

Создание игры Тетрис в Microsoft Visual Studio на языке программирования C#

Маринчук Александр Сергеевич

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

В данной статье описан процесс создания игры Тетрис в Microsoft Visual Studio на языке программирования C#. При написании использовались Windows Forms, а отрисовка происходила через класс Graphics. Реализовано несколько функций: управление фигурами и их генерация, подсчет очков, рестарт игры и пауза, коллизия. На выходе получена готовая игра Тетрис и сделаны выводы об актуальности написания различных приложений в настоящее время.

Ключевые слова: игра, тетрис, MVS, C#, программирование.

Creating a Tetris game in Microsoft Visual Studio using the C # programming language

Marinchuk Alexander Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article describes the process of creating a Tetris game in Microsoft Visual Studio using the C # programming language. When writing, we used Windows Forms, and rendering was done through the Graphics class. Several functions have been implemented: shape management and generation, scoring, game restart and pause, collision. As a result, a finished Tetris game was obtained and conclusions were drawn about the relevance of writing various applications at the present time.

Keywords: game, tetris, MVS, C #, programming.

1. Введение

1.1 Актуальность исследования

Игры давно перестали быть просто забавой или бесполезным времяпровождением. Люди делают целые состояния на играх в которые играют миллионы пользователей, а игровая индустрия, в связи с нынешней эпидемиологической ситуацией, будет только увеличиваться, так как людям необходимо чем-то занять себя на карантине. Написание программного кода, реализующего какую-либо из игр, становится как никогда актуальным, ведь в

настоящее время, сделав популярную игру, можно заработать хорошие деньги за несколько дней.

Но знать основы и принципы написания игр, которые уже обрели популярность все же стоит. Ведь понимая, что помогло автору достичь таких высоких рейтингов будет гораздо легче продумать и реализовать собственную успешную игру.

1.2 Обзор исследований

В статье Э. Э. Сейдаметова, А. Э. Шабанова представлены наиболее популярные средства разработки игр и проанализированы их характерные особенности [1]. В статье Н. А. Базеевой и Д. С. Лебедева описано исследование о начале игровой индустрии, а также рассмотрены особенности языков программирования для разработки игр [2]. А. С. Зубович и другие рассмотрели интегрированную среду разработки Microsoft Visual Studio и сделали вывод, что на сегодняшний день она является очень мощным средством разработки приложений для операционной системы Windows. Обладая большой функциональностью она предоставляет возможность создавать приложения в кратчайшие сроки [3]. В. А. Михайлов и другие в своей статье описали подход к управлению макросами файла Microsoft Excel с использованием приложения, разработанного в среде Visual Studio [4]. В исследовании Е. А. Жалниной и А. С. Зотовой приведены описания двух новых электронных учебных игр с контентом по математике школьного курса. Методическая часть статьи включает изложение построения образовательного контента, программная часть - структуры компьютерной программы, отличающейся возможностью её модификации [5]. Рассмотрены инструменты создания виртуального мира компьютерных игр в работе Г. И. Борзунова и других [6]. В статье С. К. Рыжковой и К. С. Халиковой рассмотрен проект, темой которого является привлечение школьников к выбору информатики как предмета ЕГЭ. Была достигнута цель по разработке более современного и интересного способа – компьютерная игра [7].

1.3 Цель исследования

Целью данного исследования является описание процесса создания игры Тетрис в Microsoft Visual Studio на языке программирования C#.

2. Методы исследования

В данной статье для написания игры будет рассмотрена среда разработки Microsoft Visual Studio в совокупности с языком программирования C# [8-9].

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы.

С# — компилируемый статически типизированный язык программирования общего назначения, разработанный в 1969—1973 годах сотрудником Bell Labs Деннисом Ритчи как развитие языка Би. Первоначально был разработан для реализации операционной системы UNIX, но впоследствии был перенесён на множество других платформ.

3. Результаты исследования

Для начала следует создать проект и добавить форму с несколькими label (Рис. 1).

