Титульник

Защищается студент группы Б8403а Куцелабский Егор Сергеевич по теме «Текстовый процессор для open-source движка Citrus»

Руководитель – старший преподаватель кафедры информатики, математического и компьютерного моделирования, Кленин Александр Сергеевич

Текстовые процессоры

Для редактирования текстовых данных существуют текстовые процессоры.

Они хранят текст, используя при этом различные по эффективности структуры данных, вычисляют его размеры, отрисовывают, некоторые редакторы (например, Word), позволяют, помимо редактирования текста, использовать в документе изображения, таблицы и прочие не относящиеся непосредственно к тексту элементы.

Game Forest и текст

Студия Game Forest занимается созданием игр. Текст занимает важную часть в информационных сообщениях в играх. Так же текст используется в визуальных редакторах игрового движка, разрабатываемого компанией. Текущая реализация текстового процессора в движке не обладает достаточной эффективностью, поэтому, при увеличении количества текста, изображаемого на экране, частота кадров в приложениях падает.

Цели и задачи работы

Цель работы – создание нового текстового процессора для движка.

Структуры данных

Мною были рассмотрены различные подходы к реализации текстовых редакторов. Сравнительную таблицу алгоритмических сложностей подходов для базовых операций вы можете увидеть на экране. Стоит отметить, что она не вполне отражает реальные характеристики подходов, поскольку типичная производительность подходов значительно отличается от производительности в худшем случае и её сложно верно оценить. При наполнении таблицы использовались данные для худших возможных случаев для соответствующих подходов. Подробную информацию о перечисленных в таблице подходах вы можете прочитать в отчете.

**#СДЕЛАТЬ СЛАЙД ПРО ЮНИТИ И АНРИЛ**

Функциональные требования

Эффективная обработка внутреннего представления текста -

Текст должен храниться в эффективной структуре данных, все базовые операции должны выполняться быстро, в том числе для больших объемов текстовых данных (порядка нескольких десятков мегабайт) при этом структура не должна требовать значительных объемов оперативной памяти.

Отрисовка текста с заданными параметрами –

Текст должен отрисовываться в окне заданного размера, с заданными параметрами шрифта, при этом отрисовываться должна только та часть текста, которая попадает в видимую область окна.

Обработка пользовательского ввода –

Процессор должен обрабатывать пользовательский ввод – ввод текстовых данных, перемещение курсора с помощью мыши и клавиш клавиатуры или касаний экрана выполняя при этом соответствующие операции по редактированию текста.

Архитектура системы

Система состоит из трех модулей:

Модуль, отвечающий за обработку внутреннего представления текста – обработка базовых команд, выделение, перемещение курсора, undo/redo.

В основу модуля, отвечающего за отрисовку, взят существовавший ранее код, однако он был значительно модифицирован для оптимизации (т. е. для отрисовки только видимой части текста).

Модуль, отвечающий за обработку пользовательского ввода, принимает поддерживаемые команды и передаёт их соответствующим обработчикам.

Piece Tree

В качестве подхода для реализации был выбран метод Piece Table в связке со SplayTree. Суть Piece Tree состоит в том, что вместо хранения символов в памяти хранится информация о позиции текстового фрагмента в потоке, его длине, а так же о том, к какому именно потоку относится фрагмент. В Piece Table используется два потока: первый – исходный файл, этот поток используется только для чтения. Второй – добавочный файл, весь новый текст дозаписывается в него.

Splay tree – сбалансированное бинарное дерево поиска, основное свойство которого – элементы, к которым обращались в последний раз, переносятся в корень дерева. Это свойство очень полезно при работе с текстовыми редакторами, поскольку большая часть операций вставки и удаления производится в позиции курсора, а частота её смены на значительную величину невелика по сравнению с количеством прочих операций.

Интерфейс

На экране представлен пример интерфейса. Представлен многострочный текст, часть строк достаточно длинна, чтобы к ним применялся soft wrap. Видно, что текст можно прокручивать. Это эффективная реализация прокрутки, в которой отрисовываются только строки, в данный момент помещающиеся в видимую область.

Реализация и тестирование

Физические характеристики системы приведены на экране.

Было проведено ручное тестирование скорости работы предыдущей версии редактора и моей реализации. Тестирование показало, что при работе с текстом объемом 100 тысяч строк новая версия процессора при вертикальной прокрутке отрисовывается без задержек на высокой частоте кадров, в то время как предыдущая версия процессора имеет задержку в 2-3 секунды перед отрисовкой нового кадра.

Заключение

* Выполнен анализ подходов к реализации текстовых процессоров
* Реализованы системы внутренней обработки текста и отрисовки
* Реализованные модули показывают значительное улучшение производительности по сравнению с предыдущей версией процессора
* До момента защиты дипломной работы планируется доработать модуль обработки пользовательского ввода

СПАСИБО ЗА ВНИМАНИЕ