

РОДИН С. А., БАЗАРОВА И. А.
РАЗРАБОТКА ПРОГРАММНОГО ДВИЖКА ДЛЯ ВИЗУАЛИЗАЦИИ
VOXEL 3D МОДЕЛЕЙ И УПРАВЛЕНИЯ РЕНДЕРИНГОМ

УДК УДК 025.3/.4:(084+086), ВАК 05.13.18, ГРНТИ 20.19.29

Разработка программного движка для визуализации VOXEL 3D моделей и управления рендерингом

Development of a software engine for visualizing VOXEL 3D models and rendering control

С. А. Родин, И. А. Базарова

S. A. Rodin, I. A. Bazarova

Ухтинский государственный технический университет, г. Ухта

Ukhta State Technical University, Ukhta

В статье рассматривается процесс разработки программного движка для визуализации 3D-моделей. В ходе разработки был получен программный продукт, основанный на воксельном представлении объемных данных.

This article discusses the process of developing a software engine for visualizing 3D models. In the course of development, a software product based on voxel representation of volumetric data was obtained.

Ключевые слова: разработка, воксель, 3D-графика.

Keywords: development, voxel, 3D-graphic.

Введение

В областях, где требуется визуализация и обработка больших объемов информации, применяются специализированные графические движки, которые должны обладать функционалом, способным уменьшить объем обрабатываемых данных и обладать приемлемой производительностью.

Рассматриваемый в статье графический движок разрабатывался для отдела «Лаборатория облачных систем и виртуализации сервисов» в организации «АНО ВО «Университет Иннополис», который занимается вопросами обработки данных ГИС.

Графический модуль, используемый в данном отделе, требовал замены, так как перестал удовлетворять требуемым показателям производительности. Заданная разработка должна была повысить производительность относительно показателей предыдущей реализации, а также уменьшить объем выделяемой оперативной памяти.

Были выявлены и определены слабые места предыдущей реализации:

- недостаточная производительность на больших объемах входных данных;
- трудоёмкий процесс поддержки ввиду высокой зависимости одних функций от других;
- не оптимальные алгоритмы конвертации объемного изображения.

Предпроектное исследование

Текущая реализация графического решения основана на популярной графической библиотеке VTK. VTK имеет фиксированный графический конвейер. Подобный конвейер имеет существенный недостаток - он не позволяет внедрять сторонний исполняемый код.

Решение работало на основе структуры `vtkUnstructuredGrid` – не структурированной таблицы. Данная структура является аналогом динамического массива, позволяющий хранить данные в произвольном порядке. Это подход не оптимален, поскольку тратится большое количество памяти на внутренние указатели.

Полигональное изображение строилось на основе высоты модели в данной точке пространства, то есть на каждый перепад высоты требовалось как минимум четыре вершины. Вершины – точки, на основе которых строится меш трехмерной модели, а большое количество вершин приводит к переполнению конвейера рендеринга. По этой причине конвейер визуализации VTK обрабатывал огромный объем данных.

Для текущей реализации было проведено тестирование при визуализации объемных трехмерных данных. Тестирование показало, что 10 миллионов точек программа рендерит со скоростью 14 fps и занимает 1,5 Gb оперативной памяти, что является показателями низкой производительности.

Таким образом возникла потребность для разработки графического модуля, способного визуализировать большие объемы данных с приемлемым уровнем производительности, а также с меньшим расходом оперативной памяти.

Задачами разработки являлись реализация инструментария по работе с трехмерной моделью, алгоритмов по созданию трехмерной сцены и функционала для создания интерфейса.

Проектирование программного обеспечения

На основе предпроектного обследования были определены основные технические решения. Среди них:

1. Общая архитектура движка, состоящая из двенадцати модулей и отвечающих за определенные функции движка:

- Core – является набором обязательных для старта приложения функций.
- Debug – набор классов и функций, позволяющих улучшить работу по определению ошибок и их возникновений. Также в данную подсистему включен профайлер, позволяющий проверить скорость работы функций.
- Thread – функция многопоточности. Работает на базе пула потоков [1].
- OpenGL – функции по работе с OpenGL [2].
- Render – функции и классы, позволяющие работать с трехмерной сценой.
- Render::GUI – графический пользовательский интерфейс.
- Render::Text – общение с пользователем посредством визуализации текста.
- Shader – набор функций и классов по работе с шейдерами. Поскольку приложение является графическим и реализует GUI, то работа с шейдерами является обыденной задачей.
- Voxel – набор функций, позволяющих работать с воксельной [3] моделью.