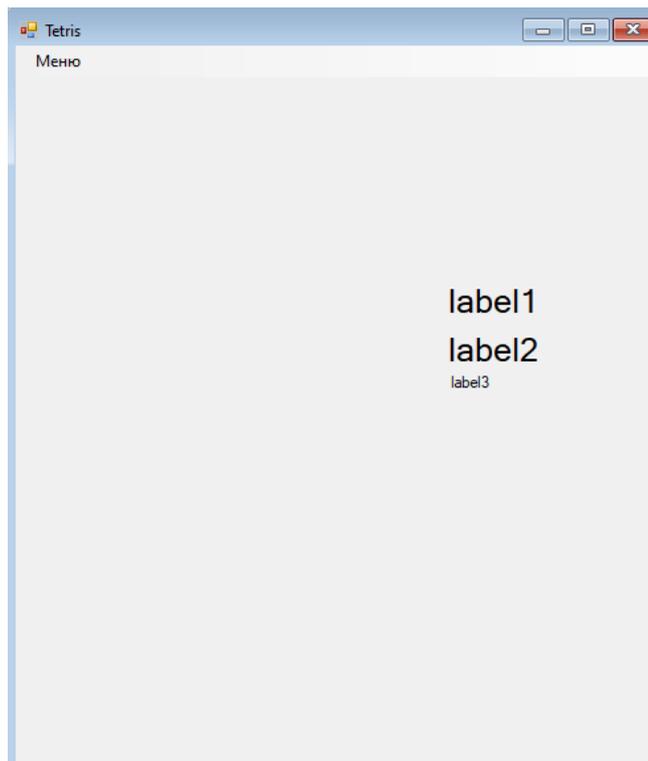


Рисунок 1 – Форма приложения

Далее следует создать класс Shape, описывающий создание фигур (Рис. 2).

```
1  ⚡ using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Tetris
8  {
9      Ссылка: 3
10     public class Shape
11     {
12         public int x;
13         public int y;
14         public int[,] matrix;
15         public int[,] nextMatrix;
16         public int sizeMatrix;
17         public int sizeNextMatrix;
18
19         public int[,] tetr1 = new int[4, 4]{
20             {0,0,1,0 },
21             {0,0,1,0 },
22             {0,0,1,0 },
23             {0,0,1,0 },
24         };
25
26         public int[,] tetr2 = new int[3, 3]{
27             {0,2,0 },
28             {0,2,2 },
29             {0,0,2 },
30         };
31
32         public int[,] tetr3 = new int[3, 3]{
33             {0,0,0 },
34             {3,3,3 },
35             {0,3,0 },
36         };
37
38         public int[,] tetr4 = new int[3, 3]{
39             { 4,0,0 },
40             {4,0,0 },
41             {4,4,0 },
42         };
43
44         public int[,] tetr5 = new int[2, 2]{
45             { 5,5 },
46             {5,5 },
47         };
48     }
49 }
```

Рисунок 2 – Класс Shape

Также здесь следует описать случайное появление фигур при игре, а также их движение (Рис. 3).

```
public int[,] GenerateMatrix()
{
    int[,] _matrix = tetr1;
    Random r = new Random();
    switch (r.Next(1, 6))
    {
        case 1:
            _matrix = tetr1;
            break;
        case 2:
            _matrix = tetr2;
            break;
        case 3:
            _matrix = tetr3;
            break;
        case 4:
            _matrix = tetr4;
            break;
        case 5:
            _matrix = tetr5;
            break;
    }
    return _matrix;
}

// ссылка: 1
public void RotateShape()
{
    int[,] tempMatrix = new int[sizeMatrix, sizeMatrix];
    for (int i = 0; i < sizeMatrix; i++)
    {
        for (int j = 0; j < sizeMatrix; j++)
        {
            tempMatrix[i, j] = matrix[j, (sizeMatrix - 1) - i];
        }
    }
    matrix = tempMatrix;
    int offset1 = (8 - (x + sizeMatrix));
    if (offset1 < 0)
    {
        for (int i = 0; i < Math.Abs(offset1); i++)
            MoveLeft();
    }

    if (x < 0)
    {
        for (int i = 0; i < Math.Abs(x)+1; i++)
            MoveRight();
    }
}
```

Рисунок 3 – Генерация фигур и управление ими

Следующим шагом будет написание кода формы. Добавим в начале игры возможность ввода имени игрока (Рис. 4-5).

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.IO;
7  using System.Linq;
8  using System.Text;
9  using System.Threading.Tasks;
10 using System.Windows.Forms;
11 using Tetris.Controllers;
12
13 namespace Tetris
14 {
15     public partial class Form1 : Form
16     {
17
18         string playerName;
19
20
21         public Form1()
22         {
23             InitializeComponent();
24             playerName = Microsoft.VisualBasic.Interaction.InputBox("Введите имя игрока", "Настройка игрока", "Новый игрок");
25             if(playerName == "")
26             {
27                 playerName = "Новый игрок";
28             }
29             this.KeyUp += new KeyEventHandler(keyFunc);
30             Init();
31         }
32     }
33 }

```

Рисунок 4 – Код инициализации начала игры

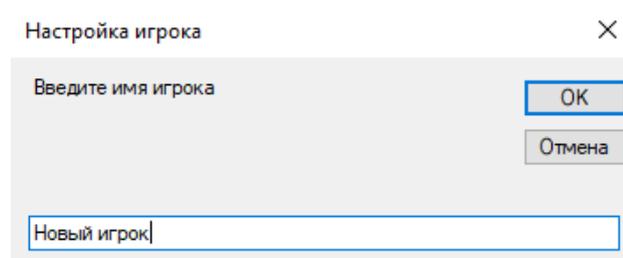


Рисунок 5 – Начальное окно игры

Функция `Init` отвечает за инициализацию начала игры, запуская таймер (Рис. 6).

