Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Операционные системы и системное программирование (ОСиСП)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту на тему

Игровое приложение «Сапер»

БГУИР КП I–40 01 01 25 ПЗ

Студент: гр. 751003 Стубеда В.Д.

Проверил: Базылев Е.Н.

Минск, 2019

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой ПОИТ

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Лапицкая Н.В. 2019г.

ЗАДАНИЕ

по курсовому проектированию

Студенту *Стубеде Вадиму Дмитриевичу* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Тема работы *Игровое приложение «Сапер»*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Срок сдачи законченной работы 30*.11.2019г.*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Исходные данные к работе *Язык программирования C++. Среда разработки Microsoft Visual Studio 2017. Реализация компьютерной игры Сапер с использованием программного интерфейса WinAPI.*

4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке)

*Введение*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*1 Анализ литературных источников\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2 Постановка задачи\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*3 Разработка программного средства\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*4 Тестирование и проверка работоспособности программного средства\_\_\_\_*

*5 Руководство по установке и использованию программного средства\_\_\_\_\_*

*Заключение \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_*

*Список использованных источников\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Приложения* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

*Схема алгоритма в формате А1*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Консультант по курсовой работе *Базылев Е.Н. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

7.Дата выдачи задания *15.09.2019г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

8. Календарный график работы над проектом на весь период проектирования

(с обозначением сроков выполнения и процентом от общего обьема работы):

*Раздел 1. Введение к 25.09.2019г. – 10 % готовности работы;\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Раздел 2 к 30.09.2019г. – 30% готовности работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Раздел 3 к 15.10.2019г. – 60% готовности работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Раздел 4 к 10.11.2019г. – 80% готовности работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Раздел 5.Заключение. Приложения к 20.11.2019г. – 90% готовности работы;*

*оформление пояснительной записки и графического материала к 23.05.2019г. – 100% готовности работы.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Защита курсового проекта с 20.11.2019г. по 1.12.2019г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

РУКОВОДИТЕЛЬ *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Базылев Е.Н.*

*(подпись)*

Задание принял к исполнению *Стубеда В.Д. 15.09.2019г.*

*(дата и подпись студента)*

Содержание

[Введение 5](#_Toc25615777)

[1 Анализ литературных источников 7](#_Toc25615778)

[1.1 Анализ существущих приложений. 11](#_Toc25615779)

[2 Постановка задачи 12](#_Toc25615780)

[3 Разработка программного средства 13](#_Toc25615781)

[3.1 Описание алгоритмов 13](#_Toc25615782)

[3.2 Структура данных 14](#_Toc25615783)

[3.2.1 Структура типов основного алгоритма 14](#_Toc25615784)

[3.2.2 Структура данных основного алгоритма 14](#_Toc25615785)

[3.2.3 Структура данных алгоритма timerTick(hWnd) 15](#_Toc25615786)

[3.2.4 Структура данных алгоритма check() 16](#_Toc25615787)

[3.2.5 Структура данных алгоритма paintGame(hWnd) 17](#_Toc25615788)

[3.2.6 Структура данных алгоритма rotate() 18](#_Toc25615789)

[3.2.7 Структура данных алгоритма startGame(hWnd) 19](#_Toc25615790)

[3.2.8 Описание графических компонентов основного окна 20](#_Toc25615802)

[3.3 Схемы алгоритмов решения задачи 21](#_Toc25615803)

[4 Тестирование и проверка работоспособности программного средства 25](#_Toc25615804)

[5 Руководство по установке и использованию программного средства 28](#_Toc25615805)

[5.1 Шаг 1. Установка 28](#_Toc25615806)

[5.2 Шаг 2. Знакомство с программой. Инструкция по эксплуатации 28](#_Toc25615807)

[Заключение 30](#_Toc25615808)

[Список использованной литературы 31](#_Toc25615809)

[Приложение А 32](#_Toc25615810)

Введение

В мире сейчас наступило время, когда появилось множество разных увлечений. Среди хобби компьютерные игры приобрели огромный размах. Не только дети, но и взрослая часть населения увлекается этим видом развлечений. Жанров игр очень много, и для каждого найдется свой. Многие игры развивают у человека реакцию (RPG и квесты), творческое начало, фантазию, помогают в освоениии компьютера. Жанр логических игр развивает у игрока логическое мышление и умение вырабатывать различные стратегии.

Игра – непременный спутник развития человечества. На стадии игры выполняется чрезвычайно важные функции. Игры использовались для социализации подрастающего поколения, для подготовки к коллективной работе, для тренировки. Игра также является основным видом деятельности ребенка. С.Л. Рубинштейн отмечал, что игра хранит и развивает. По мнению Д.Б. Эльконина, «в игре не только развиваются или заново формируются отдельные интеллектуальные операции, но и коренным образом изменяется позиции ребенка и взрослого в отношении к окружающему миру и формируется механизм возможной смены позиции и координации своей точки зрения с другими возможными точками зрения».

