МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет**

**(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

**Разработка эффективного метода реализации рендеринга**

КУРСОВАЯ РАБОТА

по дисциплине «Программная инженерия»

ЮУрГУ – 09.03.04.20251.308-15822.КР

|  |  |
| --- | --- |
| Нормоконтролер3,  преподаватель кафедры СП  \_\_\_\_\_\_\_\_\_\_\_\_П.А. Манатин  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2025 г. | Научный руководитель:  преподаватель кафедры СП  \_\_\_\_\_\_\_\_\_\_\_\_П.А. Манатин  Автор работы:  студент группы КЭ-343  \_\_\_\_\_\_\_\_\_\_Г.О. Кузьмин  Работа защищена  с оценкой: \_\_\_\_\_\_\_\_\_\_\_  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |

Челябинск, 2025 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет**

**(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

УТВЕРЖДАЮ

Зав. кафедрой СП

\_\_\_\_\_\_\_\_\_\_ Л.Б. Соколинский

15.02.2025 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине «Программная инженерия»

студенту группы КЭ-343

Кузьмину Глебу Олеговичу,

обучающемуся по направлению   
09.03.04 «Программная инженерия»

1. **Тема работы**

Разработка эффективного метода реализации рендеринга.

1. **Срок сдачи студентом законченной работы:** 31.05.2025 г.
2. **Исходные данные к работе2**
3. Ермоленко Андрей Васильевич, and Осипов Константин Сергеевич. "Параллельное программирование в контактных задачах со свободной границей" Вестник Сыктывкарского университета. Серия 1. Математика. Механика. Информатика, no. 23, 2017, pp. 85-91.
4. Fernando Berzal. 2013. Structured parallel programming by Michael McCool, James Reinders & Arch Robison. SIGSOFT Softw. Eng. Notes 38, 2 (March 2013), 35–39..
5. Документация OpenMP. [Электронный ресурс] URL:

https://homepages.math.uic.edu/~jan/mcs572f16/mcs572notes/lec09.html

1. **Перечень подлежащих разработке вопросов3**
2. Анализ предметной области.
3. Проектирование и разработка приложения.
4. Проведение тестирования приложения.
5. **Дата выдачи задания:** 15.02.2025 г.

**Научный руководитель,**

преподаватель кафедры СП П.А. Манатин

**Задание принял к исполнению** Г.О. Кузьмин

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 5](#_Toc200917588)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 7](#_Toc200917589)

[1.1. Предметная область проекта 7](#_Toc200917590)

[1.2. Сравнительный анализ аналогов 7](#_Toc200917591)

[2. ПРОЕКТИРОВАНИЕ 10](#_Toc200917592)

[2.1. Требования к программному продукту 10](#_Toc200917593)

[ЛИТЕРАТУРА 11](#_Toc200917594)

[ПРИЛОЖЕНИЯ 12](#_Toc200917595)

[Приложение А. Спецификация вариантов использования 12](#_Toc200917596)

[Приложение Б. Скриншоты приложения 15](#_Toc200917597)

# ВВЕДЕНИЕ

В данный момент потребность в вычислительной мощности со стороны программного обеспечения к ресурсам системы, на которой она выполняется, очень высока. Причин для этой проблемы много, но основной является нынешняя особенность архитектуры ЭВМ. Современные настольные компьютеры содержат многоядерные процессоры, что позволяет системе выделять процессам отдельные ядра (или потоки) для выполнения множества задач параллельно, а не по очереди. Однако в случае нехватки производительности отдельного ядра (потока), выполняемый процесс будет «тормозить», при том, что утилизация ресурса процессора не полная. Многие разработчики рассчитывают на то, что ресурсов одного логического ядра вполне хватит, однако это не всегда так.

Для решения этой конкретной проблемы используются методы параллельного программирования, позволяющие использовать больше ресурсов для выполнения задачи.

**Актуальность**

Актуальность работы заключается в создании приложения с использованием новейших методов оптимизации и параллелизации приложения. Результат работы может стать показательным для разработчиков.

