

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

Маринчук Александр Сергеевич

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

На сегодняшний день существует огромное количество игр разнообразных жанров, в которые играют миллионы людей со всего мира. Каждая игра это огромный труд зачастую даже не одного человека, а целой команды. Но создать игру бывает не так сложно, как придумать основную идею или механику, которая затянет пользователя своей уникальностью и в итоге принесет неплохую прибыль. В данной статье описано создание игры Arkanoid в среде Microsoft Visual Studio на языке программирования C#.

Ключевые слова: арканойд, C#, MVS, winforms, graphics.

Creating an Arkanoid game in Microsoft Visual Studio using the C# programming language

Marinchuk Alexander Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

Today there are a huge number of games of various genres that are played by millions of people from all over the world. Each game is a huge work, often not even by one person, but by a whole team. But creating a game is not as difficult as coming up with a basic idea or mechanic that will attract the user with its uniqueness and ultimately bring a good profit. This article describes how to create an Arkanoid game in Microsoft Visual Studio using the C # programming language.

Keywords: arkanoid, C #, MVS, winforms, graphics.

1. Введение

1.1 Актуальность исследования

На сегодняшний день существует огромное количество игр разнообразных жанров, в которые играют миллионы людей со всего мира. Каждая игра это огромный труд зачастую даже не одного человека, а целой команды. Но создать игру бывает не так сложно, как придумать основную идею или механику, которая затянет пользователя своей уникальностью и в итоге принесет неплохую прибыль.

Начинающим программистам подойдет изучение классических игр и механик, используемых в них, чтобы лучше понять, как работает язык программирования, с чего начать создание будущего проекта или попытаться улучшить старую механику, добавив в нее что-то уникальное. В данной статье описано создание игры Arkanoid в среде Microsoft Visual Studio на языке программирования C#.

1.2 Обзор исследований

В статье Н. А. Базеевой и Д. С. Лебедева описано исследование о начале игровой индустрии, а также рассмотрены особенности языков программирования для разработки игр [1]. R. Bourbia и др. в своей научной работе рассказывают об игре, цель которой улучшение навыков сборки компьютера студентами в виртуальной лаборатории [2]. Об основных вопросах разработки однопользовательских двумерных игр на языке JavaScript написано в учебном пособии С. А. Беляева [3]. Н. В. Бужинская и Д. Ю. Тазабеков в статье рассматривают процесс разработки компьютерной игры «Winduza» на игровом движке Unity с использованием языка программирования C# [4]. Обширный обзор литературы по подходу, ориентированному на разработку видеоигр представлен в исследовании у Н. Engström и др. [5]. И. С. Голубь и Д. В. Лучанинов описывают алгоритм создания развивающей игры «Парные картинки» с помощью языка программирования JavaScript [6]. В статье Э. Э. Сейдаметова, А. Э. Шабанова представлены наиболее популярные средства разработки игр и проанализированы их характерные особенности [7].

1.3 Цель исследования

Целью данной статьи является создание игры Arkanoid в среде Microsoft Visual Studio на языке программирования C#.

2. Методы исследования

В данной статье будет рассмотрен процесс создания игры Arkanoid в среде Microsoft Visual Studio (MVS) на языке программирования C# [8].

Arkanoid — видеоигра для игровых автоматов, разработанная компанией Taito в 1986 году. Игра основана на играх серии Breakout фирмы Atari. Именно её название стало нарицательным для класса подобных игр [9].

Смысл игры заключается в наборе очков, управлении платформой, которую можно двигать вправо или влево и разрушении ячеек мячиком, которому надо не дать попасть мимо платформы. В процессе игры строки с ячейками будут опускаться вниз, заставляя игрока разбивать их продуманно, чтобы не проиграть, потому что если строки ячеек достигнут низа игрового поля, то игра закончится и начнется заново.