В ходе работы над данным курсовым проектом было разработано игровое приложение “Сапер” – логическая игра, тренирующая логигу и разрабатывающая мышление.

# Анализ литературных источников

Сапер представляет собой логическую игру, где игровое поле разделено на смежные ячейки, некоторые из которых «заминированы»; количество «заминированных» ячеек известно. Целью игры является открытие всех ячеек, не содержащих мины.

Игрок открывает ячейки, стараясь не открыть ячейку с миной. Открыв ячейку с миной, он проигрывает. Если под открытой ячейкой мины нет, то в ней появляется число, показывающее, сколько ячеек, соседствующих с только что открытой, «заминировано; используя эти числа, игрок пытается рассчитать расположение мин, однако иногда даже в середине и в конце игры некоторые ячейки всё же приходится открывать наугад. Если под соседними ячейками тоже нет мин, то открывается некоторая «не заминированная» область до ячеек, в которых есть цифры. «Заминированные» ячейки игрок может пометить, чтобы случайно не открыть их. Открыв все «не заминированные» ячейки, игрок выигрывает.

История игры «Сапер» уходит в далекие 50-е года 20-го столетия. В то время это была конечно же не компьютерная игра, т.к. персональные компьютеры появились гораздо позже, а игра в большой картонной коробке.

Игровое поле представляло собой трехслойный punchboard (доска с отверстиями и вставленными фишками). В верхнем слое были отверстия, в которые нужно было ударять молотком во второй слой, который в свою очередь был защитной пленкой и скрывал под собой третий слой, на котором были изображены мины и цифры в точности под каждым отверстием первого слоя.

Если игрок пробивал все отверстия, под которыми не было мин, то он мог получить от производителя приз – еще одну такую же игру.

Таким образом, «Сапер» стал очень популярной и увлекательной игрой, а одновременно и прибыльным товаром для производителя.

С эрой появления персональных компьютереов игра «Сапер» обрела свою еще большую популярность. Самым первым компьютерным прородителем этой игры можно считать игру «Куб», созданную Дэвидом Алом.

Вскоре после этого, в 1985-м году, взяв идею игры «Куб», трое программистов из США создали игру Relentless Logic (RLogic), которая работала под операционной системой MS-DOS и все действия игры проходили не в кубе, а на прямоугольной поверхности.

Еще одним вариантом игры послужила игра Mines, разработанная Томом Андерсоном в 1987г., в основе которой была RLogic, только с немного измененными правилами.

С появлением операционной системы Windows «Сапер» приобрел свою самую большую популярность и в нее стали играть по всему миру.

Примечательно, что игра "Сапер" появилась в Windows тоже не просто так. Как сообщает компания, эта игра была нужна для того, чтобы научить игроков пользоваться двумя кнопками мыши. А поиск заминированных клеток оказался самым простым способом донести до пользователей разницу между правой и левой клавишей.

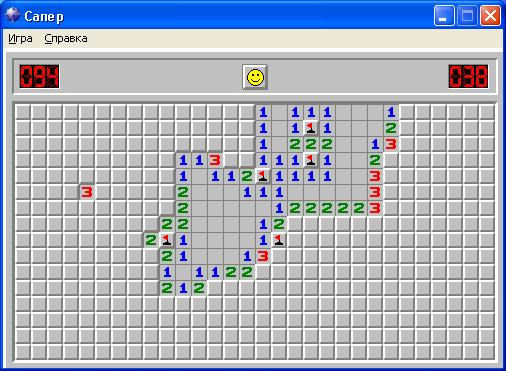


Рисунок 1.1 – Классическая игра «Сапер».

## Анализ существущих приложений.

Существует множество различных аналогов Сапера. Все они очень схожи. В основном различия составляют стили, которые заключаются в цвете, форме поля или форме ячеек. Существет также сапер с объемным полем.

