Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Операционный системы и системное программирование (ОСиСП)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему:

Программное средство «Текстовый редактор»

БГУИР КП 1-40 01 01 005 ПЗ

Студент: гр. 451003

Борикова Е. В.

Руководитель:

Зуев С. И.

Минск 2016

**СОДЕРЖАНИЕ**

Введение 4

1 Анализ предметной области 5

1.1 Анализ аналогов программного средства 5

2 Разработка программного средства 7

2.1 Общая информация 7

2.2 Графические инструменты 7

2.3 Визуальное представление 8

2.3.1 Платформа 8

2.3.2 Игровое поле 8

2.4 Игровая механика 10

2.4.1 Вспомогательные массивы 10

2.4.2 Вспомогательные глобальные переменные 11

2.4.3 Дополнительные эффекты 11

2.4.4 Физика шарика 14

2.4.5 Взаимодействие шарика с платформой 15

2.4.6 Взаимодействие шарика с блоками 16

2.4.7 Генерация дополнительных эффектов 16

2.4.8 Отображение текстовой информации 16

2.5 Описание блок-схемы 19

3 Руководство пользователя 21

3.1 Инструкция по использованию 21

Заключение 24

Список использованных источников 25

Приложение А. Исходный код программы 26

Приложение Б. Блок-схема главного цикла программы 50

**ВВЕДЕНИЕ**

Для работы с текстовыми документами существуют программы, которые называются текстовыми редакторами.

Текстовый редактор – это прикладная программа, позволяющая создавать текстовые документы, редактировать их, просматривать содержимое документа на экране, распечатывать документ. Примером простого текстового редактора является программа Блокнот.

Существуют текстовые редакторы, которые позволяют управлять не только содержанием документа, но и его внешним видом. Такие редакторы называются текстовыми процессорами. Текстовый процессор – это текстовый редактор с более широкими возможностями (форматирование текста, вставка графических объектов, проверка правописания и т.д.) Примером простого текстового процессора является программа WordPad.

Существует множество текстовых редакторов - от простейших учебных до мощных издательских систем, с помощью которых делают книги и т.п.

Наименьший элемент текста – символ. Слова – это символьные последовательности, отделяемые друг от друга пробелами или знаками препинания. Структурными единицами текста являются строка, абзац, страница, раздел. Абзац — несколько предложений, связанных по смыслу. Клавиша Enter в текстовых процессорах вставляет в текст символ конца абзаца.

Основной режим работы текстового редактора - это ввод и редактирование. При работе с текстовым редактором по экрану монитора перемещается курсор, который указывает текущую позицию для ввода. Символ, набираемый с клавиатуры помещается в позицию курсора. Под редактированием понимается внесение любых изменений в набранный текст.

Windows API ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) application programming interfaces) — общее наименование целого набора базовых функций [интерфейсов программирования приложений](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) операционных систем семейств [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows) корпорации «[Майкрософт](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D0%B9%D0%BA%D1%80%D0%BE%D1%81%D0%BE%D1%84%D1%82)». Является самым прямым способом взаимодействия приложений с Windows.

Windows API спроектирован для использования в языке [Си](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) для написания [прикладных программ](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), предназначенных для работы под управлением операционной системы MS Windows. Работа через Windows API — это наиболее близкий к операционной системе способ взаимодействия с ней из прикладных программ.  Windows API представляет собой множество функций, структур данных и числовых констант, следующих соглашениям языка Си. Все языки программирования, способные вызывать такие функции и оперировать такими типами данных в программах, исполняемых в среде Windows, могут пользоваться этим API.