```

public void Init()
{
    //RecordsController.ShowRecords(label3);
    this.Text = "Тетрис: Текущий игрок - " + playerName;
    MapController.size = 25;
    MapController.score = 0;
    MapController.linesRemoved = 0;
    MapController.currentShape = new Shape(3, 0);
    MapController.Interval = 300;
    label1.Text = "Score: " + MapController.score;
    label2.Text = "Lines: " + MapController.linesRemoved;

    timer1.Interval = MapController.Interval;
    timer1.Tick += new EventHandler(update);
    timer1.Start();

    Invalidate();
}

```

Рисунок 6 – Функция `Init`

Далее напишем функцию KeyFunc для поворота фигурок вокруг оси, их движения влево и вправо, а также напишем код, отвечающий за ускорение фигуры при нажатии на пробел (Рис. 7).

```
private void keyFunc(object sender, KeyEventArgs e)
{
    switch (e.KeyCode)
    {
        case Keys.A:
            if (!MapController.IsIntersects())
            {
                MapController.ResetArea();
                MapController.currentShape.RotateShape();
                MapController.Merge();
                Invalidate();
            }
            break;
        case Keys.Space:
            timer1.Interval = 10;
            break;
        case Keys.Right:
            if (!MapController.CollideHor(1))
            {
                MapController.ResetArea();
                MapController.currentShape.MoveRight();
                MapController.Merge();
                Invalidate();
            }
            break;
        case Keys.Left:
            if (!MapController.CollideHor(-1))
            {
                MapController.ResetArea();
                MapController.currentShape.MoveLeft();
                MapController.Merge();
                Invalidate();
            }
            break;
    }
}
```

Рисунок 7 – Функция KeyFunc

Теперь опишем функцию OnPaint, которая будет отвечать за отрисовку сетки по которой двигаются фигуры. Также создадим на форме меню, где будут располагаться 3 кнопки: пауза, начать заново и справка. Функции OnPauseButtonClick и OnAgainButton отвечают за постановку игры на паузу и ее рестарт соответственно (Рис. 8).

```
private void OnPaint(object sender, PaintEventArgs e)
{
    MapController.DrawGrid(e.Graphics);
    MapController.DrawMap(e.Graphics);
    MapController.ShowNextShape(e.Graphics);
}

ссылка: 1
private void OnPauseButtonClick(object sender, EventArgs e)
{
    var pressedButton = sender as ToolStripMenuItem;
    if (timer1.Enabled)
    {
        pressedButton.Text = "Продолжить";
        timer1.Stop();
    }
    else
    {
        pressedButton.Text = "Пауза";
        timer1.Start();
    }
}

ссылка: 1
private void OnAgainButtonClick(object sender, EventArgs e)
{
    timer1.Tick -= new EventHandler(update);
    timer1.Stop();
    MapController.ClearMap();
    Init();
}
```

Рисунок 8 – Код отрисовки сетки и кнопок рестарта и паузы

Следующим шагом будет написание кода кнопки справки (Рис. 9).

```
ссылка: 1
private void OnInfoPressed(object sender, EventArgs e)
{
    string infoString = "";
    infoString = "Для управление фигурами используйте стрелочку влево/вправо.\n";
    infoString += "Чтобы ускорить падение фигуры - нажмите 'Пробел'.\n";
    infoString += "Для поворота фигуры используйте 'A'.\n";
    MessageBox.Show(infoString, "Справка");
}
```

Рисунок 9 – Код кнопки справки

Последним шагом будет написание функции update, которая будет отвечать за обновление результата игрока, очистку сетки при проигрыше и проверку на коллизию (Рис. 10).

