

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет Вычислительной техники

Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"

Зав. кафедрой ВТ

М.А.

Митрохин

« ____ » ____ 20 ____ г

ЗАДАНИЕ

на курсовое проектирование по курсу

Программирование

Студенту Саранцеву Егору Александровичу Группа 23ВВВ3

Тема проекта: *Разработка программы сложной структуры методом нисходящего программирования.*

Исходные данные (технические требования) на проектирование
Разработать игру 2048, управление осуществляется с клавиатуры, указать счет, равный произведению всех элементов на поле. Размер поля 4x4.

Обязательные требования к программе:

1. Многомодульность.
 2. Использование сложных типов данных – структур, списков, файлов.
 3. Режим работы видеосистемы – текстовый / графический.
 4. Устройство ввода информации – клавиатура, мышь.
 5. Пользовательский интерфейс должен быть построен на основе меню и панели инструментов.
 6. Наличие заставки.
 7. Операционная система MS Windows.
 8. Язык программирования – Си и Ассемблер.
 9. Среда разработки ПО – Microsoft Visual Studio.
- Объем работы по курсу

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет Вычислительной техники

Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"

Зав. кафедрой ВТ

М.А.

Митрохин

« ____ » ____ 20 __ г

ЗАДАНИЕ

на курсовое проектирование по курсу

Программирование

Студенту Саранцеву Егору Александровичу Группа 23ВВВ3

Тема проекта: *Разработка программы сложной структуры методом нисходящего программирования.*

Исходные данные (технические требования) на проектирование
Разработать игру 2048, управление осуществляется с клавиатуры, указать счет, равный произведению всех элементов на поле. Размер поля 4x4.

Обязательные требования к программе:

1. Многомодульность.
2. Использование сложных типов данных – структур, списков, файлов.
3. Режим работы видеосистемы – текстовый / графический.
4. Устройство ввода информации – клавиатура, мышь.
5. Пользовательский интерфейс должен быть построен на основе меню и панели инструментов.
6. Наличие заставки.
7. Операционная система MS Windows.
8. Язык программирования – Си и Ассемблер.
9. Среда разработки ПО – Microsoft Visual Studio. Объем работы по курсу

Объем работы по курсу

1.Расчетная часть

Разработка программы.

2.Графическая часть

Схема данных, схема ресурсов системы, схема работы системы,
иерархическая структура программы, схема взаимодействия программы.

3.Экспериментальная часть

Отладка программы.

Срок выполнения проекта по разделам

В соответствии с графиком.

Дата выдачи задания "06" февраля 2024

Дата защиты проекта "04" 06.24

Руководитель Голотенков Н.О

Задание получил "06" февраля 2024г.

Студент Саранцев Е.А.

Содержание

Введение.....	5
1. Постановка задачи.....	6
2. Выбор решения.....	6
3. Описание разработки программы.....	9
4. Отладка и тестирование.....	11
5. Описание программы.....	13
6. Руководство пользователя.....	25
7. Заключение.....	29
Список литературы.....	30
Приложение А.....	31
Листинги программы.....	31
Файл MyForm.cpp.....	31
Файл MyForm.h.....	31
Приложение В.....	65
Результаты работы программы.....	65

Введение

Современный мир может быть очень запутанным, и порой наш мозг не успевает обрабатывать большие объёмы информации. В таких случаях нам нужен отдых, но не обязательно полное отключение от внешнего мира.

Одной из игр, которая может помочь вам расслабиться и при этом развивать полезные навыки, является 2048. Эта игра была создана два года назад и с тех пор завоевала популярность среди любителей головоломок.

Среди мобильных игр уже давно существуют аналоги, но 2048 остаётся одной из самых популярных. Она не только развлекает, но и может быть эффективным тренажёром для развития навыков планирования, творческого мышления и пространственного воображения.

Игра начинается с игрового поля, состоящего из 16 квадратиков. Каждый квадратик содержит число. Цель игры — объединить два квадратика с одинаковыми числами и получить квадратик с суммой этих чисел. Таким образом, необходимо объединить все квадратики на поле и получить квадратик с числом 2048.

Игра развивает логическое мышление и навыки планирования, так как для успешного прохождения необходимо строить определенную стратегию. Она также помогает развивать способность оценивать суть проблем и анализировать их.

Чтобы победить в игре, нужно выбрать правильную стратегию. Игроки могут перемещать квадратики вправо, вверх, влево и вниз, но нужно быть очень внимательным и не заполнять поле до того момента, пока не будет возможности провести необходимые соединения. Одна из стратегий победы — сосредоточиться на перемещении квадратиков в самый дальний угол игрового поля, чтобы самая большая плитка всегда находилась в углу.

1. Постановка задачи.

Необходимо разработать программу для игры в 2048. Программа должна иметь интуитивно понятный интерфейс для пользователя. Поле стандартных для игры размеров 4x4, управление осуществляется с клавиатуры.

Многомодульность программы. Программа должна быть поделена на логические модули. Это упростит поиск ошибок при отладке и тестировании программы, а также позволит легко расширять функционал программы.

Режим работы видеосистемы – текстовый/графический. Для начала необходимо определиться с типом интерфейса и с элементами управления, затем необходимо изучить способы их реализации.

Устройство ввода-вывода – клавиатура и мышь. Необходимо различать и идентифицировать действия, произведенные с их помощью, это облегчит использование программы.

2. Выбор решения.

В качестве платформы для разработки выбрана WinForms для создания многофункционального интерактивного интерфейса. Разработанная программа состоит из двух модулей: файл MyForm.h и файл MyForm.cpp. В файле MyForm.h описывается форма, в которой будет происходить сама игра (рисунок 1) и все ее механики: модификации клеток в зависимости от счета, слияние клеток, движение клеток, проверка на победу и проигрыш, нахождение счета и рестарт/завершение программы.

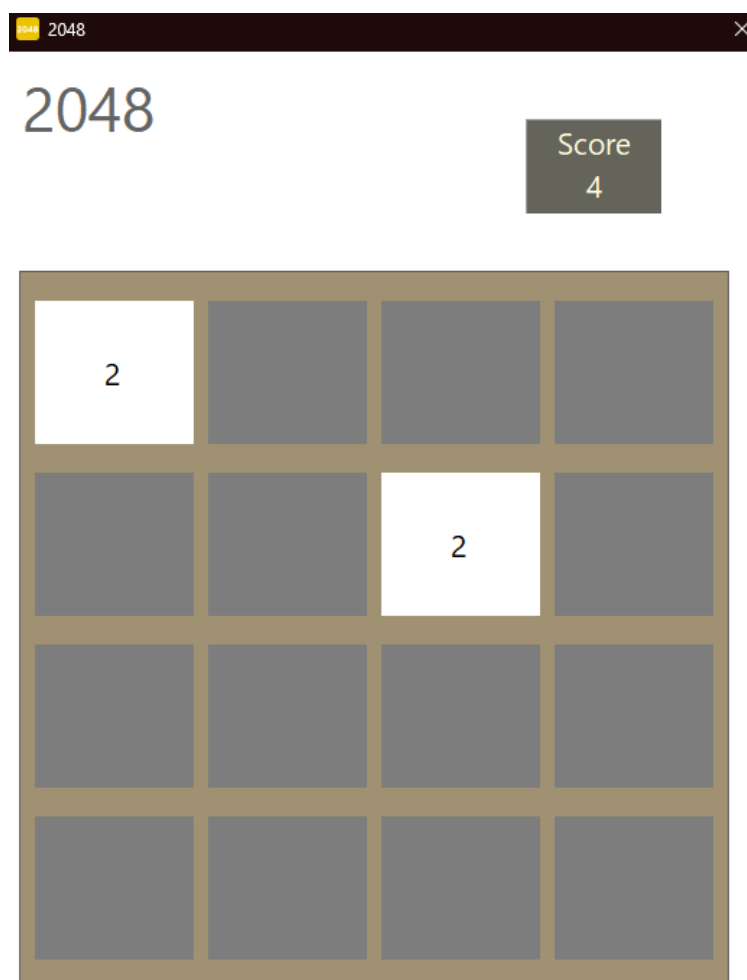


Рисунок 1- игра 2048

В файле MyForm.cpp подключается вышеуказанная библиотека и два пространства имен System System Windows Forms, и просто запускается форма. Тем самым удовлетворяется пункт многомодульности. При запуске программы появляется поле для игры и два числа в случайных клетках. Управление выбрано WASD (W- вверх, A- влево, S- вниз, D- вправо), как стандарт в последнее время,

1 Приветственный экран. Название игры и кто разработал. (рисунок 2).

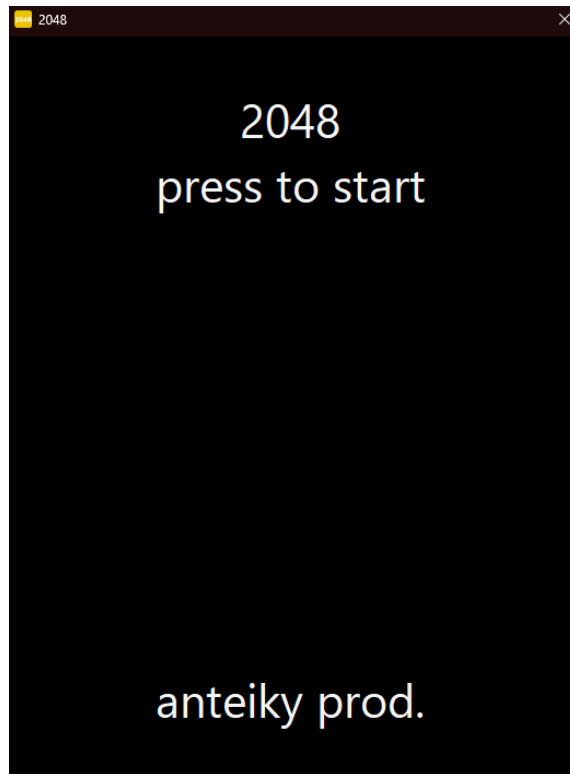


Рисунок 2- Заставка

2 В начале в двух случайных клетках появляется значение считается счет, программа просто складывает все значения лежащих в плитках и переносит его в заданное поле

3 После чего пользователь выбирает одно из 4 направлений, и в соответствии с выбранным направлением клетки сначала производят слияние, только если их значения равны, иначе происходит смещение если есть свободные клетки.

4 Модификации клеток в зависимости от счета. Так будет проще различить счет и программа становится интуитивно понятной.

5 После смещения счет считается еще раз, так как клетки соединились, а значит и счет изменился.

6 Проверка на поражение и победу.

6.1. Чтобы проверить на поражение нужно просмотреть соседей всех клеток, если хотя бы две соседних клетки будут равны, функция возвратит ложь.

6.2. Чтобы проверить на победу нужно искать клетку со значением 2048, тогда функция вернет истину, иначе если во всех клетках не оказалась такого значения вернет ложь.

Важно что проверка происходит сначала на победу, а только после нее на поражение, так как пользователь может заполнить все клетки и набрать 2048, что в обратном случае выведет поражение.