Создаём проект на C# в Microsoft Visual Studio и добавляем форму, которая будет являться игровым полем. Для начала нарисуем платформу, мячик и колонки ячеек с которыми будет взаимодействовать пользователь (Рис. 1-2).

```
public void DrawMap(Graphics g)
{
    for (int i = 0; i < mapHeight; i++)
    {
        for (int j = 0; j < mapWidth; j++)
        {
            if (map[i, j] == 9)
            {
                g.DrawImage(arcanoIdSet, new Rectangle(new Point(j*20,i*20), new Size(60, 20)), 398, 17, 150, 50,GraphicsUnit.Pixel);
            }
            if (map[i, j] == 8)
            {
                g.DrawImage(arcanoIdSet, new Rectangle(new Point(j * 20, i * 20), new Size(20, 20)), 886, 548, 73, 73, GraphicsUnit.Pixel);
            }
            if (map[i, j] == 1)
            {
                g.DrawImage(arcanoIdSet, new Rectangle(new Point(j * 20, i * 20), new Size(40, 20)), 20, 16, 170, 59, GraphicsUnit.Pixel);
            }
            if (map[i, j] == 2)
            {
                g.DrawImage(arcanoIdSet, new Rectangle(new Point(j * 20, i * 20), new Size(40, 20)), 20, 16+77*(map[i,j]-1), 170, 59, GraphicsUnit.Pixel);
            }
            if (map[i, j] == 3)
            {
                g.DrawImage(arcanoIdSet, new Rectangle(new Point(j * 20, i * 20), new Size(40, 20)), 20, 16 + 77 * (map[i, j] - 1), 170, 59, GraphicsUnit.Pixel);
            }
            if (map[i, j] == 4)
            {
                g.DrawImage(arcanoIdSet, new Rectangle(new Point(j * 20, i * 20), new Size(40, 20)), 20, 16 + 77 * (map[i, j] - 1), 170, 59, GraphicsUnit.Pixel);
            }
            if (map[i, j] == 5)
            {
                g.DrawImage(arcanoIdSet, new Rectangle(new Point(j * 20, i * 20), new Size(40, 20)), 20, 16 + 77 * (map[i, j] - 1), 170, 59, GraphicsUnit.Pixel);
            }
        }
    }
}
```

Рисунок 1 – Код отрисовки объектов игры

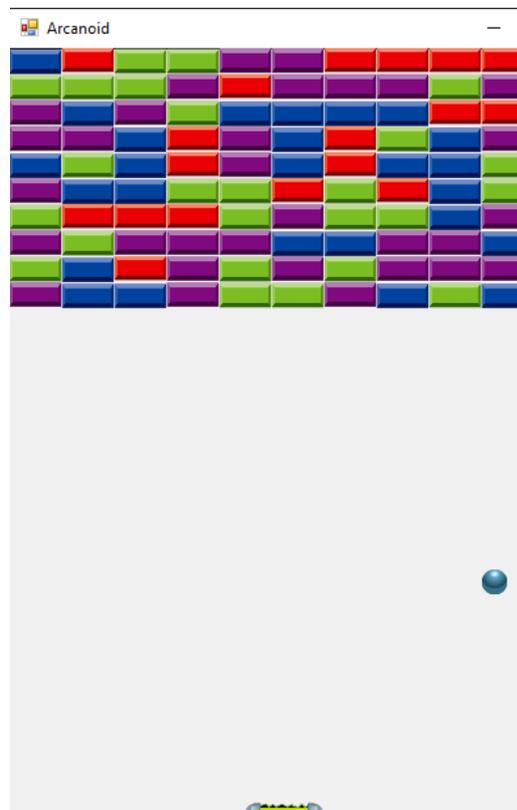


Рисунок 2 – Отрисовка объектов игры

Далее напишем код, отвечающий за отображения набранных игроком очков, а также за движение платформы и шарика (Рис. 3-4).

```
public void Init()
{
    this.Width = (mapWidth+5) * 20;
    this.Height = (mapHeight + 2) * 20;

    arcanoidSet = new Bitmap("C:\\Users\\admin\\Desktop\\arcanoid.png");
    timer1.Interval = 40;

    score = 0;

    scoreLabel.Text = "Score: "+score;

    for (int i = 0; i < mapHeight; i++)
    {
        for (int j = 0; j < mapWidth; j++)
        {
            map[i, j] = 0;
        }
    }

    platformX = (mapWidth-1)/2;
    platformY = mapHeight-1;

    map[platformY, platformX] = 9;
    map[platformY, platformX + 1] = 99;
    map[platformY, platformX + 2] = 999;

    ballY = platformY - 1;
    ballX = platformX + 1;

    map[ballY, ballX] = 8;

    dirX = 1;
    dirY = -1;

    GeneratePlatforms();

    timer1.Start();
}
```