```
private void update(object sender, EventArgs e)
{
    MapController.ResetArea();
    if (!MapController.Collide())
    {
        MapController.currentShape.MoveDown();
    }
    else
    {
        MapController.Merge();
        MapController.SliceMap(label1, label2);
        timer1.Interval = MapController.Interval;
        MapController.currentShape.ResetShape(3, 0);
        if (MapController.Collide())
        {
            MapController.ClearMap();
            timer1.Tick -= new EventHandler(update);
            timer1.Stop();
            MessageBox.Show("Ваш результат: " + MapController.score);
            Init();
        }
    }
    MapController.Merge();
    Invalidate();
}
```

Рисунок 10 – Функция update

Запустим проект и посмотрим, как выглядит готовый вид игры (Рис. 11).

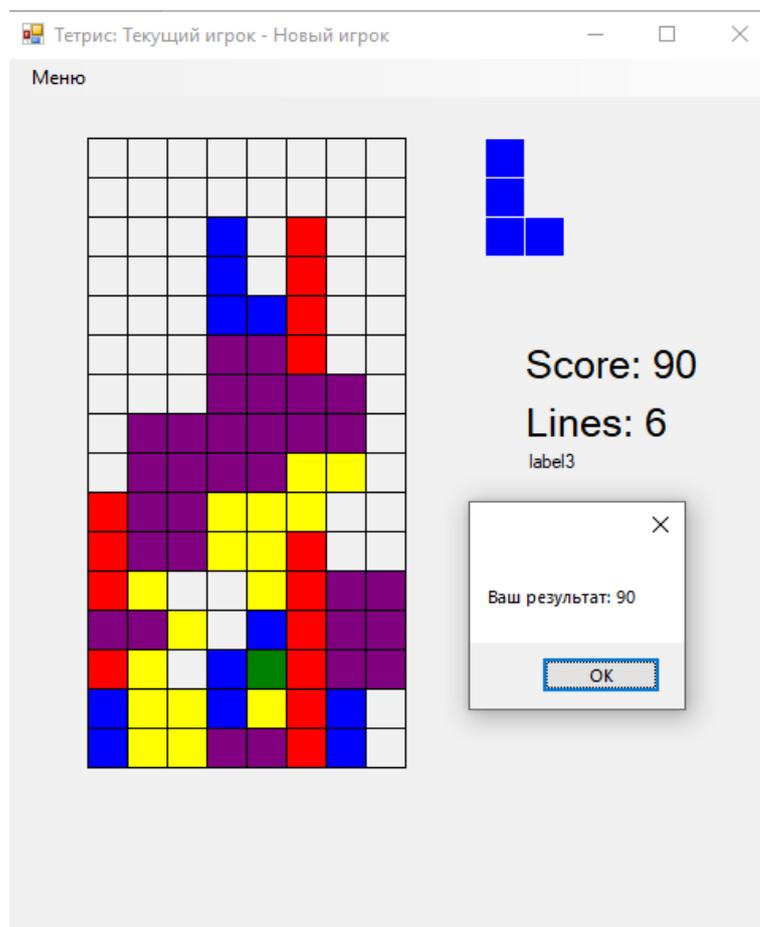


Рисунок 11 – Вид готовой игры

4. Выводы

В рамках данного исследования был рассмотрен инструмент MVS для написания игры с помощью языка программирования C#. Выделены основные аспекты на которые следует обратить внимание при создании собственного приложения. Также в работе были реализованы все планируемые функции и таким образом написана игра Тетрис.

Библиографический список

1. Сейдаметов Э. Э., Шабанов А. Э. Современные средства разработки игр // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2020. №. 1. С. 54-62
2. Базеева Н. А., Лебедев Д. С. Языки программирования для создания игр // E-Scio. 2019. №. 4 (31).
3. Зубович А. С., Егунова А. И., Аббакумов А. А. Среда разработки Microsoft Visual Studio на примере создания логической игры "крестики-нолики" // Роль инноваций в трансформации современной науки. 2017. С. 59-62.
4. Михайлов В. А., Исянова А. Ш., Михайлова Т. А. Использование возможностей среды Visual Studio для разработки приложения управления макросами файла Microsoft Excel // Актуальные проблемы науки и образования в современном ВУЗе. 2019. С. 102-106
5. Жалнина Е. А., Зотова А. С. Создание компьютерных учебных игр на программной платформе HTML // Школьные годы. 2017. №. 70. С. 51-54.
6. Борзунов Г. И., Кольцов И. А., Фирсов Д. А. Инструменты создания виртуального мира компьютерных игр // Дизайн, технологии и инновации в текстильной и легкой промышленности (ИННОВАЦИИ-2016). 2016. С. 21-24.
7. Рыжкова С. К., Халикова К. С. Проект по созданию компьютерной игры "Информатика везде" // Гуманитарные научные исследования. 2016. №. 12. С. 143-147.
8. Microsoft Visual Studio URL:
https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio (дата обращения: 12.01.2021).
9. Язык программирования C# URL:
[https://ru.wikipedia.org/wiki/C#_\(язык_программирования\)](https://ru.wikipedia.org/wiki/C#_(язык_программирования)) (дата обращения: 12.01.2021).