**Постановка задачи**

Целью курсовой работы является Разработка эффективного метода реализации рендеринга. Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ предметной области и аналогичных решений;
2. Сформулировать функциональные и нефункциональные требования к программному продукту;
3. Реализовать программу для симуляции физической модели.
4. Оптимизировать вычисления доступными методами.
5. Реализовать многопоточное выполнение программы.
6. Сравнить производительность.

**Структура и содержание работы**

Работа состоит из введения, четырех глав, заключения и списка литературы. Объем работы составляет 50 страниц, объем списка литературы – 21 источник.

В первой главе описывается …

Вторая глава посвящена …

В третьей главе …

В приложении А содержится …

Глоссарий является необязательной частью текста работы.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Предметная область проекта

Целью данной работы является разработка программы, направленной на симуляцию физической модели и изучение эффективности применения методов параллельного программирования к ней.

Параллельное программирование – это подход к разработке программ, в котором задачи выполняются на нескольких процессорах или ядрах процессора. Вместо того, чтобы выполнять код последовательно, подход позволяет распределить вычислительную нагрузку между несколькими вычислительными ресурсами, такими как процессоры, для более эффективного выполнения. Параллельное программирование позволяет кратно уменьшить время, затраченное на выполнение поставленной задачи.

Для исследования эффективности применения подхода обычно используются два показателя: эффективность распараллеливания и ускорение распараллеливания. Первый показывает, насколько хорошо применен подход, второй – как количество выделенных ресурсов влияет на эффективность [7].

1.2. Сравнительный анализ аналогов

Работы по теме параллельного программирования в основном связаны с разделением основную задачу программы на несколько подзадач, которые выполняются параллельно друг другу, что в итоге приводит к более быстрому получению результатов.

Большинство проектов с открытым исходным кодом используют общедоступные библиотеки для реализации параллельности в программном продукте, например OpenMP [1] или OpenMPI [2] из-за их простой интеграции в проект (OpenMP поддерживается в компиляторе GCC «из коробки»). В более серьезных проектах используются такие библиотеки как OpenCL или thread для получения доступа к аппаратным ресурсам на низком уровне.

В отдельных случаях разработчик проекта может применить GPGPU вычисления, что подразумевает использование ресурсов графического чипа и видеопамяти для решения задачи. Реализация такого подхода очень трудоемка, так как фактически в персональном настольном компьютере GPU является отдельным устройством и не имеет доступа к памяти программы, а программа не может напрямую взаимодействовать с ресурсами графического чипа.

В основном расчет физической модели используется в компьютерной графике для видео игр и кино, например для просчета объемного тумана или создания динамически изменяющихся объектов окружения. Существует несколько реализаций аналогичных проектов с использованием параллельного программирования, например встроенный в программный пакет Unity [3], предназначенный для разработки видео игр, модуль «particle system». Этот модуль реализован достаточно эффективно, чтобы работать в реальном времени и при этом не перегружать систему. Точных расчетов эффективности реализации «particle system» получить не представляется возможным, так как Unity [3] автоматически определяет, сколько ресурсов выделять как для отдельного кадра, так и для уникальной конфигурации системы пользователя.

Второй пример применения представлен в профильном программном пакете «houdini fx» [5], предназначенном для создания реалистичных объёмных или двумерных компьютерных эффектов. В «houdini fx» используются OpenCL [6] и OpenMP[1] с точной настройкой процесса просчета, программный пакет предоставляет возможность использования нескольких процессоров, видеокарт и специализированных ускорителей.

Основным методом оценки успеха в распараллеливании программы являются две метрики: ускорение и эффективность ускорения. Ускорение вычисляется делением времени выполнения непараллельной программы на время выполнения параллельной программы. Под эффективностью распараллеливания понимается отношение получаемого ускорения к числу задействованных процессов [4].

В результате обзора литературы был выявлен набор инструментальных средств для проведения исследования эффективности распараллеливания задач расчета и отображения симуляции физической модели.

Однопоточный вариант проекта будет реализован на языке программирования C++, с использованием системы сборки Cmake и компилятором GCC.

Наиболее подходящим вариантом для реализации проекта с поддержкой параллельных вычислений является библиотека OpenMP для распараллеливания на CPU.