Рисунок 3 – Код, отвечающий за расположение и счетчик очков

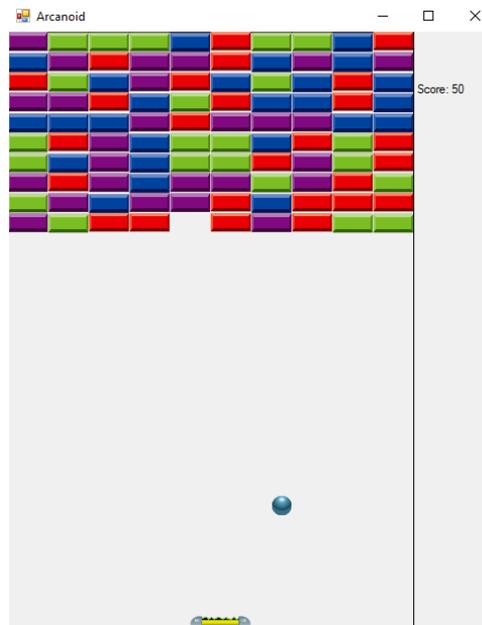


Рисунок 4 – Отображение счетчика очков на игровом поле

Следующим шагом будет написание обработчика нажатия клавиш влево и вправо для взаимодействия игрока с платформой через клавиатуру (Рис. 5).

```
private void inputCheck(object sender, KeyEventArgs e)
{
    map[platformY, platformX] = 0;
    map[platformY, platformX + 1] = 0;
    map[platformY, platformX + 2] = 0;
    switch (e.KeyCode)
    {
        case Keys.Right:
            if(platformX+1<mapWidth-1)
                platformX++;
            break;
        case Keys.Left:
            if(platformX>0)
                platformX--;
            break;
    }
    map[platformY, platformX] = 9;
    map[platformY, platformX + 1] = 99;
    map[platformY, platformX + 2] = 999;
}
```

Рисунок 5 – Обработчик нажатия клавиш

Далее необходимо написать код, который будет проверять шарик на коллизию с другими объектами (Рис. 6).

```
public bool IsCollide()
{
    bool isColliding = false;
    if(ballX+dirX>mapWidth-1 || ballX + dirX < 0)
    {
        dirX *= -1;
        isColliding = true;
    }
    if (ballY + dirY < 0)
    {
        dirY *= -1;
        isColliding = true;
    }

    if (map[ballY + dirY, ballX] != 0)
    {
        bool addScore = false;
        isColliding = true;

        if(map[ballY + dirY, ballX] > 10 && map[ballY + dirY, ballX] < 99)
        {
            map[ballY + dirY, ballX] = 0;
            map[ballY + dirY, ballX-1] = 0;
            addScore = true;
        }else if (map[ballY + dirY, ballX] < 9)
        {
            map[ballY + dirY, ballX] = 0;
            map[ballY + dirY, ballX + 1] = 0;
            addScore = true;
        }
        if (addScore)
        {
            score += 50;
            if (score % 200 == 0 && score > 0)
            {
                AddLine();
            }
        }
        dirY *= -1;
    }
    if (map[ballY, ballX+dirX] != 0)
    {
        bool addScore = false;
        isColliding = true;

        if (map[ballY, ballX + dirX] > 10 && map[ballY + dirY, ballX]<99)
        {
            map[ballY, ballX + dirX] = 0;
            map[ballY, ballX+dirX-1] = 0;
            addScore = true;
        }
        else if (map[ballY, ballX + dirX] < 9)
        {
            map[ballY, ballX + dirX] = 0;
            map[ballY, ballX + dirX+1] = 0;
            addScore = true;
        }
        if (addScore)
        {
            score += 50;
            if (score % 200 == 0 && score > 0)
            {
                AddLine();
            }
        }
        dirX *= -1;
    }
    scoreLabel.Text = "Score: " + score;
    return isColliding;
}
```

