

Курсовая работа

Визуализация поведения робота Ф-2

Студент:

Комаров Никита Сергеевич ИУ7-52Б

Научный руководитель:

Волкова Лилия Леонидовна

Москва – 2022 г.

Введение

Цель работы: разработать ПО с пользовательским интерфейсом, позволяющее получить динамическое изображение трёхмерной сцены с анимированной моделью робота Ф-2.

Задачи:

1. Описать состав сцены и выбрать форму представления её объектов.
2. Выбрать алгоритм удаления невидимых линий и поверхностей.
3. Выбрать модель освещения.
4. Выбрать способ анимации.
5. Разработать модель описания сцены и анимации.
6. Разработать алгоритм получения изображения.
7. Реализовать пользовательский интерфейс.
8. Выполнить тестирование и анализ быстродействия программы.

Робот Ф-2



Сцена

- Модель робота Ф-2
 - Голова – параллелепипед
 - Руки и тело – цилиндры
- 2 модели людей
 - Голова – сфера
 - Руки и тело – цилиндры
- Плоскость пола
 - Параллелепипед
 - Цвет
- Источник направленного света
 - Направление
 - Интенсивность

Алгоритмы удаления невидимых линий и поверхностей

- Z-буфер
- Алгоритм Художника
- **Обратная трассировка лучей**
- Алгоритм Робертса
- Алгоритм Маршировки лучей

Модели освещения

1. Модель Ламберта
2. **Модель Фонга**
3. Глобальная модель

Методы анимации

- **Метод ключевых кадров**
- Метод кривых движения

Выбранные алгоритмы и методы

- Аналитическое представление объектов
 - + Выигрыш по памяти
 - + Производительность
- Обратная трассировка лучей
 - + Работает с выбранной формой представления объектов
 - + Производительность
- Модель освещения Фонга
 - + Производительность
 - + Качество изображения
- Анимация методом ключевых кадров
 - + Не требует аналитического задания траекторий движения точек

Формат описания сцены/анимации

• /camera 800 600 0.5	Добавить камеру
• /box gr 2500 50 2500 0 -130 200 0 0 0	Добавить параллелепипед (плоскость пола)
• /paint gr 0 200 0	Покрасить параллелепипед
• /robot r1 0 0 200 0 180 0	Добавить модель робота
• /human h1 300 0 150 0 90 0	Добавить модель человека
• /amb_light 0.2	Добавить рассеянное освещение
• /dir_light 0.75 0 -0.2 1	Добавить источник направленного света
• /anim r1	Начать анимацию робота
• /key 0 r1	Добавить в анимацию робота ключевой кадр со значениями по умолчанию
• /key 5 r1 0 0 200 0 135 0	Добавить в анимацию ключевой кадр анимации с указанными значениями
• /anim r1.lhand	Начать анимацию левой руки робота
• /key 0 r1.lhand	Добавить в анимацию левой руки робота ключевой кадр со значениями по умолчанию
• /key 5 r1.lhand	Добавить в анимацию левой руки робота ключевой кадр со значениями по умолчанию
• /key 15 r1.lhand 0 0 0 120 45 0	Добавить в анимацию левой руки робота ключевой кадр с указанными значениями
• /anim r1.rhand	Начать анимацию правой руки робота
• /key 0 r1.rhand	Добавить в анимацию правой руки робота ключевой кадр со значениями по умолчанию
• /key 5 r1.rhand	Добавить в анимацию правой руки робота ключевой кадр со значениями по умолчанию
• /key 15 r1.rhand 0 0 0 120 -45 0	Добавить в анимацию правой руки робота ключевой кадр с указанными значениями

Разработка программного обеспечения

- Язык программирования: C++
- Фреймворк: Qt 5.8
- Среда разработки: Visual Studio 2022 Preview
- В процессе разработки были использованы возможности стандартной библиотеки C++ по организации потоков (`std::thread`). Это позволило применить параллельные вычисления в алгоритме генерации изображения. Кроме того, вынесение вычислений в отдельный поток было необходимо для устранения эффекта зависания программного обеспечения во время отрисовки сцены.