# 2. ПРОЕКТИРОВАНИЕ

2.1. Требования к программному продукту

Для реализации приложения были выведены следующие требования:

**Функциональные:**

1. Симуляция двумерной физической модели.
2. Поддержка многопоточной симуляции с использованием библиотеки OpenMP.
3. Режим работы без пользователя для сбора информации о модели.
4. Запись результатов выполнения теста производительности программы в файл CSV формата.
5. Поддержка взаимодействия пользователя и симуляции.

**Нефункциональные:**

* Отображение текущей производительности.

Эти требования описывают минимальный функционал программы.

2.2. Варианты использования программы

Исходя из поставленных требований была создана диаграмма использования приложения, она представлена на рисунке 1.

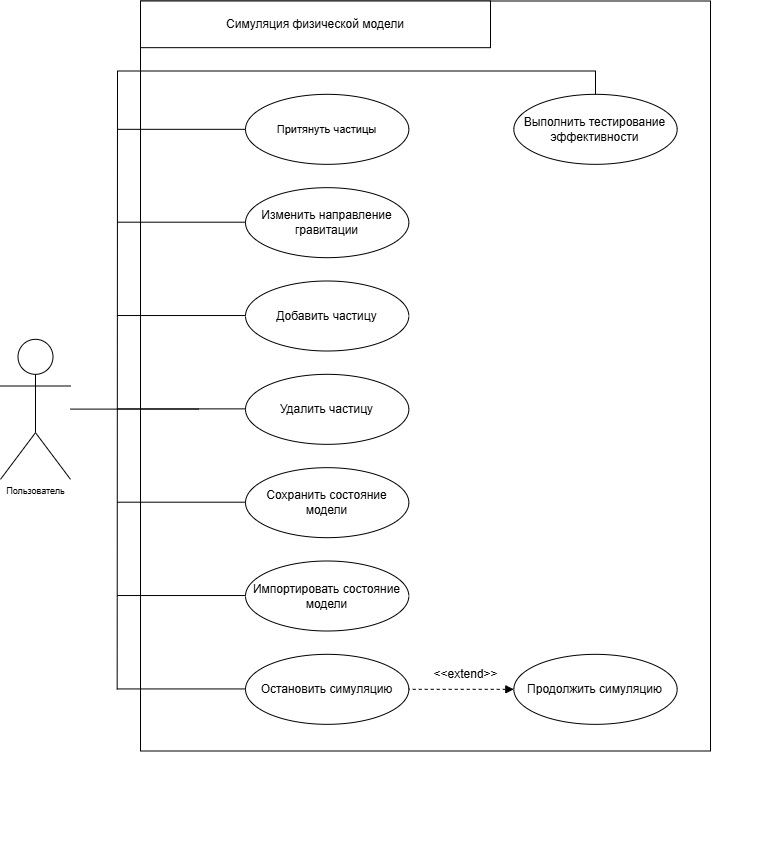


Рисунок 1 – Диаграмма вариантов использования

Пользователь является основным актером, который будет пользоваться системой, однако во время тестирования производительности пользователь не может взаимодействовать с ней.

Спецификация вариантов использования находится в приложении А.

**Краткое описание всех вариантов использования приложения:**

Притянуть частицы: Пользователь создает локальную временную точку притяжения.

Изменить направление гравитации: Пользователь меняет направление гравитации.

Добавить частицу: Пользователь добавляет новую частицу на место курсора.

Удалить частицу: Пользователь убирает существующую частицу на месте курсора из симуляции.

Сохранить состояние модели: Пользователь сохраняет состояние модели в файл на устройстве.

Импортировать состояние модели: Пользователь импортирует сохраненную модель с устройства.

Остановить симуляцию: Пользователь приостанавливает просчет физики в системе.

Продолжить симуляцию: Пользователь возобновляет просчет физики в системе.

Выполнить тестирование эффективности: Система выполняет симуляцию по заранее определенным параметрам.

Сохранить результаты в файл: Система сохраняет результаты тестирования в файл.