1. Анализ предметной области
   1. Анализ аналогов программного средства

**«**Notepad++**»** — свободный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса большого количества языков программирования и разметки. Поддерживает открытие более 100 форматов. Базируется на компоненте Scintilla, написан на C++ с использованием STL, а также Windows API и распространяется под лицензией GNU General Public License. Базовая функциональность программы может быть расширена как за счёт плагинов, так и сторонних модулей, таких как компиляторы и препроцессоры. [1]

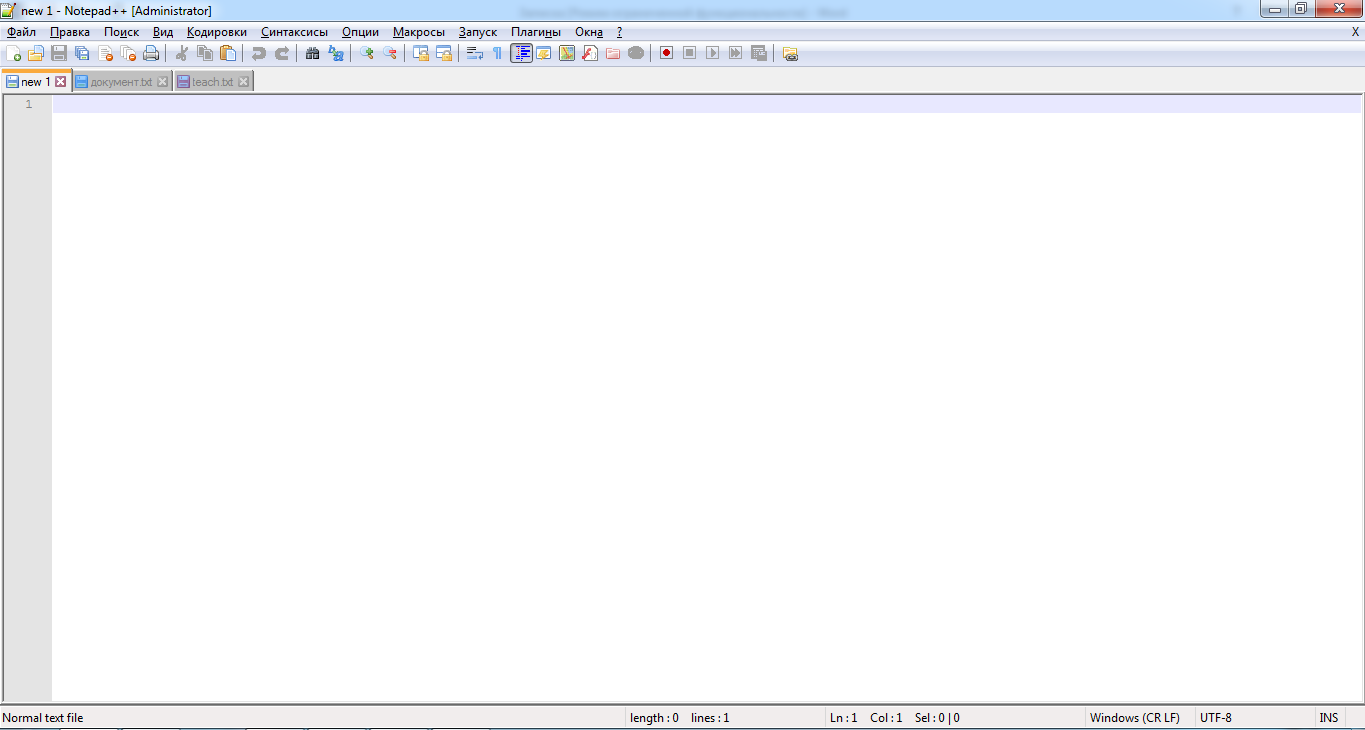


Рисунок 1.1 – Notepad ++

Пользователь имеет ряд возможностей таких как [1]:

* Подсветка синтаксиса
* Сворачивание кода
* Автодополнение и автоматическое закрытие скобок и тэгов (если активировано)
* Закладки
* Регулярные выражения для поиска и замены
* Запись и воспроизведение макросов
* Сравнение файлов
* Менеджер проектов
* Карта документа
* Переопределение любых горячих клавиш
* Резервное копирование сохраняемых файлов (включается в настройках)
* Трансформация текста при помощи подключённого плагина TextFX
* Поддержка и конвертирование кодировок ANSI, UTF-8 и UCS-2
* Блоковое выделение текста, одновременное выделение нескольких разных мест (с Ctrl)
* Мультистрочное редактирование (с использованием Alt)

**«**Sublime Text**»** — кроссплатформенный текстовый редактор. Поддерживает плагины на языке программирования Python. Программа часто используется как редактор исходного кода или HTML.

