

## Разработка игры «Ударь крота» на C#

*Ульянов Егор Андреевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### **Аннотация**

В данной статье рассматривается и описывается разработка игры «Ударь крота» на языке программирования C#. Практическим результатом является разработанная игра.

**Ключевые слова:** Игра «Ударь крота», C#, Visual Studio

## Developing a «Whac-A-Mole» game in C #

*Ulianov Egor Andreevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### **Abstract**

This article discusses and describes the development of the game «Whac-A-Mole» in the C # programming language. The bottom line is a developed game.

**Keywords:** Whac-A-Mole, C#, Visual Studio

«Whac-A-Mole» - популярная аркадная игра. Изобретена в 1975 году Кадзуо Ямада, основана на карандашных набросках дизайнера 1974 года. Машина «Whac-A-Mole» состоит из шкафа на уровне пояса, с игровой зоной и дисплеем очков, а также большого мягкого черного молотка, в более поздних моделях есть экран с лучшим результатом дня. Пять отверстий в верхней части игровой площадки заполнены маленькими игрушечными кротами, которые выскакивают наугад. Очки начисляются путем ударов каждому кроту по мере появления. Чем быстрее реакция и удар, тем больше баллов. Если игрок не ударит крота в течение определенного времени или ударит слабо, крот вернется обратно в нору и не принесет очков. Игровой процесс начинается достаточно медленно, чтобы большинство людей могло поразить всех появляющихся кротов, постепенно скорость увеличивается, и увеличивается одновременное количество кротов на игровом поле. По истечении назначенного времени игра заканчивается. Окончательный счет основан на количестве очков. В дополнение к описанной выше однопользовательской игре существует многопользовательская игра, которую чаще всего можно найти в парках развлечений. В этой версии есть большое количество отдельных машин «Whac-A-Mole», связанных друг с другом, и цель состоит в том, чтобы первым набрать определенное количество очков. В большинстве версий удар по кроту приносит десять

очков, и побеждает тот игрок, который первым наберет 150 очков (то есть 15 кротов). Со временем параметры игрового процесса стали более настраиваемыми, что позволяет оператору или владельцу выборочно изменять победный счет, очки попаданий, скорость появления кротов, а также время игры.

Целью данной статьи является создание игры «Ударь крота» в среде разработки «Visual Studio» на языке программирования C#.

В своей работе Н. Н.Додобоев, О.И.Кукарцева, Я.А.Тынченко рассмотрели вопросы появления различных языков программирования (в частности C#), определения особенностей этих языков, а также составления основных видов и классификаций языков программирования[1]. З.С.Магомадова рассмотрела языки программирования высокого уровня, особенности, недостатки и сложности в изучении, а также описала несколько легких алгоритмов [2]. В своей работе И.Ю.Просвирнина создала приложение «Морской бой», обладающее игровым искусственным интеллектом, в котором предусмотрен режим игры «Игрок против компьютера» [3]. В статье Н.А.Базеевой и Д.С.Лебедева описано начало игровой индустрии, а также рассмотрены особенности языков программирования для разработки игр [4].

Создаем проект «Windows Forms App» и называем его. Далее необходимо для проекта нарисовать ресурсы будущей игры, а именно изображение крота, изображение ударенного крота и игровое поле см. рисунок 1.

