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

по курсовому проекту

на тему

**ИГРОВОЕ ПРИЛОЖЕНИЕ «2048»**

БГУИР КП 1-40 01 01 628 ПЗ

Выполнил

студент: гр. 851006 Шустал А.Ю.

Проверил: Жиденко А.Л.

Минск 2020

Учреждение образования

«Белорусский государственный университет информатики и

радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой ПОИТ

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

(подпись)

Лапицкая Н.В. 2020г.

ЗАДАНИЕ

по курсовому проектированию

Студенту *Шусталу Алексею Юрьевичу* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Тема работы *Игровое приложение «2048» \_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

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

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

2. Срок сдачи законченной работы *03.12.2020г.*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Исходные данные к работе *Среда программирования VisualStudio. Язык программирования C++. Интерфейс программирования Windows API. Реализовать следующий функционал: загрузку таблицы лидеров из файла. Игровое поле с возможностью перемещения содержимого клеток в сторону, соответствующую клавише. Предусмотреть масштабирование окна. Реализовать систему генерации нового содержимого клеток. В случае, если закончились попытки или уровни – необходимо дать пользователю возможность сохранить своё имя и счёт в таблице лидеров. Реализовать возможность посмотреть список лидеров. \_*

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

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

*1 Анализ предметной области* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

*4 Руководство по 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 *Жиденко А.Л.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

7.Дата выдачи задания *26.10.2020г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и процентом от общего объёма работы):

*Раздел 1. Введение к 31.10.2020г. – 10 % готовности работы;\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Раздел 2 к 7.11.2020г.– 30% готовности работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Раздел 3 к 9.11.2020г. – 60% готовности работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Раздел 4 к 19.11.2020г. – 80% готовности работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Раздел 5.Заключение. Приложения к 23.11.2020г. – 90% готовности работы;*

*оформление пояснительной записки и графического материала к 27.11.2020г. – 100% готовности работы.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Защита курсового проекта с 3.12.2020г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

РУКОВОДИТЕЛЬ *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Жиденко А.Л.*

*(подпись)*

Задание принял к исполнению *\_\_\_\_\_\_\_\_\_\_\_Шустал А.Ю. 26.10.2020г.*

*(дата и подпись студента)*

Содержание

[Введение 5](#_Toc57843711)

[1 Анализ предметной области 6](#_Toc57843712)

[1.1 Информация об игре «2048» 6](#_Toc57843713)

[1.2 Анализ существующих приложений 6](#_Toc57843714)

[2 Постановка задачи 13](#_Toc57843717)

[3 Разработка программного средства 14](#_Toc57843718)

[3.1 Информационная база задачи 14](#_Toc57843719)

[3.2 Графический интерфейс 15](#_Toc57843730)

[3.3 Принцип работы таблицы лидеров 17](#_Toc57843735)

[3.4 Принцип работы программы 17](#_Toc57843736)

[4 Руководство по использованию программного средства 18](#_Toc57843737)

[4.1 Системные требования 18](#_Toc57843738)

[4.2 Работа с программным средством 18](#_Toc57843739)

[Заключение 22](#_Toc57843742)

[Список использованной литературы 23](#_Toc57843743)

[Приложение А 24](#_Toc57843744)

Введение

Компьютерные игры являются очень важной частью жизни современного человека, ведь почти каждый человек вырос, играя в компьютерные игры. Первые видеоигры были на игровых автоматах и приставках, из-за недоступности персональных компьютеров пользователю, однако, как только ПК стал более общедоступным, видеоигры нашли свой путь и туда. Внедрение графических интерфейсов во многом поспособствовало этому.

На сегодняшний день компьютерные игры на ПК очень популярны, в основном благодаря своей доступности. В современном мире превалируют игры, которые довольно просты в понимании и прохождении, которые помогут расслабиться или занять себя на короткий период времени. И одна из таких игр - это “2048”.

Игра “2048” является браузерной, которая была написана на языке программирования JavaScript в 2014 году. Сама игра представляет собой игровое поле с размерностью 4 х 4 состоит из квадратов, на котором появляются цифры 2 или 4. Цель игры – получить плитки с номиналом 2048, при желании можно продолжить дальше. Прообразом этой игры является коммерческая игра “Threes”, создатели которой были недовольны успехом “2048” и называли её “испорченным плагиатом”. Игра “2048” только за неделю после выпуска набрала более 33 миллионов посещений. Однако эта игра работает только в браузере, из-за языка программирования, на котором она была написана.

В результате вышеизложенного, было принято решение создать игровое приложение “2048” для компьютеров под управлением Windows с применением интерфейса программирования Windows API.

# 1 Анализ предметной области

## Информация об игре «2048»

Видеоигра игра «2048» была написана в качестве упражнения в программировании в 2014 году.

Принцип игры заключается в следующем:

В каждом раунде появляется плитка номинала 2 (с вероятностью 90%) или 4 (с вероятностью 10%). Нажатием стрелки игрок может скинуть все плитки игрового поля в одну из 4 сторон. Если при сбрасывании две плитки одного номинала налетают друг на друга, то они превращаются в одну, номинал которой равен сумме соединившихся плиток. После каждого хода на свободной секции поля появляется новая плитка номиналом 2 или 4. Если в одной строчке или столбце находится более двух плиток одного номинала, то при сбрасывании они начинают соединяться с той стороны, в которую были направлены. Например, после хода влево при строке (4, 4, 4) результат будет равен (8, 4). Данная обработка позволяет более точно формировать стратегию игры. За каждое соединение игровые очки увеличиваются на номинал получившейся плитки. Игра заканчивается поражением, если после очередного хода невозможно совершить действие.

Данная игра как было сказано выше, не является оригинальным проектом, так как принципы игры были заложены в игре «Threes», однако именно реализация в виде «2048» стала самой популярной, c огромным количеством аналогов данной игры.

## Анализ существующих приложений

### Игровое приложение «2048»

В данной реализации пользователю представлено игровое поле. Интерфейс интуитивно понятен и не вызывает сложностей, есть описание действий ниже игрового поля. Данная игра работает только в браузере.

В главном меню игры (см. рисунок 1.1) вам доступны следующие действия:

* перемещать клетки с помощью стрелок;
* начать новую игру, с помощью кнопки «New Game».

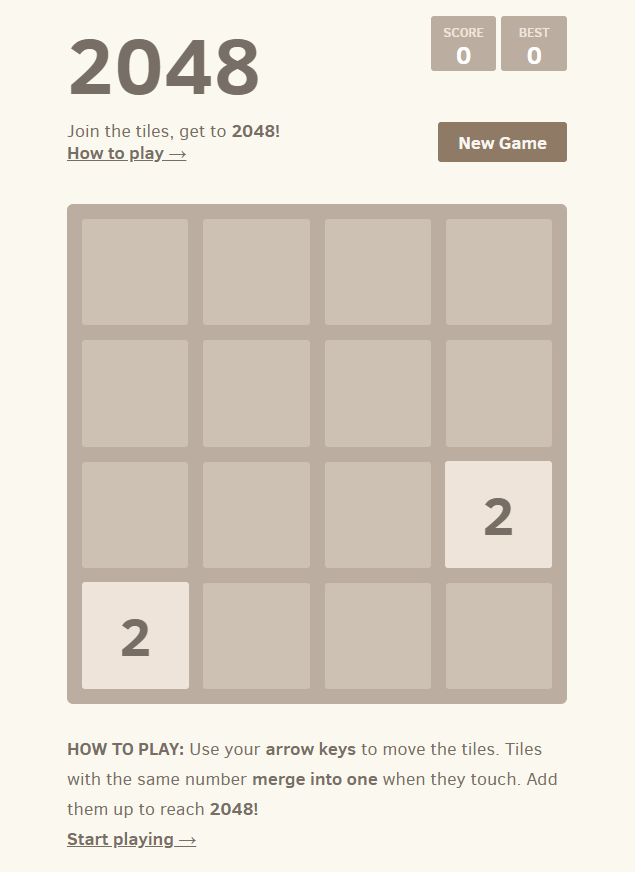


Рисунок 1.1 – Интерфейс главного меню «2048»

В ходе игрового процесса пользователь перемещает клетки в ту сторону, в которую пользователь нажал стрелку. При поражении (см. рисунок 1.2) и на экране появляется сообщение о проигрыше.

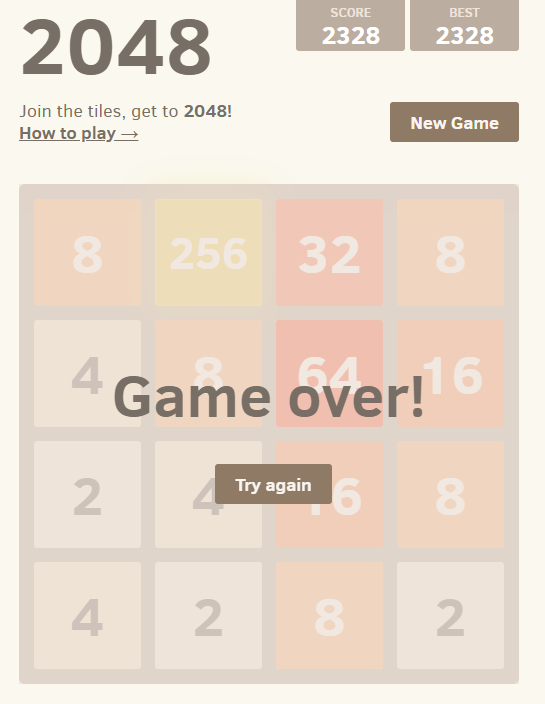


Рисунок 1.2 – Сообщение о проигрыше «Bricks DEMOLITION»

При нажатии на кнопку «New Game» поле очищается и появляются 2 клетки рисунок 1.3.