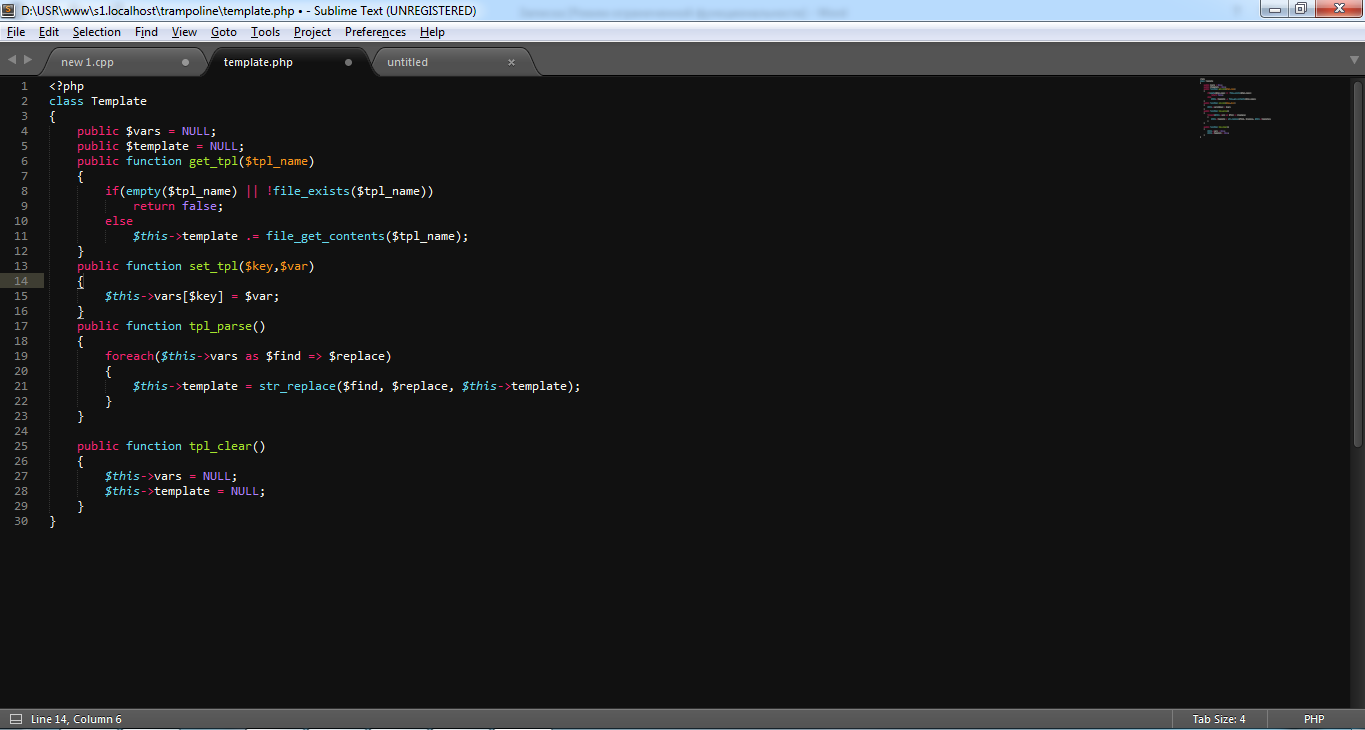


Рисунок 1.2 – Sublime Text

На данный момент уже доступна версия Sublime Text 3. Данный текстовый редактор является очень мощным текстовым редактором, в котором существует огромные дополнительные возможности.

* 1. Постановка задачи

Задачей данной курсовой работы является создание упрощенной версии программного средства текстовый редактор.

Реализовать следующие функции: создание, редактирование, удаление, загрузки из файла, сохранение текста, а также поддержка возможности распечатать текстовый документ.