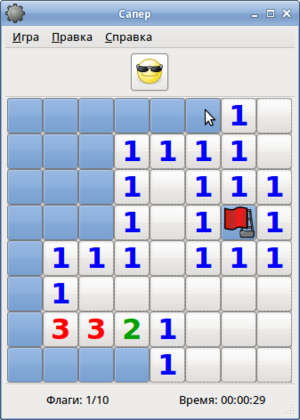


Рисунок 1.1 – Простой Сапер

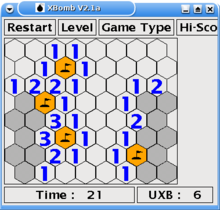


Рисунок 1.2 – Сапер с шестиугольными ячейками

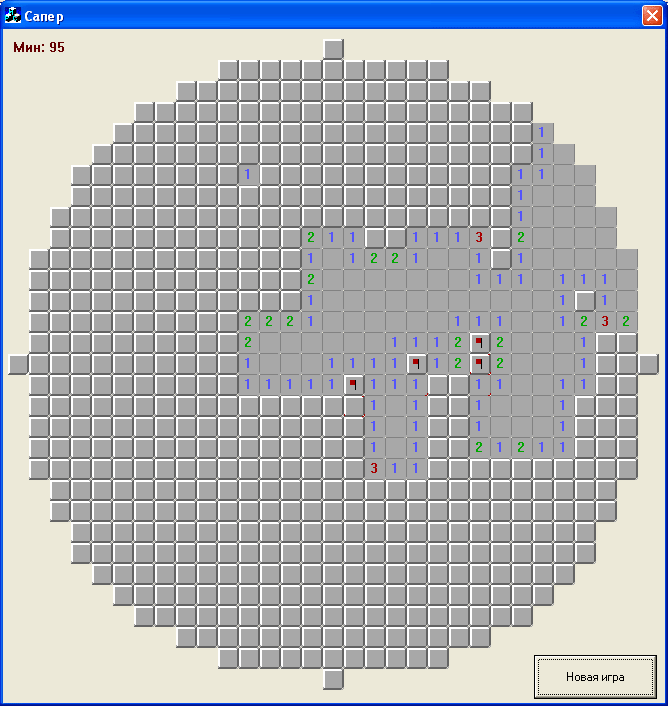


Рисунок 1.3 - Сапер с круглым полем

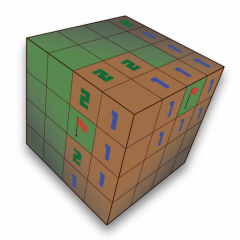


Рисунок 1.4 – Сапер с объемным полем

# 2 Постановка задачи

Написать игровое приложение “Сапер” с использованием прикладного программного интерфеса Windows API. Необходимо реализовать графическую часть и управление кодом игрового процесса с помощью мыши.

При запуске приложения автоматически генерируется поле с уже заполненными клетками, таким образом игрок может сразу присткпать к игре. По нажатию левой кнопки мыши ячейка открывается, по нажатию левой – ставится флаг (пометка для игрока, где предположительно стоит бомба). Нажатие левой кнопкой мыши на такую клетку не обрабатывается. Если игрок нажал левой кнопкой мыши на бомбу, то игра заканчивается, всплывает сообщение о проигрыше. Если игрок нашел все бомбы, то всплывает сообщение о победе. После окончания игры поле вскрывается и видна все подсказки и бомбы (либо флаги, если выигрыш). Чтобы начать новую игру надо щелкнуть два раза левой кнопкой мыши.

# 3 Разработка программного средства

## 3.1 Описание алгоритмов

Таблица 1 – Описание основной программы

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование алгоритма | Назначение алгоритма | Формальные параметры |
| 1. | Draw\_new\_field (HDC hdc); | Прорисовка поля игры (бомбы, цифры, пустые ячейки) | hdc – дескриптор контекста устройства |
| 2. | Create\_map() | Генерирование карты с бомбами и подсказками |  |
| 3. | Choose\_color(int cell) | Возвращает цвет, которым будет выведена цифра-подсказка | cell – цифра-подсказка, хранящаяся в ячейке |
| 4. | Delete\_cell(int x, int y) | Окрытие ячейки с переданными индексами (освобождение памяти, занятой кнопкой) | x,y – индексы ячейки |
| 5. | Check\_near\_cells(int x, int y); | Открытие пустых ячеек вокруг нажатой с переданными индексами (освобождение памяти, занятой кнопками) | x,y – индексы ячейки |

## 3.2 Структура данных

### 3.2.1 Структура типов основного алгоритма

При разработке программного средства использовалась такая структура данных, как двумерный массив. В одном хранятся дескрипторы окон-кнопок, играющих роль ячеек, во втором – карта бомб и подсказок.

Таблица 2 – Структура типов основного алгоритма

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| cellhWnd | HWND \*\* | Точка для хранения дескрипторов |
| map | BYTE \*\* | Точка для хранения карты бомб |

### 3.2.2 Структура данных основного алгоритма

Таблица 3 – Структура данных основного алгоритма

|  |  |  |
| --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение |
| hInst | HINSTANCE | Указатель на экземпляр |
| hWnd | HWND | Дескриптор главного окна |
| hBrush | HBRUSH | Идентификатор кисти |
| hPen | HPEN | Идентификатор логического пера |
| XCellNum | int | Кол-во ячеек по горизонтали |
| YCellNum | int | Кол-во ячеек по вертикали |

Продолжение таблицы 3

|  |  |  |
| --- | --- | --- |
| BombsNum | int | Кол-во бомб |
| GAME\_IN\_PROCESS | BOOL | Указывает, запущена ли игра |
| dx[8] = { 0,0,1,-1,1,1,-1,-1 } | int[] | Константы для изменения индексов ячеек по горизонтали при проверке пустых ячеек |
| dy[8] = { 1,-1,1,1,0,-1,-1,0 } | int[] | Константы для изменения индексов ячеек по вертикали при проверке пустых ячеек |

