Маятник Ньютона

КУРСОВОЙ ПРОЕКТ ПО ДИСЦИПЛИНЕ: «КОМПЬЮТЕРНАЯ ГРАФИКА»

СТУДЕНТ: СУКОЧЕВА АЛИС

РУКОВОДИТЕЛЬ: ВОЛКОВА ЛИЛИЯ ЛЕОНИДОВНА

Цель работы:

Разработать программу, использующую обратную трассирову лучей для визуализации маятника Ньютона с использованием параллельных вычислений.

Задачи курсового проекта:

1. описать предметную область работы;
2. рассмотреть существующие алгоритмы построения реалистичных изображений;
3. выбрать и обосновать выбор реализуемых методов или алгоритмов;
4. подробно изучить выбранный алгоритм;
5. разработать программу на основе одного из существующих алгоритмов;
6. увеличить скорость работы выбранного алгоритма;
7. выбрать и обосновать выбор языка программирования, для решения данной задачи.

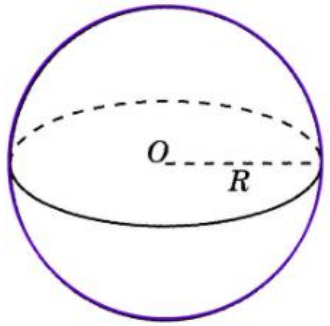
Маятник Ньютона

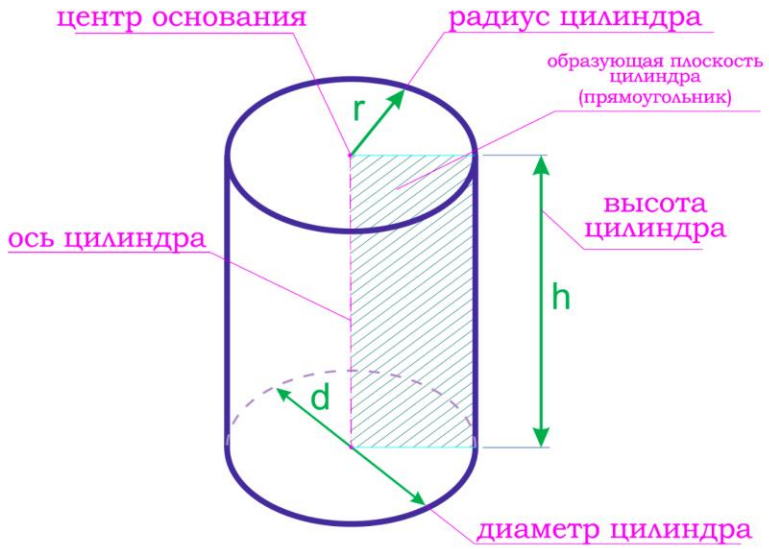
Колыбе́ль Ньютона (маятник Ньютона) — названная в честь Исаака Ньютона механическая система, предназначенная для демонстрации преобразования энергии различных видов друг в друга: кинетической в потенциальную и наоборот. В отсутствие противодействующих сил, таких как трение, система могла бы действовать вечно.

Визуализируемые объекты

Сфера

Задается центром, цветом и радиусом. Нормалю в найденной точке сферы является вектор, испущенный из центра сферы к найденной точке.





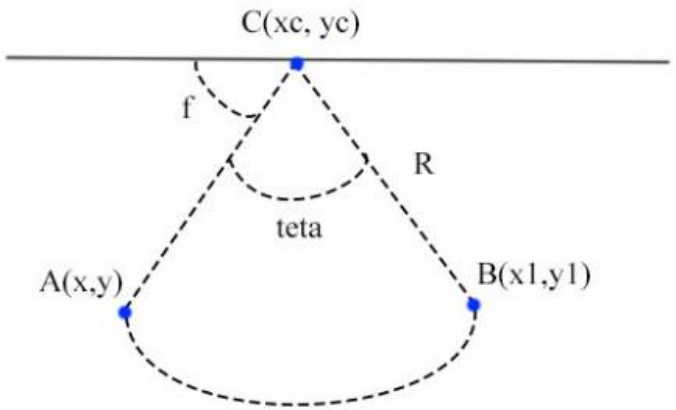
Цилиндр

Задается центром, цветом, радиусом, минимальным и максиальным значением высоты и осью, образующие цилиндра параллельны ей.

Нормаль в найденной точке цилиндра вычисляется аналогично сфере, за исключением того, что одну компонента центра цилинд-ра нужно поднять до точки пересечения.

Движение механической системы

Для движения механической системы были произведены преобразования над центром сферы, а именно поворот. Для поворота использовалась некоторая опорная точка и угол.



