

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

**Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»**

**Лабораторная работа №4
по курсу «Параллельная обработка данных»**

**Моделирование и визуализация системы N взаимодействующих тел с
использованием технологий OpenGL и CUDA.**

Выполнил: К.М. Воронов
Группа: 8О-407Б
Преподаватель: А.Ю. Морозов

Москва, 2022

Условие

Цель работы. Использование GPU для моделирования и визуализации системы N взаимодействующих тел. Взаимодействие технологий CUDA и OpenGL: vbo + texture. Решение проблемы коллизий множества объектов. Создание простейшей “игры”.

Программное и аппаратное обеспечение

GPU:

- Название NVIDIA GeForce RTX 3050
- Compute capability: 8.6
- Графическая память: 4100456448
- Разделяемая память: 49152
- Константная память: 65536
- Количество регистров на блок: 65536
- Максимальное количество нитей: (1024, 1024, 64)
- Максимальное количество блоков: (2147483647, 65535, 65535)
- Количество мультипроцессоров: 16

Сведения о системе:

- Процессор: Intel Core i5-11400H 2.70GHz
- ОЗУ: 32 ГБ
- SSD 1ТБ

Программное обеспечение:

- OS: Linux Mint 21
- Текстовый редактор: Sublime text
- Компилятор: nvcc

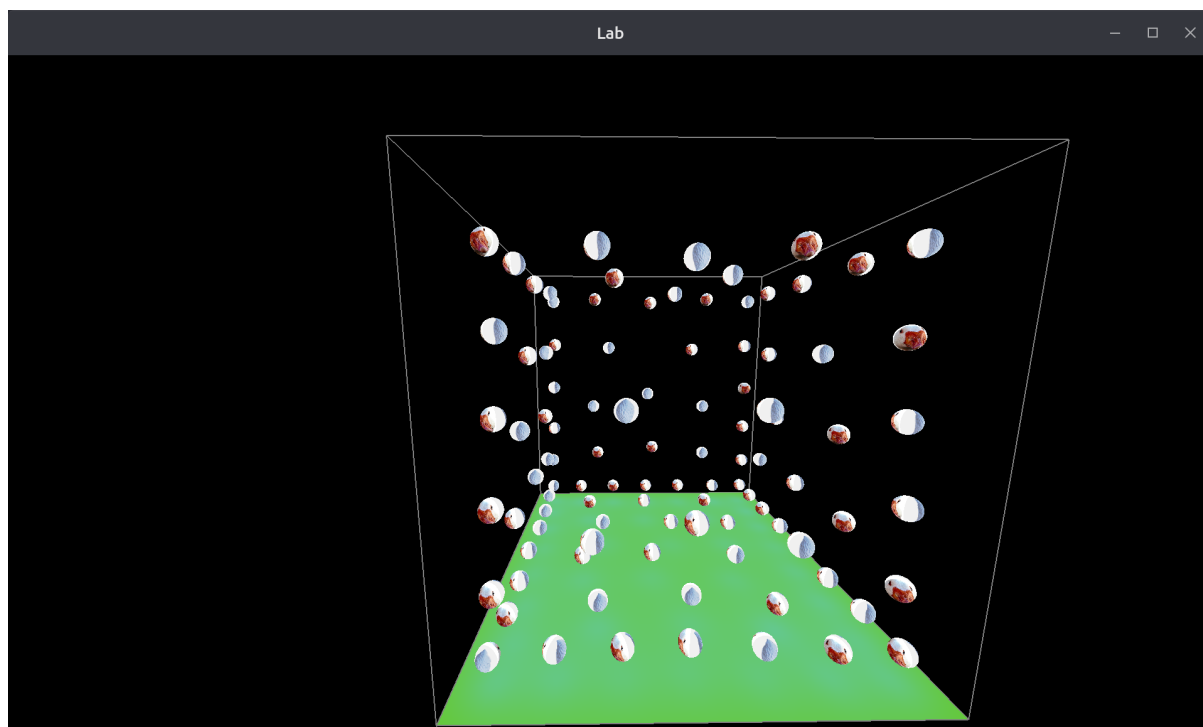
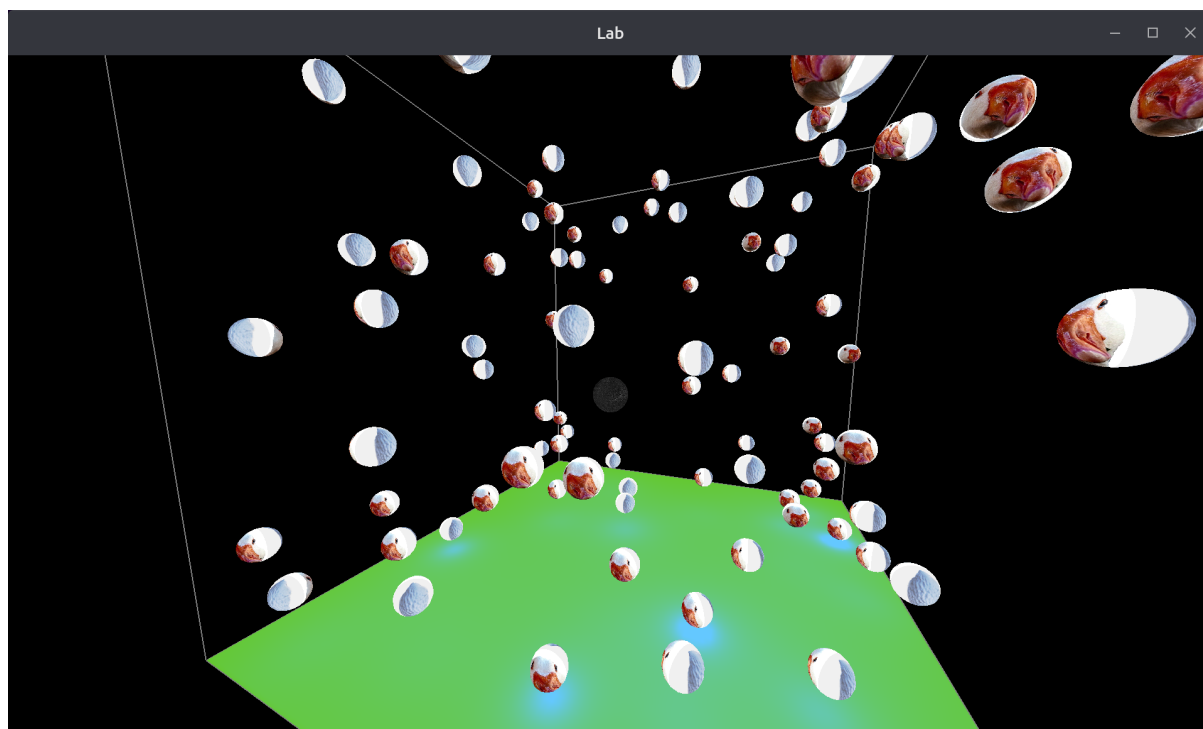
Метод решения