### 3.2.3 Структура данных алгоритма Draw\_new\_field (hdc)

Таблица 4 – Структура данных алгоритма Draw\_new\_field (hdc)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Комментарии |
| hdc | HDC | Дескриптор контекста устройства | Формальный параметр |
| x | int | Координата ячейки на поле | Локальная переменная |
| y | int | Координата ячейки на поле | Локальная переменная |
| rt | RECT | Определяет координаты прямоугольника | Локальная переменная |

Продолжение таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| s[2] | TCHAR | Хранит строковое значение цифры-подсказки | Локальная переменная |
| hbbomb | HBITMAP | Указатель на битовую карту картинки | Локальная переменная |
| hMemDC | HDC | Дескриптор контекста устройства в памяти | Локальная переменная |
|  | | | |

### 3.2.4 Структура данных алгоритма Choose\_color(cell)

Таблица 5 – Структура данных алгоритма Choose\_color(cell)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Комментарии |
| cell | int | Значение цифры-подсказки | Формальный параметр |
| COLORREF | rgb | Цвет цифры-подсказки | Локальная переменная |
|  |  |  |  |

### 3.2.5 Структура данных алгоритма Delete\_cell(x, y)

Таблица 6 – Структура данных алгоритма Delete\_cell(x, y)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Комментарии |
| x | int | Параметр для цикла обхода всех ячеек | Формальный параметр |
| y | int | Параметр для цикла обхода всех ячеек | Формальный параметр |
| XCellNum | int | Кол-во ячеек по горизонтали | Глобальная переменная |
| YCellNum | int | Кол-во ячеек по вертикали | Глобальная перменная |
| cellhWnd | HWND\*\* | Массив дескрипторов окон-кнопок, играющих роль ячеек | Глобальная перменная |

### 3.2.6 Структура данных алгоритма Check\_near\_cells(x, y)

### Таблица 7 – Структура данных алгоритма Check\_near\_cells(x,y)

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Комментарии |
| x | int | Координата пустой ячейки | Формальный параметр |
| y | int | Координата пустой чейки | Формальный параметр |
| cellhWnd | HWND \*\* | Массив дескрипторов окон-кнопок, играющих роль ячеек | Глобальная перменная |
| map | BYTE \*\* | Массив значений карты бомб | Глобальная перменная |
| dx[8] = { 0,0,1,-1,1,1,-1,-1 } | int[] | Константы для изменения индексов ячеек по горизонтали при проверке пустых ячеек | Глобальная перменная |
| dy[8] = { 1,-1,1,1,0,-1,-1,0 } | int[] | Константы для изменения индексов ячеек по вертикали при проверке пустых ячеек | Глобальная перменная |

### 3.2.7 Структура данных алгоритма Create\_map()

Таблица 8 – Структура данных алгоритма Create\_map()

|  |  |  |  |
| --- | --- | --- | --- |
| Элементы данных | Рекомендуемый тип | Назначение | Комментарии |
| x | int | Координата ячейки с бомбой | Локальная переменная |
| y | int | Координата ячейки с бомбой | Локальная переменная |
| map | BYTE \*\* | Массив значений карты бомб | Глобальная перменная |
| XCellNum | int | Кол-во ячеек по горизонтали | Глобальная переменная |
| YCellNum | int | Кол-во ячеек по вертикали | Глобальная перменная |
| count | int | Счетчик цифры-подсказки | Локальная переменная |



### Описание графических компонентов основного окна



Рисунок 3.1 – Вид окна приложения

Составляющие формы главного окна Minesweeper:

–Кнопки в виде ячеек, при нажатии на которые ячейка либо открывается, либо ставится флаг(если нажать левой кнопкой мыши)

После нажатия на левую кнопку мыши ячейки открываются. Если попалась пустая чейка, то окрываются еще все пустые вокруг

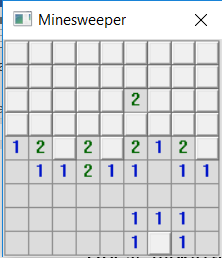


Рисунок 3.2 – Открытые ячейки

При нажатии на правую кнопку мыши выставляется флаг

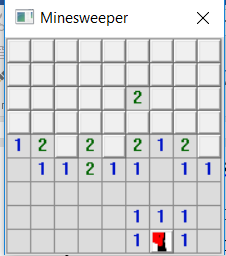


Рисунок 3.3 – Выставлен флаг

## 3.3 Схемы алгоритмов решения задачи



Рисунок 3.3 – Создание карты бомб



Рисунок 3.4 – Прорисовка поля



Рисунок 3.5 – Открытие пустых клеток

# 4 Тестирование и проверка работоспособности программного средства

В ходе тестирования были рассмотрены в действии все функциональные возможности программы «Сапер». Было изучено поведение программы при различных действиях пользователя.

