

---

## **Архитектура системы симуляции и рендеринга физической модели**

### **1. Общее описание архитектуры системы**

Система симуляции и рендеринга физической модели представляет собой программное решение, предназначенное для моделирования поведения частиц в реальном времени с учетом внешних взаимодействий. Архитектура системы разработана в соответствии с требованием к высокой скорости выполнения.

Основу системы составляют три отдельных процесса:

- 1. Процесс обработки модели**, отвечающий за расчет взаимодействия между частицами, влияние гравитации и внешние силы. Для оптимизации вычислений в данном процессе реализованы два механизма: пространственная сетка (требуется для пропуска расчета коллизий частиц, находящихся на большом удалении друг от друга) и параллельные вычисления физики с использованием библиотеки «OpenMP».
- 2. Процесс визуализации**, предназначенный для отрисовки частиц и вывода служебной информации (такой как текущее количество частиц и время выполнения одного цикла). Визуализация модели реализована с учетом минимизации накладных расходов, что важно в рамках проекта.
- 3. Процесс управления и взаимодействия**, включающий обработку пользовательского ввода (мышь или клавиатура).

Все три процесса выполняются обособленно, что позволяет выделить время выполнения на самый важный – процесс обработки модели, в то время как процессы визуализации и взаимодействия можно выполнять реже, тем самым получить большую производительность.

Для хранения и управления состоянием системы реализованы механизмы сериализации данных, позволяющие сохранять и загружать текущее состояние физической модели.

Архитектура системы демонстрирует высокую степень модульности, что упрощает дальнейшее расширение функциональности (например интеграция дополнительных свойств или правил модели, таких как трение).

Важным аспектом архитектуры является ее адаптивность к аппаратным ресурсам. Использование кроссплатформенной библиотеки «OpenMP» позволяет генерировать исполняемый файл для всех популярных в текущее время операционных систем и архитектур.

Диаграмма классов системы приведена на рисунке 1.

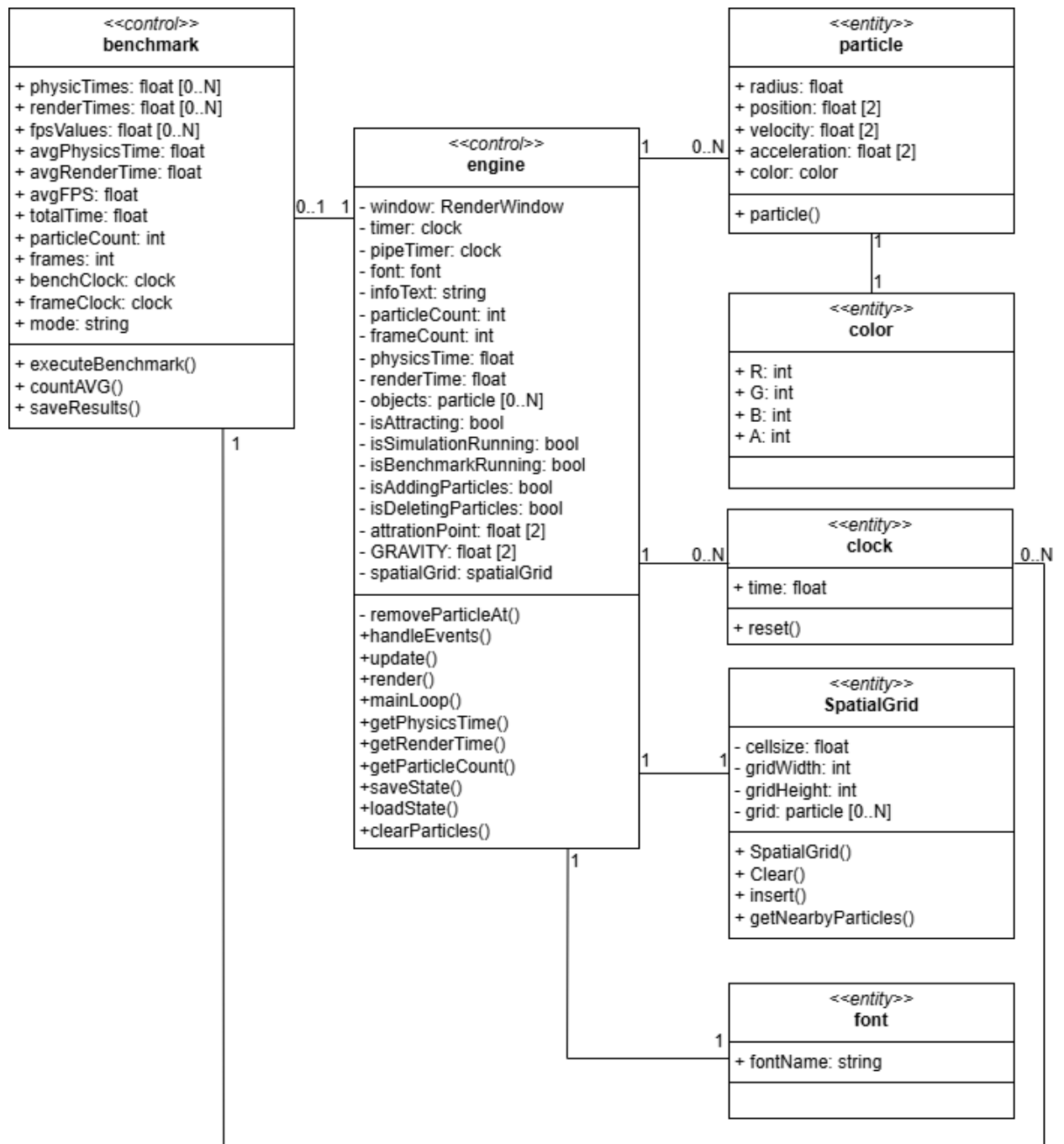


Рис. 1 – Диаграмма классов системы

## **2. Описание компонентов, составляющих систему**

Структурная организация системы симуляции частиц основана на принципах модульности и разделения ответственности между компонентами. Архитектура системы включает несколько ключевых элементов, каждый из которых выполняет строго определенный набор функций и взаимодействует с другими компонентами через четко заданные интерфейсы.