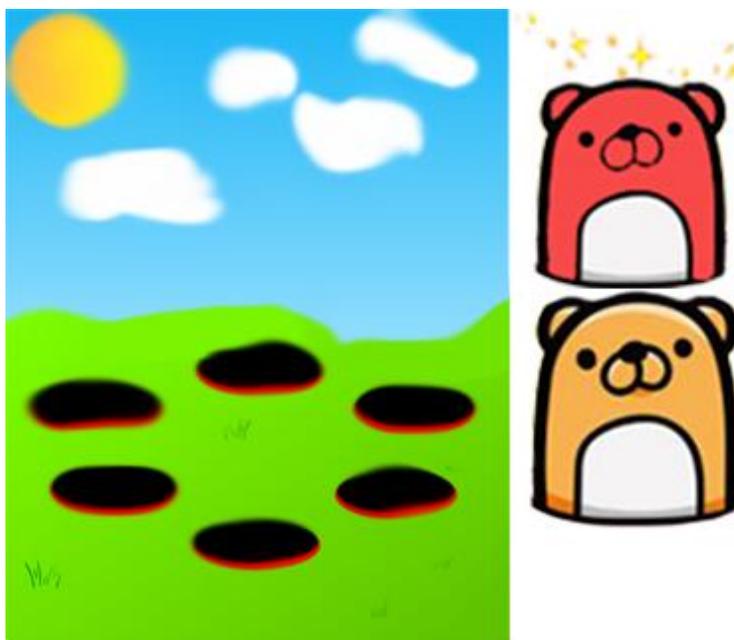


Рис. 1. Ресурсы игры

Далее необходимо добавить ресурсы в проект. Для этого кликаем два раза по файлу «resources.resx», ждем добавить ресурс, в выпадающем списке выбираем добавить существующий файл и добавляем подготовленные файлы см. рисунок 2.

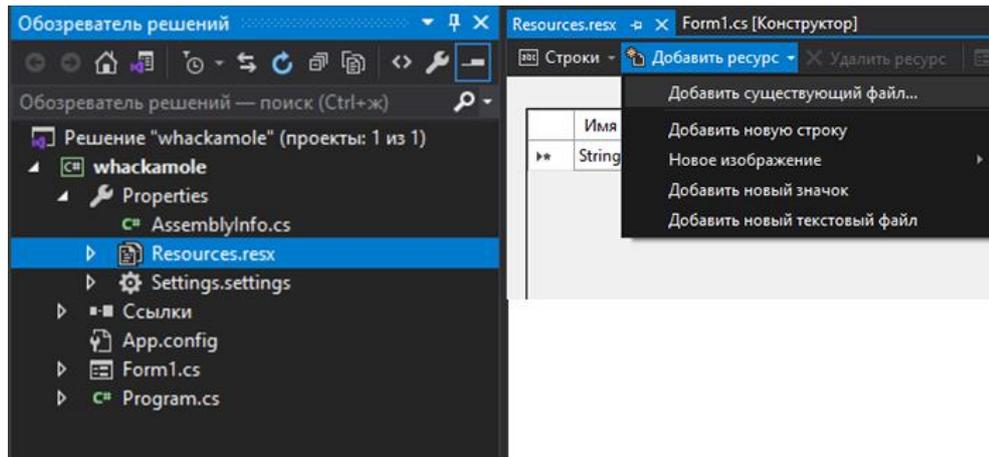


Рис. 2. Добавление ресурсов

Возвращаемся к форме, меняем значение в пункте «Text» на «Whack a Mole», а размер ставим 607, 586. Далее находим пункт «Background», нажимаем обзор, ставим изображение «ground» и меняем параметр фонового изображения на «Stretch» см. рисунок 3 и 4.

