно, поскольку это приведёт к излишним затратам по времени и как следствие - к замедлению работы программы.

Алгоритм трассировки лучей — при построении изображения луч посылается в заданном направлении для оценки приходящей оттуда световой энергии. Эта энергия определяется освещённостью первой поверхности, встретившейся на пути луча. Из каждого источника света выпускаются лучи во все стороны. При пересечении луча с поверхностью получают отражённый и преломлённый лучи, а если моделировать диффузное освещение, то и пучок лучей, равномерно распространяющийся во все стороны. Данный метод имеет наивысшую степень реализма и вместе с этим высокую сложность реализации и затрат ресурсов при работе программы. [6]

Алгоритм Робертса — Алгоритм Робертса представляет собой первое известное решение задачи об удалении невидимых линий. Это математически элегантный метод, работающий

в объектном пространстве. Алгоритм прежде всего удаляет из каждого тела те ребра или грани, которые экранируются самим телом. Пусть F – некоторая грань многогранника. Плоскость, несущая эту грань, разделяет пространство на два подпространства. Назовем положительным то из них, в которое смотрит внешняя нормаль к грани. Если точка наблюдения – в положительном подпространстве, то грань – лицевая, в противном случае – нелицевая. Если многогранник выпуклый, то удаление всех нелицевых граней полностью решает задачу визуализации с удалением невидимых граней. Невыпуклые тела должны быть разбиты на выпуклые части. В этом алгоритме выпуклое многогранное тело с плоскими гранями должно представиться набором пересекающихся плоскостей. [6]

Алгоритм Варнака — алгоритм Варнака является ещё одним примером алгоритма, основанного на разбиении картинной плоскости на части, для каждой из которых исходная задача может быть решена достаточно просто. Разобьём видимую часть картинной плоскости на 4 равные части. В случаях, когда часть полностью накрывается проекцией ближайшей грани и часть не накрывается проекцией ни одной грани, вопрос о закрасивании соответствующей части решается тривиально. В случае, когда ни одно из этих условий не выполнено, данная часть разбивается на 4 части, для каждой из которых проверяется выполнение этих условий, и так далее. Очевидно, что разбиение имеет смысл проводить до тех пор, пока размер части больше, чем размер пиксела. [7]

Алгоритм, использующий Z-буфер — данный алгоритм является одним 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 x и y наибольшего значения функции $z(x, y)$. [6]

Поскольку в данной задаче важна скорость построения изображения для динамической отрисовки происходящего на экране, используется алгоритм z-буфера.

3.2.3 Анализ методов закрашивания

В компьютерной графике для расчета освещенности граней объектов применяется цветовая модель RGB. Интенсивность отраженного света точек объектов вычисляют отдельно для каждой из трех составляющих цветовых компонент, а затем объединяют в результирующую тройку цветов. При расчете освещенности граней применяют следующие типы освещения и отражения света от поверхностей:

- рассеянное;
- диффузное;
- зеркальное.

Матовые поверхности обладают свойством диффузного отражения, т. е. равномерного по всем направлениям рассеивания света. Поэтому кажется, что поверхности имеют одинаковую яркость независимо от угла обзора. Для таких поверхностей справедлив закон косинусов Ламберта, устанавливающий соответствие между количеством отраженного света и косинусом угла θ между направлением на точечный источник света интенсивности I_p и нормалью к поверхности. При этом количество отраженного света не зависит от положения наблюдателя. Освещенность рассеянным светом вычисляется по формуле. 3.1

$$I_d = I_p \cdot k_d \cdot \cos\theta. \quad (3.1)$$

Значение коэффициента диффузного отражения k_d является константой в диапазоне $(0, 1)$ и зависит от материала [5].

