

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

**Институт №8 «Информационные технологии и прикладная математика»  
Кафедра 806 «Вычислительная математика и программирование»**

**Лабораторная работа №5  
по курсу «Параллельная обработка данных»**

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

Выполнил: А. В. Куликов

Группа: 8О-408Б

Преподаватели: К.Г. Крашенинников,  
А.Ю. Морозов

Москва, 2020

## 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”.

## Задание.

**Сцена.** Виртуальный куб, у которого отрисовывается только нижняя грань (пол) и ребра. Внутри куба находятся N частиц (текстурированные сферы), которые отталкиваются друг от друга и от стенок куба. Условно предполагается что стенки куба и частицы одноименно заряжены. Для частиц учитывается ускорение свободного падения. Нижняя грань куба закрашивается в соответствии с напряженностью электрического поля создаваемого частицами (строится карта напряженности).

**Игрок.** Камера может перемещаться по пространству без каких-либо ограничений. Управление кнопками и мышкой. При приближении камеры к частицам, они должны “убегать” от неё (предполагается наличия большого одноименного заряда у игрока). По нажатию кнопки мыши, игрок совершает “выстрел” сильно заряженной частицей в направлении взгляда. Эта частица движется равномерно и действует только на другие частицы, а на неё саму никто не влияет.

## Программное и аппаратное обеспечение

Видеокарта	GeForce GTX 1650
Compute capability	7.5
Графическая память	3911 Мб
Разделяемая память	48 Кб
Константная память	64 Кб
Количество регистров на блок	65536
Максимальное кол-во блоков	2147483647*65535*65535
Максимальное кол-во нитей в блоке	1024
Кол-во мультипроцессоров	16
Ядер CUDA	896

Процессор	AMD Ryzen 5 3550H
ОЗУ	8 Гб
ЖД	

Операционная система	Ubuntu 20.04.6 LTS
IDE	VS Code
Компилятор	nvcc V10.1, mpi V3.3.2

## Метод решения

Основная логика программы разделена на 3 независимые части: визуализация, симуляция, обработка пользовательского ввода.

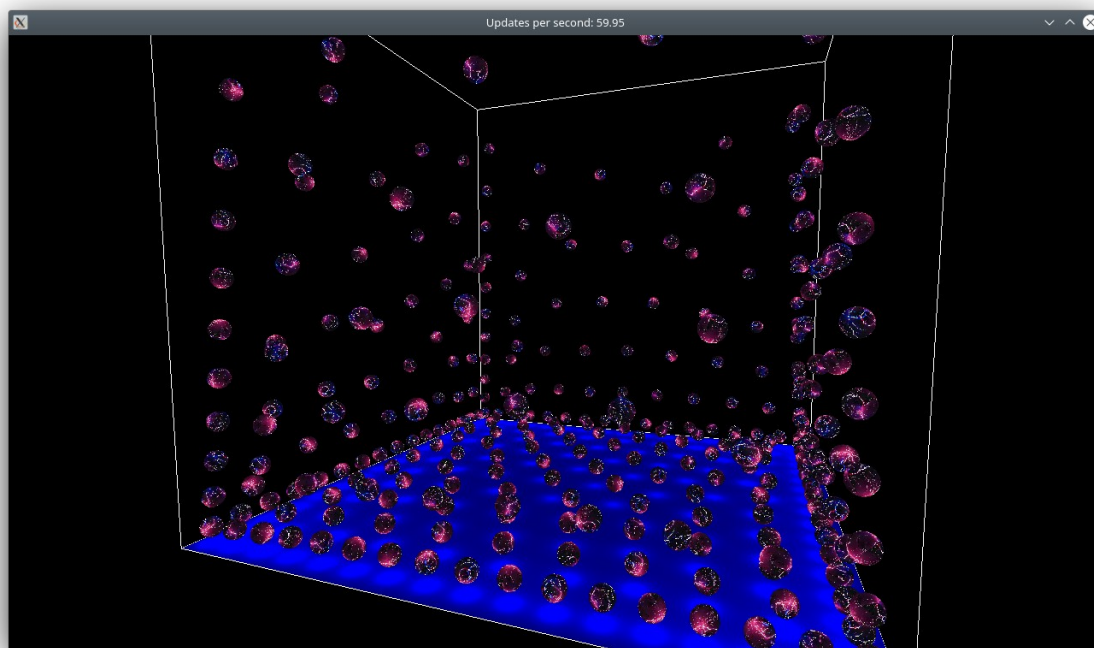
Симуляция отвечает за расчет всех взаимодействий между объектами и определении их координаты в пространстве в текущий момент времени. Визуализация, в свою очередь отвечает за вывод на экран 3D-сцены со всеми частицами, каркасом куба и картой напряженности. Обработка пользовательского ввода в зависимости от того какая клавиша нажата, координаты мыши изменяет параметры, отвечающие за перемещение и поворот камеры в пространстве. Так же при нажатии на левую кнопку производится выстрел заряженной частицей в направлении камеры.