7 Выход из программы осуществляется по кнопке 'q' при победе или поражении.

3. Описание разработки программы.

Разработанная программа состоит из нескольких модулей, которые реализовались в следующем порядке:

- 1) MyForm.cpp- запуск формы
- 2) MyForm.h- определение формы, и всех функций работающих
 - 2.1 Создание формы с помощью Winforms.
 - 2.2 Создание поля, счета и надписи с названием игры
 - 2.3 Создание ивента привязанного к нажатию пользователя
 - 2.4 Соединение и передвижение клеток
 - 2.5 Модификация клеток
 - 2.6 Счет
 - 2.7 Победа/Поражение.

По порядку реализации модулей можно понять, что использовался нисходящий подход реализации программы. Этот метод основан на том, что сначала проектируются основные компоненты программы, а затем уже дорабатываются ее мелкие детали. Это позволяет упростить ее тестирование, так как каждая часть программы тестируется сразу после ее реализации, что помогает в создании более структурированных программ.

Для создания программы была выбрана платформа WinForms, которая позволяет на языке C++ создавать полноценные приложения с графическим

интерфейсом. Хотя изначально игра была сделана в консоли, чтобы разобраться с механиками и возможными “подводными камнями”. После создания проекта, сразу создавалась форма и начата работа в конструкторе. Конструктор- это визуальное меню, где можно добавлять всевозможные поля(такие как кнопка, текстовое поле и тд.). За основу заставки взята кнопка, которая была бы с размер окна и закрывала бы все его внутренности. По нажатию программа удаляет эту кнопку и пользователь приступает к игре.

После добавлено игровое поле, взят массив типа `label`, и размещен как клетки на панели. Приступив к разработке движений и соединений, важно было понять принцип по которому клетки движутся и как они соединяются, например если на поле в строке находится четыре 2, и их соединить должна возникнуть ситуация где две четверки рядом, а значит важно что сначала клетки соединяются а затем двигаются. Принцип слияния клеток организован на сравнении значений внутри клетки, и при равенстве соседней, или равны клетки через одну и между ними клетки пуста и тд, необходимо одну клетку приравнять к 0, а другую увеличить в 2 раза, так нужно выполнить со всеми направлениями. Принцип передвижения основан на наличии пустых клеток между клеткой которую двигают в определенную сторону и границей поля.

Ассемблерная вставка- в программировании возможность компилятора встраивать низкоуровневый код, написанный на ассемблере, в программу, написанную на языке высокого уровня, например, Си. На ассемблере была реализована функцию `check_win(int zxc)`. Данная функция является проверкой на победу, в нее входит значение из ячейки, и в случае равенства значение с 2048, функция вернет 1, иначе 0. Она необходима для функции `is_win()`, которая проверяет возвращаемое значение из `check_win`, и определяет победил ли игрок.

4. Отладка и тестирование.

В качестве среды разработки была выбрана программа Microsoft Visual Studio 2022. Программа обладает всеми средствами необходимыми при разработке и отладке программы. Для отладки использовались несколько возможностей Visual Studio: точка останова, трассировка, анализ содержимого переменных.

Тестирование проводилось во время разработки, а также после завершения. В ходе нее было выявлено и исправлено достаточное количество ошибок и недочетов, связанных с ООП и определением победы/поражения.

Для отладки нужно запустить приложение с отладчиком, подключенным к процессу приложения. Для этого: нажмите клавишу F5 (Отладка > Начать отладку), которая является наиболее распространенным методом.

Однако чтобы отладка работала корректно нужно установить точки останова, ее можно задать слева от строки кода. Точки останова — это самая основная и важная функция надежной отладки. Точка останова указывает, где Visual Studio следует приостановить выполнение кода, чтобы вы могли проверить значения переменных или поведение памяти либо выполнение ветви кода.

Нажмите клавишу F5 (Отладка > запуск отладки), а отладчик запускается в первую точку останова, с которой она сталкивается. Если приложение еще не запущено, при нажатии клавиши F5 запускается отладчик и выполняется остановка в первой точке останова.

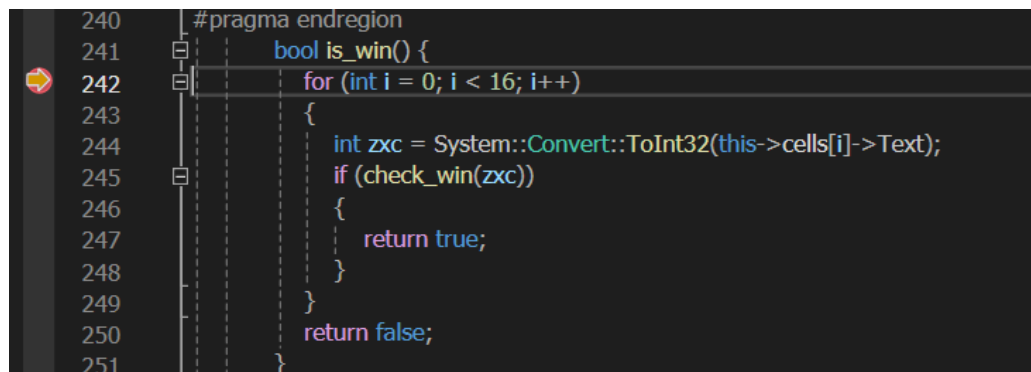


Рисунок 3-Проверка массива клеток на искомое значение с помощью точки останова

Для запуска приложения с подключенным отладчиком нажмите клавишу F11 (Отладка > Шаг с заходом). F11 — это команда Шаг с заходом, которая выполняет приложение с переходом к следующему оператору. При запуске приложения с помощью клавиши F11 отладчик останавливается на первом выполняемом операторе.

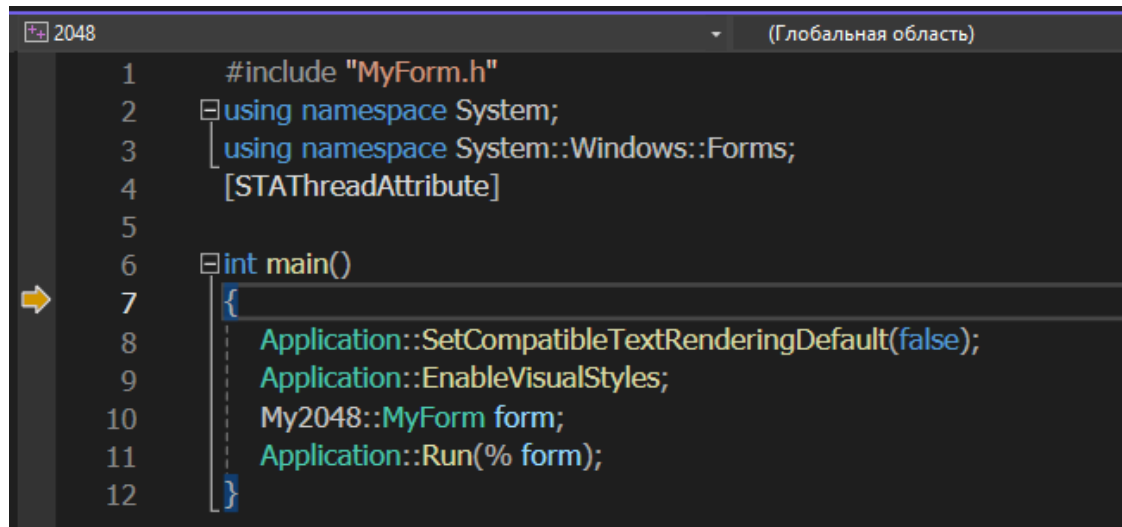


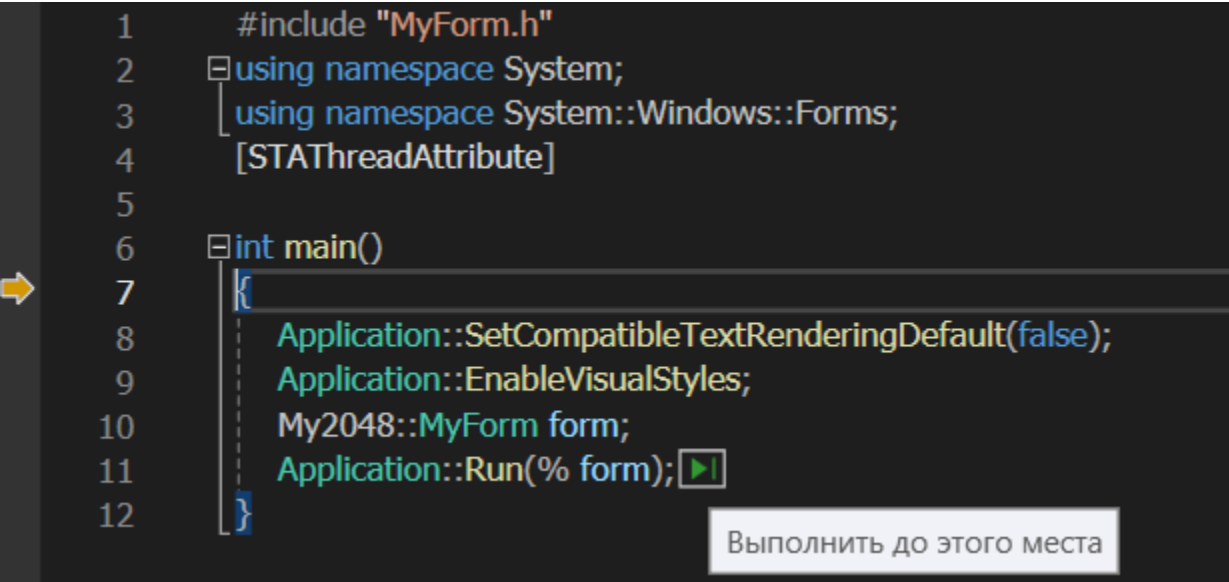
Рисунок 4-указатель на следующую исполняемую строку

Желтая стрелка представляет оператор, на котором приостановлен отладчик. В этой же точке приостанавливается выполнение приложения (этот оператор пока не выполнен). Клавишу F11 удобно использовать для более детальной проверки потока выполнения.

Для продолжения выполнения отладчика без захода в функции или методы в коде приложения (код продолжает выполняться). Нажав клавишу F10, вы можете обойти код, который вас не интересует. Так можно быстро перейти к важному для вас коду.

Использование кнопки Выполнение до щелкнутого аналогично установке временной точки останова. Кроме того, эта команда удобна для быстрой работы в видимой области кода приложения. Выполнение до щелкнутого можно использовать в любом открытом файле.

Пока в отладчике наведете указатель мыши на строку кода, пока не появится кнопка "Запустить" (выполнить выполнение здесь).



```
1  #include "MyForm.h"
2  using namespace System;
3  using namespace System::Windows::Forms;
4  [STAThreadAttribute]
5
6  int main()
7  {
8      Application::SetCompatibleTextRenderingDefault(false);
9      Application::EnableVisualStyles;
10     My2048::MyForm form;
11     Application::Run(% form);
12 }
```

Рисунок 5-Кнопка «Быстрого выполнения»

5. Описание программы.