Таблица 9 – Тест 1

|  |  |
| --- | --- |
| Тестовая ситуация | Запуск приложения |
| Ожидаемый результат: | Начало игры |
| Фактический результат: |  |

Таблица 10 – Тест 2

|  |  |
| --- | --- |
| Тестовая ситуация | Ход игрока левой кнопкой мыши |
| Ожидаемый результат: | Открытие клеток/ Проигрыш |
| Фактический результат: |  |

Таблица 11 – Тест 3

|  |  |
| --- | --- |
| Тестовая ситуация: | Ход игрока правой кнопкой мыши |
| Ожидаемый результат: | Установка флага |
| Фактический результат: |  |

Таблица 12 – Тест 4

|  |  |
| --- | --- |
| Тестовая ситуация: | Нажатие левой кнопкой мыши на кнопку с флагом |
| Ожидаемый результат: | Нет реакции |
| Фактический результат: |  |

Таблица 13 – Тест 5

|  |  |
| --- | --- |
| Тестовая ситуация: | Попадание на бомбу |
| Ожидаемый результат: | Сообщение о проигрыше |
| Фактический результат: |  |

Таблица 14 – Тест 6

|  |  |
| --- | --- |
| Тестовая ситуация: | Прохождение игры |
| Ожидаемый результат: | Сообщение о победе |
| Фактический результат: |  |

Таблица 15 – Тест 7

|  |  |
| --- | --- |
| Тестовая ситуация: | Двойной щелчок для начала новой игры |
| Ожидаемый результат: | Новая игра |
| Фактический результат: |  |

# 5 Руководство по установке и использованию программного средства

## 5.1 Шаг 1. Установка

Для открытия приложения необходимо распаковать архив и запустить файл Minesweeper.exe. Далее появится главное окно программы, представленное на рисунке 5.1.



Рисунок 5.1 – Главное окно

## 5.2 Шаг 2. Знакомство с программой. Инструкция по эксплуатации

При нажатии на правую кнопку мыши выставится флаг, теперь ячейку под флагом нельзя открыть. Чтобы урать флаг, снова нажмите правую кнопку мыши.

Для открытия ячейки нажмите левую кнопку мыши.

Чтобы начать новую игру, дважды щелкните по полю.

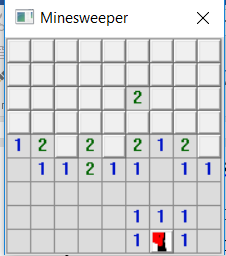


Рисунок 5.2– Окно в процессе игры

Заключение

В результате выполнения данной работы был получен игровой приложение, которое позволяет играть в классический «Сапер». Были реализованы механизмы генерации карты бомб, прорисовки поля, создания массива ячеек-кнопок, открытия пустых ячеек, процедуры обработки пользовательских действий. Код выполнен с помощью языка программирования С++. Также были изучены и применены различные базовые функции и интерфейсы для разработки приложений под платформу Windows API.

Список использованной литературы

[1] Э. Таненбаум – Операционные системые. Четвёртое издание, 2015г. – 1100с.

[2] Ч. Петцольд – Programming Windows, 1998г. – 1100с.

[3] MSDN [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/en-us/windows/win32/apiindex/windows-api-list>.

[4] First Steps [Сайт] – Режим доступа: <http://www.firststeps.ru/mfc/winapi/r.php?1>.

[5] Vsokovikov [Сайт] – Режим доступа: <http://vsokovikov.narod.ru/New_MSDN_API/Win_class>.

.

Приложение А

(обязательное)

Исходный код программы

#define CELL\_SIDE\_LENGTH 20

#define IDB\_TOP\_Cell\_Click 1111

#define IDB\_BOTTON\_Cell\_Click 2222

#include <windows.h>

#include <malloc.h>

#include <math.h>

#include <iomanip>

HINSTANCE hInst;

HWND hWnd;

HBRUSH hBrush;

HPEN hPen;

int Cmd;

int XCellNum = 9, YCellNum = 9, BombsNum = 10;

BYTE \*\*map;

HWND \*\*cellhWnd;

BOOL GAME\_IN\_PROCESS = false;

int dx[8] = { 0,0,1,-1,1,1,-1,-1 };

int dy[8] = { 1,-1,1,1,0,-1,-1,0 };

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam);

ATOM InitializeMainWnd(HINSTANCE hInstance);

void Draw\_new\_field(HDC hdc);

void Create\_map();

COLORREF Choose\_color(int cell);

void Delete\_Cells();

void Clean\_field();

void Delete\_cell(int x, int y);

void Check\_near\_cells(int x, int y);

int APIENTRY WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)