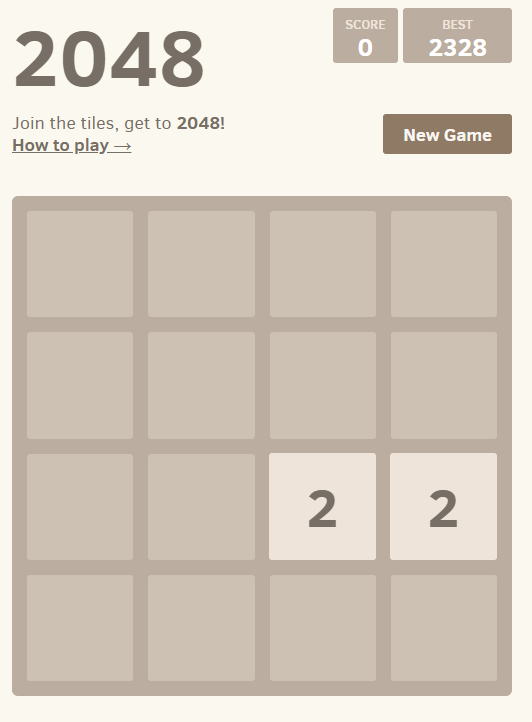


Рисунок 1.3 – Результат нажатия на «New Game»

Анализируя данное игровое приложения можно определить следующие недостатки:

* отсутствие таблицы лидеров.

К достоинствам данного приложения можно отнести:

* интуитивно понятный интерфейс;
* модель бесплатного распространения приложения.

### Игровое приложение «2048» (мобильное приложение)

Данная игра минимально отличается от компьютерной версии, в основном интерфейсом.

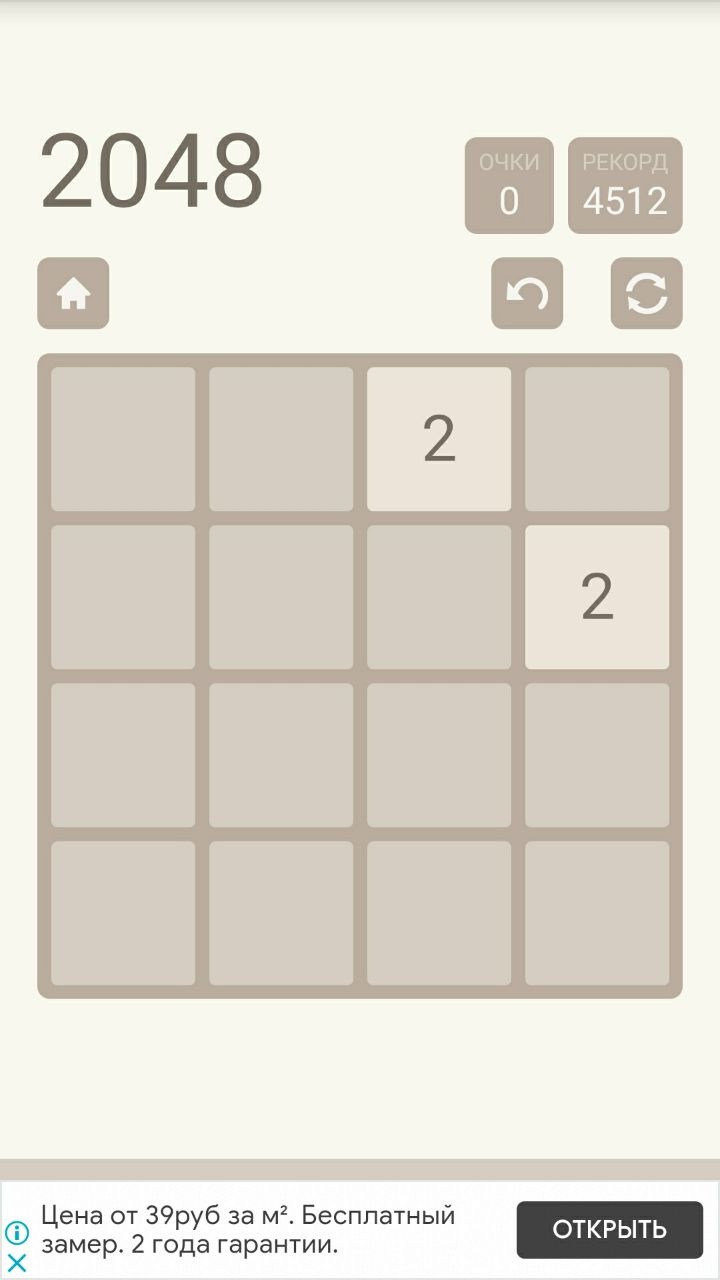


Рисунок 1.4 – Главное меню «2048» мобильной версии

В ходе игрового процесса пользователь перемещает клетки в ту сторону, в которую пользователь провёл палец. При поражении (см. рисунок 1.5) и на экране появляется сообщение о проигрыше.

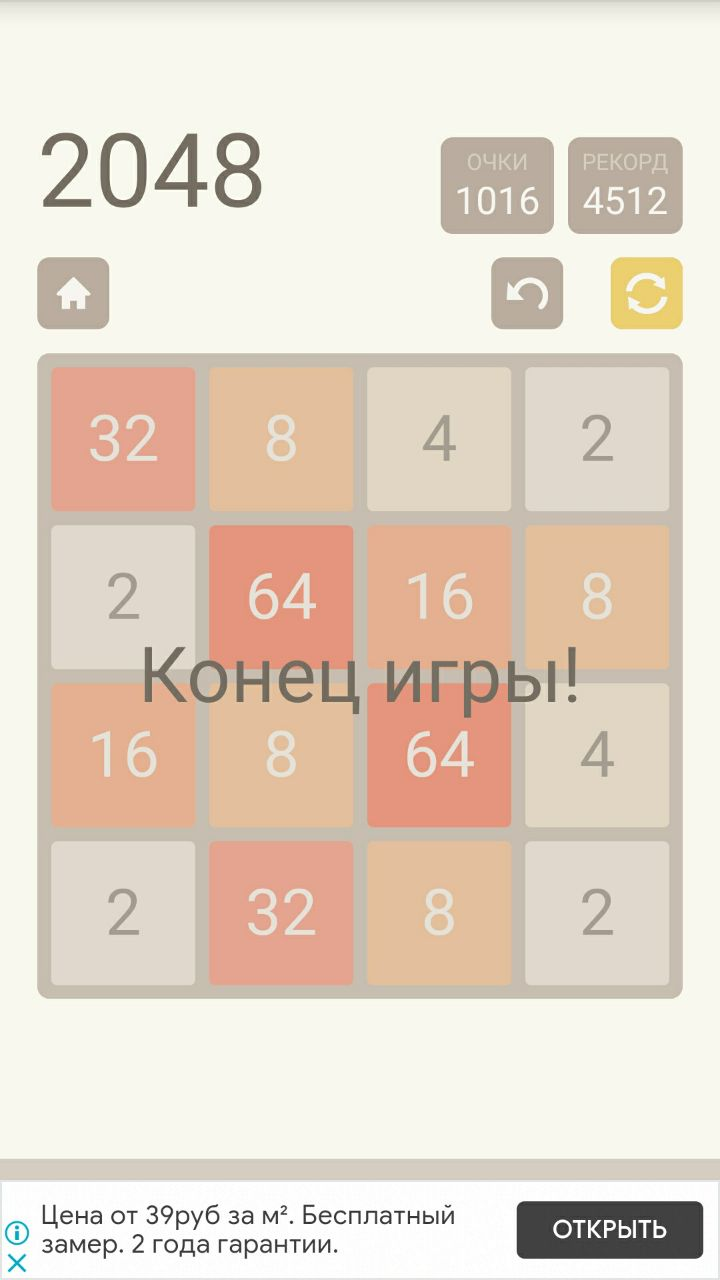


Рисунок 1.5 – Сообщение конца игры «2048»

При нажатии на кнопку с круговой стрелкой поле очищается и появляются 2 клетки (см. рисунок 1.6).