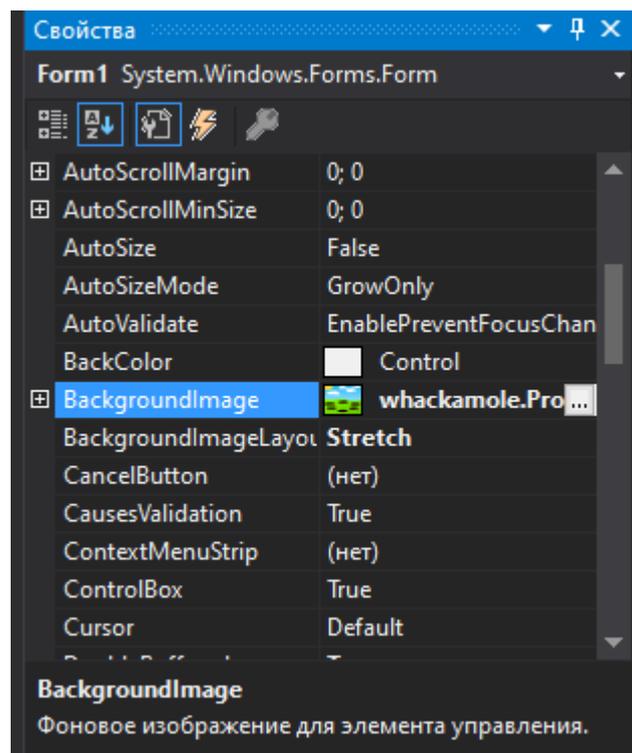


Рис. 3. Настройка заднего фона

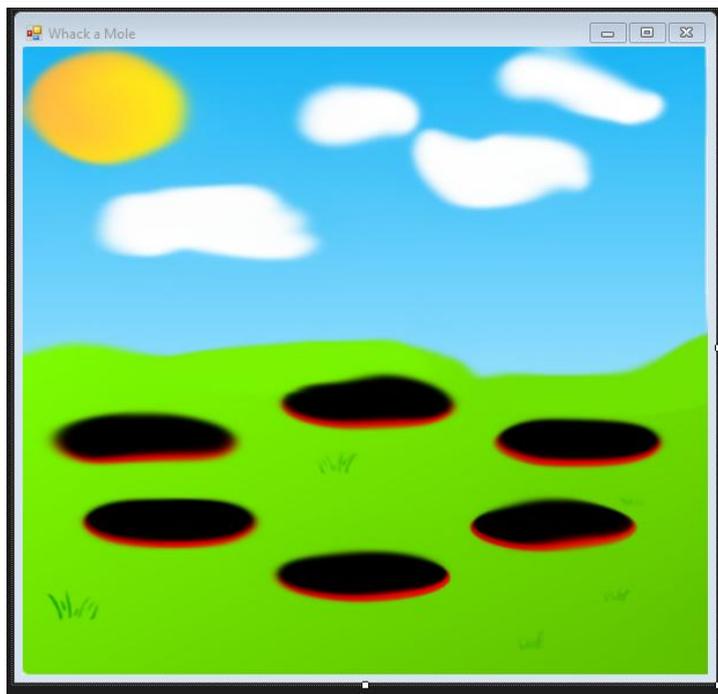


Рис. 4. Результат

Теперь нужно добавить метку, с текстом «Hit:», указывающая количество попаданий игрока. Настроим шрифт и размер текста, и в поле «name» ставим значение «lblHit» см. рисунок 5.

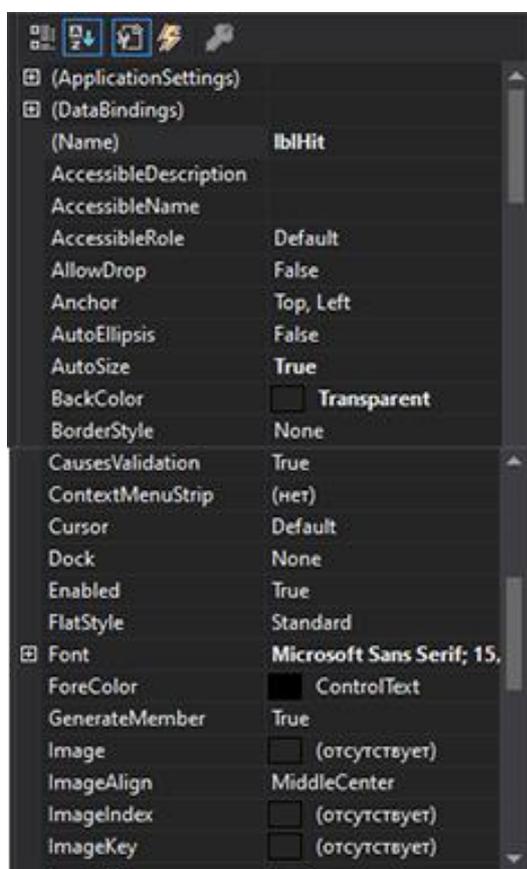


Рис. 5. Метка попаданий

Дублируем метку, в поле текст пишем «Miss:», а в поле «name» - «lblMiss». Данная метка будет показывать количество промахов игрока см. рисунок 6 и 7.