{

hInst = hInstance;

Cmd = nCmdShow;

InitializeMainWnd(hInstance);

hWnd = CreateWindow("HelloWorldClass", "Minesweeper", WS\_OVERLAPPEDWINDOW & ~WS\_MAXIMIZEBOX & ~WS\_MINIMIZEBOX & ~WS\_SIZEBOX, CW\_USEDEFAULT, 0,

(CELL\_SIDE\_LENGTH + 1) \* XCellNum + 2, (CELL\_SIDE\_LENGTH + 1) \* YCellNum + GetSystemMetrics(SM\_CYSIZE) + 2, NULL, NULL, hInstance, NULL);

ShowWindow(hWnd, nCmdShow);

UpdateWindow(hWnd);

MSG msg;

while (GetMessage(&msg, NULL, 0, 0))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

return (int)msg.wParam;

}

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)

{

switch (message)

{

case WM\_COMMAND:

{

if (wParam == IDB\_TOP\_Cell\_Click)

{

HWND hCurrWnd = (HWND)lParam;

int x = (int)GetProp(hCurrWnd, "ind\_x");

int y = (int)GetProp(hCurrWnd, "ind\_y");

if ((wParam & MK\_LBUTTON) == MK\_LBUTTON)

{

if (!(bool)GetProp(cellhWnd[x][y], "flag"))

{

if ((bool)GetProp(cellhWnd[x][y], "bomb")) {

MessageBox(NULL, "You lose!", "Game end", MB\_ICONEXCLAMATION | MB\_OK);

GAME\_IN\_PROCESS = false;

DeleteObject(map);

Delete\_Cells();

}

else

{

Check\_near\_cells(x,y);

if (BombsNum == 0)

{

Clean\_field();

MessageBox(NULL, "You win!", "Game end", MB\_ICONEXCLAMATION | MB\_OK);

}

}

}

}

}

if (wParam == IDB\_BOTTON\_Cell\_Click)

{

HWND hCurrWnd = (HWND)lParam;

int x = (int)GetProp(hCurrWnd, "ind\_x");

int y = (int)GetProp(hCurrWnd, "ind\_y");

if (!GetProp(cellhWnd[x][y], "flag"))

{

SetProp(cellhWnd[x][y], "flag", (HANDLE)true);

if ((bool)GetProp(cellhWnd[x][y], "bomb"))

BombsNum--;

HBITMAP hbFlag = (HBITMAP)LoadImage(hInst, "flag.bmp", IMAGE\_BITMAP, 0, 0, LR\_LOADFROMFILE);

SendMessage(cellhWnd[x][y], BM\_SETIMAGE, IMAGE\_BITMAP, (LPARAM)hbFlag);

}

else {

if ((bool)GetProp(cellhWnd[x][y], "bomb"))

BombsNum++;

SetProp(cellhWnd[x][y], "flag", (HANDLE)false);

SendMessage(cellhWnd[x][y], BM\_SETIMAGE, IMAGE\_BITMAP, 0);

}

}

}

break;

case WM\_CONTEXTMENU:

{

RECT wind\_param;

GetWindowRect(hWnd, &wind\_param);

int xPos = (int)LOWORD(lParam) - wind\_param.left - 2;

int yPos = (int)HIWORD(lParam) - wind\_param.top - GetSystemMetrics(SM\_CYCAPTION) - 2;

int x = (int)floor(xPos / (CELL\_SIDE\_LENGTH));

int y = (int)floor(yPos/ (CELL\_SIDE\_LENGTH));

if(cellhWnd[x][y]!= NULL)

SendMessage(hWnd, WM\_COMMAND, IDB\_BOTTON\_Cell\_Click, (LPARAM)cellhWnd[y][x]);

}

break;

case WM\_LBUTTONDBLCLK:

{

if (GAME\_IN\_PROCESS)

{

DeleteObject(map);

Delete\_Cells();

}

SendMessage(hWnd, WM\_CREATE, wParam, lParam);

}

break;

case WM\_DESTROY:

{

if (GAME\_IN\_PROCESS) {

DeleteObject(map);

Delete\_Cells();

}

PostQuitMessage(0);

}

break;

case WM\_CREATE:

{

GAME\_IN\_PROCESS = true;

Create\_map(XCellNum, YCellNum, BombsNum);

cellhWnd = (HWND\*\*)malloc(YCellNum \* sizeof(HWND\*));

for (int i = 0; i < YCellNum; i++)

{

cellhWnd[i] = (HWND\*)malloc(XCellNum \* sizeof(HWND));

}

for (int i = 0; i < XCellNum; i++)

for (int j = 0; j < YCellNum; j++)