Повернув одну сферу на некоторый угол все изображение нужно было повторно визуализировать.

Выбор алгоритма построения реалистичных изображений

Алгоритм Робертса

Алгоритм Варнока

Алгоритм Вейлера-Азертона

Алгоритм художника

Алгоритм Z-буфера

Алгоритм прямой трассировки лучей

Алгоритм обратной трассировки лучей

Обратная трассировка лучей

Из некоторой точки пространства, называемой виртуальным глазом или камерой, через каждый пиксель изображения испускается луч и находится точка пересечения с объектом сцены. Далее из найденной точки пересечения испускаются лучи до каждого источника освещения. Если данные лучи пересекают другие объекты сцены, значит точка пересечения находится в тени, относительно рассматриваемого источника освещения, и освещать ее не нужно. Освещение со всех видимых источников света складываются. Далее, если материал рассматриваемого объекта имеет отражающие или преломляющие свойства, из точки испускается отраженный луч и для него вся процедура трассировки рекурсивно повторяется.

Общий алгоритм визуализации сцены

1. Получение информации о сцене.

2. Отрисовка последовательных изображений с использованием параллельной трассировки лучей и сохранение их в памяти.

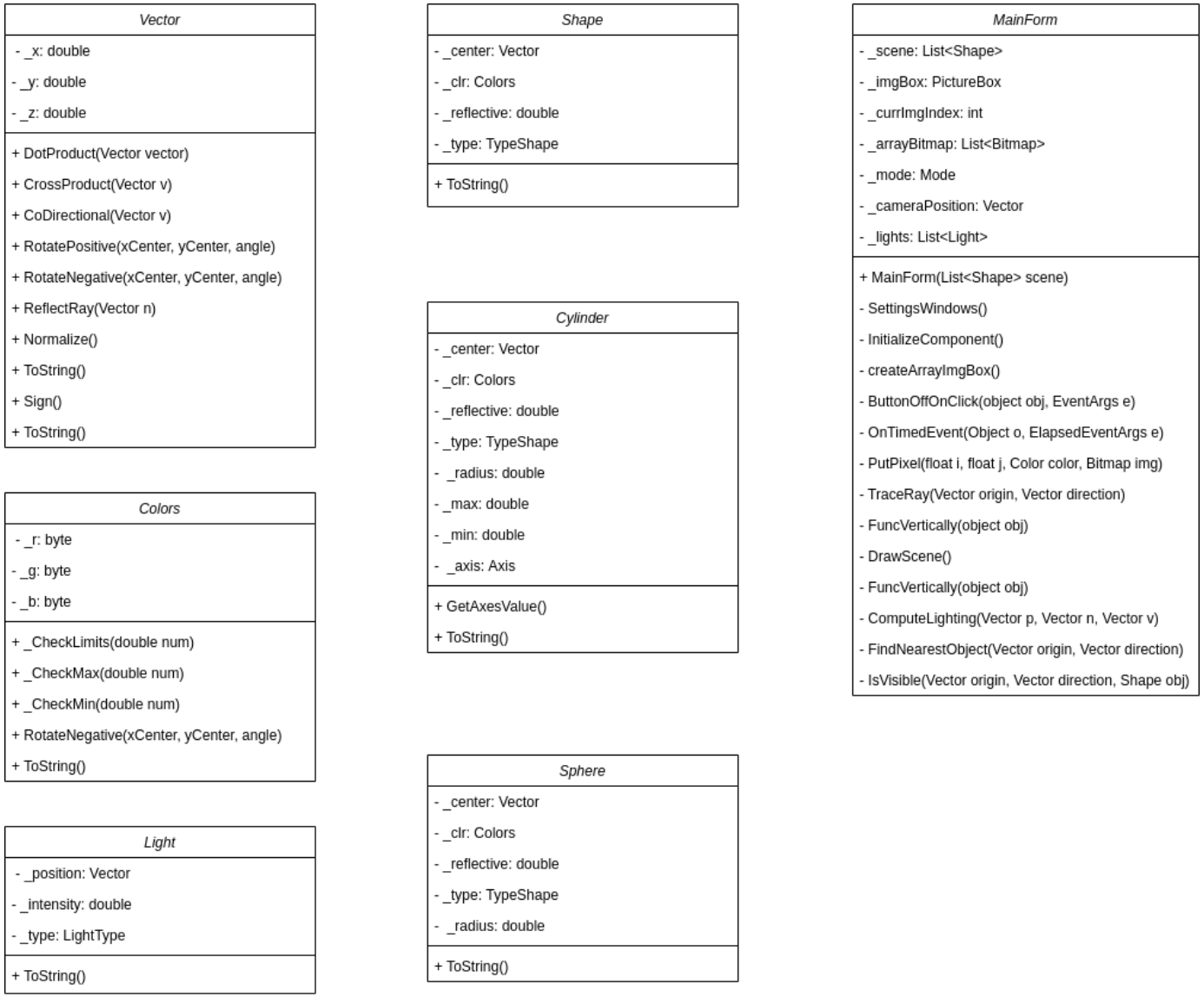
3. Запуск механической системы и поочередная визуализация отрендеренных изображний.