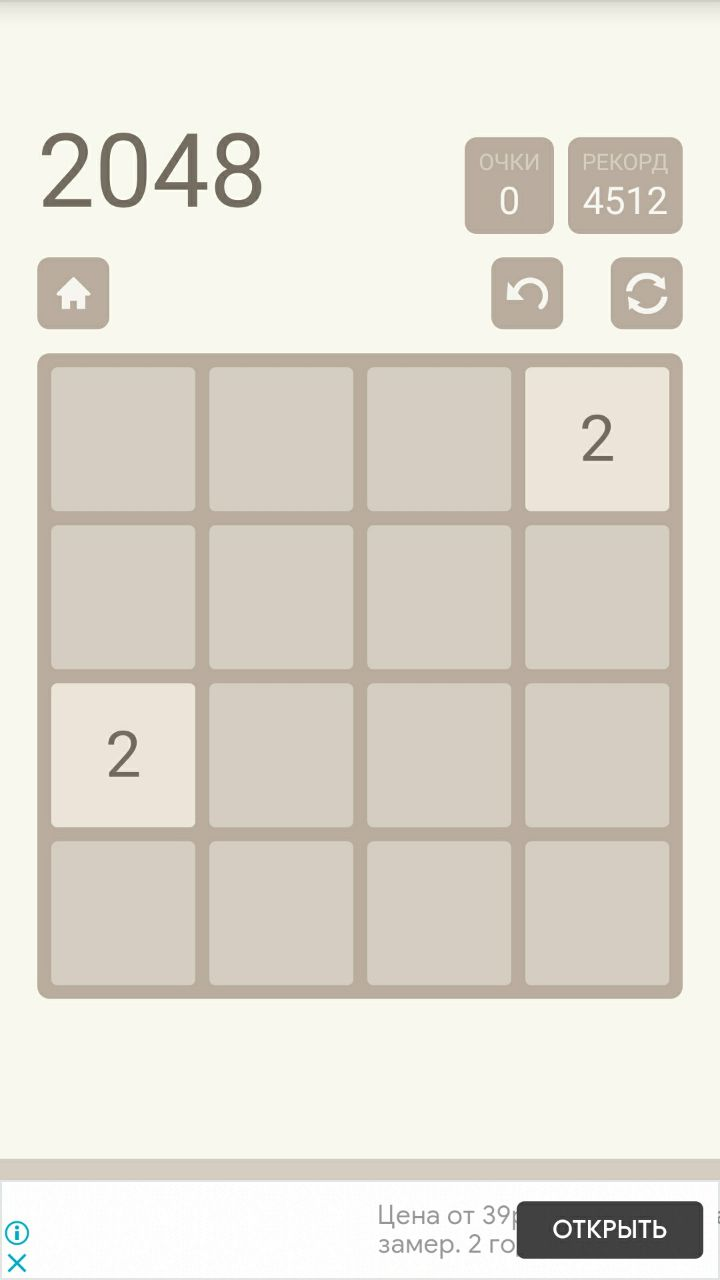


Рисунок 1.6 – Результат нажатия на кнопку с круговой стрелкой

Анализируя данное игровое приложения можно определить следующие недостатки:

* отсутствие таблицы лидеров;
* постоянное появление рекламы.

К достоинствам данного приложения можно отнести:

* интуитивно понятный интерфейс;
* модель бесплатного распространения приложения.

# Постановка задачи

Разработать игровое приложение на языке c++, с использованием интерфейса программирования Windows API, назначением которого является предоставление пользователю возможности, при наличии компьютера под управлением Windows, сыграть в игру «2048». В программе должно быть реализовано:

* игровое поле 4 х 4;
* возможность посмотреть таблицу лидеров;
* возможность ввести имя и сохранить результат в таблицу рекордов;
* возможность играть до поражения;
* управление клавиатурой;
* масштабирование интерфейса и игровой зоны под размеры окна.

При разработки игрового приложения использовать среду разработки VisualStudio. Таблицу лидеров необходимо сохранять в файл.

# Разработка программного средства

## Информационная база задачи

### Структура Cell

Данная структура отвечает за хранение областей и координат игрового поля самой игры.

### Функция MyRegisterClass

Данная функция регистрирует класс главного окна, структура которого в ней определена. Содержит структуру окна и дескриптор экземпляра приложения.

### Функция InitInstance

Данная функция создаёт главное окно и отображает его. Содержит дескриптор экземпляра приложения.

### Функция WndProc

Данная функция обрабатывает сообщения Windows для основного окна. Содержит цикл обработки сообщений, дескриптор окна.

### Функция About

Данная функция обрабатывает сообщения Windows для окна “О программе 2048”. Содержит дескриптор соответствующего диалогового окна.

### Функция ScoreBoard

Данная функция обрабатывает сообщения Windows для окна “Scoreboard”. Содержит дескриптор соответствующего диалогового окна.

### Функция TableGeneration

Данная функция служит для отрисовки игрового поля самой программы главного окна. Содержит дескриптор главного окна, структуры для отрисовки поля и его содержимого (такие как LOGBRUSH, HFONT).

### Функция ScoreBoardPrint

Данная функция служит для вывода таблицы лидеров на диалоговое окно. Содержит дескриптор диалогового окна.

### Файл исполняемого кода 2048.cpp

Содержит в себе точку входа и функции обработки сообщений к основному окну и диалоговому окну. Также содержит основную логику программы.

## Графический интерфейс

### Главное окно

При открытии приложения Вы увидите главное окно (см. рис. 3.3).

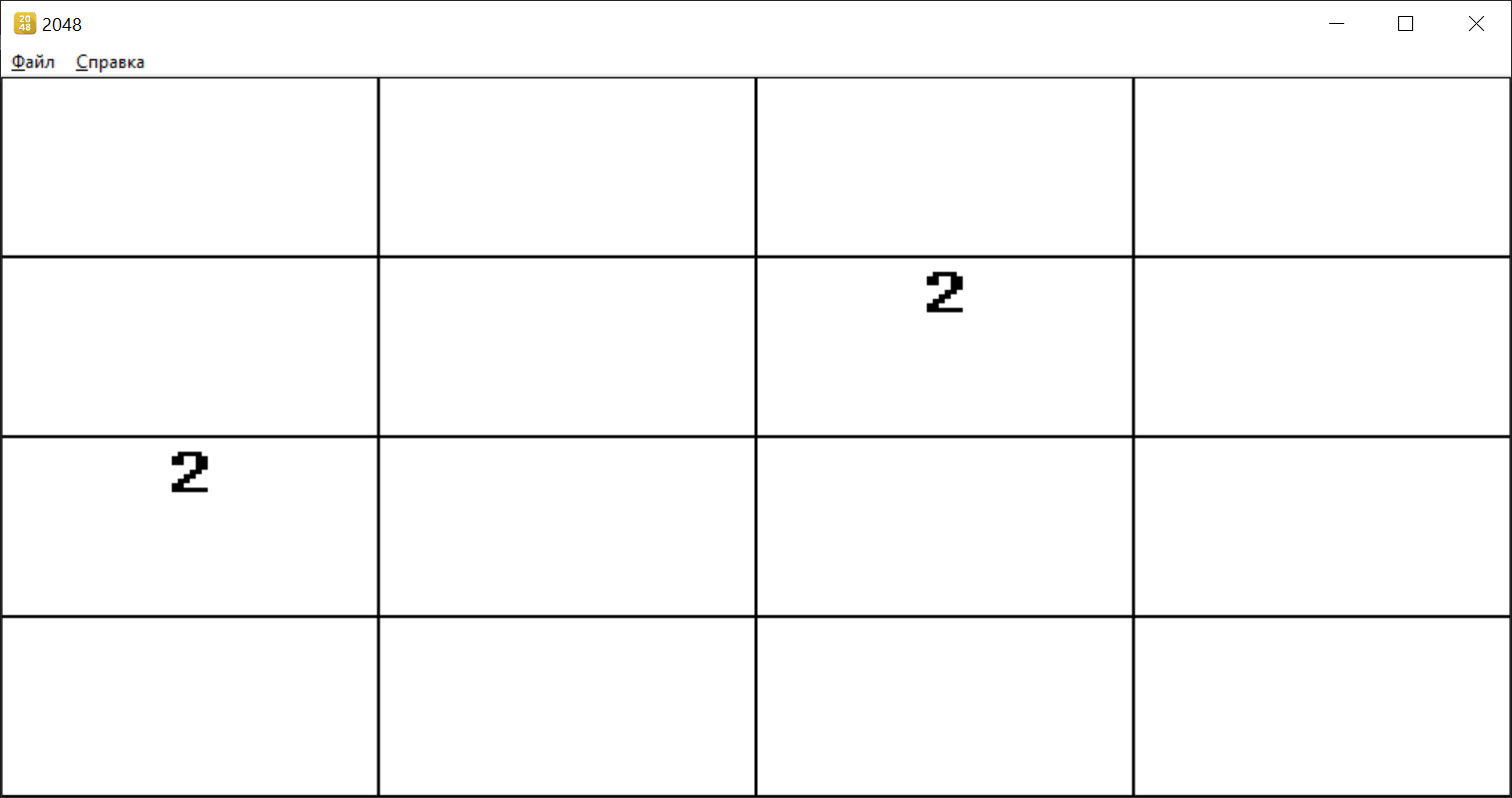


Рисунок 3.1 – Интерфейс главного окна

Составляющие главного окна:

* фон, масштабирующийся в зависимости от размеров окна (см. рисунок 3.4);
* игровое поле;
* меню главного окна.