Приложение 2048.exe является основным модулем программы. При запуске программы происходит вывод заставки-приветствия, после чего можно сразу играть, выход из игры осуществляется кнопкой ‘q’ после проигрыша или победы. Описание состояний программы выполнено в таблице ниже.

Таблица 1. – Работа заставки

Клавиши, вызывающие событие	Действие пользователя	Действие программы
ЛКМ	Нажатие на окно приложения	Убирается заставка методом delete и запускается игра(см. таблица 2)

Таблица 2. – Работа функций и методов для выполнения самой игры(Modify, KeyPress, is_win, is_lose, randomcell, score, check_win)

Клавиши, вызывающие событие	Действие пользователя	Действие программы
W/w,A/a,S/s,D/d	Выбор подходящего направления	Исходя из направления происходит, слияние и сдвиг клеток, изменение цветов клеток, поиск счета, проверка на победу и поражение
R/r	Выбран рестарт	Скидывает поле до начального состояния и позволяет начать сначала
Q/q	Выбран выход	Выход из приложения



Рисунок 6 – Схема функции файла MyForm.cpp()



Рисунок 7 – Схема функции Initialize Components()

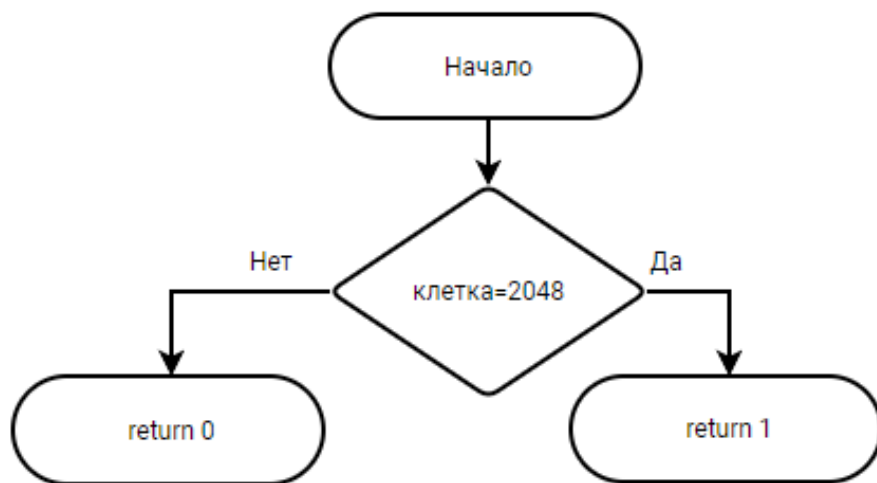


Рисунок 8 – Схема функции `check_win()`

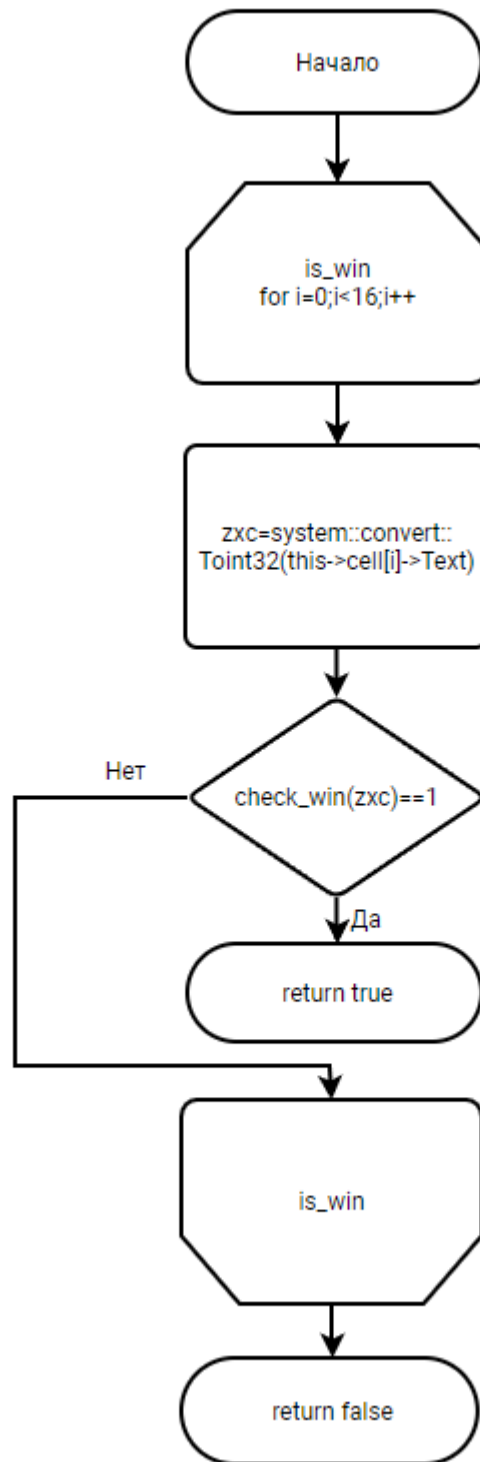


Рисунок 9- Схема функции is_win()

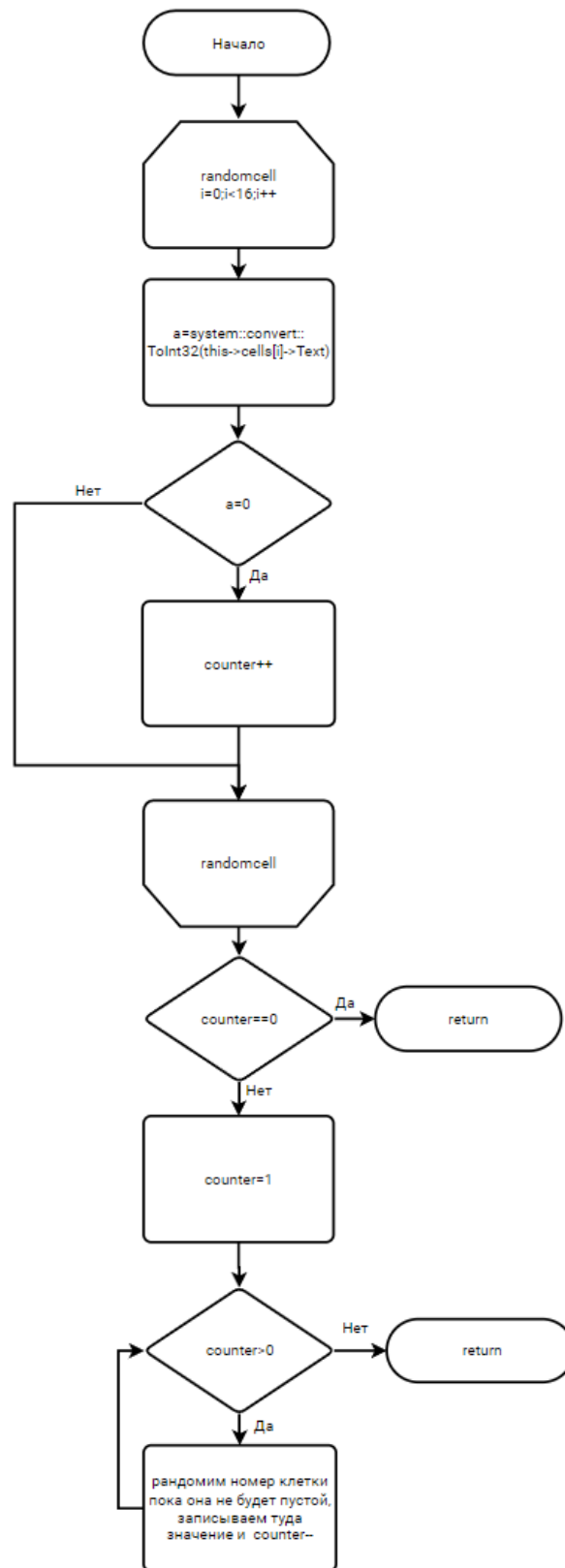


Рисунок 10- Схема функции randomcell()



Рисунок 11 - Схема функции is_lose()

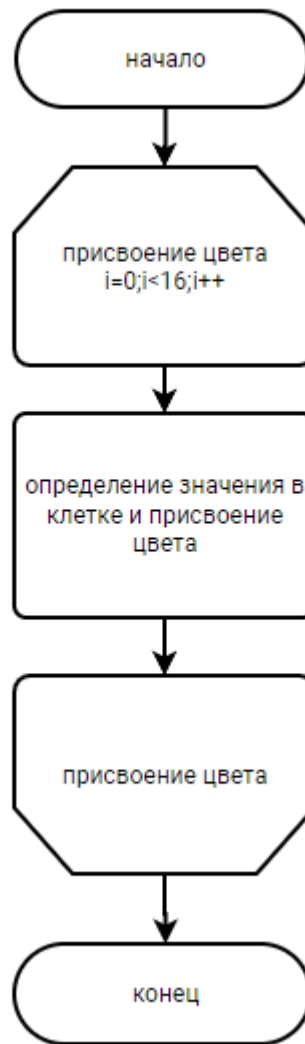


Рисунок 12- Схема функции `Modify()`

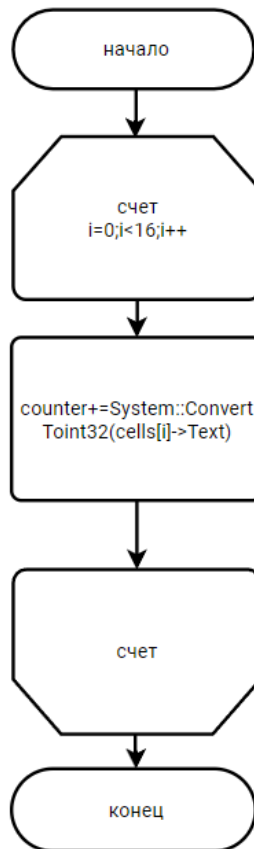


Рисунок 13- Схема функции Score()