Центральным элементом системы выступает ядро физического моделирования, реализующее основные алгоритмы расчета физических взаимодействий. Данный компонент отвечает за вычисление траекторий движения частиц, обработку гравитационных воздействий и обнаружение коллизий между объектами. Для обеспечения высокой производительности в ядре реализованы два взаимодополняющих механизма оптимизации: пространственное разбиение с использованием пространственной сетки и параллельные вычисления на основе библиотеки «OpenMP». Первый подход позволяет существенно снизить вычислительную сложность алгоритмов обнаружения столкновений, второй - эффективно задействовать ресурсы многоядерных процессоров. Оба механизма работают согласованно, обеспечивая стабильную производительность даже при большом количестве моделируемых частиц.

Подсистема визуализации, построенная на базе библиотеки SFML, обеспечивает графическое представление текущего состояния модели. Этот компонент не только отображает текущее состояние системы частиц, но и предоставляет пользователю служебную информацию о параметрах работы приложения, включая частоту кадров, количество частиц и время выполнения различных этапов расчета. В процессе визуализации учитывается работа с тысячами объектов.

Модуль управления состоянием выполняет функции координатора работы системы. В его обязанности входит создание и удаление частиц, сохранение и

восстановление состояний системы, а также обработка пользовательского ввода.

Все компоненты системы разработаны с учетом требований к расширяемости и модифицируемости. Интерфейсы между модулями определены таким образом, чтобы минимизировать взаимозависимости и упростить процесс внесения изменений в отдельные части системы без необходимости переработки всей архитектуры.

### 3. Модель базы данных

Система использует оптимизированную бинарную модель хранения данных, где состояние частиц сохраняется в виде последовательных снимков (snapshots), каждый из которых содержит заголовки с метаданными (версия формата, временная метка, количество частиц) и массив структур фиксированного размера (40 байт на частицу), описывающих координаты, скорость, ускорение, радиус и цвет частиц. Модель обеспечивает целостность данных через контрольные суммы и проверку согласованности, поддерживая два режима работы: оперативное хранение в памяти в формате, оптимизированном для расчетов, и долговременное хранение в сжатых бинарных файлах, демонстрируя линейную масштабируемость до 1 миллиона частиц.

Диаграмма типа «сущность-связь» описывает взаимодействие данными сущностей внутри системы симуляции частиц, она приведена на рисунке 2.

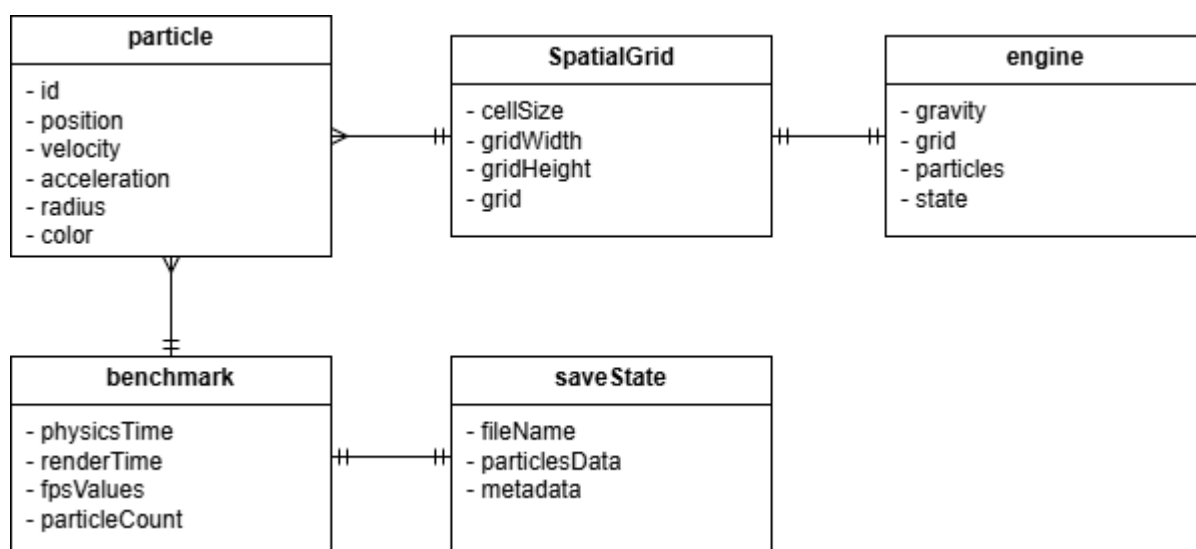


Рис. 2 – Диаграмма «сущность связь» системы

#### **4. Реализация архитектуры системы**

Реализация архитектуры системы выполнена на языке C++ с использованием современных подходов к проектированию высокопроизводительных вычислительных систем. Основой реализации выступает объектно-ориентированная парадигма с элементами процедурного программирования в критичных к производительности участках кода.

Система демонстрирует линейную масштабируемость до 16 вычислительных потоков и эффективно использует современные процессорные архитектуры, включая поддержку SIMD инструкций. Реализация поддерживает модульное тестирование всех компонентов и интеграцию с системами непрерывной интеграции.