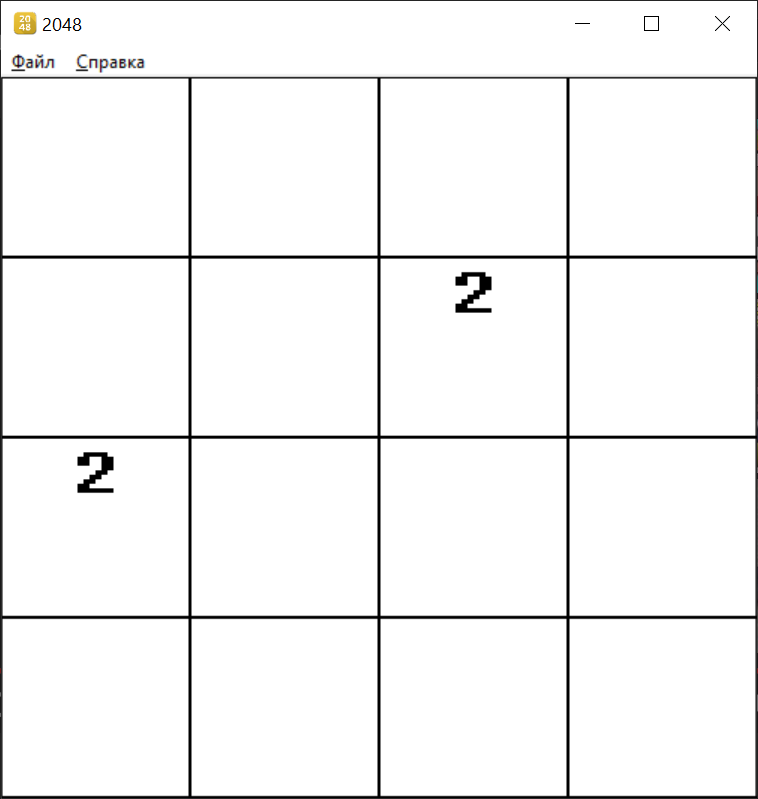


Рисунок 3.2 – Пример подстройки фона и игровой зоны под размеры окна

### Окно таблицы лидеров

При нажатии клавиши «Пробел» на главном окне появляется диалоговое окно с таблицей лидеров (см. рисунок 3.5).

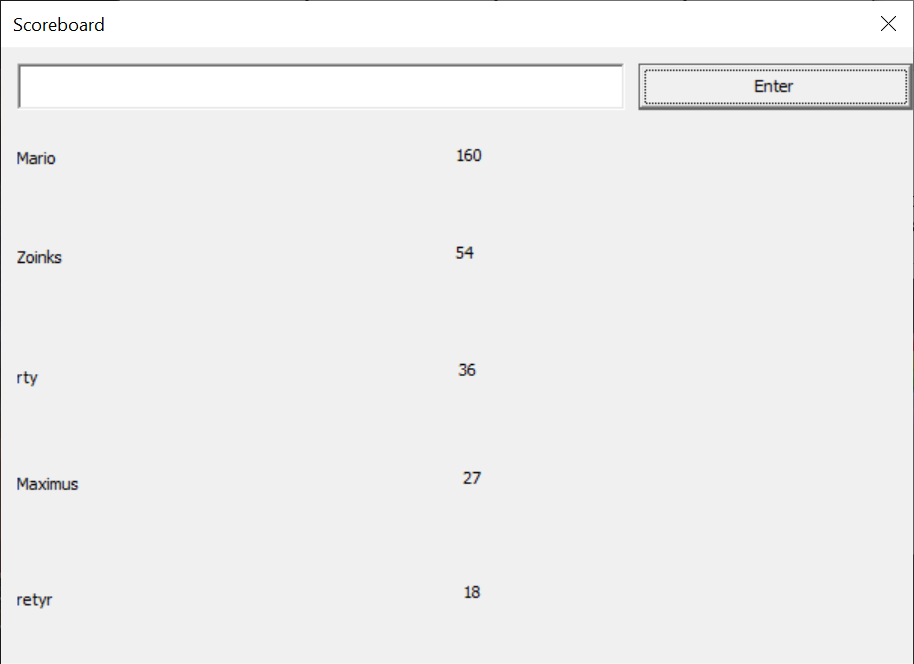


Рисунок 3.3 – Таблица лидеров

### Окно таблицы лидеров при вводе имени и нажатии на кнопку

При вводе имени и нажатии на «Enter», высчитывается Ваши очки и, если Вы попадаете в таблицу лидеров, записывает и выводит на диалоговое окно (см. рисунок 3.6).



Рисунок 3.4 – Обновлённая таблица лидеров

### Диалоговое окно «О программе 2048»

Диалоговое окно с информацией о приложении. (см. рисунок 3.7).

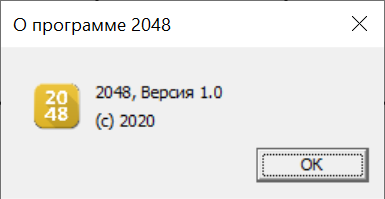


Рисунок 3.5 – Диалоговое окно «О программе 2048»

## Принцип работы таблицы лидеров

При нажатии кнопки «Пробел», с жесткого диска считывается таблица лидеров. В случае её отсутствия, – файл создается. Занесение в таблицу лидеров производится при нажатии на кнопку «Enter». Высчитываются очки пользователя и если больше чем в таблице, заносятся в соответствующее место вместе с именем. Затем выполняется перерисовка таблицы.

## Принцип работы программы

В начале работы программы генерируются два числа (2 или 4) и ставятся в случайные ячейки игрового поля. Затем при нажатии на стрелку на клавиатуре все ячейки с содержимым сдвигаются в соответствующую сторону. Если во время сдвига две ячейки с одинаковым содержимым находились рядом друг с другом, то происходит их слияние с суммой их содержимого. Если присуутствует ячейка с содержимым равным 2048, то программа оповещает пользователя о победе, а если же у пользователя закончилась возможность слияния ячеек, и всё поле занято, то программа оповещает пользователя о поражении.

# Руководство по использованию программного средства

## Системные требования

Для нормальной работы программного средства необходимы следующие минимальные системные требования:

* операционная система: Windows XP, Windows 7, Windows 8 или Windows 10;
* оперативная память 106 МБ;
* свободное место на жестком диске: 4 МБ.

## Работа с программным средством

### Запуск приложения

После запуска приложения появится главное окно, изображённое на рисунке 4.1

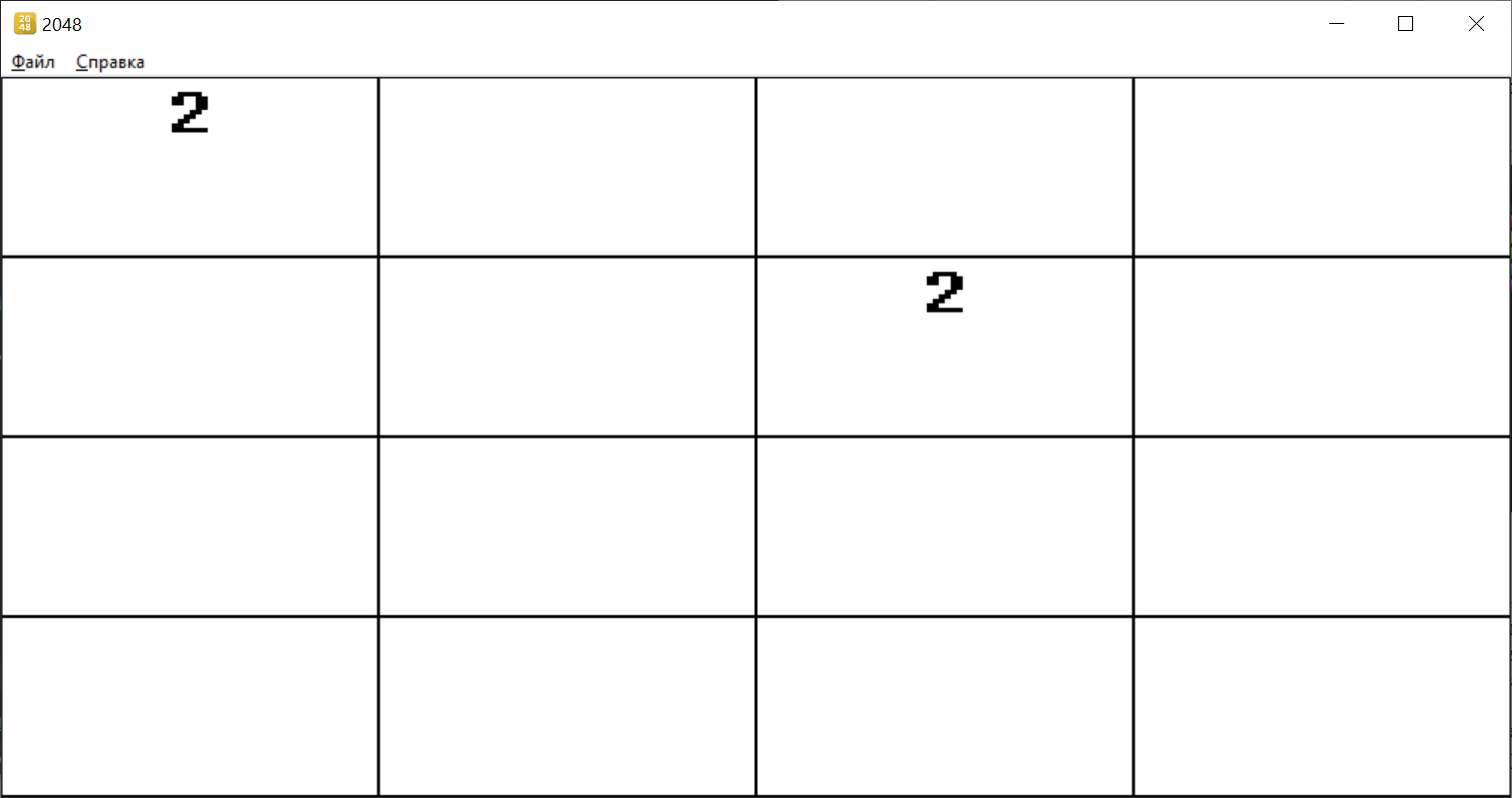


Рисунок 4.1 – Главное окно

На данном этапе работы с приложением, пользователь может перемещать ячейки с помощью стрелок на клавиатуре. По нажатию на «Пробел» откроется таблица лидеров, а по нажатию на «Справка» и «О программе…» откроется окно с информацией о приложении.