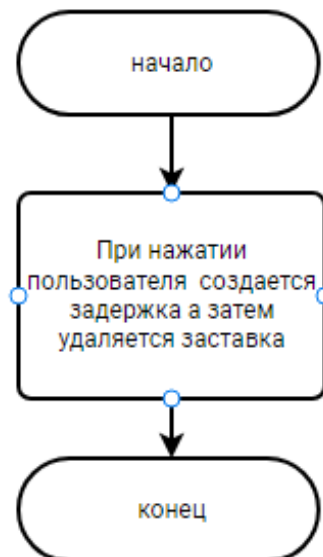
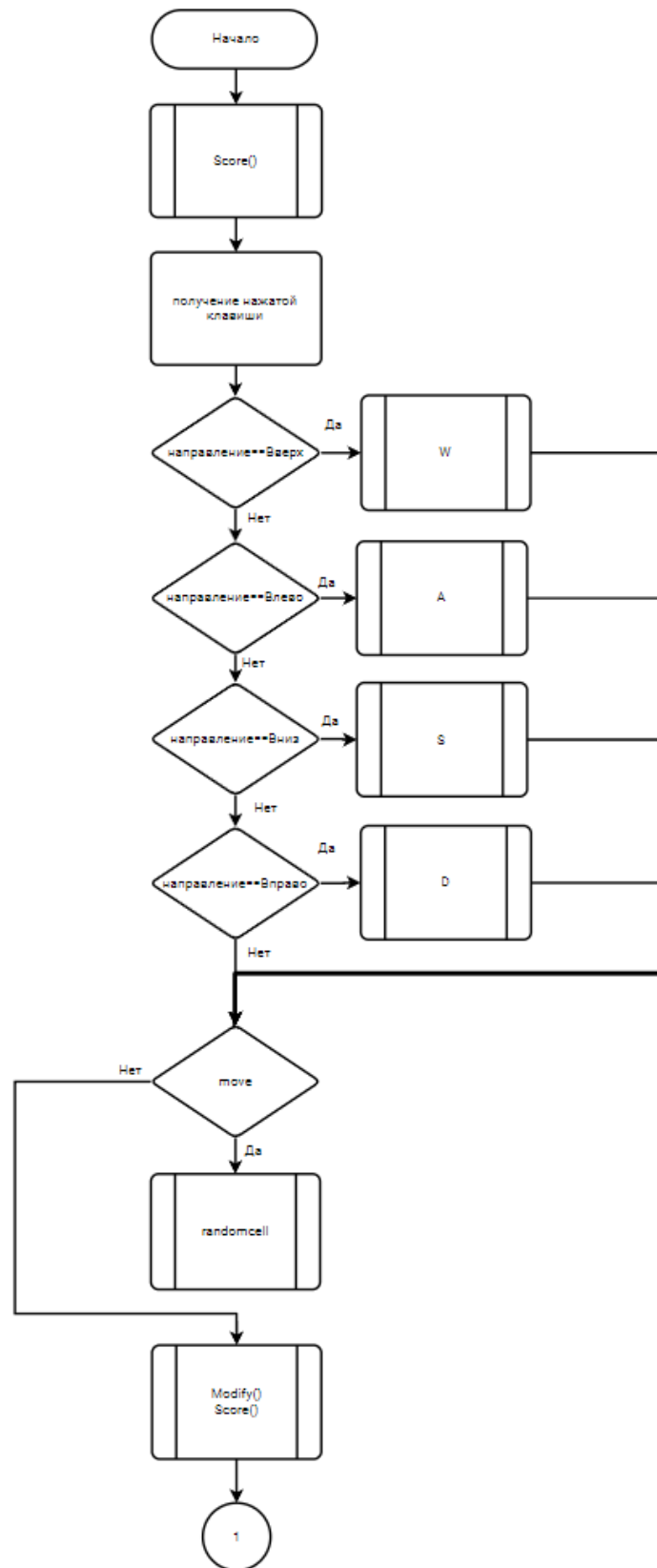
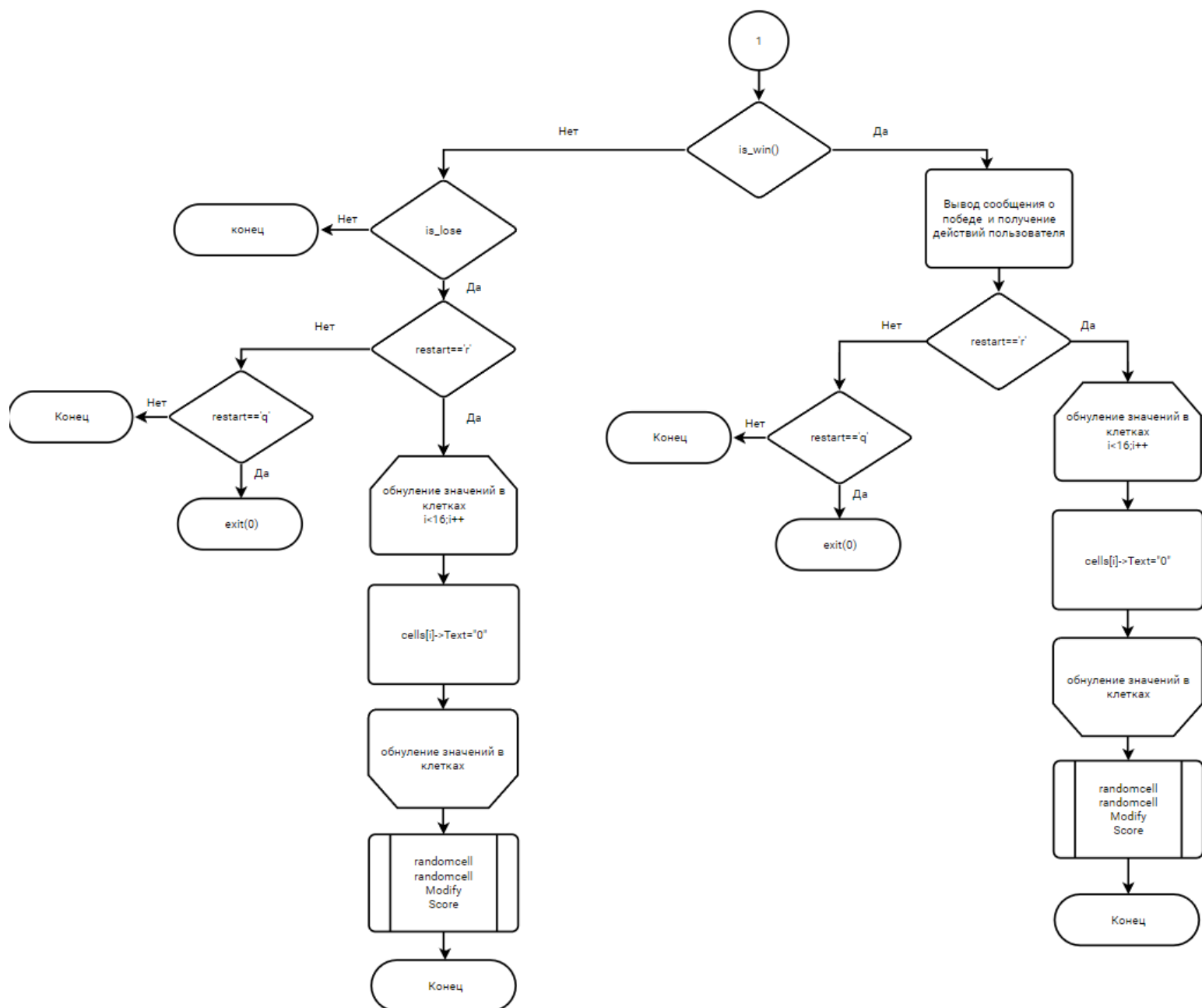


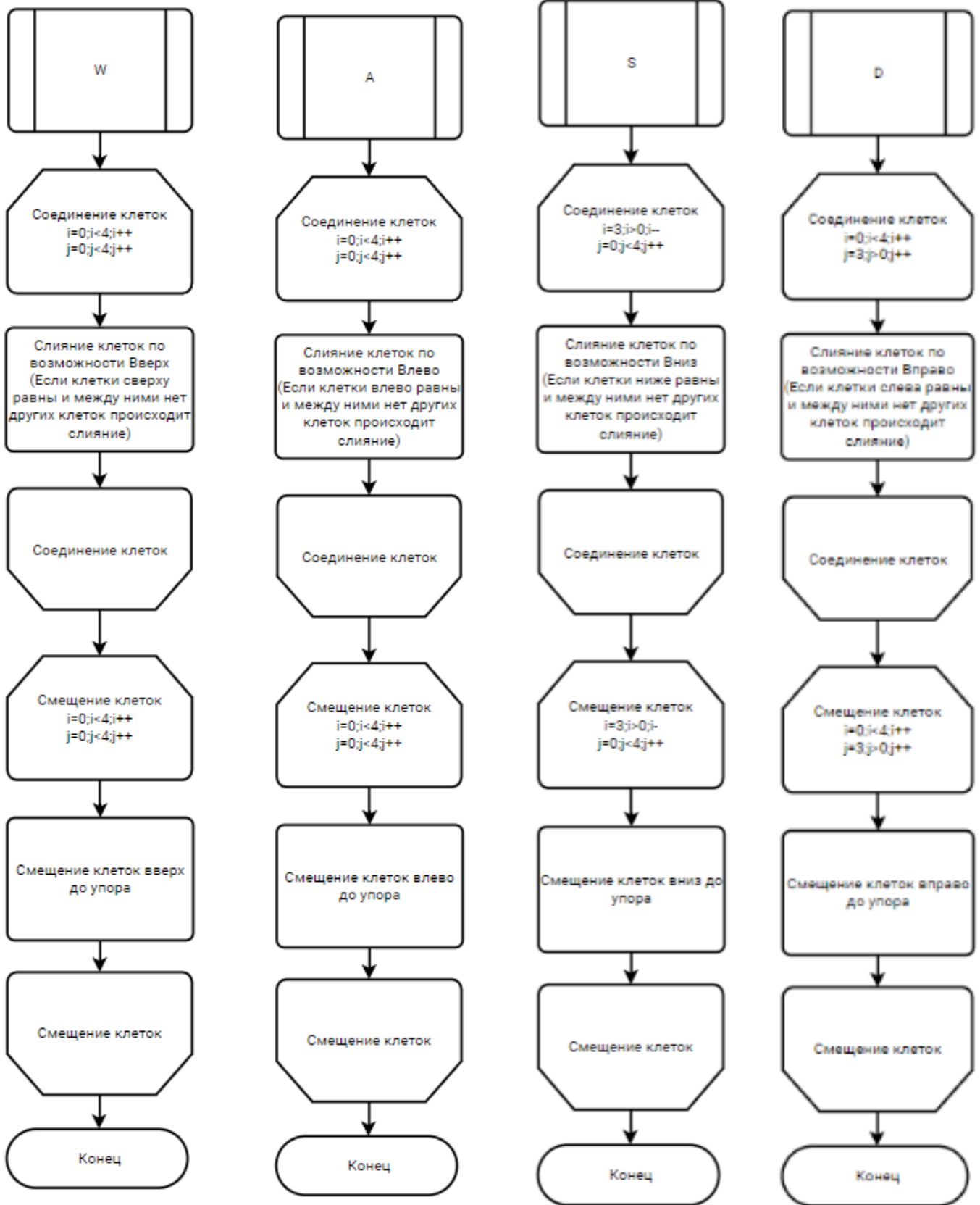
Рисунок 14- Схема функции Myform_Opening(System::Object^ sender, System::EventArgs^ e)





Рисунки 15, 16- Схема функции `MyForm_KeyPress(System::Object^ sender, System::Windows::Forms::KeyPressEventArgs e)`

Рисунки 17-20- Функции направления.



6. Руководство пользователя.