Закраска по Гуро — алгоритм закрашки поверхности по Гуро позволяет устранить дискретность изменения интенсивности. Процесс закрашки по методу Гуро осуществляется в четыре этапа:

- 1) вычисляются нормали ко всем полигонам;
- 2) определяются нормали в вершинах путем усреднения нормалей по всем полигональным граням, которым принадлежит вершина;
- 3) используя нормали в вершинах и применяя произвольный метод закрашки, вычисляются значения интенсивности в вершинах;
- 4) Каждый многоугольник закрашивается путем линейной интерполяции значений интенсивностей в вершинах сначала вдоль каждого ребра, а затем и между ребрами вдоль каждой сканирующей строки.

Закраска по Фонгу — В методе закрашки Фонга используется интерполяция вектора нормали к поверхности вдоль видимого интервала на сканирующей строке внутри многоугольника, а не интерполяция интенсивности. Интерполяция выполняется между начальной и конечной нормалью, которые сами тоже являются результатами интерполяции вдоль ребер многоуголь-

ника между нормальными в вершинах. Нормали в вершинах, в свою очередь, вычисляются так же, как в методе закраски, построенном на основе интерполяции интенсивности. В каждом пикселе вдоль сканирующей строки новое значение интенсивности вычисляется с помощью любой модели закраски. Заметные улучшения по сравнению с интерполяцией интенсивности наблюдаются в случае использования модели с учетом зеркального отражения, т. к. при этом более точно воспроизводятся световые блики. Однако даже если зеркальное отражение не используется, интерполяция векторов нормали приводит к более качественным результатам, чем интерполяция интенсивности, поскольку аппроксимация нормали в этом случае осуществляется в каждой точке. [5]

В данной задаче использовать сложные закраски с применением интерполяции смысла не имеет, так как основной целью виртуальной трёхмерной среды является симуляция работы робота в реальном времени. Поэтому, так как для построения достаточно детального изображения не требуется расчёт отражений и бликов, используется простая диффузная закраска с использованием закона Ламберта.

3.3 Вывод

В данном разделе была рассмотрена трёхмерная виртуальная среда. Рассмотрены объекты, содержащиеся в виртуальной среде, включающие в себя виртуальную сцену, являющуюся контейнером для остальных объектов, аватар робота Ф-2, являющийся эмуляцией реального прототипа робота Ф-2 и аватар пользователя, являющийся источником данных для виртуального аватара робота Ф-2. Описаны данные, которыми обмениваются аватар робота Ф-2 и аватары пользователя в процессе проведения экспериментов, включающие в себя положение лица пользователя и фразы на натуральном языке. Проанализированы алгоритмы и методы, используемые при реализации трёхмерной виртуальной среды, в результате для представления примитивов был выбран список трёхмерных полигонов, в качестве алгоритма удаления невидимых линий и поверхностей выбран алгоритм, использующий Z-буфер, в качестве алгоритма закраски был выбран алгоритм основанный на законе косинусов Ламберта.

Вывод

Список использованных источников

1. Робот Ф-2 [Электронный ресурс] – 2021. – Режим доступа: <http://f2robot.com/robot/> (дата обращения: 29.11.2021)
2. Кибрик А. Е. Язык // Языкознание. Большой энциклопедический словарь. — Большая Российская энциклопедия, 1998. — С. 605.
3. Педди, Джон. История визуальной магии в компьютерах : как создаются красивые изображения в САПР, 3D, VR и AR. - Лондон: Спрингер, 2013. - С. 25.
4. Роджерс Д., Адамс Дж. Математические основы машинной графики. - Москва, Мир, 2001. - С. 604.
5. Шилов А. В., Лесковец И. В. Компьютерная графика - Могилев, Государственное учреждение высшего профессионального образования «Белорусско-Российский университет», 2014 - С. 38.
6. Шишкин Е. В., Боресков А. В., Зайцев А. А. - Москва, "ДИАЛОГ-МИФИ 1993. - С. 135.
7. Шишкин Е. В., Боресков А. В. - Москва, "ДИАЛОГ-МИФИ 1996. - С. 290.