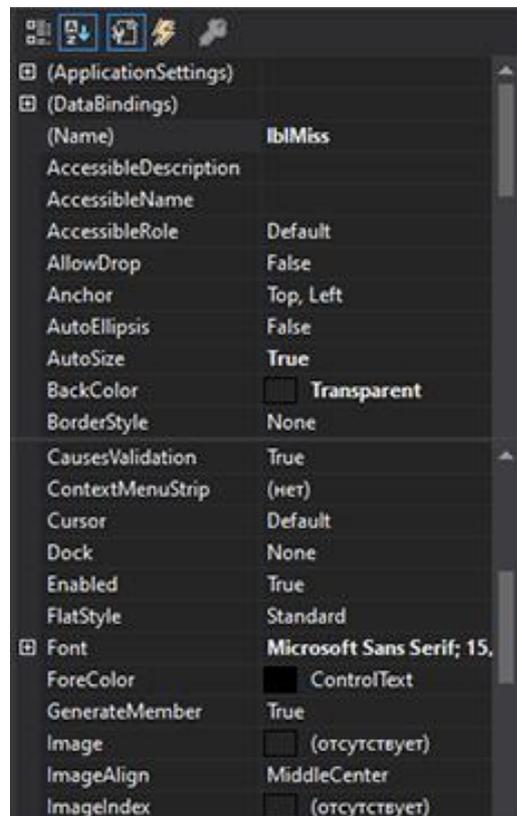


Рис. 6. Метка промахов

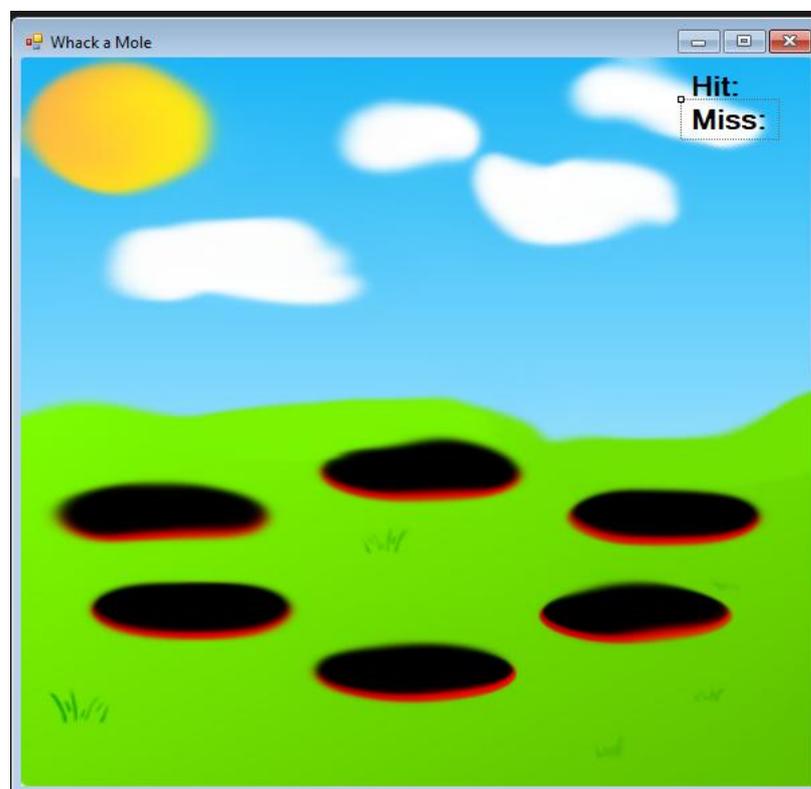


Рис. 7. Результат

Далее, в окне панель элементов, кликаем два раза по «PictureBox», и настраиваем объект: «Name - Mole», «Back Color – Transparent», «Image – alive», размер - 90, 100. Также добавим таймер в форму, с такими настройками «Enabled – True», «Interval – 2000». Для игры понадобится один крот. Так как, логика игры заключается в том, что крот будет физически перемещаться над каждой норой, по заранее записанным значениям X и Y, перемещаться между этими ячейками будет случайным образом см. рисунок 8 и 9.