Программа 2048.exe предназначена для игры в 2048. Программа имеет интуитивно понятный интерфейс и содержит саму игру. Для начала игры нужно нажать на появившееся окно (рисунок 21).

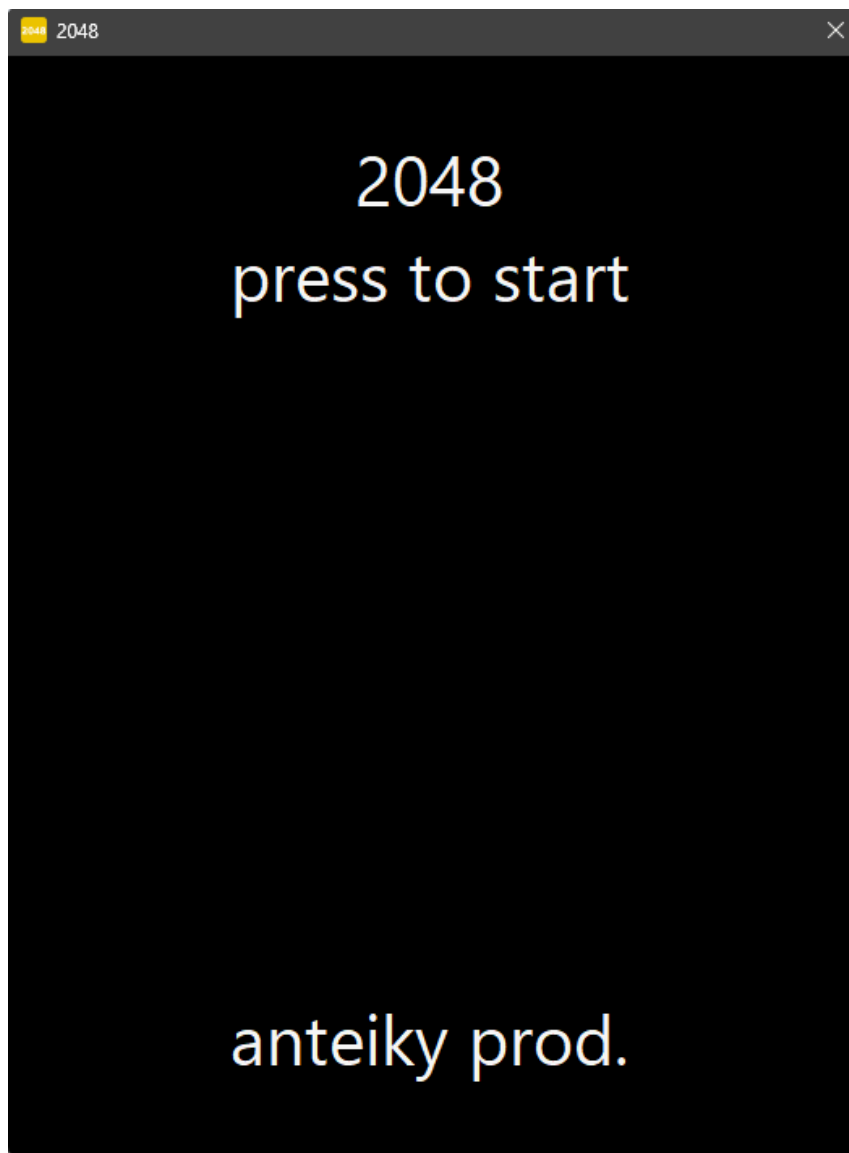


Рисунок 21- Начальная заставка

После чего перед пользователем появляется поле в котором написаны название игры, его счет и поле для игры (рисунок 22).

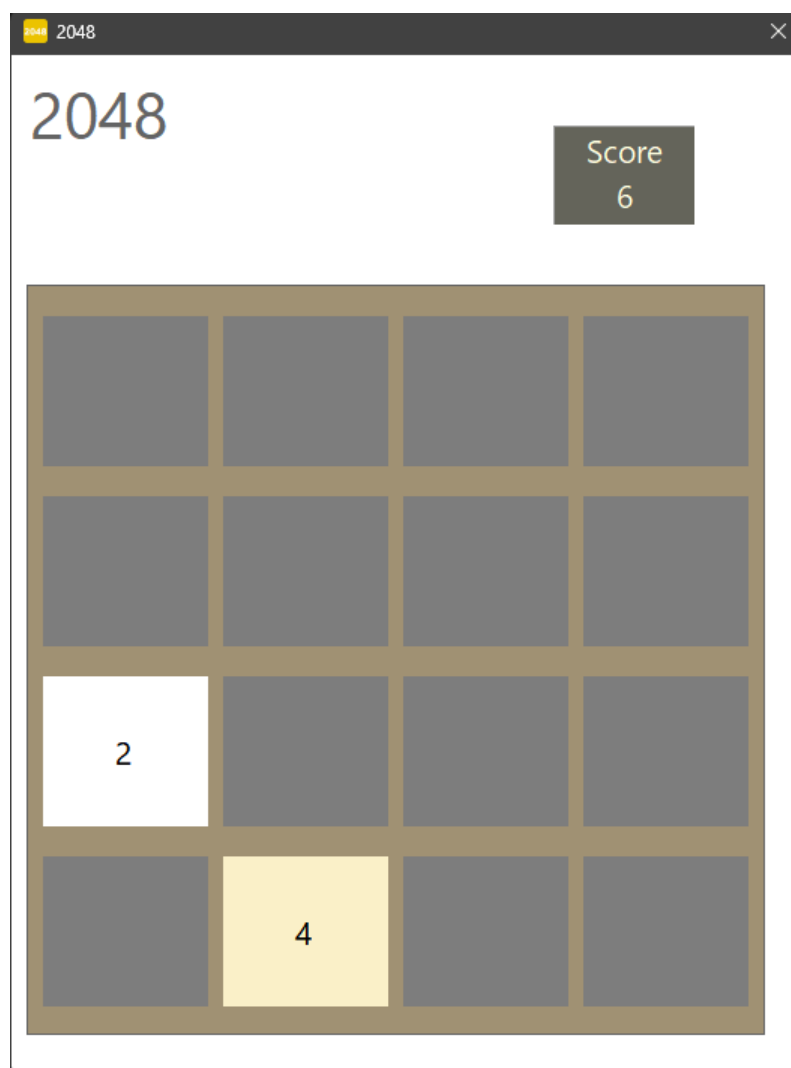


Рисунок 22- Игровое поле

Пользователь выбирает одно из 4 направлений и нажимает соответствующую клавишу(W-вверх, A-влево, S-вниз, D- вправо), тогда программа по возможности соединяет и смещает клетки в указанном направлении до границы(границей может быть как граница поля, так и клетка с иным значением внутри), а так же генерирует в одной случайной клетке значение, в 10% случаев это 4, в 90% это 2. Предположим пользователь выбрал направление вверх, тогда клетка со значением 2 и клетка со значением 4 сместиться вверх в 1 строку и появится значение в случайной клетке (рисунок 23).

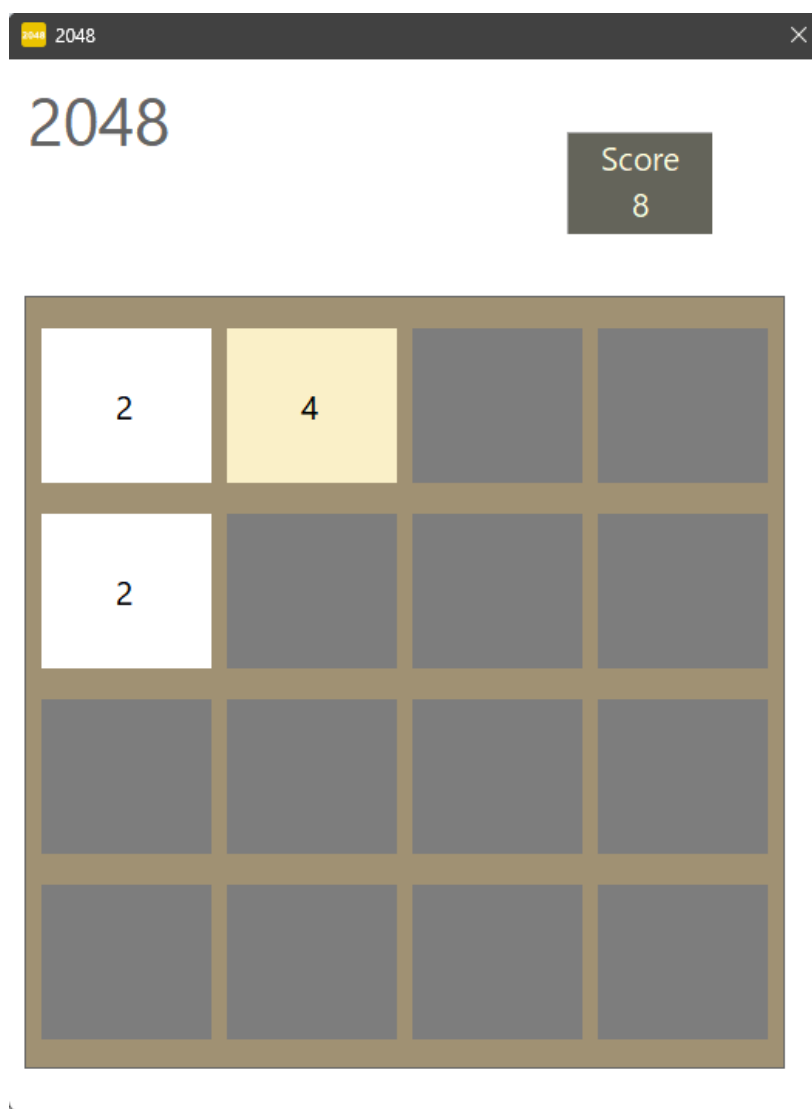


Рисунок 23- смещение клеток вверх

Клетка появилась прямо под двойкой, их значения равны а значит при нажатии вверх/вниз они соединятся, выберем направление вниз (рисунок 24).

При поражении или победе, у пользователя будет возможность сыграть снова или выйти из программы. При нажатии клавиши R/r происходит рестарт игры, а при нажатии Q/q выход (рисунки 25, 26). Поражением считается, когда у пользователя не осталось ходов. Победой же считается набор значения 2048 в одной из клеток.