Выбор языка программирования и среды разработки

В качестве языка программирования был выбран язык программирования - C#

В качестве среды разработки был использован Visual Studio Code

Структура и состав классов

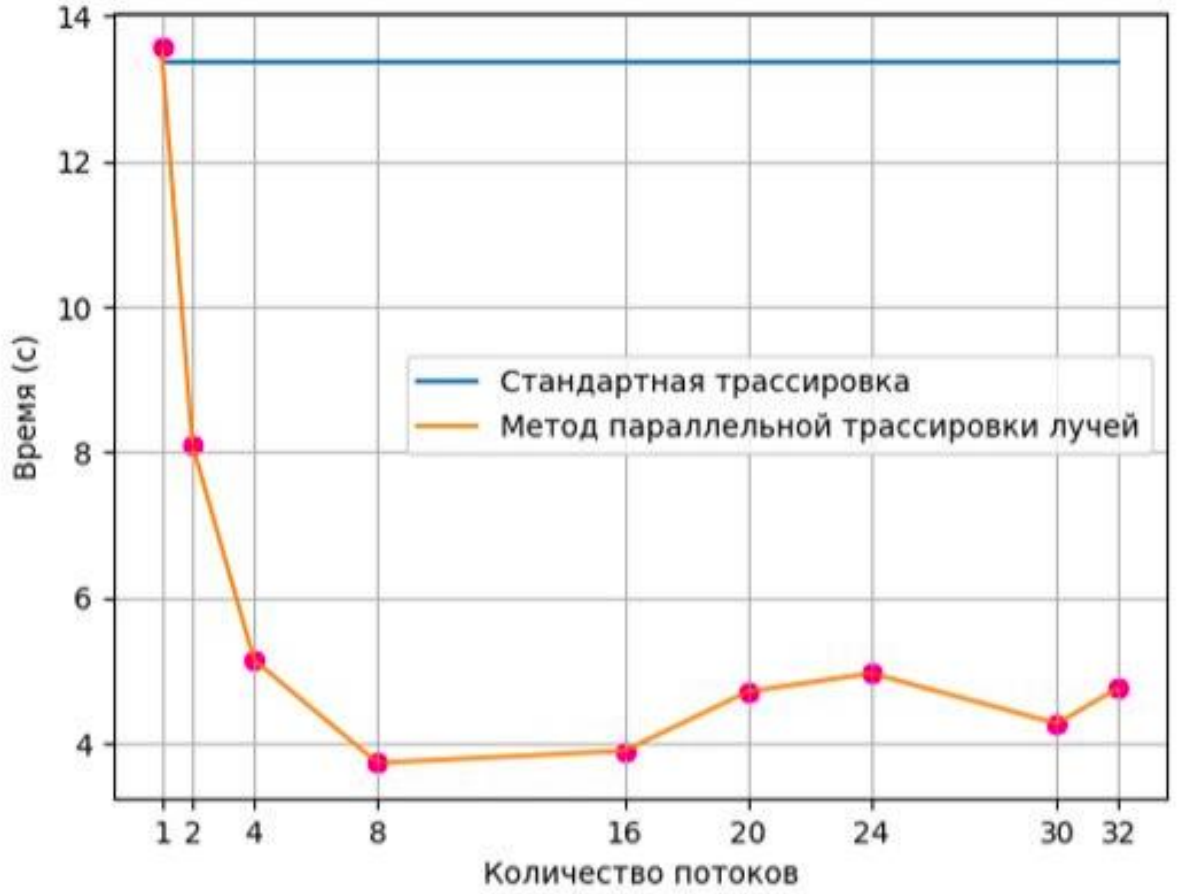


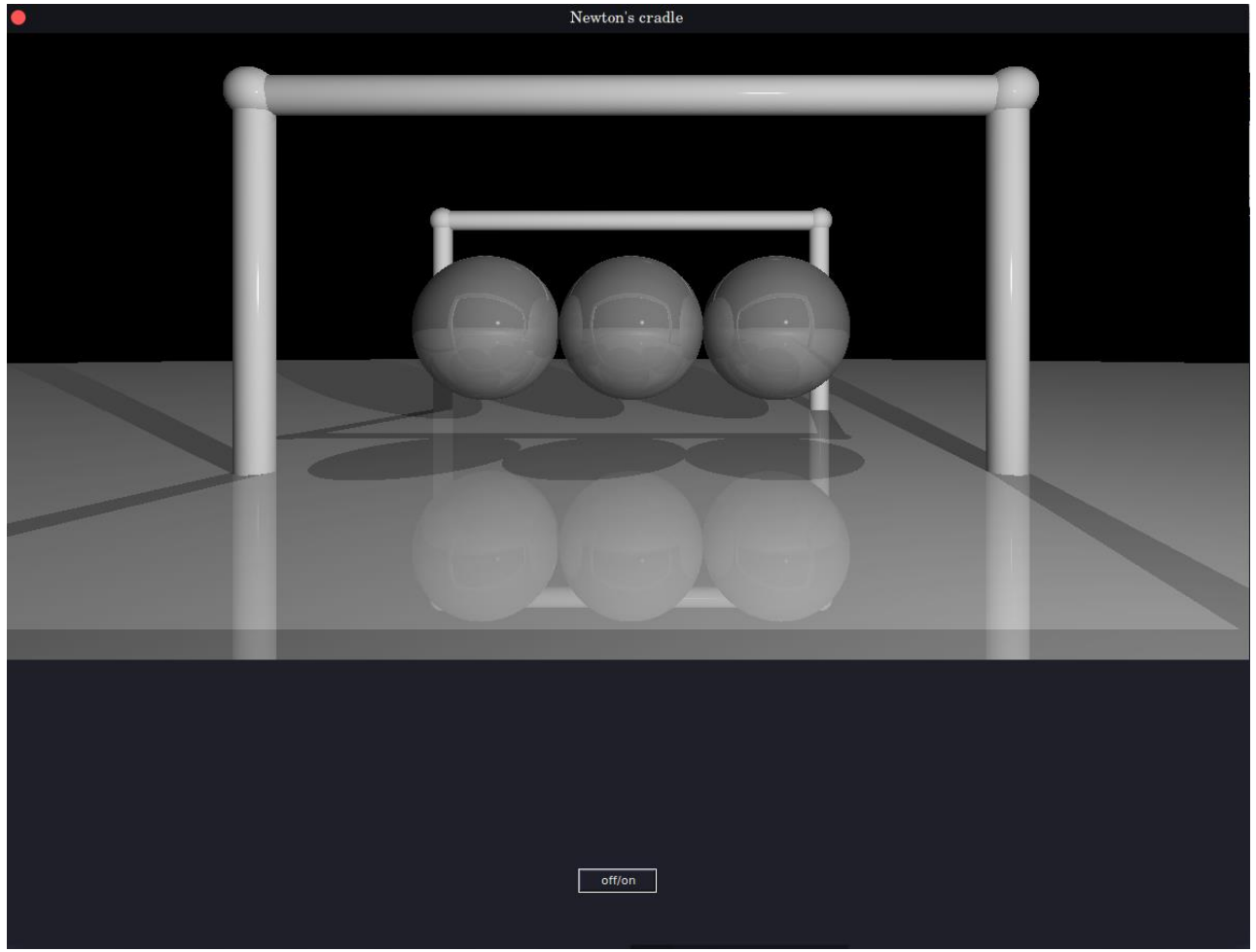
Снижение времени отрисовки сцены.

Поскольку алгоритм обратной трассировки лучей обрабатывает каждый пиксель экрана независимо, можно распараллелить обработку всего экрана, разбив его на некоторые части, что позволит снизить время отрисовки сцены.

Замеры времени

Замеры времени показывают выигрыш параллельной реализации алгоритма трассировки лучей. Рекомендуется число потоков, равное числу логических ядер на Персональном компьютере





Результат работы

Заключение (1/2)

В данном проекте была достигнута цель, а именно: разработана программа, которая успешно выполняет визуализацию маятника Ньютона с использованием трассировки лучей. Также скорость работы алгоритма увеличена благодаря использованию параллельных вычислений.

Заключение (2/2)

В рамках выполнения курсовой работы были решены следующие задачи:

1. описана предметная область работы;
2. рассмотрены существующие алгоритмы построения реалистичных изображений;
3. выбран и обоснован выбор реализуемых алгоритмов;
4. подробно изучен выбранный алгоритм;
5. разработана программа;
6. увеличена скорость работы выбранного алгоритма;
7. выбран и обоснован выбор языка программирования, для решения поставленной задачи.