{

cellhWnd[i][j] = CreateWindow("button", "", WS\_CHILD | WS\_VISIBLE | BS\_PUSHBUTTON | BS\_BITMAP, j \* (CELL\_SIDE\_LENGTH - 1) + 2, i \* (CELL\_SIDE\_LENGTH - 1) + 2, CELL\_SIDE\_LENGTH - 1, CELL\_SIDE\_LENGTH - 1, hWnd, (HMENU)IDB\_TOP\_Cell\_Click, hInst, NULL);

SetProp(cellhWnd[i][j], ("ind\_x"), (HANDLE)i);

SetProp(cellhWnd[i][j], ("ind\_y"), (HANDLE)j);

SetProp(cellhWnd[i][j], ("flag"), (HANDLE)false);

if (map[i][j] == 9)

{

SetProp(cellhWnd[i][j], ("bomb"), (HANDLE)true);

}

else

SetProp(cellhWnd[i][j], ("bomb"), (HANDLE)false);

}

}

case WM\_PAINT:

{

PAINTSTRUCT ps;

HDC hdc = BeginPaint(hWnd, &ps);

Draw\_new\_field(hdc, XCellNum, YCellNum, BombsNum);

EndPaint(hWnd, &ps);

}

break;

default:

return DefWindowProc(hWnd, message, wParam, lParam);

}

return 0;

}

void Draw\_new\_field(HDC hdc)

{

hPen = CreatePen(PS\_SOLID, 1, RGB(140, 140, 140));

SelectObject(hdc, hPen);

hBrush = CreateSolidBrush(RGB(220, 220, 220));

SelectObject(hdc, hBrush);

int y = 1;

int x = 1;

for (int i = 0; i < XCellNum; i++) {

for (int j = 0; j < YCellNum; j++)

{

Rectangle(hdc, x, y, x + CELL\_SIDE\_LENGTH, y + CELL\_SIDE\_LENGTH);

RECT rt;

SetRect(&rt, x + 1, y + 1, x + CELL\_SIDE\_LENGTH - 1, y + CELL\_SIDE\_LENGTH - 1);

SetBkMode(hdc, TRANSPARENT);

if ((map[i][j] > 0) & (map[i][j] < 9)) {

SetTextColor(hdc, Choose\_color(map[i][j]));

TCHAR s[2];

\_itoa\_s(map[i][j], s, 16);

DrawText(hdc, s , -1, &rt, DT\_CENTER | DT\_VCENTER);

}

if (map[i][j] == 9) {

HBITMAP hbbomb = (HBITMAP)LoadImage(hInst, "bomb.bmp", IMAGE\_BITMAP, 0, 0, LR\_LOADFROMFILE);

HDC hMemDC = CreateCompatibleDC(hdc);

BITMAP info;

GetObject(hbbomb, sizeof(info), &info);

SelectObject(hMemDC, hbbomb);

TransparentBlt(hdc, x, y, CELL\_SIDE\_LENGTH-1, CELL\_SIDE\_LENGTH-1, hMemDC, 0, 0, info.bmWidth, info.bmHeight, RGB(255,255,255));

}

x += CELL\_SIDE\_LENGTH - 1;

}

x = 1;

y += CELL\_SIDE\_LENGTH - 1;

}

return;

}

void Create\_map()

{

int pi, pj;

srand(time(0));

map = (byte\*\*)malloc(XCellNum \* sizeof(byte\*));

for (int i = 0; i < YCellNum; i++)

{

map[i] = (byte\*)malloc(XCellNum \* sizeof(byte));

}

for (int i = 0; i < XCellNum; i++)

for (int j = 0; j < YCellNum; j++)

{

map[i][j] = 0;

}

for (int i = 0; i < BombsNum; i++)

{

do {

pi = rand() % XCellNum;

pj = rand() % YCellNum;

} while (map[pi][pj] == 9);

map[pi][pj] = 9;

}

for (int i = 0; i < XCellNum; i++)

for (int j = 0; j < YCellNum; j++)

{

if (map[i][j] != 9)

{

int count = 0;

if (i != 0 && j != 0 && map[i - 1][j - 1] == 9)

count++;

if (i != 0 && map[i - 1][j] == 9)

count++;

if (j != 0 && map[i][j - 1] == 9)

count++;

if (i != XCellNum - 1 && j != YCellNum - 1 && map[i + 1][j + 1] == 9)

count++;

if (i != XCellNum - 1 && map[i + 1][j] == 9)

count++;

if (j != YCellNum - 1 && map[i][j + 1] == 9)

count++;

if (i != XCellNum - 1 && j != 0 && map[i + 1][j - 1] == 9)

count++;

if (i != 0 && j != YCellNum - 1 && map[i - 1][j + 1] == 9)

count++;

map[i][j] = count;

}

}

return;