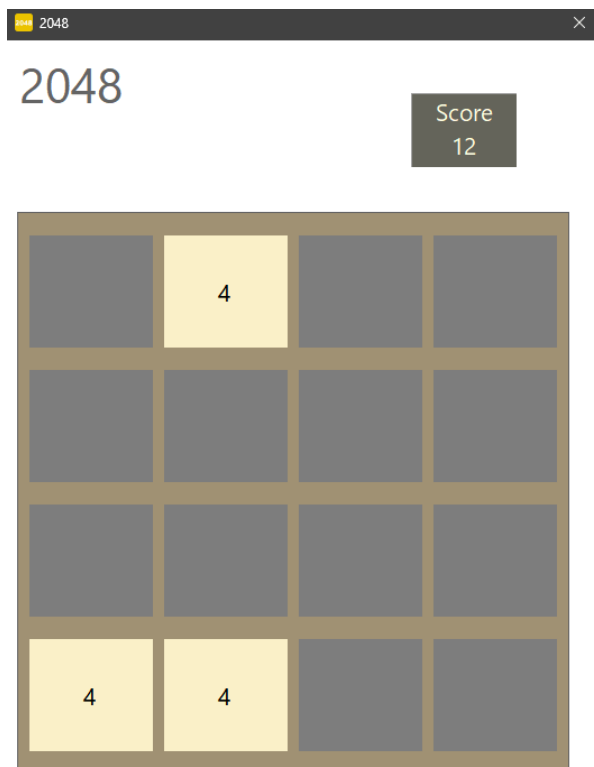


Рисунок 24- смещение с последующим соединением

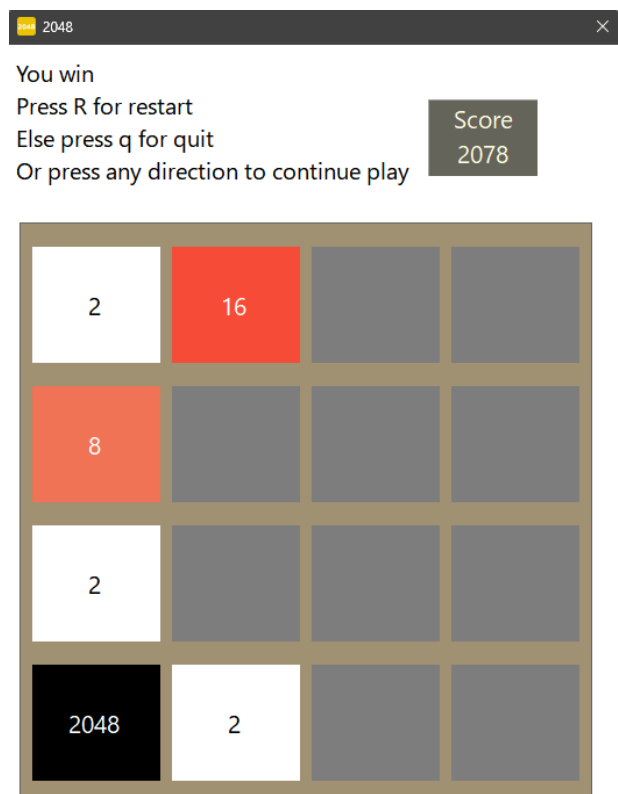


Рисунок 25- Поражение/Победа с возможностью рестарта/выхода

7. Заключение

В ходе лабораторной работы были получены навыки ООП и разработки многомодульных программ. Были изучены способности обработки событий с мышки и клавиатуры. Также был получен опыт в создании графического интерфейса.

Данный курсовой проект был выполнен в соответствии поставленной задачи, в среде Microsoft Visual Studio 2022. Было проведено исследование компонентов программной среды Microsoft Visual Studio 2022, которые использовались при создании программы, также использовалось множество функций. В результате выполнения данной курсовой работы, мной была разработана игровая программа под названием “2048”

Чтобы улучшить данный проект можно добавить поля разного размера, а также анимации движения/появления/соединения ячеек.

Список литературы.

1. Роберт Лафоре. Объектно-ориентированное программирование в С++
2. Керниган Б. Ритчи Д. Язык программирования С. 1985 г.
3. Стивен Прата. Язык программирования С++. Лекции и упражнения
4. Галисеев Г.В. Ассемблер для Win32. Самоучитель
5. Скотт Мэйерс — Эффективное использование С++. 55 верных советов улучшить структуру и код ваших программ

Приложение А

Листинги программы

Файл MyForm.cpp

```
#include "MyForm.h"
using namespace System;
using namespace System::Windows::Forms;
[STAThreadAttribute]
int main()
{
    Application::SetCompatibleTextRenderingDefault(false);
    Application::EnableVisualStyles;
    My2048::MyForm form;
    Application::Run(% form);
}
```

Файл MyForm.h

```
#pragma once
#include <time.h>
#include <cstdlib>
#include <iostream>
#pragma unmanaged
int check_win(int zxc) {
    __asm {
        cmp zxc, 2048
        jne not_win
        mov zxc, 1
        not_win:
    }
    if (zxc == 1) return 1;
    else return 0;
}
#pragma managed
namespace My2048 {
```

```

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

public ref class MyForm : public
System::Windows::Forms::Form
{
public:
    MyForm(void)
    {
        srand(time(NULL));
        InitializeComponent();
    }
protected:
    /// <summary>
    /// Освободить все используемые ресурсы.
    /// </summary>
    ~MyForm()
    {
        if (components)
        {
            delete components;
        }
    }
private: System::Windows::Forms::Label^ cap;
private: System::Windows::Forms::Label^ score;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Panel^ panel1;
private: System::Windows::Forms::Button^ Opening;
private: array<System::Windows::Forms::Label^>^ cells;

```

```

private: int a;
private: int b;
private: int c;
private: int d;
private: int e;
private: int wins=0;
protected:
private: System::ComponentModel::Container^ components;
#pragma region Windows Form Designer generated code
    void InitializeComponent()
    {

System::ComponentModel::ComponentResourceManager^ resources =
(gcnew
System::ComponentModel::ComponentResourceManager(MyForm::typeid))
;

        this->Opening = (gcnew
System::Windows::Forms::Button());
        this->cap = (gcnew
System::Windows::Forms::Label());
        this->score = (gcnew
System::Windows::Forms::Label());
        this->panell = (gcnew
System::Windows::Forms::Panel());
        this->cells = (gcnew
array<System::Windows::Forms::Label^>(16));
        this->panell->SuspendLayout();
        this->SuspendLayout();
        //
        // Opening
        //

        this->Opening->BackColor =
System::Drawing::Color::Black;

```

```

        this->Opening->FlatAppearance->BorderColor =
System::Drawing::Color::Black;

this->Opening->FlatAppearance->MouseDownBackColor =
System::Drawing::Color::Black;

this->Opening->FlatAppearance->MouseOverBackColor =
System::Drawing::Color::Black;

        this->Opening->FlatAppearance->BorderSize = 0;
        this->Opening->FlatStyle =
System::Windows::Forms::FlatStyle::Flat;
        this->Opening->Location =
System::Drawing::Point(0, 0);

        this->Opening->Font = (gcnew
System::Drawing::Font(L"Yu Gothic UI", 32.25F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));

        this->Opening->ForeColor =
System::Drawing::Color::WhiteSmoke;

        this->Opening->Name = L"Opening";
        this->Opening->Size =
System::Drawing::Size(526, 683);

        this->Opening->TabIndex = 0;
        this->Opening->Text = L"2048\press to
start\n\n\n\n\n\n\n\n\nnanteiky prod.";

        this->Opening->Click += gcnew
System::EventHandler(this, &MyForm::Myform_Opening);
        //
        // cap
        //
        this->cap->AutoSize = true;

```

```

                                this->cap->BackColor          =
System::Drawing::Color::White;

                                this->cap->Font              =      (gcnew
System::Drawing::Font(L"Yu      Gothic      UI",      32.5F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                                static_cast<System::Byte>(204)));
                                this->cap->Location = System::Drawing::Point(2,
9);

                                this->cap->Name = L"cap";
                                this->cap->Size  = System::Drawing::Size(117,
59);

                                this->cap->TabIndex = 0;
                                this->cap->Text = L"2048";
                                //
                                // score
                                //

                                this->score->BackColor        =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_
c_cast<System::Byte>(102)),
static_cast<System::Int32>(static_cast<System::Byte>(100)),

static_cast<System::Int32>(static_cast<System::Byte>(91))));
                                this->score->BorderStyle      =
System::Windows::Forms::BorderStyle::Fixed3D;
                                this->score->FlatStyle         =
System::Windows::Forms::FlatStyle::System;
                                this->score->Font              =      (gcnew
System::Drawing::Font(L"Yu      Gothic      UI",      15.75F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
                                static_cast<System::Byte>(204)));

```

```

        this->score->ForeColor          =
System::Drawing::SystemColors::Info;
        this->score->Location          =
System::Drawing::Point(361, 47);
        this->score->Name = L"score";
        this->score->Size = System::Drawing::Size(95,
67);
        this->score->TabIndex = 1;
        this->score->Text = L"SCORE";
        this->score->TextAlign          =
System::Drawing::ContentAlignment::MiddleCenter;
        //
        // panel1
        //
        this->panel1->BackColor          =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static
c_cast<System::Byte>(163)),
static_cast<System::Int32>(static_cast<System::Byte>(145)),
static_cast<System::Int32>(static_cast<System::Byte>(118)));
        this->panel1->BorderStyle          =
System::Windows::Forms::BorderStyle::FixedSingle;
        this->panel1->Location          =
System::Drawing::Point(10, 153);
        this->panel1->Name = L"panel1";
        this->panel1->Size = System::Drawing::Size(492,
500);
        this->panel1->TabIndex = 3;
        srand(time(NULL));
        for (int i = 0; i < 16; i++)
        {
            this->cells[i]          =          gcnew
System::Windows::Forms::Label();

```

```

        this->cells[i]->Text = "0";
    }
    randomcell();
    randomcell();
    for (int i = 0; i < 16; i++)
    {

this->panel1->Controls->Add(this->cells[i]);

        this->cells[i]->Font      =      (gcnew
System::Drawing::Font(L"Yu      Gothic      UI",      15.75F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        if (i < 4)      this->cells[i]->Location =
System::Drawing::Point(10 + 120 * i, 20);
        else if (i < 8)      this->cells[i]->Location
= System::Drawing::Point(10 + 120 * (i - 4), 140);
        else if (i < 12)      this->cells[i]->Location
= System::Drawing::Point(10 + 120 * (i - 8), 260);
        else if (i < 16)      this->cells[i]->Location
= System::Drawing::Point(10 + 120 * (i - 12), 380);
        this->cells[i]->Size      =
System::Drawing::Size(110, 100);

        if
(System::Convert::ToInt32(cells[i]->Text) == 0)
        {
            this->cells[i]->ForeColor      =
System::Drawing::Color::Gray;
            this->cells[i]->BackColor      =
System::Drawing::Color::Gray;
        }
    }

```

```

else if
(System::Convert::ToInt32(cells[i]->Text) == 2)
{
    this->cells[i]->BackColor =
System::Drawing::Color::White;
    this->cells[i]->ForeColor =
System::Drawing::Color::Black;
}

else if
(System::Convert::ToInt32(cells[i]->Text) == 4)
{
    this->cells[i]->BackColor =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_
c_cast<System::Byte>(252)),
static_cast<System::Int32>(static_cast<System::Byte>(244)),

static_cast<System::Int32>(static_cast<System::Byte>(200))));
    this->cells[i]->ForeColor =
System::Drawing::Color::Black;
}

else if
(System::Convert::ToInt32(cells[i]->Text) == 8)
{
    this->cells[i]->BackColor =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_
c_cast<System::Byte>(243)),
static_cast<System::Int32>(static_cast<System::Byte>(115)),

static_cast<System::Int32>(static_cast<System::Byte>(87))));
    this->cells[i]->ForeColor =
System::Drawing::Color::White;
}