Функциональность графического интерфейса пользователя

- Загрузка сцены из файла
- Запуск анимации
- Регулирование скорости просмотра
- Настройка камеры
- Отображение сгенерированного изображения сцены

Загрузить сцену

Пауза/Продолжить

Заново

Скорость просмотра

Позиция камеры

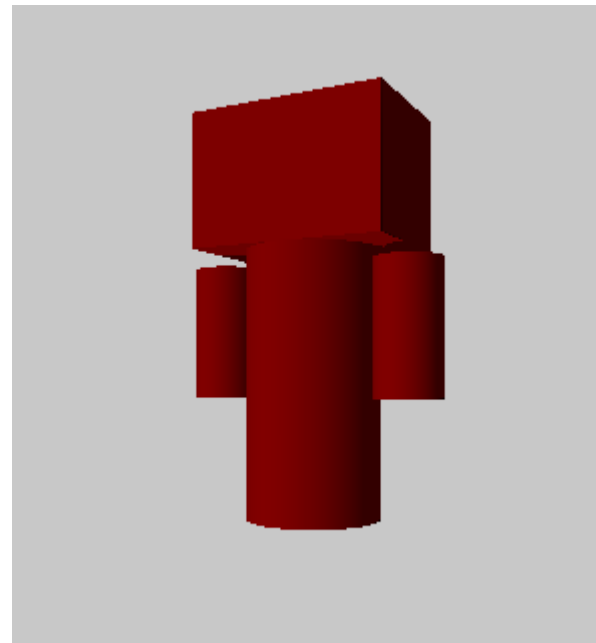
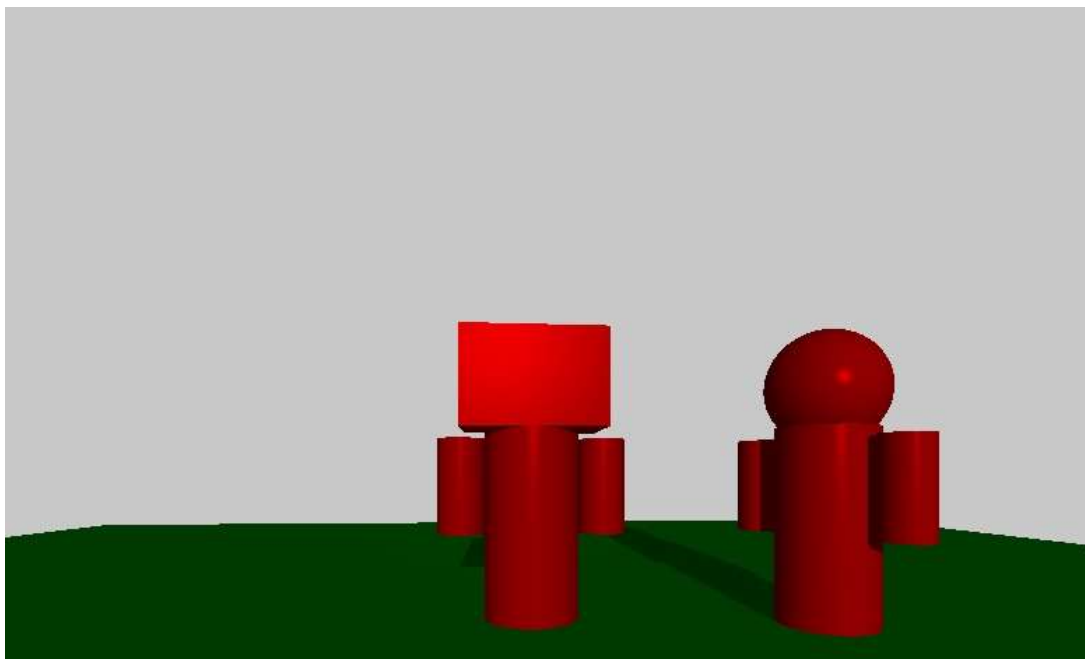
X: 0,00 Y: 0,00 Z: 0,00

Поворот камеры

X: 0,00 Y: 0,00 Z: 0,00

Установить камеру

Пример получаемых изображений



Измерение числа кадров, генерируемых в секунду (FPS)

Количество роботов	1	2	3	4	5
FPS	5.50	4.64	2.55	1.12	0.79

Заключение

Цель достигнута.

Поставленные задачи выполнены:

1. Описан состав сцены и выбрана форма представления объектов сцены.
2. Выбран алгоритм удаления невидимых линий и поверхностей.
3. Выбрана модель освещения.
4. Выбран способ анимации.
5. Разработана модель описания сцены и анимации.
6. Разработан алгоритм получения изображения.
7. Реализован пользовательский интерфейс.
8. Выполнены тестирование и анализ быстродействия программы.

Разработанное программное обеспечение применимо как модуль в проекте Ф-2 для первичного тестирования сценариев человеко-машинной коммуникации в виртуальном режиме, на модели.