ВВЕДЕНИЕ.

Краткая постановка задачи сформулирована в техническом задании проекта. Цель введения — раскрыть и дополнить техническое задание, а также выделить абстракции, понятия и сокращения, которые будут действительны на протяжении всего дальнейшего изложения.

Приведу формулировку из ТЗ:

«Разработать программу трехмерного моделирования и анимации открытого огня. Программа должна реалистично отрисовывать процесс горения, как в пустом пространстве, так и при наличии окружения. При этом огонь может взаимодействовать с окружением в качестве источника света. Также необходимо реализовать генерацию дыма, получаемого при горении».

Здесь каждое предложение несет в себе законченную мысль. Каждое можно трактовать как подзадачу всего проекта. Однако среди них выделяется главная — отрисовка трехмерной реалистичной анимации. Всё остальное (второстепенные подзадачи) будет реализовано как расширения главной задачи.

Отсюда следует очень простая и важная мысль: отрисовка трехмерной анимации – это ядро системы. Дальше, над этим ядром надстраиваются подзадачи:

1) Трехмерная сцена.  
2) Несложное статическое окружение.  
3) Получение реалистичного огня (пламя, всполохи, спектры цветов …).  
4) Рисование дыма, получаемого при горении.  
5) Взаимодействие между огнем и окружением, такое, что огонь является источником света.

Можно видеть, что система разбита на два уровня: уровень ядра и уровень подзадач, в свою очередь разбитый на подзадачи. Большой плюс подобного разбора в том, что его можно безболезненно перенести на построение архитектуры программного решения. В таком случае — выделенные компоненты (подзадачи и ядро) становятся доменами.

Замечу, что здесь представлена вся рассматриваемая система, хоть и в самом абстрактном её понимании. Никаких других подзадач или уровней в дальнейшем вводиться не будет. То есть, вся последующая работа проводится вглубь обозначенных пунктов.

Вся расчетно-пояснительная записка поделена на разделы, каждый из которых описывает определенный аспект проведенной работы.

Анализ предметной области, выделение базовых программных сущностей, выбор инструментов разработки — находятся в аналитическом разделе.

Математическая модель ядра описана и проанализирована в конструкторском разделе.  
Программная реализация, тестирование и исследование всего перечисленного составляют часть конструкторского раздела и полностью весь технический раздел.

2. АНАЛИТИЧЕСКИЙ РАЗДЕЛ.

**2.1. Анализ предметной области.**

Графическая симуляция огня необходима достаточно часто. Огонь — это слишком опасная стихия, для того чтобы проводить с ним реальные эксперименты. Заказчиками технологии выступают в основном индустрия развлечений и министерство по чрезвычайным ситуациям. Первым огонь нужен в компьютерных играх, фильмах (различные спецэффекты), мультанимациях. Вторым — для систем предсказания лесных пожаров, компьютерных систем обучения сотрудников.

Для рисования огня в компьютерной графике было изобретено достаточно много методов. Американский инженер H.R. Baum составил их классификацию в своей работе «Computer fire simulation»:

1) Particle system based.

Огонь представляется в виде большого количества частиц. Каждая имеет свой единственный цвет и некоторую степень прозрачности. При большом количестве частиц, сильно нагружает компьютер, при этом не давая ни внешней ни поведенческой реалистичности. Считается способом для новичков, так как прост в реализации.

2) Noise-synthesized.

Здесь огонь получается путем наложения друг на друга нескольких сеток шумов. Выглядит хорошо, однако трудно воспроизвести реальное поведение.

3) Geometry-skeleton.

Используется лучевой каркас, который затем покрывается некоторой поверхностью. Данный метод подкупает своей простотой реализации и простотой расчетов. Других преимуществ не имеет.

4) Data-driven.

Допустим имеется возможность наблюдать реальный огонь в реальном мире и фиксировать его с помощью фото или видеосъемки. (Что случается достаточно редко). Собранный материал можно использовать для построения компьютерной модели. Метод нуждается в большом хранилище данных и заведомо реалистичен. Однако, чтобы добавить новое поведение огня, нужно снова проводить реальные эксперименты и увеличивать объем базы данных.

5) Physically-based.

К последнему классу относятся наиболее трудные для моделирования и программирования методы. Они же дают самый качественный результат, то есть самое реалистичное изображение. Поэтому, исследование «физически-обоснованных» способов представляет огромный интерес для всех, кто нуждается в искусственной симуляции огня на компьютере. Направление это довольно молодое, но в нем уже есть признанные мэтры, такие как Jos Stam, Ronald Fedkiw, Duc Quang Nguyen.