```

```

else if
(System::Convert::ToInt32(cells[i]->Text) == 16)
{
    this->cells[i]->BackColor =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_
c_cast<System::Byte>(245)),
static_cast<System::Int32>(static_cast<System::Byte>(76)),

static_cast<System::Int32>(static_cast<System::Byte>(58))));
    this->cells[i]->ForeColor =
System::Drawing::Color::White;
}

else if
(System::Convert::ToInt32(cells[i]->Text) == 32)
{
    this->cells[i]->BackColor =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_
c_cast<System::Byte>(240)),
static_cast<System::Int32>(static_cast<System::Byte>(56)),

static_cast<System::Int32>(static_cast<System::Byte>(41))));
    this->cells[i]->ForeColor =
System::Drawing::Color::White;
}

else if
(System::Convert::ToInt32(cells[i]->Text) == 64)
{
    this->cells[i]->BackColor =
System::Drawing::Color::Red;
    this->cells[i]->ForeColor =
System::Drawing::Color::White;
}

```

```

else if
(System::Convert::ToInt32(cells[i]->Text) == 128)
{
    this->cells[i]->BackColor =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_
c_cast<System::Byte>(193)),
static_cast<System::Int32>(static_cast<System::Byte>(183)),

static_cast<System::Int32>(static_cast<System::Byte>(87))));
    this->cells[i]->ForeColor =
System::Drawing::Color::White;
}

else if
(System::Convert::ToInt32(cells[i]->Text) == 256)
{
    this->cells[i]->BackColor =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_
c_cast<System::Byte>(207)),
static_cast<System::Int32>(static_cast<System::Byte>(179)),

static_cast<System::Int32>(static_cast<System::Byte>(34))));
    this->cells[i]->ForeColor =
System::Drawing::Color::White;
}

else if
(System::Convert::ToInt32(cells[i]->Text) == 512)
{
    this->cells[i]->BackColor =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_
c_cast<System::Byte>(183)),
static_cast<System::Int32>(static_cast<System::Byte>(120)),

static_cast<System::Int32>(static_cast<System::Byte>(19))));

```

```

                                this->cells[i]->ForeColor    =
System::Drawing::Color::White;
                                }
                                else if
(System::Convert::ToInt32(cells[i]->Text) == 1024)
                                {
                                this->cells[i]->BackColor    =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static
c_cast<System::Byte>(106)),
static_cast<System::Int32>(static_cast<System::Byte>(70)),
static_cast<System::Int32>(static_cast<System::Byte>(11))));;
                                this->cells[i]->ForeColor    =
System::Drawing::Color::White;
                                }
                                else if
(System::Convert::ToInt32(cells[i]->Text) >= 2048)
                                {
                                this->cells[i]->BackColor    =
System::Drawing::Color::Black;
                                this->cells[i]->ForeColor    =
System::Drawing::Color::White;
                                }
                                this->cells[i]->TextAlign    =
System::Drawing::ContentAlignment::MiddleCenter;

                                }
Score();
//
// MyForm
//
                                this->AutoScaleDimensions    =
System::Drawing::SizeF(6, 13);

```

```

this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
this->BackColor =
System::Drawing::Color::White;
this->ClientSize = System::Drawing::Size(526,
681);

this->Controls->Add(this->Opening);
this->Controls->Add(this->panell);
this->Controls->Add(this->labell);
this->Controls->Add(this->score);
this->Controls->Add(this->cap);

this->Icon =
(cli::safe_cast<System::Drawing::Icon^>(resources->GetObject(L"$t
his.Icon")));

this->Font = (gcnew System::Drawing::Font(L"Yu
Gothic UI", 8.25F, System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
this->FormBorderStyle =
System::Windows::Forms::FormBorderStyle::FixedDialog;
this->MaximizeBox = false;
this->MinimizeBox = false;
this->Name = L"MyForm";
this->StartPosition =
System::Windows::Forms::FormStartPosition::CenterScreen;
this->Text = L"2048";
this->panell->ResumeLayout(false);
this->KeyPress += gcnew
System::Windows::Forms::KeyPressEventHandler(this,
&MyForm::MyForm_KeyPress);
this->ResumeLayout(false);
this->PerformLayout();
}

```

```

#pragma endregion

    bool is_win() {
        for (int i = 0; i < 16; i++)
        {
            int zxc =
System::Convert::ToInt32(this->cells[i]->Text);
            if (check_win(zxc))
            {
                return true;
            }
        }
        return false;
    }

    void randomcell()
    {

        int counter = 0;
        for (int i = 0; i < 16; i++)
        {
            int a =
(System::Convert::ToInt16(this->cells[i]->Text));
            if (a == 0)
            {
                counter++;
            }
        }
        if (counter == 0) return;
        else
        {
            counter = 1;
            while (counter > 0)
            {
                int x = rand() % 16;

```

```

int a =
System::Convert::ToInt32(this->cells[x]->Text);
    if (a == 0)
    {
        if (rand() % 100 <= 10)
this->cells[x]->Text = System::Convert::ToString(4);
        else this->cells[x]->Text =
System::Convert::ToString(2);
        counter--;
    }
}
}

bool is_lose()
{
    bool lose = false;
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            this->a =
System::Convert::ToInt32(this->cells[i * 4 + j]->Text);
            if (a == 0) return false;
        }
    }
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            this->a =
System::Convert::ToInt32(this->cells[i * 4 + j]->Text);

```

```

        if (i < 3) this->b =
System::Convert::ToInt32(this->cells[(i + 1) * 4 + j]->Text);
        if (i > 0) this->c =
System::Convert::ToInt32(this->cells[(i - 1) * 4 + j]->Text);
        if (j < 3) this->d =
System::Convert::ToInt32(this->cells[i * 4 + j + 1]->Text);
        if (j > 0) this->e =
System::Convert::ToInt32(this->cells[i * 4 + j - 1]->Text);
        if (a == b && i < 3) return false;
        else if (a == c && i > 0) return
false;

        else if (a == d && j < 3) return
false;

        else if (a == e && j > 0) return
false;

    }

}

return true;
}

void Modify() {
    for (int i = 0; i < 16; i++)
    {
        if
(System::Convert::ToInt32(cells[i]->Text) == 0)
        {
            this->cells[i]->ForeColor =
System::Drawing::Color::Gray;
            this->cells[i]->BackColor =
System::Drawing::Color::Gray;
        }
        else if
(System::Convert::ToInt32(cells[i]->Text) == 2)
        {

```

```

                                this->cells[i]->BackColor    =
System::Drawing::Color::White;
                                this->cells[i]->ForeColor    =
System::Drawing::Color::Black;
                                }
                                else if
(System::Convert::ToInt32(cells[i]->Text) == 4)
                                {
                                    this->cells[i]->BackColor    =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_
c_cast<System::Byte>(252)),
static_cast<System::Int32>(static_cast<System::Byte>(244)),
static_cast<System::Int32>(static_cast<System::Byte>(200))));
                                    this->cells[i]->ForeColor    =
System::Drawing::Color::Black;
                                }
                                else if
(System::Convert::ToInt32(cells[i]->Text) == 8)
                                {
                                    this->cells[i]->BackColor    =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_
c_cast<System::Byte>(243)),
static_cast<System::Int32>(static_cast<System::Byte>(115)),
static_cast<System::Int32>(static_cast<System::Byte>(87))));
                                    this->cells[i]->ForeColor    =
System::Drawing::Color::White;
                                }
                                else if
(System::Convert::ToInt32(cells[i]->Text) == 16)
                                {

```

```

                                this->cells[i]->BackColor    =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static
c_cast<System::Byte>(245)),
static_cast<System::Int32>(static_cast<System::Byte>(76)),

static_cast<System::Int32>(static_cast<System::Byte>(58))));;
                                this->cells[i]->ForeColor    =
System::Drawing::Color::White;
                                }

                                else                                if
(System::Convert::ToInt32(cells[i]->Text) == 32)
                                {
                                    this->cells[i]->BackColor    =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static
c_cast<System::Byte>(240)),
static_cast<System::Int32>(static_cast<System::Byte>(56)),

static_cast<System::Int32>(static_cast<System::Byte>(41))));;
                                    this->cells[i]->ForeColor    =
System::Drawing::Color::White;
                                    }

                                else                                if
(System::Convert::ToInt32(cells[i]->Text) == 64)
                                {
                                    this->cells[i]->BackColor    =
System::Drawing::Color::Red;
                                    this->cells[i]->ForeColor    =
System::Drawing::Color::White;
                                    }

                                else                                if
(System::Convert::ToInt32(cells[i]->Text) == 128)
                                {

```

```

                                this->cells[i]->BackColor    =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_
c_cast<System::Byte>(193)),
static_cast<System::Int32>(static_cast<System::Byte>(183)),

static_cast<System::Int32>(static_cast<System::Byte>(87))));;
                                this->cells[i]->ForeColor    =
System::Drawing::Color::White;
                                }

                                else                                if
(System::Convert::ToInt32(cells[i]->Text) == 256)
                                {
                                    this->cells[i]->BackColor    =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_
c_cast<System::Byte>(207)),
static_cast<System::Int32>(static_cast<System::Byte>(179)),

static_cast<System::Int32>(static_cast<System::Byte>(34))));;
                                    this->cells[i]->ForeColor    =
System::Drawing::Color::White;
                                    }

                                else                                if
(System::Convert::ToInt32(cells[i]->Text) == 512)
                                {
                                    this->cells[i]->BackColor    =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static_
c_cast<System::Byte>(183)),
static_cast<System::Int32>(static_cast<System::Byte>(120)),

static_cast<System::Int32>(static_cast<System::Byte>(19))));;
                                    this->cells[i]->ForeColor    =
System::Drawing::Color::White;
                                    }

```

```

else if
(System::Convert::ToInt32(cells[i]->Text) == 1024)
{
    this->cells[i]->BackColor =
System::Drawing::Color::FromArgb(static_cast<System::Int32>(static
c_cast<System::Byte>(106)),
static_cast<System::Int32>(static_cast<System::Byte>(70)),
static_cast<System::Int32>(static_cast<System::Byte>(11))));
    this->cells[i]->ForeColor =
System::Drawing::Color::White;
}
else if
(System::Convert::ToInt32(cells[i]->Text) >= 2048)
{
    this->cells[i]->BackColor =
System::Drawing::Color::Black;
    this->cells[i]->ForeColor =
System::Drawing::Color::White;
}
}
void Score()
{
    int counter = 0;
    for (int i = 0; i < 16; i++)
    {
        counter +=
System::Convert::ToInt32(this->cells[i]->Text);
    }
    this->score->Text = "Score\n" +
System::Convert::ToString(counter);
}