По формулам электростатики рассчитывается взаимодействие объектов с игроком, стенками куба и игроком. Соответственно все имеет заряд, который и определяет силу взаимодействия. При приближении объектов и пули к полу, его цвет тоже меняется по законам электростатики (напряжённость).

Описание программы

Программа имеет 3 ядра. Первое - `kernel_floor` - отвечает за изменение цвета при приближении объектов. Второе - `kernel_move` - за взаимодействие объектов с друг другом, игроком, пулей и стенами. Третье - `kernel_res` - изменение положения объектов. Параметры игры задаются в виде переменных в начале программы. Объекты хранятся в структуре `Item`, которая имеет поля, отвечающие за скорость объекта, его положение, заряд и угол.

Результаты работы



Замеры 100 кадров(сумма по времени)

Количество объектов	Время, мс
100	577.956482 ms
1000	1058.502563 ms
2000	1876.561035 ms
3000	2887.378418 ms

Выводы

Выполнив данную лабораторную, я научился писать программы на OpenGL с использованием CUDA. Лабораторная оказалась увлекательной, так как с результатами можно было “взаимодействовать”. Также интересным этапом работы был выбор текстур для объектов и пули. Обработка данных в реальном времени с использованием видеокарты очень поднимает FPS, так как вычисления происходят быстрее.