}

ATOM InitializeMainWnd(HINSTANCE hInstance) {

WNDCLASSEX wcex;

wcex.cbSize = sizeof(WNDCLASSEX);

wcex.style = CS\_DBLCLKS;

wcex.lpfnWndProc = WndProc;

wcex.cbClsExtra = 0;

wcex.cbWndExtra = 0;

wcex.hInstance = hInstance;

wcex.hIcon = LoadIcon(NULL, IDI\_APPLICATION);

wcex.hCursor = LoadCursor(NULL, IDC\_ARROW);

hBrush = CreateSolidBrush(RGB(220, 220, 220));

wcex.hbrBackground = hBrush;

wcex.lpszMenuName = NULL;

wcex.lpszClassName = "HelloWorldClass";

wcex.hIconSm = wcex.hIcon;

return RegisterClassEx(&wcex);

}

COLORREF Choose\_color(int cell)

{

COLORREF rgb;

switch (cell) {

case 1:

rgb = RGB(0, 20, 200);

break;

case 2:

rgb = RGB(0, 100, 0);

break;

case 3:

rgb = RGB(255, 0, 0);

break;

case 4:

rgb = RGB(0, 0, 120);

break;

case 5:

rgb = RGB(120, 0, 0);

break;

case 6:

rgb = RGB(0, 100, 170);

break;

case 7:

rgb = RGB(120, 0, 130);

break;

case 8:

rgb = RGB(50, 0, 0);

break;

}

return rgb;

}

void Delete\_Cells() {

for (int i = 0; i < XCellNum; i++)

for (int j = 0; j < YCellNum; j++)

{

if (cellhWnd[i][j] != NULL)

{

RemoveProp(cellhWnd[i][j], "flag");

RemoveProp(cellhWnd[i][j], "bomb");

RemoveProp(cellhWnd[i][j], "ind\_x");

RemoveProp(cellhWnd[i][j], "ind\_y");

DestroyWindow(cellhWnd[i][j]);

cellhWnd[i][j] = NULL;

}

}

return;

}

void Delete\_cell(int x, int y) {

RemoveProp(cellhWnd[x][y], "flag");

RemoveProp(cellhWnd[x][y], "bomb");

RemoveProp(cellhWnd[x][y], "ind\_x");

RemoveProp(cellhWnd[x][y], "ind\_y");

DestroyWindow(cellhWnd[x][y]);

cellhWnd[x][y] = NULL;

return;

}

void Clean\_field() {

for (int i = 0; i < XCellNum; i++)

for (int j = 0; j < YCellNum; j++)

{

if ((cellhWnd[i][j] != NULL) && (!(bool)GetProp(cellhWnd[i][j], "bomb")))

{

RemoveProp(cellhWnd[i][j], "flag");

RemoveProp(cellhWnd[i][j], "bomb");

RemoveProp(cellhWnd[i][j], "ind\_x");

RemoveProp(cellhWnd[i][j], "ind\_y");

DestroyWindow(cellhWnd[i][j]);

cellhWnd[i][j] = NULL;

}

}

return;

}

void Check\_near\_cells(int x, int y)

{

if (cellhWnd[x][y] == NULL) return;

Delete\_cell(x, y);

if (map[x][y] == 0)

{

for (int k = 0; k < 8; k++)

{

if (( (x + dx[k])>= 0) && ((x + dx[k] < XCellNum)) && (((y + dy[k]) >=0) && ((y +dy[k]) < YCellNum)))

Check\_near\_cells(x + dx[k], y + dy[k]);

}

}

}

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Обозначение | | | | Наименование | | | | Дополнительные сведения | | | |
|  | | | | Текстовые документы | | | |  | | | |
|  | | | |  | | | |  | | | |
| БГУИР КП 1–40 01 01 025 ПЗ | | | | Пояснительная записка | | | | 36 с. | | | |
|  | | | |  | | | |  | | | |
|  | | | |  | | | |  | | | |
|  | | | | Графические документы | | | |  | | | |
|  | | | |  | | | |  | | | |
| БГУИР 751003 008 СП | | | | Игра «Сапер». | | | | Формат А1 | | | |
|  | | | |  | | | |  | | | |
|  | | | |  | | | |  | | | |
|  | | | |  | | | |  | | | |
|  | | | |  | | | |  | | | |
|  | | | |  | | | |  | | | |
|  | | | |  | | | |  | | | |
|  | | | |  | | | |  | | | |
|  | | | |  | | | |  | | | |
|  | | | |  | | | |  | | | |
|  | | | |  | | | |  | | | |
|  |  |  |  |  | БГУИР КП 1-40 01 01 025 Д1 | | | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
| Изм. | Л. | № докум. | Подп. | Дата | Игровое приложение «Сапер» Ведомость курсового  проекта |  | | | | Лист | Листов |
| Разраб. | | Стубеда В.Д |  | 30.11.19 |  |  |  | |  | 37 | 37 |
| Пров. | | Базылев Е.Н. |  | 30.11.19 |  | Кафедра ПОИТ  гр. 751003 | | | | | |