# ЛИТЕРАТУРА

1. Богданов Д.В. Оптимальный способ хранения и обработки древовидных структур в базах данных. // Программные продукты и системы, 2009. – № 1. – С. 140–142.
2. ГОСТ 33245-2015 (ISO/IEC TR 29163-1:2009) Информационные технологии (ИТ). Эталонная модель распределенного объекта контента (SCORM®) 2004 3-я редакция. Часть 1. Обзор. Версия 1.1. [Электронный ресурс] URL: http://docs.cntd.ru/document/1200127254 (дата обращения: 10.05.2020 г.).
3. Жигальская Н.С. Модель вариантов использования универсальной среды электронного обучения UniCST. // Инновационные технологии обучения: проблемы и перспективы: Материалы всерос. науч.-метод. конф. (29-30 марта 2008 г., Липецк) – Липецк: Изд-во ЛГПУ, 2008. – С. 204–207.
4. Маликов А.В. Ориентированные графы в реляционных базах данных. // Доклады ТУСУР, 2008. – № 2(18). – Ч. 2. – С. 100–104.
5. Desktop Browser Market Share Worldwide | StatCounter Global Stats. [Электронный ресурс] URL: http://gs.statcounter.com/browser-market-share/desktop/worldwide (дата обращения: 24.05.2020 г.).
6. Dracula Graph Library. [Электронный ресурс] URL: https://www.graphdracula.net (дата обращения: 16.03.2020 г.).
7. Ivanova O.N., Silkina N.S. Competence-Oriented Model of Representation of Educational Content. // Proceedings of the 40th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO'2017, Opatija, Croatia, May 22–26, 2017. IEEE, 2017. – 791–794 pp.
8. Юнг К.Г. Психологические типы / под ред. Зеленского В. – СПб: Азбука, 2001. – 736 с.

# ПРИЛОЖЕНИЯ

Приложение А. Спецификация вариантов использования

Спецификация вариантов использования (ВИ) приложения приведена в таблицах 1–4.

Таблица 1 – Спецификация ВИ «Выполнить тестирование эффективности»

|  |
| --- |
| ***UseCase:* *Выполнить тестирование эффективности*** |
| *ID:* 1 |
| *Аннотация:* Пользователь дает команду системе на выполнение тестирования производительности. |
| *Главные актеры:* Пользователь. |
| *Второстепенные актеры:* Нет. |
| *Предусловия:* Нет. |
| *Основной поток:*   1. Пользователь нажимает на кнопку «B». 2. Система сбрасывает текущее состояние модели. 3. Система выполняет тестирование производительности. 4. Система записывает результаты тестирования в файл на устройстве пользователя. |
| *Постусловия:* Началось тестирование эффективности. |
| *Альтернативные потоки:* Нет. |

Таблица 2 – Спецификация ВИ «Притянуть частицы»

|  |
| --- |
| ***UseCase:* *Притянуть частицы*** |
| *ID:* 2 |
| *Аннотация:* Пользователь создает локальную точку притяжения. |
| *Главные актеры:* Пользователь. |
| *Второстепенные актеры:* Нет. |
| *Предусловия:* Нет. |
| *Основной поток:*   1. Пользователь перемещает курсор на место, куда хочет установить точку притяжения. 2. Пользователь зажимает среднюю кнопку мыши. 3. Система создает точку притяжения на месте курсора. 4. Пользователь отпускает среднюю кнопку мыши. 5. Система удаляет точку притяжения. |
| *Постусловия:* Точка притяжения была создана и удалена. |
| *Альтернативные потоки:* Нет. |

Продолжение приложения А

Таблица 3 – Спецификация ВИ «Изменить направление гравитации»

|  |
| --- |
| ***UseCase:* *Изменить направление гравитации*** |
| *ID:* 3 |
| *Аннотация:* Пользователь изменяет направление гравитации. |
| *Главные актеры:* Пользователь. |
| *Второстепенные актеры:* Нет. |
| *Предусловия:* Нет. |
| *Основной поток:*   1. Пользователь нажимает стрелку на клавиатуре. 2. Система изменяет направление гравитации в соответствии с направлением нажатой стрелки. |
| *Постусловия:* Направление гравитации изменено. |
| *Альтернативные потоки:* Нет. |

Таблица 4 – Спецификация ВИ «Добавить частицу»

|  |
| --- |
| ***UseCase:* *Добавить частицу*** |
| *ID:* 4 |
| *Аннотация:* Пользователь добавляет новую частицу в систему. |
| *Главные актеры:* Пользователь. |
| *Второстепенные актеры:* Нет. |
| *Предусловия:* Нет. |
| *Основной поток:*   1. Пользователь перемещает курсор желаемое место появления новой частицы. 2. Пользователь нажимает левую кнопку мыши. 3. Система создает новую частицу на месте курсора. |
| *Постусловия:* Создана новая частица. |
| *Альтернативные потоки:* Нет. |