1. разработка алгоритма
   1. Общая информация
   2. Визуальное представление

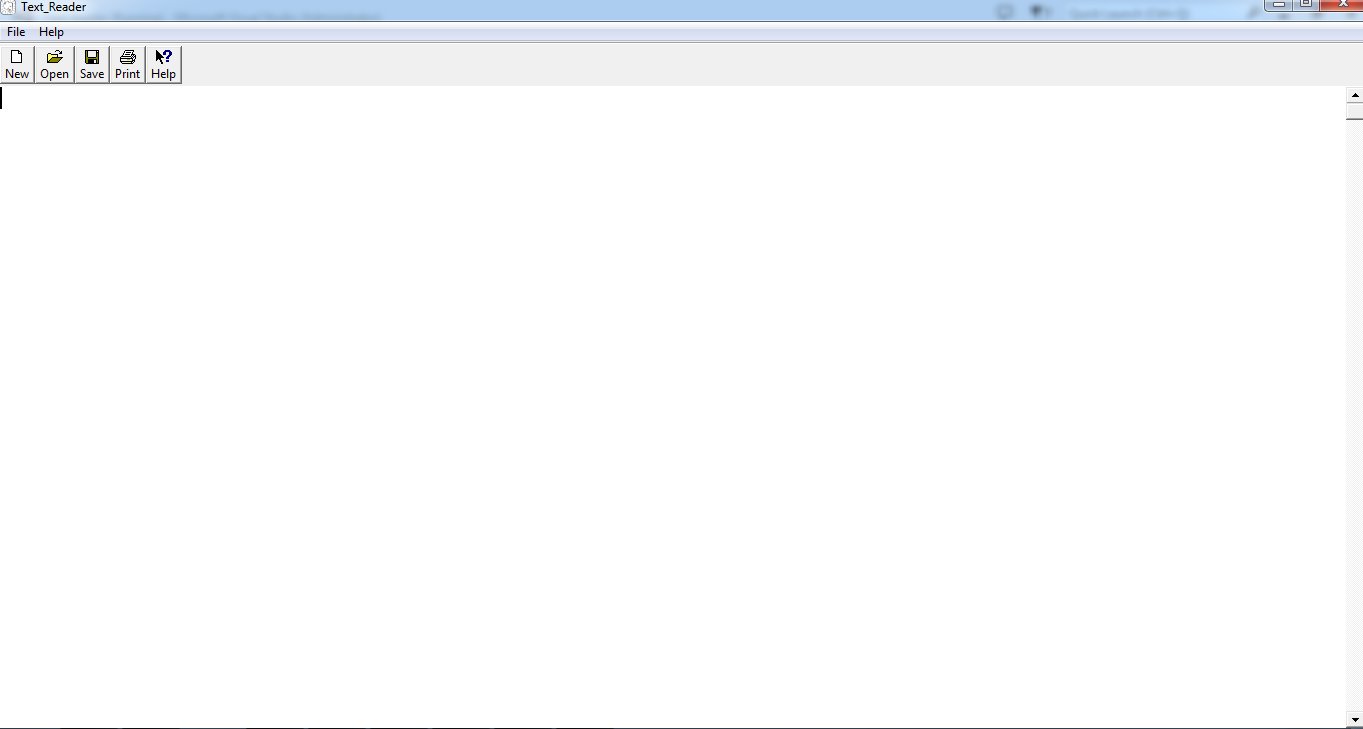


Рисунок 2.1 - Внешний вид приложения

* 1. Описание блок-схемы

На блок-схеме представлен алгоритм главного цикла программы. При работе с графическим движком SFML все графические изменения происходят пока открыто окно. Алгоритм цикла "Пока открыто окно":

1. Получить время. С помощью средств движка SFML на каждой итерации вычисляется время, прошедшее с прошлого измерения времени в микросекундах.
2. Обработка нажатия клавиш. Набор действий, реагирующих на нажатия клавиш клавиатуры и выполняющих соответствующие преобразования.
3. Обработка движения курсора. Поскольку платформа управляется с помощью мыши, необходимо обрабатывать перемещения курсора, перемещая соответственно платформу.
4. Обработка нажатия кнопки мыши. Левая кнопка мыши отвечает за начало игры, а также за стрельбу из лазерной пушки.
5. Поимка бонусов. Проверяет, словил ли игрок выпавший блок с дополнительным эффектом.
6. Цикл, обрабатывающий движение каждого шарика, имеющегося на игровом поле. Если шарик сталкивается с блоком или со стенкой, выполняются соответствующие преобразования. Также если таймер движения становится больше, чем заранее определенная игровая скорость (частота обновления игрового поля в микросекундах), выполняется изменение координат шариков.
7. Если на игровом поле имеется блок с дополнительным эффектом, изменяются его координаты.
8. Если таймер движения больше, чем игровая скорость, таймер движения обнуляется.
9. Очистка окна.
10. Прорисовка всех компонентов: карты блоков, шариков, блоков-бонусов, платформы и текстовой информации.
11. Разработка программного средства