Недостатки метода: сложная программная реализация, возможная расходимость применяемого метода расчета состояний модели и долгий расчет этой самой модели.

**2.1. Обзор и анализ существующих программных систем и обоснование необходимости разработки.**

Для рендеринга огня могут быть использованы пакеты редактирования компьютерной графики: Autodesk Maya или Blender3D. Качество получаемых результатов для обоих пакетов приблизительно одинаково.

Недостатками обеих пакетов можно назвать негибкость параметров огня, невозможность поэкспериментировать с огнем и тем как он взаимодействует с окружением.

**2.2. Выбор, обоснование метода моделирования и алгоритма.**

Для реализации ядра системы автором был выбран Physically-based method. Причиной выбора послужило то, что хоть это и самый сложный метод, но он дает самый реалистичный и качественный результат. Также данный метод позволяет решать и некоторые дополнительные задачи, к примеру распространение огня, тление и затухание. Это значит, что в будущем разработанная система может быть расширена до полноценного фреймворка (если такое потребуется).

Существует занимательный факт, который несколько упрощает реализацию ядра. Дело в том, что движение жидкостей, газов и плазмы в поле скоростей можно описать одной и той же системой векторных уравнений. Система называется уравнениями Навье-Стокса, в честь французского физика Навье и ирландского математика Стокса. Конечно, так можно делать только тогда, когда не требуется физическая достоверность моделирования (что определенно наш случай). Имеется дополнительное ограничение — описываемая жидкость или газ должны быть несжимаемыми. На практике это не играет роли, так как на дозвуковых скоростях отклонением модели от реальности можно пренебречь.

Огонь, в первом приближении, является плазмой, а дым — газом. Значит оба явления можно описать с помощью одной и той же математической модели. Разница будет лишь в коэффициентах расчета, таких как вязкость, диффузия, цветовые спектры и некоторые другие. Замечу, что при реализации анимации огня другими методами, дым пришлось бы прорабатывать с нуля.

Для реализации ядра в этой работе, был взят за основу программный движок, описанный Jos Stam в его статье Real-time fluid dynamics for games.

3. КОНСТРУКТОРСКИЙ РАЗДЕЛ.

**3.1. Математические основы метода математического моделирования.**

Уравнения Навье-Стокса выглядят следующим образом.

(1)

Это нелинейная система векторных уравнений в частных производных. Верхнее описывает поведение поля скоростей, а нижнее — поле плотностей. Система справедлива как для двумерного, так и для трехмерного случая (меняется только количество компонент векторов).

Рассмотрим сначала второе уравнение. Оно утверждает, что изменение скалярного поля плотностей во времени происходит из-за трех причин, соответствующих трем слагаемым в правой части. Первая — это так называемая адвекция (advection), в сущности, перемещение вещества под действием поля скоростей. Вторая — диффузия вещества, где каппа — коэффициент диффузии. И, наконец, третья причина — поле источников плотности, обозначаемое S.

Первое уравнение очень похоже на второе. И действительно, три слагаемых в правой части имеют ту же самую смысловую нагрузку: адвекция, диффузия, где ню — коэффициент, и источники поля (по сути — это поле сторонних сил).

Ядро программного решения призвано решить данную систему для заданных коэффициентов и полей.

**3.2. Разработка алгоритма метода моделирования.**

Алгоритм симуляции огня представляет собой безусловный цикл по времени. Внутри цикла выполняется перерасчет полей плотности и скорости, а также отрисовка огня на сцене. Помимо этого, производится калибровка поля скоростей, для того чтобы эмулировать действие температуры. Это необходимо делать, потому что температура не входит явно в систему уравнений (1).

Алгоритм изображен на рисунке 1.

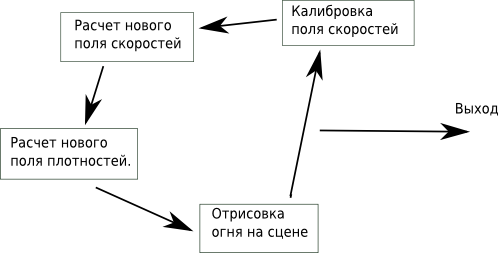


Рисунок 1.

**3.3. Разработка и обоснование используемых типов и структур данных.**

**3.4. Разработка структуры программного комплекса.**

4. ТЕХНИЧЕСКИЙ РАЗДЕЛ.