Продолжение приложения А

Таблица 5 – Спецификация ВИ «Удалить частицу»

|  |
| --- |
| ***UseCase:* *Удалить частицу*** |
| *ID:* 5 |
| *Аннотация:* Пользователь удаляет частицу из системы. |
| *Главные актеры:* Пользователь. |
| *Второстепенные актеры:* Нет. |
| *Предусловия:* Нет. |
| *Основной поток:*   1. Пользователь перемещает курсор на место существующей точки. 2. Пользователь нажимает правую кнопку мыши. 3. Система перемещает точку притяжения на новое место. |
| *Постусловия:* Точка притяжения перемещена. |
| *Альтернативные потоки:* Нет. |

Таблица 6 – Спецификация ВИ «Сохранить состояние модели»

|  |
| --- |
| ***UseCase:* *Сохранить состояние модели*** |
| *ID:* 6 |
| *Аннотация:* Пользователь дает команду системе о сохранении состояния модели. |
| *Главные актеры:* Пользователь. |
| *Второстепенные актеры:* Нет. |
| *Предусловия:* Нет. |
| *Основной поток:*   1. Пользователь нажимает кнопку «S» на клавиатуре. 2. Система сохраняет состояние в файл. |
| *Постусловия:* Файл состояния системы сохранен на устройстве пользователя. |
| *Альтернативные потоки:* Нет. |

Продолжение приложения А

Таблица 7 – Спецификация ВИ «Импортировать состояние модели»

|  |
| --- |
| ***UseCase:* *Импортировать состояние модели*** |
| *ID: 7* |
| *Аннотация:* Пользователь дает команду системе об импорте сохраненной модели с устройства. |
| *Главные актеры:* Пользователь. |
| *Второстепенные актеры:* Нет. |
| *Предусловия:* Нет. |
| *Основной поток:*   1. Пользователь нажимает кнопку «L» на клавиатуре. 2. Система загружает состояние модели из файла. |
| *Постусловия:* Файл состояния системы успешно импортирован. |
| *Альтернативные потоки:* Нет. |

Таблица 8 – Спецификация ВИ «Остановить симуляцию»

|  |
| --- |
| ***UseCase:* *Остановить симуляцию*** |
| *ID:* 8 |
| *Аннотация:* Пользователь останавливает обновление модели. |
| *Главные актеры:* Пользователь. |
| *Второстепенные актеры:* Нет. |
| *Предусловия:* Нет. |
| *Основной поток:*   1. Пользователь нажимает кнопку «space» на клавиатуре. 2. Система прекращает обновление физической модели. |
| *Постусловия:* Обновление модели приостановлено. |
| *Альтернативные потоки:* Нет. |

Окончание приложения А

Таблица 9 – Спецификация ВИ «Продолжить симуляцию»

|  |
| --- |
| ***UseCase:* *Продолжить симуляцию*** |
| *ID:* 9 |
| *Аннотация:* Пользователь возобновляет обновление модели. |
| *Главные актеры:* Пользователь. |
| *Второстепенные актеры:* Нет. |
| *Предусловия:* Модель должна быть приостановлена. |
| *Основной поток:*   1. Пользователь нажимает кнопку «space» на клавиатуре. 2. Система возобновляет обновление физической модели. |
| *Постусловия:* Обновление модели возобновлено. |
| *Альтернативные потоки:* Нет. |

# Приложение Б. Скриншоты приложения

Скриншоты разработанного приложения приведены на рисунках 1–10.

Нумерация таблиц и рисунков в приложении начинается с единицы и является сквозной по всем приложениям.

Если приложение одно, то оно не нумеруется. Если приложений несколько, то в содержание выносится отдельная позиция «ПРИЛОЖЕНИЯ» (заголовок первого уровня), приложения нумеруются буквами А, Б, В, Г и т.д. (заголовки второго уровня).

В тексте в приложениях применяются те же правила оформления, что и к основному тексту курсовой работы.