- Input – функции получения ответов об изменении состоянии мыши и клавиатуры.
- EventSystem – функции, позволяющие получать callback после выполнения события.
- FileManager – функции, работающие с файловой системой ОС.

2. Общий алгоритм работы движка, включающий в себя процесс инициализации и общий цикл работы движка.

Процесс инициализации начинается с запуска программы. Инициализируется пул потоков, настраивается работа с логгером и консолью. Далее происходит настройка окна – загрузка указателей на функции OpenGL, создание два контекста рендеринга и инициализация системы событий. Включается расширение DebugOutput, устанавливаются первичные флаги рендера, инициализируются стандартный и мультисемпловый буфер. После этого загружаются и настраиваются шейдеры, создаются основная сцена и сцена фильтрации, а также инициализируется графический интерфейс.

Процесс рендеринга делится на два этапа: обновление и подготовка данных, и, собственно, сам процесс визуализации. В первый этап входит обновление камеры, обновление значений плоскостей отсечений, которые важны для алгоритма frustum culling [4], и обновление буферов данных, расчет видимости воксельной модели, то есть отбраковка невидимых граней. На этом этапе обновления приостанавливается и начинается этап визуализации.

Выполняется проверка флагов, которые установил пользователь и установка параметров визуализации согласно настройкам, выполнение отрисовки воксельной модели, отрисовка пользовательского интерфейса. На этом цикл визуализации считается завершенным. Этап обновления продолжается - от окна получают сообщения, после обработки которых выполняется замена буфера кадра на новоотрисованный, и цикл рендера начинается снова.

Основным нововведением в программе является процесс создания трехмерной модели, который основан на воксельном представлении объемных данных. Модель состоит из совокупности контейнеров – нод, которые хранят в себе воксели. Воксель – единица измерения трехмерного пространства, которая хранит в себе цвет.

Каждый контейнер, помимо своих достоинств, также имеет недостатки, которые компенсируют другие контейнеры. Тип контейнера определяется на основе количества вокселей в контейнере, а также однотипности цвета в нем.

Результаты разработки системы

Разработанная программа соответствует всем требованиям заказчика и реализует заявленные функции, а именно:

- Загрузка данных о координатной сетке и о её параметрах.
- Построение трехмерной модели.
- Визуализация трехмерной модели.
- Сохранение трехмерной модели.
- Фильтрация по параметру.

Проведенное тестирование программы показало значительный рост производительности (85 fps к 14 fps) и уменьшение объема выделяемой оперативной памяти (130 Мб к 1.5 Gb).

В дальнейшем, несмотря на уже полученные результаты работы программы, предстоит её отладка и работа по улучшению взаимодействия с пользователем. Разработка является универсальной и при незначительной доработке может использоваться и в других областях применения.

Список использованных источников и литературы

1. Пул потоков в C++ 11 – Web-Answers [Электронный ресурс]. – Режим доступа: <https://web-answers.ru/c/pul-potokov-v-c-11.html> (Дата обращения: 16.05.2020).
2. Реестр Khronos OpenGL® [Электронный ресурс]. – Режим доступа: https://www.khronos.org/registry/OpenGL/index_gl.php (Дата обращения: 18.05.2020).
3. Воксельная графика / Обзор существующих алгоритмов / Форум / 3D движок с волновым алгоритмом растеризации связанных тетраэдров / GameDev.ru – Разработка игр [Электронный ресурс]. – Режим доступа: <https://www.gamedev.ru/pages/tetrawaves/forum/?id=168132> (Дата обращения: 09.05.2020).
4. Отбор фрустума / Общее / Статьи / Программирование игр / Разработка игр [Электронный ресурс]. – Режим доступа: <https://gamedev.ru/code/articles/FrustumCulling?page=1> (Дата обращения: 13.05.2020).

List of references

1. Thread Pool in C ++ 11 - Web-Answers, <https://web-answers.ru/c/pul-potokov-v-c-11.html>, accessed 05/16/2020.
2. Khronos OpenGL® Registry, https://www.khronos.org/registry/OpenGL/index_gl.php, accessed 05/18/2020.
3. Voxel graphics / Overview of existing algorithms / Forum / 3D engine with a wave algorithm for rasterizing related tetrahedrons / GameDev.ru, <https://www.gamedev.ru/pages/tetrawaves/forum/?id=168132>, accessed 05/09/2020.
4. Frustum culling, <https://gamedev.ru/code/articles/FrustumCulling?page=1>, accessed 05/13/2020.