При проектировании программы была выбрана методика нисходящего проектирования: сначала была проработана основная концепция программы (обработка нажатие клавиш, отображение символов).

В рамках данного курсового проекта планируется разработать программное средство.

Для разработки программного средства использовать язык программирования C++ . Среда для разработки – Microsoft Visual Studio 2013.

3.1 Написание кода

* + 1. Вспомогательные переменные

Программа использует глобальные переменные:

* Дескриптор главного окна (HWND).
* HINSTANCE hInst - главная сущность.
* TCHAR szTitle– текст заголовка тулбара.
* TCHAR szWindowClass – имя главного окна.
* HIMAGELIST g\_hImageList – лист картинок, отображаемые на тулбаре.
* OPENFILENAME ofn – указатель на структуру, используемую для открытия файла.
* HFONT font – указатель на шрифт.
* RECT rect - структура содержит координаты верхнего левого и нижнего правого углов прямоугольника.
* list<Glyph\*> glyph\_list – указатель на список glyph.
* list<Glyph\*>::iterator CarriagePosition – указатель, который используется кареткой, которая перемещается перечислением glyph’ов из списка.
* SCROLLINFO si – указатель на структуру для колеса прокрутки.
  + 1. Модуль «Glyph.h»

Модуль «Glyph.h» состоит из одного класса Glyph, который содержит открытые свойства и методы: деконструктор, получить высоту и ширину, метод рисовать, с помощью которого мы выводим символ на экран.

Данный класс нам необходим, чтобы создать список вводимых символов с клавиатуры, т.е. нам необходимо хранить где-то символы, которые мы ввели, для этого мы и будем использовать список типа Glyph.

Также для реализации передвижения каретки нам необходимо использовать list<Glyph\*>::iterator. Тип итератора для управляемой последовательности. Описывает тип объекта Glyph\*, который можно использовать как произвольно-доступный итератор для контролируемой последовательности.

* + 1. Модуль «Char.h»

Модуль «Char.h» находится один класс Char, который наследуется от класса Glyph и дополняется скрытыми методами и свойствами: вычислить область экрана, высота, ширина, размер, состояние перевода на новую строку, значение символа, шрифт, цвет.

Реализация всех методов находится в файле «Char.cpp». В данном файле у нас находится конструктор для инициализации объекта Char:

Char::Char(int size, HFONT font, wchar\_t symbol, COLORREF rgb, bool state)

{

this->size = size;

this->font = font;

this->symbol = symbol;

this->rgb = rgb;

this->state = state;

computeRect();

}

Символ имеет тип данных wchar\_t. В Windows API тип wchar\_t именуется как WCHAR и имеет фиксированный размер 16 бит, что не позволяет кодировать весь набор символов Unicode (больше 1 миллиона). Поэтому нарушается стандарт ANSI/ISO C, который требует, чтобы символьный тип wchar\_t поддерживал все представимые в системе символы в одном объекте wchar\_t. По сути, в WinAPI под WCHAR подразумевается 2-байтное слово из кодировки UTF-16LE (как тип WORD), поэтому символы с кодами выше FFFF16кодируются парой WCHAR (так называемые «суррогаты») и всем API-функциям передаётся не количество символов, а размер символьного массива в машинных словах.[4] В общем, символ строки может быть представлен в виде 1-го байта и 2-х байтов.   
Обычно одно-байтовый символ это символ кодировки ANSI- в этой кодировке представлены все английские символы. А 2-х байтовый символ это кодировка UNICODE, в которой могут быть представлены все остальные языки в мире. В общем, символ строки может быть представлен в виде 1-го байта и 2-х байтов.   
Обычно одно-байтовый символ это символ кодировки ANSI- в этой кодировке представлены все английские символы. А 2-х байтовый символ это кодировка UNICODE, в которой могут быть представлены все остальные языки в мире. Для этого работе с файлом мы должны учитывать кодировку, т.е. при открытии файла необходимо указать с какой кодировкой будем работать:

\_wfopen\_s(&outFile, (LPWSTR)fullpath, L"r+,ccs=UTF-16LE");

Таким образом данное программное средство может поддерживать все языки мира и корректно отображать, сохранять и записывать любые символы.

1. Разработка программного средства

В качестве основы для проведённого функционального тестирования, были взяты функциональные требования, описывающие необходимые задачи, которые решает ПО. Результаты приведены в таблице 4.1.

Таблица 4.1 – Тестовые сценарии

|  |  |  |  |
| --- | --- | --- | --- |
| **Функциональность** | **Выполняемые действия** | **Ожидаемый результат** | **Результат теста** |
| Отображение набранного текста | Ввод текста с клавиатуры | Вывод текста на экран | Пройден |
| Переход на новую строку | Нажать на клавиатуре «Enter» | Перевод каретки на новую строку | Пройден |
| Вставка вертикальной табуляции | Нажать на клавиатуре «Tab» | Вставка вертикальной табуляции | Пройден |
| Вставка спец. символов | Ввод с клавиатуры символов: ! « № ; % : ? \* ( ) \_ + @ # $ ^ & {} [] ,. < > / \ | “ ` ~ | Отображение в рабочей области: ! « № ; % : ? \* ( ) \_ + @ # $ ^ & {} [] ,. < > / \ | “ ` ~ | Пройден |
| Отображение символов на арабском языке | Ввод символов с клавиатуры на арабском языке: ابينيبمسي ميستبىءزؤ مسيتب | Отображение символов в рабочей области на арабском языке: ابينيبمسي ميستبىءزؤ مسيتب | Пройден |
| Напечатать текстовый документ | Нажатие на панели инструментов кнопки «Print» или через контекстное меню File- > Print | Открытие другой программы, которая обрабатывает текстовый документ, который мы передели | Пройден |
| Создание нового документа | Нажатие комбинации клавиш Ctrl + N, или через контекстное меню File->New, или на панели инструментов нажать кнопку «New» | Отображение листа символов, который предварительно очистился | Пройден |
| Открытие текстового документа | Нажатие комбинации клавиш Ctrl + O, или через контекстное меню File->Open, или на панели инструментов нажать кнопку «Open» | Отображение текстового документа | Пройден |
| Продолжение таблицы 4.1 – Тестовые сценарии | | | |
| Сохранение текстового документа | Нажатие комбинации клавиш Ctrl + S, или через контекстное меню File->Save, или на панели инструментов нажать кнопку «Save» | Сохранение текстового документа | Пройден |
| Сохранение и отображение текстового документа только на английском языке | Нажатие комбинации клавиш Ctrl + S, или через контекстное меню File->Save, или на панели инструментов нажать кнопку «Save» | Сохранение и отображение текстового документа | Пройден |
| Сохранение и отображение текстового документа только на русском языке | Нажатие комбинации клавиш Ctrl + S, или через контекстное меню File->Save, или на панели инструментов нажать кнопку «Save» | Сохранение и отображение текстового документа | Пройден |
| Сохранить текстовый документ в уже существующий файл | Нажатие комбинации клавиш Ctrl + S, или через контекстное меню File->Save, или на панели инструментов нажать кнопку «Save». Далее указать имя файла, который уже существует | Замена и сохранение документа | Пройден |
| Просмотр информации о программе | Нажатие на панели инструментов кнопки «About» или через контекстное меню File- > About | Появление нового окна с информацией об авторе, год, имя программы | Пройден |
| Передвижение каретки влево | Нажатие на клавиатуре стрелочки влево | Перемещение каретки | Пройден |
| Позиция каретки не изменяется влево, если дошли до начала листа символов | Нажатие на клавиатуре стрелочки влево в начале листа символов | Позиция каретки не изменяется | Пройден |
| Передвижение каретки вправо | Нажатие на клавиатуре стрелочки вправо | Перемещение каретки | Пройден |
| Позиция каретки не изменяется вправо, если дошли до конца листа символов | Нажатие на клавиатуре стрелочки вправо в конце листа символов | Позиция каретки не изменяется | Пройден |
| Передвижение каретки вверх | Нажатие на клавиатуре стрелочки вверх | Перемещение каретки | Пройден |
| Продолжение таблицы 4.1 – Тестовые сценарии | | | |
| Позиция каретки не изменяется вверх, если каретка на самом верху | Нажатие на клавиатуре стрелочки вверх, когда находимся на самой первой строке | Позиция каретки не изменяется | Пройден |
| Передвижение каретки вниз | Нажатие на клавиатуре стрелочки вниз | Перемещение каретки | Пройден |
| Позиция каретки не изменяется вниз, если каретка находится на последнее позиции | Нажатие на клавиатуре стрелочки вниз, когда находимся на последней позиции | Позиция каретки не изменяется | Пройден |
| Удаление символов | Нажатие клавиши на клавиатуре «Delete» | Удаление символа, стоящего после каретки, позиция каретки не изменяется | Пройден |
| Удаление символов, которых не существует | Нажатие клавиши на клавиатуре «Delete» в позиции, где после каретки не следует символов | Обработка данной исключительной ситуации. Исключения не прервали работу | Пройден |
| Удаление символов | Нажатие клавиши на клавиатуре «Backspace» | Удаление символа, стоящего перед кареткой, позиция каретки смещается влево на одну позицию | Пройден |
| Удаление символов, которых не существует | Нажатие клавиши на клавиатуре «Backspace» в позиции где перед кареткой не следует символов | Обработка данной исключительной ситуации. Исключения не прервали работу | Пройден |

1. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Кликните двойным щелчком на ярлык с надписью «Text Reader» (рис. 5.1).

Рисунок 5.1 – Ярлык программы

Программа запустится на вашем компьютере. Запустив приложение, пользователь увидит главное окно (рис 5.2).

Рисунок 5.2 – Главное окно программы

После открытия окна в рабочей области уже можно увидеть каретку. Для создания текста необходимо начать вводить текст с помощью клавиатуры. Текст сразу будет отображаться на экране в рабочей области.

Если нам необходимо сохранить наш текст, то это можно сделать нажатием комбинации клавиш Ctrl + S, нажатием кнопки «Save» на панели инструментов (рис. 5.5) или через контекстное меню: File-> Save. (рис. 5.3 Далее открывается окно для задания местоположения и имени файла. Для подтверждения выбора файла необходимо нажать кнопку «ОК». Выполняется сохранение текста в файл. Если ввести в диалоговое окно «Save file as…» (рис. 5.3) несуществующий файл, то программное средство создаст этот файл, что очень удобно и позволяет сохранить немного времени.

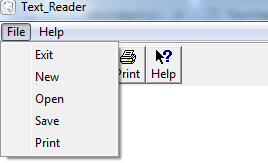


Рис. 5.3 – Контекстное меню

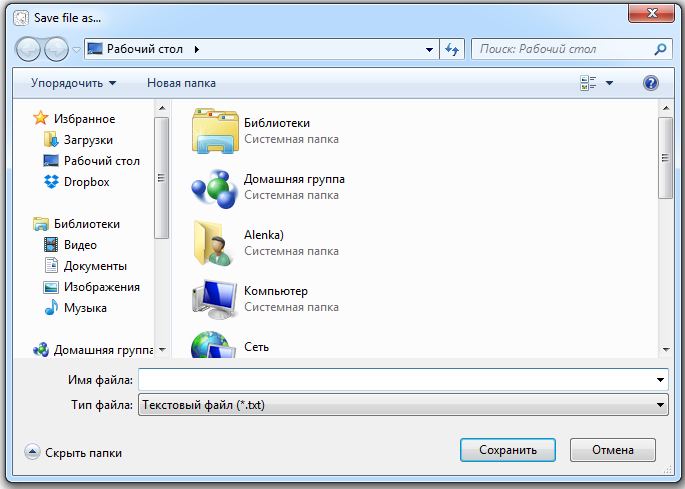


Рис. 5.4 – Окно «Save file as…»

Если нам необходимо создать новый текстовый документ, то это можно сделать нажатием комбинации клавиш Ctrl + N, нажатием кнопки «New» на панели инструментов (рис 5.5) или через контекстное меню: File-> New(рис 5.3). Рабочая область и список символов очищается, каретка перемещается в верхний левый угол рабочей области.



