
Реализация и тестирование системы

Система симуляции частиц представляет собой приложение на C++, которое моделирует поведение частиц в двумерном пространстве с учетом физических взаимодействий, таких как гравитация, столкновения и силы притяжения. Для визуализации используется библиотека SFML, а для параллельной обработки вычислений — OpenMP.

1. Реализация компонентов системы (с примерами исходного кода)

Управление частицами

Основой системы является структура «Particle», определенная в файле `particle.hpp`, которая описывает свойства частицы: позиция (`sf::Vector2f`) задает координаты, скорость определяет движение, ускорение изменяется под действием сил, радиус используется для столкновений и рендеринга, а цвет (`sf::Color`) обеспечивает визуальное различие. Частицы создаются динамически через функцию «`createParticle`» и хранятся в векторе «`std::vector`» в классе «Engine», с максимальным количеством 1000, заданным параметром «`g_maximumParticles`» в файле «`parameters.hpp`». Цвет частиц генерируется на основе номера кадра, что улучшает визуальную идентификацию.

Физическая симуляция, реализованная в файле «`updateOpenMP.cpp`», управляет движением и взаимодействием частиц. Функция «`applyGravity`» добавляет к ускорению каждой частицы вектор гравитации (GRAVITY), направление и величина которого могут изменяться пользователем с помощью клавиш стрелок. По умолчанию гравитация направлена вниз, обеспечивая естественное падение частиц, но пользователь может моделировать различные

сценарии, такие как невесомость или боковые силы. Функция «checkCollisions» обрабатывает столкновения между частицами, используя класс «SpatialGrid» для оптимизации вычислений. «SpatialGrid» делит пространство на ячейки, ограничивая проверку столкновений соседними частицами, что снижает вычислительную сложность с $O(n^2)$ до $O(n)$, особенно важно при максимальном количестве частиц. При обнаружении столкновения вычисляется импульс на основе относительной скорости и нормали столкновения, а позиции частиц корректируются для устранения перекрытия.

Параллельная обработка с помощью OpenMP, реализованная через директиву «#pragma omp parallel for», ускоряет этот процесс. Функция «applyAttraction» активирует силу притяжения при нажатии средней кнопки мыши, притягивая частицы в заданном радиусе к курсору с силой, уменьшающейся с расстоянием, создавая эффект, подобный гравитационному притяжению. Функция «checkBorders» ограничивает движение частиц рамками окна, инвертируя скорость при достижении границы и уменьшая ее за счет коэффициента затухания, что обеспечивает реалистичный отскок.

Тестирование производительности

Класс «Benchmark» измеряет производительность системы, записывая время обработки физики и рендеринга за 1000 кадров. Средние значения FPS, времени физики и рендеринга сохраняются в CSV-файл для последующего анализа. Класс позволяет сравнить производительность при использовании OpenMP (параллельная обработка) и без него, демонстрируя эффективность параллелизации.

Рендер

Функция «render» в классе «Engine» отображает частицы как круги, используя их радиус и цвет. Экран очищается перед каждым кадром, после чего рисуются все частицы. В верхнем левом углу окна отображаются метрики

производительности: текущий FPS, количество частиц, время обработки физики и рендеринга. Это позволяет пользователю отслеживать производительность системы в реальном времени.

2. Реализация интерфейса системы (пользовательский и/или программный интерфейс)

Система реализована как настольное приложение с использованием SFML, без веб-интерфейса. Пользователь взаимодействует с симуляцией через окно приложения, созданное классом «Engine». Окно приложения представлено на рисунке 1.