## Описание программы

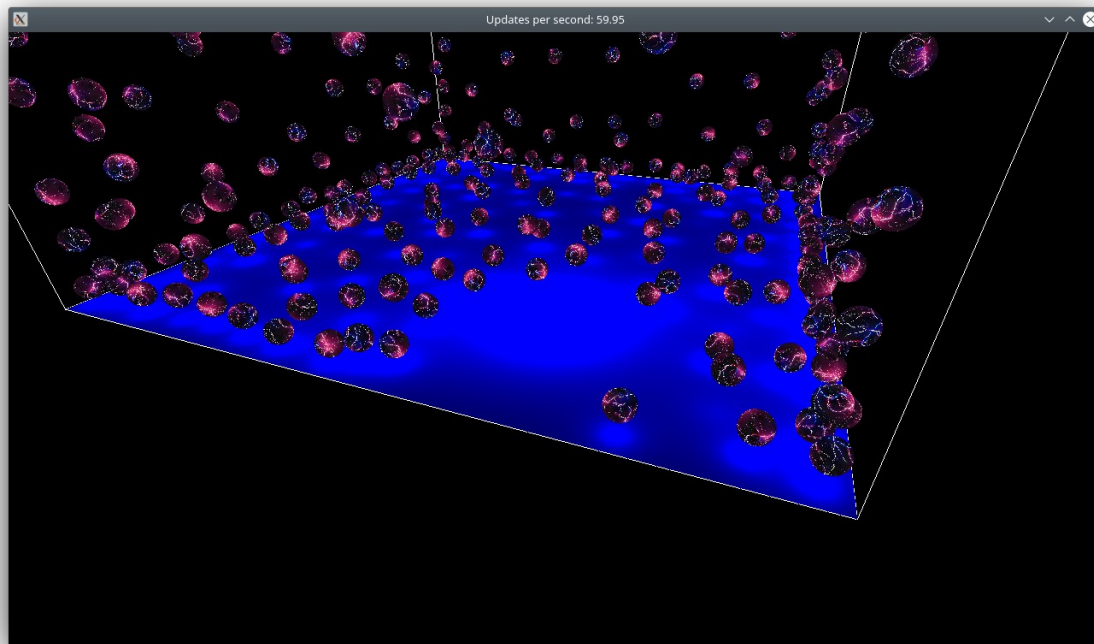
Вся программа реализована в одном файле `main.cu`. Функция `update` отвечает за симуляцию, `display` – за визуализацию. Функции `mouse_movement`, `mouse_clicks`, `keys` отвечают за обработку пользовательского ввода. Расчет всей физики производится на CPU. Расчет карты напряженности производится на GPU с помощью ядра `kernel`.

## Результаты

*Система в состоянии покоя*



*Состояние системы после выстрела заряженной частицей*



## **Выводы**

Данная программа является достаточно простой симуляцией системы из нескольких взаимодействующих тел с возможностью визуализации.

Основные сложности в реализации программы возникли при вспоминании технологии OpenGL, а так же при установлении причины неизвестной ошибки, возникающей при регистрации OpenGL буфера в CUDA.