Рис.5.5– Панель инструментов

Для редактирования текста реализованы удобные и привычные нам функции. Например, для удаления символа необходимо нажать клавишу «Backspace» или «Delete». Чтобы переместить каретку в нужную позицию необходимо использовать клавиши стрелка вниз, вверх, вправо и влево.

Если нам необходимо распечатать наш текстовый документ, то это можно сделать с помощью кнопки «Print» на панели инструментов (рис. 5.5) или через контекстное меню File-> Print (рис. 5.3). Далее выбираем программу, через которую хотим распечатать текстовый документ и работаем уже с ней.

Программное средство предоставляет возможность узнать информацию об авторе, иконку, год издания и версию программы. Это можно сделать с помощью контекстного меню Help-> About или на панели инструментоыв нажатием кнопки «About». Откроется диалоговое окно (рис. 5.6).

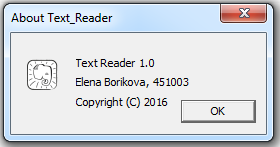


Рис. 5.6 – диалоговое окно «About»

**ЗАКЛЮЧЕНИЕ**

Ещё с древних времён люди находили различные способы развлечения. Один из них – игры. Сложно представить человека, который ни разу не играл в игры, будь то компьютерные или настольные. Целью данной курсовой работы – разработать программное средство, которое позволит создавать, редактировать, сохранять и считывать текст.

Результатом данной курсовой работы является программное средство «Text Reader», созданное в интегрированной среде разработки Microsoft Visual Studio 2013 на языке с++. Приложение опирается на платформу WinAPI32.

В ходе разработки было изучено большое количество литературы из различных источников. Были усовершенствованы знания о работе с файлами. Получены новый знания связанные с обработкой клавиш, а также с работой каретки. Полезным было получить навык с работой с различными кодировками.

Возможно дальнейшее усовершенствование проекта:

* Добавление колеса прокрутки;
* Добавление функции «Отмена шага»;
* Обработать сообщения поступающие от мышки;
* Подсветка синтаксиса;

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Текстовые редакторы[Электронный ресурс]. – Текстовый редактор. Текстовый процессор. Режим доступа: <http://school497.ru/download/u/02/les11/les.html>. Дата доступа: 12.12.2016
2. Википедия Свободная энциклопедия [Электронный ресурс]. - Notepad++. Режим доступа: https://ru.wikipedia.org/wiki/Notepad%2B%2B. Дата доступа: 12.12.2016.
3. Википедия Свободная энциклопедия [Электронный ресурс]. - Sublime Text. Режим доступа: - <https://ru.wikipedia.org/wiki/Sublime_Text>. Дата доступа: 12.12.2016.
4. Википедия Свободная энциклопедия [Электронный ресурс]. – Широкий символ, wchar\_t. Режим доступа: - <https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%80%D0%BE%D0%BA%D0%B8%D0%B9_%D1%81%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB>. Дата доступа: 12.12.2016.
5. Макс Шлее, Профессиональное программирование на С++ – М.: БХВ-Петербург, 2010. – 883 с.
6. Федоренко Ю.П., Алгоритмы и программы на C++ Builder –М.: ДМК Пресс,2010. –544 с.
7. Мартин.Р., Чистый код. Создание, анализ и рефакторинг: Петербург,2015. –464 с
8. Джеффри Рихтер, Windows для профессионалов: создание эффективных Win32\_приложений с учетом специфики 64 разрядной версии Windows / Пер. с англ. — 4-е изд. — Спб.: Питер; М.: Издательство «Русская Редакция»; 2008. — 720 стр.: ил.

Приложение А. Исходный код программы

Main.cpp

Map.h