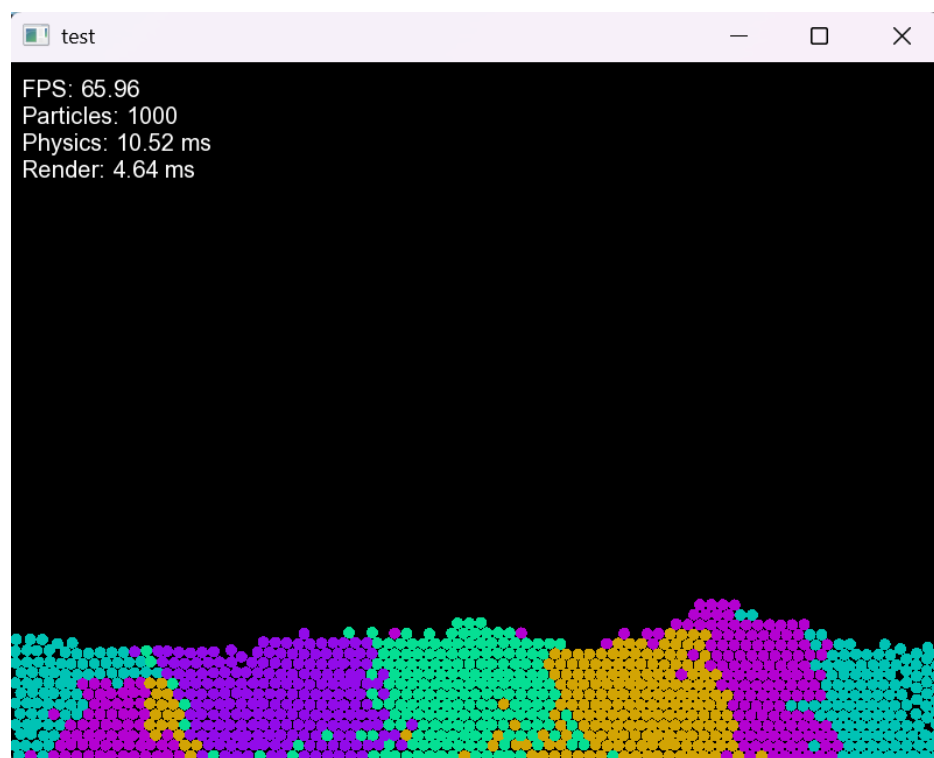


Рис. 1 - Окно приложения

Класс «Engine» управляет визуализацией и пользовательским вводом. Функция «render» отображает частицы как круги, используя их радиус и цвет, очищая экран перед каждым кадром. В верхнем левом углу окна отображаются метрики производительности: FPS, количество частиц, время физики и рендеринга, что позволяет отслеживать производительность в реальном времени. Функция «handleEvents» обрабатывает события

клавиатуры и мыши: пробел приостанавливает симуляцию, клавиши S и L сохраняют и загружают состояние частиц, стрелки изменяют гравитацию, клавиша B запускает бенчмарк через класс «Benchmark», левый клик добавляет частицы, правый клик удаляет, а средний клик активирует притяжение. Функции «saveState» и «loadState» обеспечивают сохранение и загрузку состояния частиц в бинарный файл.

3. Тестирование системы

Тестирование системы проводилось для проверки функциональности, производительности и удобства пользовательского интерфейса, охватывая основные аспекты, включая создание частиц, физические взаимодействия, управление и производительность. Тестовые случаи подтвердили корректность работы системы: запуск симуляции приводит к созданию частиц до достижения лимита (1000); нажатие клавиш стрелок изменяет направление гравитации, и частицы начинают двигаться в новом направлении; при столкновении частиц они отскакивают с учетом физики без перекрытия; при достижении границ окна частицы отскакивают с уменьшением скорости за счет затухания; нажатие средней кнопки мыши притягивает частицы к курсору в заданном радиусе; клавиша Space приостанавливает и возобновляет симуляцию; клавиши S и L корректно сохраняют и загружают состояние частиц; левый и правый клики мыши добавляют и удаляют частицы в позиции курсора; клавиша B запускает бенчмарк, который выполняется 1000 кадров и сохраняет результаты в CSV-файл; метрики производительности (FPS, количество частиц, время физики и рендеринга) корректно отображаются в верхнем левом углу окна.

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
1	Создание частиц	1. Запустить программу. 2. Наблюдать автоматическое создание частиц.	Частицы создаются до тех пор, пока их не будет 1000.	Да
2	Изменение гравитации	1. Запустить программу. 2. Нажать одну из четырех стрелок на клавиатуре.	Частицы начинают двигаться в другом направлении гравитации.	Да

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
3	Столкновение частиц	1. Запустить программу. 2. Дать частицам столкнуться.	Частицы отскакивают друг от друга, не перекрывают друг друга.	Да
4	Столкновение с границами	1. Запустить программу. 2. Дать частицам столкнуться с границей.	Частицы отталкиваются от стенок окна.	Да
5	Сила притяжения	1. Запустить программу. 2. Дождаться появления нескольких частиц. 3. Зажать среднюю кнопку мыши. 4. Переместить курсор.	Частицы притягиваются к курсору.	Да
6	Пауза симуляции	1. Запустить программу. 2. Нажать клавишу «space» на клавиатуре.	Симуляция останавливается.	Да
7	Сохранение и загрузка состояния	1. Запустить программу. 2. Нажать кнопку «S» на клавиатуре. 3. Нажать кнопку «L» на клавиатуре.	Система записывает состояние модели в файл при нажатии «S» и загружает при нажатии «L».	Да
8	Добавление частицы	1. Запустить программу. 2. Нажать левую кнопку мыши.	Система создает новую частицу на месте курсора.	Да
9	Тестирование производительности	1. Запустить программу. 2. Нажать кнопку «B» на клавиатуре.	Система очищает симуляцию, отрисовывает тысячу кадров и сохраняет результаты теста в файл.	Да
10	Отображение метрик производительности	1. Запустить программу. 2. Наблюдать левый верхний угол окна.	Отображаются корректные значения производительности.	Да