### Игровой процесс

При нажатии в сторону, где в ряд находятся две ячейки с одинаковым содержимым происходит слияние ячеек как показано на рисунках 4.2 и 4.3.

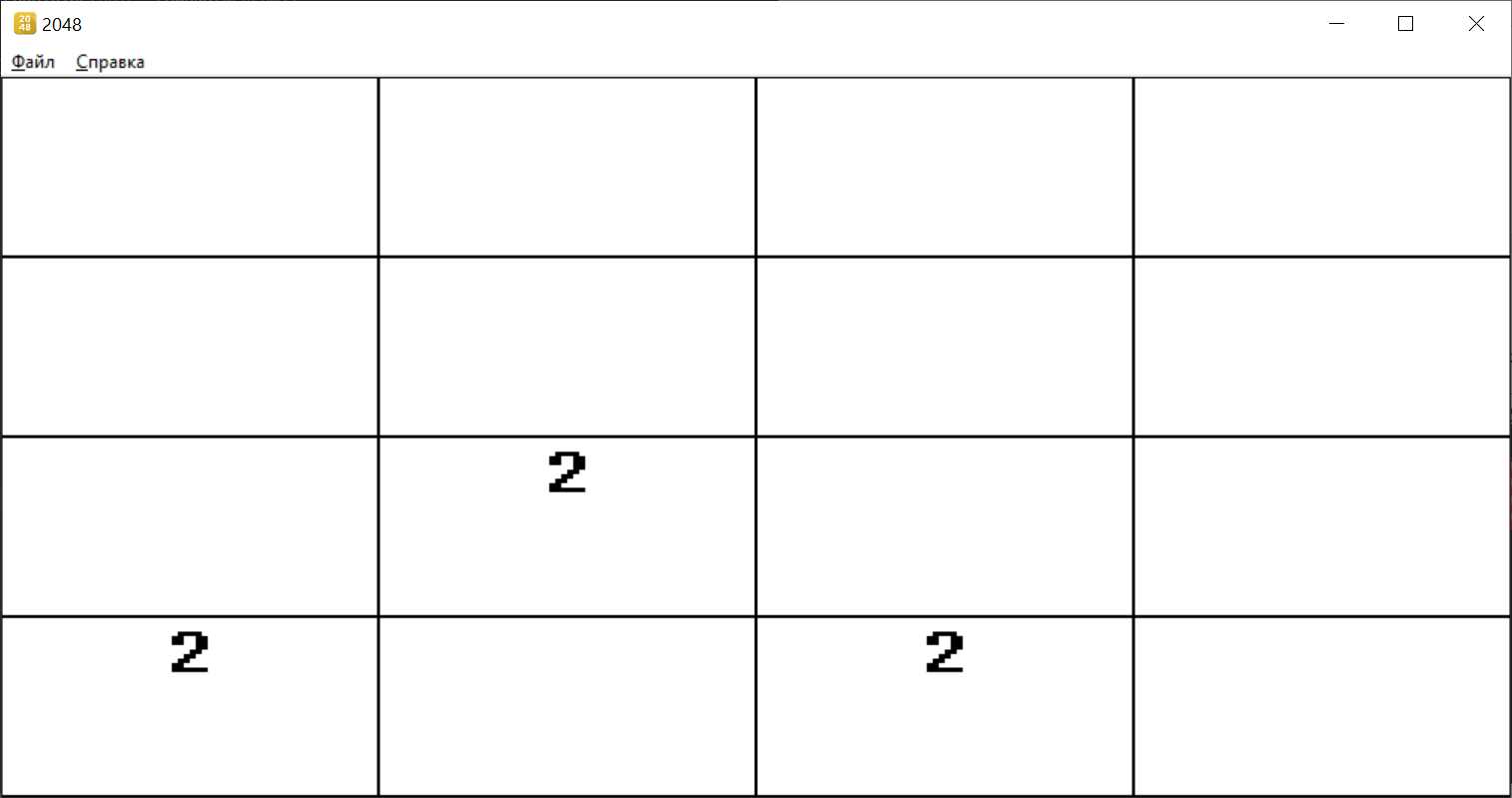


Рисунок 4.2 – Игровое поле до слияния

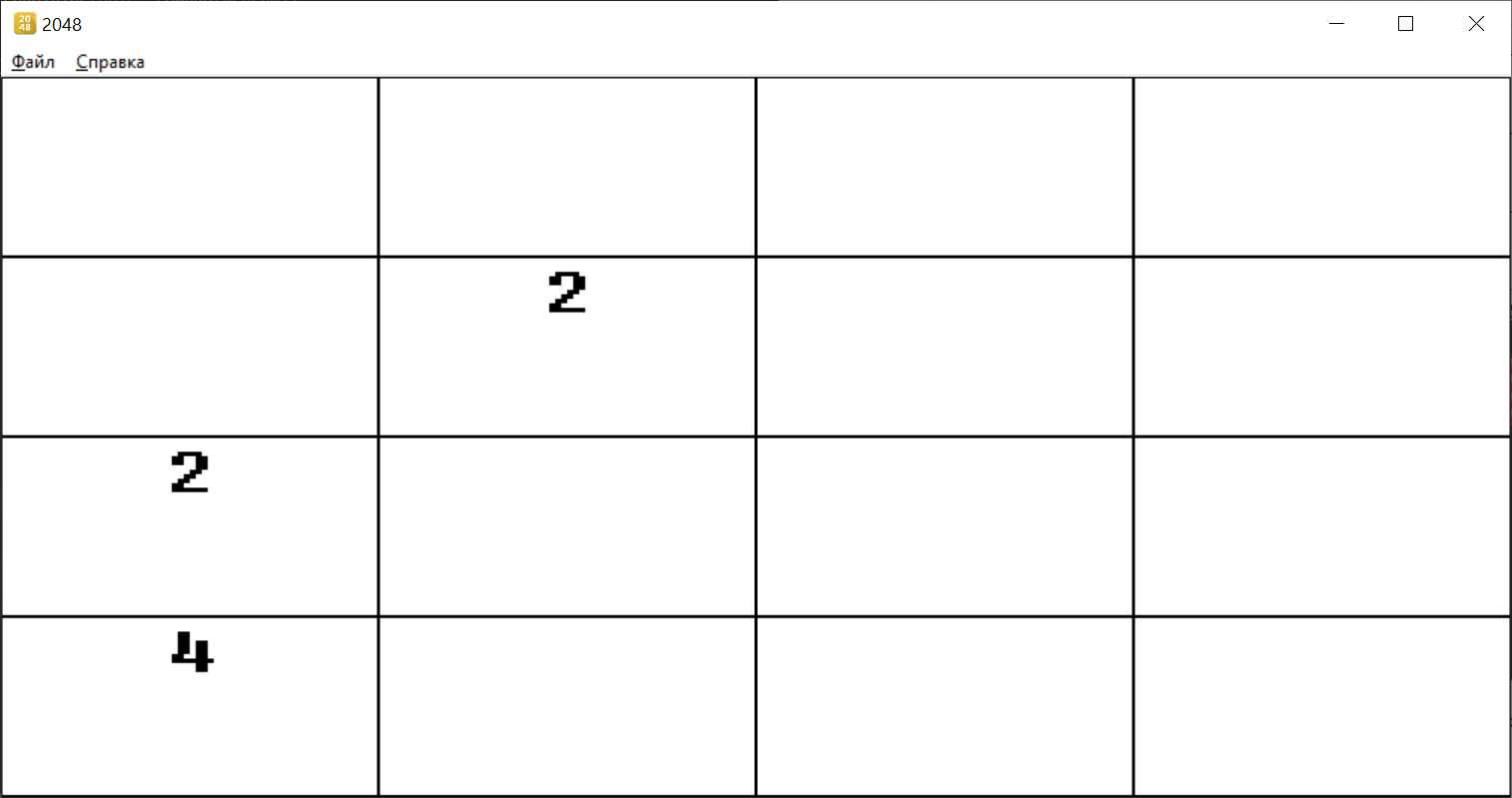


Рисунок 4.3 – Игровое поле после слияния

Следует отметить, что после каждого хода на поле случайным образом появляется в свободном месте новая ячейка (либо 2, либо 4).

При проигрыше, пользователь получает следующее сообщение (рисунок 4.4).

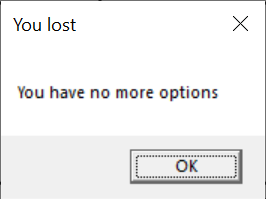


Рисунок 4.4 – Сообщение о поражении

Соответственно при победе, пользователь получает следующее сообщение (рисунок 4.5).