Рисунок 6 – Вид готовой игры

Теперь можно написать функцию обновления главного окна при различных ситуациях: столкновении шарика с различными объектами и проигрыше (Рис. 7).

```
private void update(object sender, EventArgs e)
{
    if(ballY + dirY > mapHeight - 1){
        Init();
    }

    map[ballY, ballX] = 0;
    if (!IsCollide())
        ballX += dirX;
    if (!IsCollide())
        ballY += dirY;
    map[ballY, ballX] = 8;

    map[platformY, platformX] = 9;
    map[platformY, platformX + 1] = 99;
    map[platformY, platformX + 2] = 999;

    Invalidate();
}
```

Рисунок 7 – Код обновления окна игры

Последним шагом будет добавление кода, отвечающего за добавление новой линии препятствий (Рис. 8).

```
public void AddLine()
{
    for (int i = mapHeight-2; i > 0; i--)
    {
        for (int j = 0; j < mapWidth; j += 2)
        {
            map[i, j] = map[i-1, j];
        }
    }
    Random r = new Random();
    for (int j = 0; j < mapWidth; j += 2)
    {
        int currPlatform = r.Next(1, 5);
        map[0, j] = currPlatform;
        map[0, j + 1] = currPlatform + currPlatform * 10;
    }
}
```

Рисунок 8 – Код добавления новой линии препятствий

На выходе игра выглядит следующим образом (Рис. 9).

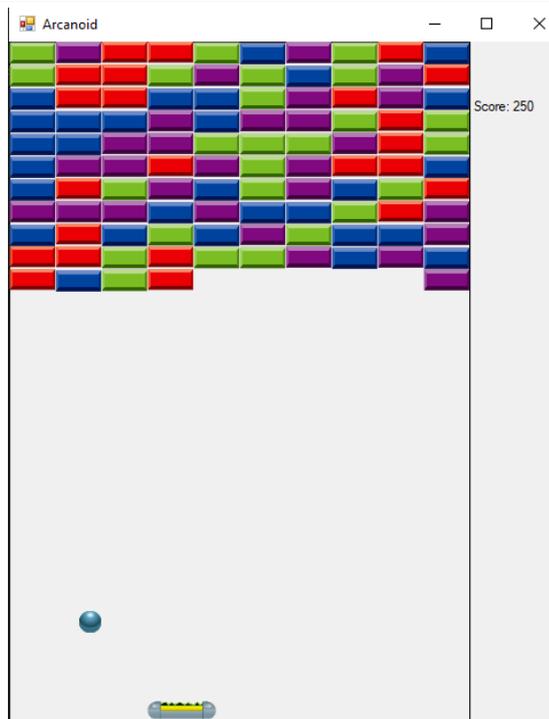


Рисунок 9 – Вид готовой игры

3. Выводы

Таким образом, была написана игра Arkanoid в среде Microsoft Visual Studio на языке программирования C#. Данная статья будет полезна начинающим программистам для понимания механик игр подобного жанра, а также изучения языка программирования C#.

Библиографический список

1. Базеева Н. А., Лебедев Д. С. Языки программирования для создания игр //E-Scio. 2019. №. 4 (31).
2. Bourbia R. et al. Development of serious game to improve computer assembly skills //Procedia-Social and Behavioral Sciences. 2014. Т. 141. С. 96-100.
3. Беляев С. А. Разработка игр на языке JavaScript. 2016.
4. Бужинская Н. В., Тазабеков Д. Ю. Разработка компьютерной обучающей игры "WINDUZA" //Заметки ученого. 2019. №. 2. С. 16-21.
5. Engström H. et al. Game development from a software and creative product perspective: A quantitative literature review approach //Entertainment Computing. 2018. Т. 27. С. 10-22.
6. Голубь И. С., Лучанинов Д. В. Разработка обучающей игры «Парные картинки» с помощью JavaScript //Постулат. 2016. №. 12.
7. Сейдаметов Э. Э., Шабанов А. Э. Современные средства разработки игр //Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2020. №. 1. С. 54-62.
8. Microsoft Visual Studio URL: <https://visualstudio.microsoft.com/ru/downloads/>
9. Arkanoid URL: <https://ru.wikipedia.org/wiki/Arkanoid>