```

```

        private: System::Void Myform_Opening(System::Object^ sender,
System::EventArgs^ e) {
            _sleep(100);
            delete(this->Opening);
        };

        private:      System::Void      MyForm_KeyPress(System::Object^
sender, System::Windows::Forms::KeyPressEventArgs^ e) {
            Score();
            bool move = false;
            char direction = (char)tolower(e->KeyChar);
            if (direction == 'w')
            {
                for (int i = 0; i < 4; i++)
                {
                    for (int j = 0; j < 4; j++)
                    {
                        this->a =
System::Convert::ToInt32(cells[i * 4 + j]->Text);
                        if (i < 3) this->b =
System::Convert::ToInt32(this->cells[(i + 1) * 4 + j]->Text);
                        if (i < 2) this->c =
System::Convert::ToInt32(this->cells[(i + 2) * 4 + j]->Text);
                        if (i == 0) this->d =
System::Convert::ToInt32(this->cells[(i + 3) * 4 + j]->Text);
                        if (a & b && i < 3)
                        {
                            this->cells[i * 4 + j]->Text =
System::Convert::ToString(a << 1);
                            this->cells[(i + 1) * 4 + j]->Text
= "0";
                            move = true;
                        }
                    }
                }
            }

```

```

else if (a & c && b == 0 && i < 2)
{
    this->cells[i * 4 + j]->Text =
System::Convert::ToString(a << 1);
    this->cells[(i + 2) * 4 + j]->Text
= "0";

    move = true;
}
else if (a & d && b == 0 && c == 0 && i
== 0)
{
    this->cells[i * 4 + j]->Text =
System::Convert::ToString(a << 1);
    this->cells[(i + 3) * 4 + j]->Text
= "0";

    move = true;
}
}
}
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        this->a =
System::Convert::ToInt32(this->cells[i * 4 + j]->Text);
        if (i < 3) this->b =
System::Convert::ToInt32(this->cells[(i + 1) * 4 + j]->Text);
        if (i < 2) this->c =
System::Convert::ToInt32(this->cells[(i + 2) * 4 + j]->Text);
        if (i == 0) this->d =
System::Convert::ToInt32(this->cells[(i + 3) * 4 + j]->Text);
        if (a == 0)
        {

```

```

        if (b == 0)
        {
            if (c == 0)
            {
                if (d == 0) continue;
                if (i == 0)
                {
                    this->cells[i * 4 +
j]->Text = this->cells[(i + 3) * 4 + j]->Text;
                    this->cells[(i + 3)
* 4 + j]->Text = "0";

                    move = true;
                }
            }
            else
            {
                if (c == 0) continue;
                if (i < 2)
                {
                    this->cells[i * 4 +
j]->Text = this->cells[(i + 2) * 4 + j]->Text;
                    this->cells[(i + 2)
* 4 + j]->Text = "0";

                    move = true;
                }
            }
        }
    }
    else {
        if (b == 0) continue;
        if (i < 3)
        {
            this->cells[i * 4 +
j]->Text = this->cells[(i + 1) * 4 + j]->Text;

```



```

    }
    else if (a & c && b == 0 && j < 2)
    {
        this->cells[i * 4 + j]->Text =
System::Convert::ToString(a << 1);
        this->cells[i * 4 + j + 2]->Text =
"0";

        move = true;
    }
    else if (a & d && b == 0 && c == 0 && j
== 0)
    {
        this->cells[i * 4 + j]->Text =
System::Convert::ToString(a << 1);
        this->cells[i * 4 + j + 3]->Text =
"0";

        move = true;
    }
}

for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        this->a =
System::Convert::ToInt32(this->cells[i * 4 + j]->Text);
        if (j < 3) this->b =
System::Convert::ToInt32(this->cells[i * 4 + j + 1]->Text);
        if (j < 2) this->c =
System::Convert::ToInt32(this->cells[i * 4 + j + 2]->Text);
        if (j == 0) this->d =
System::Convert::ToInt32(this->cells[i * 4 + j + 3]->Text);
        if (a == 0)

```

```

{
    if (b == 0)
    {
        if (c == 0)
        {
            if (d == 0) continue;
            if (j == 0)
            {
                this->cells[i * 4 +
j]->Text = this->cells[i * 4 + j + 3]->Text;
                this->cells[i * 4 +
j + 3]->Text = "0";

                move = true;
            }
        }
        else
        {
            if (c == 0) continue;
            if (j < 2)
            {
                this->cells[i * 4 +
j]->Text = this->cells[i * 4 + j + 2]->Text;
                this->cells[i * 4 +
j + 2]->Text = "0";

                move = true;
            }
        }
    }
    else {
        if (b == 0) continue;
        if (j < 3)
        {

```



```

        else if (a & c && b == 0 && i > 1)
        {
            this->cells[i * 4 + j]->Text =
System::Convert::ToString(a << 1);
            this->cells[(i - 2) * 4 + j]->Text
= "0";

            move = true;
        }
        else if (a & d && b == 0 && c == 0 && i
== 3)
        {
            this->cells[i * 4 + j]->Text =
System::Convert::ToString(a << 1);
            this->cells[(i - 3) * 4 + j]->Text
= "0";

            move = true;
        }
    }

    for (int i = 3; i > 0; i--)
    {
        for (int j = 0; j < 4; j++)
        {
            this->a =
System::Convert::ToInt32(this->cells[i * 4 + j]->Text);
            if (i != 0) this->b =
System::Convert::ToInt32(this->cells[(i - 1) * 4 + j]->Text);
            if (i > 1) this->c =
System::Convert::ToInt32(this->cells[(i - 2) * 4 + j]->Text);
            if (i == 3) this->d =
System::Convert::ToInt32(this->cells[(i - 3) * 4 + j]->Text);
            if (a == 0)
            {

```

```

        if (b == 0)
        {
            if (c == 0)
            {
                if (d == 0) continue;
                if (i == 3)
                {
                    this->cells[i * 4 +
j]->Text = this->cells[(i - 3) * 4 + j]->Text;
                    this->cells[(i - 3)
* 4 + j]->Text = "0";

                    move = true;
                }
            }
            else
            {
                if (c == 0) continue;
                if (i > 1)
                {
                    this->cells[i * 4 +
j]->Text = this->cells[(i - 2) * 4 + j]->Text;
                    this->cells[(i - 2)
* 4 + j]->Text = "0";

                    move = true;
                }
            }
        }
    }
    else {
        if (b == 0) continue;
        if (i > 0)
        {
            this->cells[i * 4 +
j]->Text = this->cells[(i - 1) * 4 + j]->Text;

```

```

        this->cells[(i - 1) * 4 +
j]->Text = "0";

        move = true;
    }
}
}
}
}
else if (direction == 'd')
{
    for (int i = 0; i < 4; i++)
    {
        for (int j = 3; j > 0; j--)
        {
            this->a =
System::Convert::ToInt32(cells[i * 4 + j]->Text);
            if (j != 0) this->b =
System::Convert::ToInt32(this->cells[i * 4 + j - 1]->Text);
            if (j > 1) this->c =
System::Convert::ToInt32(this->cells[i * 4 + j - 2]->Text);
            if (j == 3) this->d =
System::Convert::ToInt32(this->cells[i * 4 + j - 3]->Text);
            if (a & b && j != 0)
            {
                this->cells[i * 4 + j]->Text =
System::Convert::ToString(a << 1);
                this->cells[i * 4 + j - 1]->Text =
"0";

                move = true;
            }
            else if (a & c && b == 0 && j > 1)
            {

```

```

        this->cells[i * 4 + j]->Text =
System::Convert::ToString(a << 1);
        this->cells[i * 4 + j - 2]->Text =
"0";

        move = true;
    }
    else if (a & d && b == 0 && c == 0 && j
== 3)
    {
        this->cells[i * 4 + j]->Text =
System::Convert::ToString(a << 1);
        this->cells[i * 4 + j - 3]->Text =
"0";

        move = true;
    }
}

for (int i = 0; i < 4; i++)
{
    for (int j = 3; j > 0; j--)
    {
        this->a =
System::Convert::ToInt32(this->cells[i * 4 + j]->Text);
        if (j != 0) this->b =
System::Convert::ToInt32(this->cells[i * 4 + j - 1]->Text);
        if (j > 1) this->c =
System::Convert::ToInt32(this->cells[i * 4 + j - 2]->Text);
        if (j == 3) this->d =
System::Convert::ToInt32(this->cells[i * 4 + j - 3]->Text);
        if (a == 0)
        {
            if (b == 0)
            {

```

```

        if (c == 0)
        {
            if (d == 0) continue;
            if (j == 3)
            {
                this->cells[i * 4 +
j]->Text = this->cells[i * 4 + j - 3]->Text;
                this->cells[i * 4 +
j - 3]->Text = "0";

                move = true;
            }
        }
    else
    {
        if (c == 0) continue;
        if (j > 1)
        {
            this->cells[i * 4 +
j]->Text = this->cells[i * 4 + j - 2]->Text;
            this->cells[i * 4 +
j - 2]->Text = "0";

            move = true;
        }
    }
}
else {
    if (b == 0) continue;
    if (j > 0)
    {
        this->cells[i * 4 +
j]->Text = this->cells[i * 4 + j - 1]->Text;
        this->cells[i * 4 + j -
1]->Text = "0";
    }
}

```

```

        move = true;
    }
}
}
}
}
}
if (move) randomcell();
Modify();
Score();
if (is_win())
{
    this->cap->Text = "You win\nPress R for
restart\nElse press q for quit\nOr press any direction to
continue play";
    this->cap->Font = (gcnew
System::Drawing::Font(L"Yu Gothic UI", 14.5F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
static_cast<System::Byte>(204)));
    char restart = (char)tolower(e->KeyChar);
    if (restart == 'r')
    {
        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < 4; j++)
            {
                this->cells[i * 4 + j]->Text =
"0";
            }
        }
        randomcell();
        randomcell();
    }
}

```

```

        Modify();
        this->cap->Text = "2048";
        this->cap->Font = (gcnew
System::Drawing::Font(L"Yu Gothic UI", 32.5F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
        Score();
    }
    else if (restart == 'q') exit(0);
    this->wins++;
}
else if (is_lose())
{
    this->cap->Text = "You lose\nPress R for
restart\nElse press q for quit";
    this->cap->Font = (gcnew
System::Drawing::Font(L"Yu Gothic UI", 14.5F,
System::Drawing::FontStyle::Regular,
System::Drawing::GraphicsUnit::Point,
        static_cast<System::Byte>(204)));
    char restart = (char)tolower(e->KeyChar);
    if (restart == 'r')
    {
        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < 4; j++)
            {
                this->cells[i * 4 + j]->Text = "0";
            }
        }
        randomcell();
        randomcell();
    }
}

```

```
        Modify();  
        this->cap->Text = "2048";  
        Score();  
    }  
    else if (restart == 'q') exit(0);  
}  
  
}  
};  
}
```

Приложение В

Результаты работы программы



You win

Press R for restart

Else press q for quit

Or press any direction to continue play

Score
4126