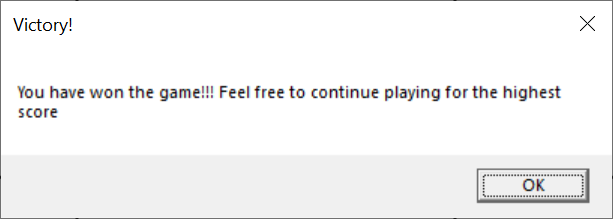


Рисунок 4.5 – Сообщение о победе

При нажатии на главном окне программы на «Пробел» появляется таблица рекордов (рисунок 4.6)

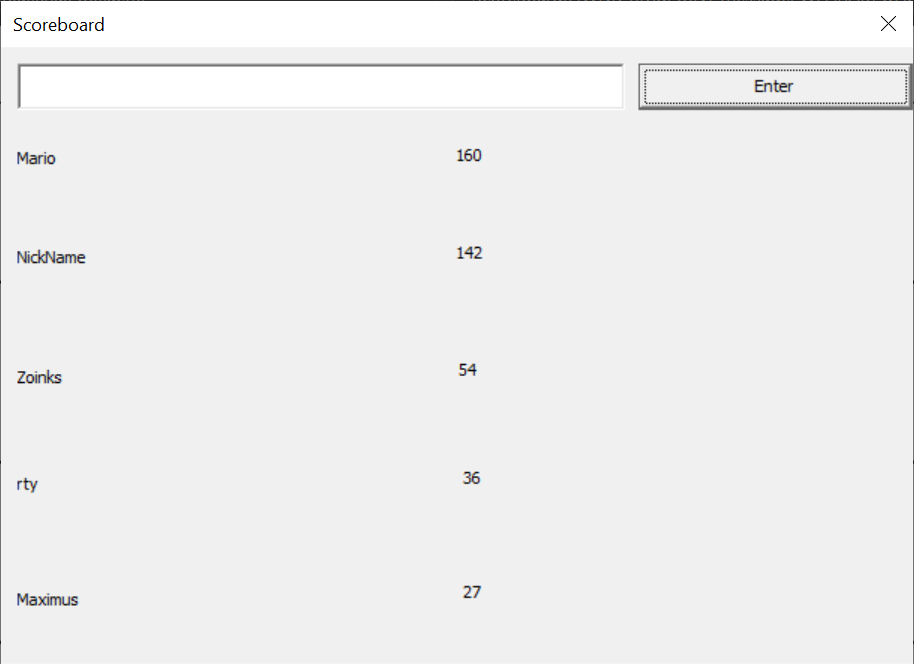


Рисунок 4.6 – Диалоговое окно таблица рекордов

При вводе имени и нажатии на кнопку «Enter» высчитываются очки пользователя и, если они достаточно высоки, чтобы находиться в таблице, записываются в таблицу рекордов (рисунок 4.7).

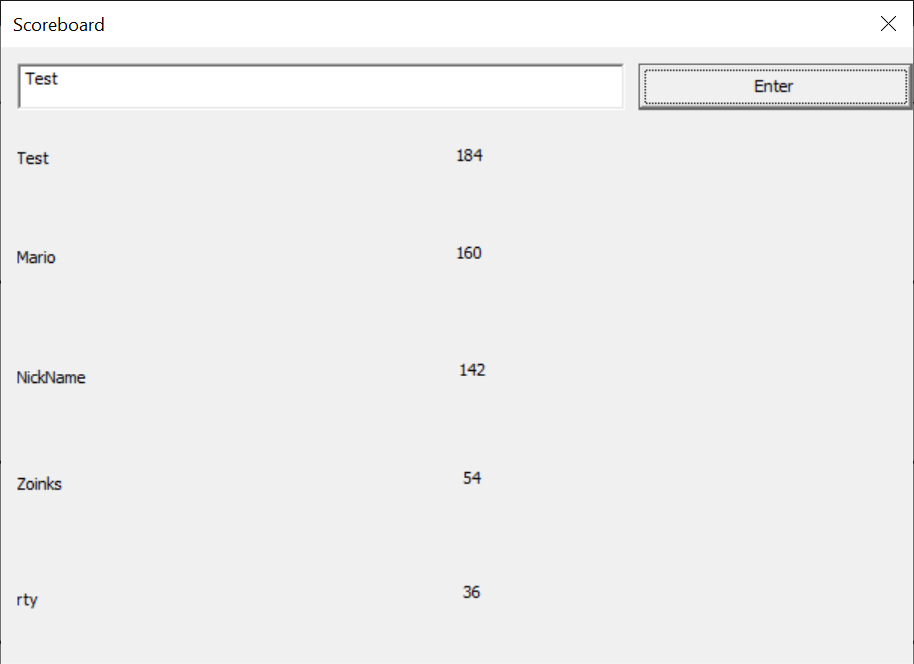


Рисунок 4.7 – Диалоговое окно обновлённая таблица рекордов

Заключение

В ходе разработки приложения был создан продукт, сочетающий в себе лучшие стороны от аналогов, с устраненными проблемными местами и свежим дизайном. С помощью использования современных сред разработки удалось достичь высокой производительности и отзывчивости. Во время реализации данного проекта мне удалось развить навыки разработки приложений на языке c++. Данное приложение справляется со всеми своими основными задачами.

В результате работы удалось добиться работы программы с низким потреблением ресурсов, благодаря чему программа будет работать без проблем на любом современном компьютере с операционной системой Windows.

Данное приложение может быть модернизировано с помощью добавления анимации и заполнением ячеек различными цветами, усовершенствованием метода случайной генерации чисел.

Список использованной литературы

[1] Основы программирования для Win32 API [Электронный ресурс] Режим доступа: https://dims.karelia.ru/win32.

[2] Создание Windows приложений [Электронный ресурс] Режим доступа: https://prog-cpp.ru/winelements.

[3] Microsoft Documentation [Электронный ресурс] Режим доступа: https://docs.microsoft.com/en-us/windows/win32.

[4] Win 32 API по шагам [Электронный ресурс] Режим доступа: https://www.firststeps.ru.

[5] Статья об истории развития компьютерных игр [Электронный ресурс] Режим доступа: https://hype.tech/@id103/istoriya-razvitiya-i-evolyuciya-videoigr-v-cifrah-i-kartinkah-w4e9feon.

[6] Справочник по функциям Windows API [Электронный ресурс] Режим доступа: http://rusproject.narod.ru/winapi/winapi.htm.

Приложение А

(обязательное)

Исходный код программы

**Файл 2048.cpp:**

#include "framework.h"

#include "2048.h"

#include "windowsx.h"

#include "sstream"

#include "time.h"

#include "iostream"

#include "fstream"

#include "vector"

#include "string"

#define MAX\_LOADSTRING 100

HINSTANCE hInst;

WCHAR szTitle[MAX\_LOADSTRING];

WCHAR szWindowClass[MAX\_LOADSTRING];

int base[4][4] = { 0 };

bool victoryCheck = false;

bool flag = false;

bool defeatCheckDown = false;

bool defeatCheckUp = false;

bool defeatCheckLeft = false;

bool defeatCheckRight = false;

int space = 0;

LOGFONT font;

struct Cell {

int top;

int left;

int right;

int bottom;

RECT content;

};

Cell Grid[4][4];

bool victory = true;

PAINTSTRUCT paintstruct;

char const\* victoryBoxName = "Victory!";

char const\* victoryBoxCaption = "You have won the game!!! Feel free to continue playing for the highest score";

char const\* defeatBoxName = "You lost";

char const\* defeatBoxCaption = "You have no more options";

ATOM MyRegisterClass(HINSTANCE hInstance);

BOOL InitInstance(HINSTANCE, int);

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);

INT\_PTR CALLBACK About(HWND, UINT, WPARAM, LPARAM);

INT\_PTR CALLBACK ScoreBoard(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam);

void TableGeneration(HWND hWnd);

void ScoreBoardPrint(HWND hDlg);

int APIENTRY wWinMain(\_In\_ HINSTANCE hInstance,

\_In\_opt\_ HINSTANCE hPrevInstance,

\_In\_ LPWSTR lpCmdLine,

\_In\_ int nCmdShow)

{

UNREFERENCED\_PARAMETER(hPrevInstance);

UNREFERENCED\_PARAMETER(lpCmdLine);

font.lfCharSet = DEFAULT\_CHARSET;

font.lfPitchAndFamily = FIXED\_PITCH;

font.lfHeight = 50;

font.lfWidth = 30;

font.lfWeight = 10;

font.lfEscapement = 0;

srand(time(0));

int i, j, i2, j2;

do

{

i = 0 + rand() % 3;

j = 0 + rand() % 3;

i2 = 0 + rand() % 3;

j2 = 0 + rand() % 3;

} while (i == i2 && j == j2);

base[i][j] = 2;

base[i2][j2] = 2;

LoadStringW(hInstance, IDS\_APP\_TITLE, szTitle, MAX\_LOADSTRING);

LoadStringW(hInstance, IDC\_MY2048, szWindowClass, MAX\_LOADSTRING);

MyRegisterClass(hInstance);

if (!InitInstance (hInstance, nCmdShow))

{

return FALSE;

}

HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC\_MY2048));

MSG msg;

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

{

if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

}

return (int) msg.wParam;

}

ATOM MyRegisterClass(HINSTANCE hInstance)

{

WNDCLASSEXW wcex;

wcex.cbSize = sizeof(WNDCLASSEX);

wcex.style = CS\_HREDRAW | CS\_VREDRAW;

wcex.lpfnWndProc = WndProc;

wcex.cbClsExtra = 0;

wcex.cbWndExtra = 0;

wcex.hInstance = hInstance;

wcex.hIcon = LoadIcon(hInstance, MAKEINTRESOURCE(IDI\_ICON2));

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

wcex.hbrBackground = (HBRUSH)(COLOR\_WINDOW+1);

wcex.lpszMenuName = MAKEINTRESOURCEW(IDC\_MY2048);

wcex.lpszClassName = szWindowClass;

wcex.hIconSm = LoadIcon(wcex.hInstance, MAKEINTRESOURCE(IDI\_ICON2));

return RegisterClassExW(&wcex);

}

BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)

{

hInst = hInstance;

HWND hWnd = CreateWindowW(szWindowClass, szTitle, WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT, 0, CW\_USEDEFAULT, 0, nullptr, nullptr, hInstance, nullptr);

if (!hWnd)

{

return FALSE;

}

ShowWindow(hWnd, nCmdShow);

UpdateWindow(hWnd);

return TRUE;

}

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

{

switch (message)

{

case WM\_COMMAND:

{

int wmId = LOWORD(wParam);

switch (wmId)

{

case IDM\_ABOUT:

DialogBox(hInst, MAKEINTRESOURCE(IDD\_ABOUTBOX), hWnd, About);

break;

case IDM\_EXIT:

DestroyWindow(hWnd);

break;

default:

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

}

}

break;

case WM\_PAINT:

{

TableGeneration(hWnd);

}

break;

case WM\_SIZE:

InvalidateRect(hWnd, NULL, FALSE);

break;

case WM\_GETMINMAXINFO: {

LPMINMAXINFO info = (LPMINMAXINFO)lParam;

info->ptMinTrackSize.x = 520;

info->ptMinTrackSize.y = 540;

}

break;

case WM\_KEYDOWN:

switch (wParam) {

case VK\_DOWN:

for (int j = 3; j >= 0; j--) {

for (int i = 3; i >= 0; i--) {

if (base[i][j] != 0 && i < 3) {

int counter = 0;

for (int p = i; p <= 3; p++) {

if (base[p][j] == 0) {

counter++;

}

}

if (base[i + counter][j] == 0) {

base[i + counter][j] = base[i][j];

base[i][j] = 0;

}

}

}

}

defeatCheckDown = true;

for (int j = 3; j >= 0; j--) {

int temp = base[3][j];

for (int i = 2; i >= 0; i--) {

if (temp == base[i][j] && temp != 0) {

base[i + 1][j] = temp \* 2;

defeatCheckDown = false;

defeatCheckUp = false;

defeatCheckLeft = false;

defeatCheckRight = false;

if (base[i + 1][j] == 2048)

victoryCheck = true;

for (int p = i; p >= 1; p--) {

base[p][j] = base[p - 1][j];

if (p == 1)

base[0][j] = 0;

}

}

temp = base[i][j];

}

}

flag = false;

for (int i = 0; i <= 3; i++) {

for (int j = 0; j <= 3; j++) {

if (base[i][j] == 0)

flag = true;

}

}

while(flag){

int i = 0 + rand() % 4;

int j = 0 + rand() % 4;

if (base[i][j] == 0 && i < 4 && j < 4) {

if ((1 + rand() % 10) == 1)

base[i][j] = 4;

else

base[i][j] = 2;

flag = false;

}

}

InvalidateRect(hWnd, NULL, FALSE);

if (victoryCheck && victory) {

MessageBoxA(hWnd, victoryBoxCaption, victoryBoxName, MB\_OK);

victory = false;

}

else if (defeatCheckDown && defeatCheckUp && defeatCheckLeft && defeatCheckRight) {

MessageBoxA(hWnd, defeatBoxCaption, defeatBoxName, MB\_OK);

}

break;

case VK\_UP:

for (int j = 3; j >= 0; j--) {

for (int i = 0; i <= 3; i++) {

if (base[i][j] != 0 && i > 0) {

int counter = 0;

for (int p = i; p >= 0; p--) {

if (base[p][j] == 0) {

counter++;

}

}

if (base[i - counter][j] == 0) {

base[i - counter][j] = base[i][j];

base[i][j] = 0;

}

}

}

}

defeatCheckUp = true;

for (int j = 3; j >= 0; j--) {

int temp = base[0][j];

for (int i = 1; i <= 3; i++) {

if (temp == base[i][j] && temp != 0) {

base[i - 1][j] = temp \* 2;

defeatCheckDown = false;

defeatCheckUp = false;

defeatCheckLeft = false;

defeatCheckRight = false;

if (base[i - 1][j] == 2048)

victoryCheck = true;

for (int p = i; p <= 2; p++) {

base[p][j] = base[p + 1][j];

if (p == 2)

base[3][j] = 0;

}

}

temp = base[i][j];

}

}

space = 0;

flag = false;

for (int i = 0; i <= 3; i++) {

for (int j = 0; j <= 3; j++) {

if (base[i][j] == 0)

flag = true;

}

}

while (flag) {

int i = 0 + rand() % 4;

int j = 0 + rand() % 4;

if (base[i][j] == 0 && i < 4 && j < 4) {

if ((1 + rand() % 10) == 1)

base[i][j] = 4;

else

base[i][j] = 2;

flag = false;

}

}

InvalidateRect(hWnd, NULL, FALSE);

if (victoryCheck && victory) {

MessageBoxA(hWnd, victoryBoxCaption, victoryBoxName, MB\_OK);

victory = false;

}

else if (defeatCheckDown && defeatCheckUp && defeatCheckLeft && defeatCheckRight) {

MessageBoxA(hWnd, defeatBoxCaption, defeatBoxName, MB\_OK);

}

break;

case VK\_LEFT:

for (int i = 0; i <= 3; i++) {

for (int j = 0; j <= 3; j++) {

if (base[i][j] != 0 && j > 0) {

int counter = 0;

for (int p = j; p >= 0; p--) {

if (base[i][p] == 0) {

counter++;

}

}

if (base[i][j - counter] == 0) {

base[i][j - counter] = base[i][j];

base[i][j] = 0;

}

}

}

}

defeatCheckLeft = true;

for (int i = 3; i >= 0; i--) {

int temp = base[i][0];

for (int j = 1; j <= 3; j++) {

if (temp == base[i][j] && temp != 0) {

base[i][j - 1] = temp \* 2;

defeatCheckDown = false;

defeatCheckUp = false;

defeatCheckLeft = false;

defeatCheckRight = false;

if (base[i][j - 1] == 2048)

victoryCheck = true;

for (int p = j; p <= 2; p++) {

base[i][p] = base[i][p + 1];

if (p == 2)

base[i][3] = 0;

}

}

temp = base[i][j];

}

}

space = 0;

flag = false;

for (int i = 0; i <= 3; i++) {

for (int j = 0; j <= 3; j++) {

if (base[i][j] == 0)

flag = true;

}

}

while (flag) {

int i = 0 + rand() % 4;

int j = 0 + rand() % 4;

if (base[i][j] == 0 && i < 4 && j < 4) {

if ((1 + rand() % 10) == 1)

base[i][j] = 4;

else

base[i][j] = 2;

flag = false;

}

}

InvalidateRect(hWnd, NULL, FALSE);

if (victoryCheck && victory) {

MessageBoxA(hWnd, victoryBoxCaption, victoryBoxName, MB\_OK);

victory = false;

}

else if (defeatCheckDown && defeatCheckUp && defeatCheckLeft && defeatCheckRight) {

MessageBoxA(hWnd, defeatBoxCaption, defeatBoxName, MB\_OK);

}

break;

case VK\_RIGHT:

for (int i = 0; i <= 3; i++) {

for (int j = 3; j >= 0; j--) {

if (base[i][j] != 0 && j < 3) {

int counter = 0;

for (int p = j; p <= 3; p++) {

if (base[i][p] == 0) {

counter++;

}

}

if (base[i][j + counter] == 0) {

base[i][j + counter] = base[i][j];

base[i][j] = 0;

}

}

}

}

defeatCheckRight = true;

for (int i = 0; i <= 3; i++) {

int temp = base[i][3];

for (int j = 2; j >= 0; j--) {

if (temp == base[i][j] && temp != 0) {

base[i][j + 1] = temp \* 2;

defeatCheckDown = false;

defeatCheckUp = false;

defeatCheckLeft = false;

defeatCheckRight = false;

if (base[i][j + 1] == 2048)

victoryCheck = true;

for (int p = j; p >= 1; p--) {

base[i][p] = base[i][p - 1];

if (p == 1)

base[i][0] = 0;

}

}

temp = base[i][j];

}

}

space = 0;

flag = false;

for (int i = 0; i <= 3; i++) {

for (int j = 0; j <= 3; j++) {

if (base[i][j] == 0)

flag = true;

}

}

while (flag) {

int i = 0 + rand() % 4;

int j = 0 + rand() % 4;

if (base[i][j] == 0 && i < 4 && j < 4 && flag) {

if ((1 + rand() % 10) == 1)

base[i][j] = 4;

else

base[i][j] = 2;

flag = false;

}

}

InvalidateRect(hWnd, NULL, FALSE);

if (victoryCheck && victory) {

MessageBoxA(hWnd, victoryBoxCaption, victoryBoxName, MB\_OK);

victory = false;

}

else if (defeatCheckDown && defeatCheckUp && defeatCheckLeft && defeatCheckRight) {

MessageBoxA(hWnd, defeatBoxCaption, defeatBoxName, MB\_OK);

}

break;

case VK\_SPACE:

DialogBox(hInst, MAKEINTRESOURCE(IDD\_DIALOG1), hWnd, ScoreBoard);

break;

}

break;

case WM\_DESTROY:

PostQuitMessage(0);

break;

default:

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

}

return 0;

}

void TableGeneration(HWND hWnd) {

RECT clientRect;

GetClientRect(hWnd, &clientRect);

int clientHeight = clientRect.bottom - clientRect.top;

int clientWidth = clientRect.right - clientRect.left;

HDC currDC = BeginPaint(hWnd, &paintstruct);

HDC virtDC = CreateCompatibleDC(currDC);

HBITMAP bitmap = CreateCompatibleBitmap(currDC, clientWidth, clientHeight);

SelectBitmap(virtDC, bitmap);

HPEN newPen = CreatePen(PS\_SOLID, 1, RGB(0, 0, 0));

HPEN pen = SelectPen(virtDC, newPen);

LOGBRUSH brush;

brush.lbStyle = BS\_SOLID;

brush.lbColor = RGB(255, 255, 255);

HBRUSH hBrush;

hBrush = CreateBrushIndirect(&brush);

FillRect(virtDC, &clientRect, hBrush);

DeleteObject(hBrush);

HFONT hFont = CreateFontIndirect(&font);

SelectObject(virtDC, hFont);

SetTextColor(virtDC, RGB(0, 0, 0));

SetBkColor(virtDC, RGB(255, 255, 255));

Rectangle(virtDC, 0, 0, clientWidth, clientHeight);

int cellWidth = clientWidth / 4;

int cellHeight = clientHeight / 4;

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

Grid[i][j].left = j \* cellWidth;

Grid[i][j].right = (j + 1) \* cellWidth;

Grid[i][j].top = i \* cellHeight;

Grid[i][j].bottom = (i + 1) \* cellHeight;

Grid[i][j].content.left = Grid[i][j].left + cellWidth / 100;

Grid[i][j].content.top = Grid[i][j].top + cellHeight / 100;

Grid[i][j].content.right = Grid[i][j].right - cellWidth / 100;

Grid[i][j].content.bottom = Grid[i][j].bottom - cellHeight / 100;

}

}

char temp;

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

Rectangle(virtDC, Grid[i][j].left, Grid[i][j].top, Grid[i][j].right, Grid[i][j].bottom);

std::stringstream s;

s << std::scientific << base[i][j];

temp = base[i][j];

if (base[i][j] != 0)

DrawTextA(virtDC, s.str().c\_str(), -1, (LPRECT)&Grid[i][j].content, DT\_CENTER | DT\_WORDBREAK | DT\_EDITCONTROL);

}

}

SetStretchBltMode(virtDC, COLORONCOLOR);

BitBlt(currDC, 0, 0, clientWidth, clientHeight, virtDC, 0, 0, SRCCOPY);

DeleteObject(hFont);

DeleteObject(bitmap);

DeletePen(SelectPen(virtDC, pen));

DeleteDC(virtDC);

EndPaint(hWnd, &paintstruct);

}

INT\_PTR CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)

{

UNREFERENCED\_PARAMETER(lParam);

switch (message)

{

case WM\_INITDIALOG:

return (INT\_PTR)TRUE;

case WM\_COMMAND:

if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL)

{

EndDialog(hDlg, LOWORD(wParam));

return (INT\_PTR)TRUE;

}

break;

}

return (INT\_PTR)FALSE;

}

void ScoreBoardPrint(HWND hDlg) {

std::vector<int> intVector;

std::vector<std::string\*> strVector;

std::ifstream readFile;

readFile.open("Scoreboard.txt");

std::string\* s = new std::string();

readFile >> \*s;

int i = 0;

while (!readFile.eof()) {

strVector.push\_back(s);

SetDlgItemTextA(hDlg, IDC\_STATIC1 + i, (LPCSTR)s->c\_str());

i++;

s = new std::string();

readFile >> \*s;

SetDlgItemTextA(hDlg, IDC\_STATIC1 + i, (LPCSTR)s->c\_str());

i++;

intVector.push\_back(std::stol(\*s));

s = new std::string();

readFile >> \*s;

}

readFile.close();

delete s;

}

INT\_PTR CALLBACK ScoreBoard(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)

{

UNREFERENCED\_PARAMETER(lParam);

switch (message)

{

case WM\_INITDIALOG:

return (INT\_PTR)TRUE;

case WM\_COMMAND:

if (LOWORD(wParam) == IDOK || LOWORD(wParam) == IDCANCEL || LOWORD(wParam) == IDCLOSE)

{

EndDialog(hDlg, LOWORD(wParam));

return (INT\_PTR)TRUE;

}

if (LOWORD(wParam) == IDC\_BUTTON1) {

LPSTR str = new char[10];

GetDlgItemTextA(hDlg, IDC\_EDIT1, str, 10);

std::vector<int> intVector;

std::vector<std::string\*> strVector;

std::ifstream readFile;

readFile.open("Scoreboard.txt");

std::string\* s = new std::string();

readFile >> \*s;

int i = 0;

while (!readFile.eof()) {

strVector.push\_back(s);

s = new std::string();

readFile >> \*s;

intVector.push\_back(std::stol(\*s));

s = new std::string();

readFile >> \*s;

}

readFile.close();

int sum = 0;

for (int k = 0; k <= 3; k++) {

for (int p = 0; p <= 3; p++) {

sum += base[k][p];

}

}

std::ofstream file;

file.open("Scoreboard.txt");

for (int k = 0; k < intVector.size(); k++) {

if (intVector[k] < sum) {

file << str << "\n";

file << sum << "\n";

for (int p = k; p <= 3; p++) {

file << \*strVector[p] << "\n";

if (p == 3)

file << intVector[p];

else

file << intVector[p] << "\n";

}

break;

}

else {

file << \*strVector[k] << "\n";

if (k == 4)

file << intVector[k];

else

file << intVector[k] << "\n";

}

}

file.close();

delete str;

readFile.open("Scoreboard.txt");

s = new std::string();

readFile >> \*s;

i = 0;

while (!readFile.eof()) {

SetDlgItemTextA(hDlg, IDC\_STATIC1 + i, (LPCSTR)s->c\_str());

i++;

s = new std::string();

readFile >> \*s;

SetDlgItemTextA(hDlg, IDC\_STATIC1 + i, (LPCSTR)s->c\_str());

i++;

s = new std::string();

readFile >> \*s;

}

readFile.close();

delete s;

}

break;

case WM\_PAINT:

ScoreBoardPrint(hDlg);

break;

}

return (INT\_PTR)FALSE;

}

**Файл Resource.h:**

//{{NO\_DEPENDENCIES}}

// Включаемый файл, созданный в Microsoft Visual C++.

// Используется 2048.rc

//

#define IDC\_MYICON 2

#define IDD\_MY2048\_DIALOG 102

#define IDS\_APP\_TITLE 103

#define IDD\_ABOUTBOX 103

#define IDM\_ABOUT 104

#define IDM\_EXIT 105

#define IDI\_MY2048 107

#define IDI\_SMALL 108

#define IDC\_MY2048 109

#define IDR\_MAINFRAME 128

#define IDI\_ICON1 129

#define IDI\_ICON2 130

#define IDD\_DIALOG1 131

#define IDC\_BUTTON1 1000

#define IDC\_EDIT1 1001

#define IDC\_LIST1 1002

#define IDC\_STATIC1 1005

#define IDC\_STATIC2 1006

#define IDC\_STATIC3 1007

#define IDC\_STATIC4 1008

#define IDC\_STATIC5 1009

#define IDC\_STATIC6 1010

#define IDC\_STATIC7 1011

#define IDC\_STATIC8 1012

#define IDC\_STATIC9 1013

#define IDC\_STATIC10 1014

#define IDC\_STATIC -1

// Next default values for new objects

//

#ifdef APSTUDIO\_INVOKED

#ifndef APSTUDIO\_READONLY\_SYMBOLS

#define \_APS\_NO\_MFC 1

#define \_APS\_NEXT\_RESOURCE\_VALUE 132

#define \_APS\_NEXT\_COMMAND\_VALUE 32771

#define \_APS\_NEXT\_CONTROL\_VALUE 1015

#define \_APS\_NEXT\_SYMED\_VALUE 110

#endif

#endif

Ведомость документов

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Обозначение | | | | | Наименование | | | | Дополнительные сведения | |
|  | | | | | Текстовые документы | | | |  | |
|  | | | | |  | | | |  | |
| БГУИР КП 1–40 01 01 628 ПЗ | | | | | Пояснительная записка | | | | 36 с. | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | | Графические документы | | | |  | |
|  | | | | |  | | | |  | |
| ГУИР.851006-01 СА | | | | | Отрисовка содержимого файла | | | | Формат А1 | |
|  | | | | | Схема алгоритма | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | |  | | | |  | |
|  |  |  |  |  | БГУИР КП I- 40 01 01 628 ПЗ | | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
| Изм | Лист | № докум. | Подп. | Дата | Игровое приложение «2048»  Ведомость курсового  проекта | Литера | | | Лист | Листов |
| Разраб. | | Шустал А.Ю. |  |  | Т |  |  | 36 | 36 |
| Провер. | | Жиденко А.Л. |  |  | Кафедра ПОИТ  гр. 851006 | | | | |
|  | |  |  |  |
|  | |  |  |  |
|  | |  |  |  |