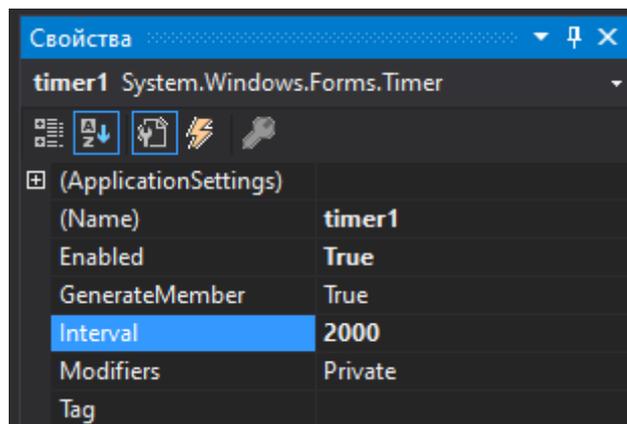
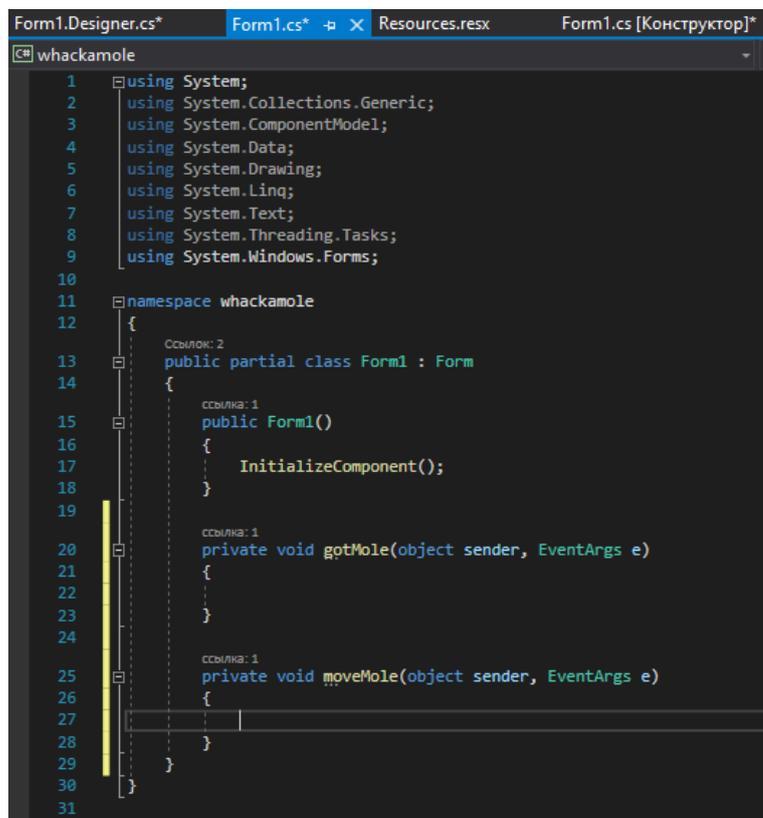


Рис. 9. Добавление таймера



Рис. 8. Размещение крота в стандартном положении

Кликаем по кроту, и переходим в настройку события. В поле «Tick» вводим значение «gotMole». С таймером делаем тоже самое, но значение будет «moveMole». Это создаст в коде, два события см. рисунок 10.



```
Form1.Designer.cs* Form1.cs* -p x Resources.resx Form1.cs [Конструктор]*
whackamole
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace whackamole
12 {
13     Ссылка: 2
14     public partial class Form1 : Form
15     {
16         Ссылка: 1
17         public Form1()
18         {
19             InitializeComponent();
20         }
21         Ссылка: 1
22         private void gotMole(object sender, EventArgs e)
23         {
24         }
25         Ссылка: 1
26         private void moveMole(object sender, EventArgs e)
27         {
28         }
29     }
30 }
31
```

Рис. 10. Создание событий

Приступаем к написанию кода, для начала создадим несколько переменных и напишем функцию перемещения крота см. рисунок 11.

```
private void moveMole(object sender, EventArgs e)
{
    isHit = false;
    Mole.Enabled = true;
    Mole.Image = Properties.Resources.alive;
    Mole.BackColor = System.Drawing.Color.Transparent;

    locationNum = rnd.Next(1, 7);

    switch (locationNum)
    {
        case 1:
            Mole.Left = 434;
            Mole.Top = 249;
            break;

        case 2:
            Mole.Left = 257;
            Mole.Top = 211;
            break;

        case 3:
            Mole.Left = 58;
            Mole.Top = 240;
            break;

        case 4:
            Mole.Left = 85;
            Mole.Top = 318;
            break;

        case 5:
            Mole.Left = 254;
            Mole.Top = 364;
            break;

        case 6:
            Mole.Left = 416;
            Mole.Top = 323;
            break;

        default:
            break;
    }
}
```

Рис. 11. Функция перемещения

Далее напишем событие попадания по кроту см. рисунок 12.

```
ссылка: 1
private void gotMole(object sender, EventArgs e)
{
    score++;

    Mole.Image = Properties.Resources.dead;

    isHit = true;

    Mole.Enabled = false;
}
```

Рис. 12. Программирование события попадания

И наконец, напишем счетчик попаданий и промахов, а также правило победы или поражения см. рисунок 13.

```
lblHit.Text = "Hit: " + score;  
lblMiss.Text = "Miss: " + misses;  
  
if (isHit == false)  
{  
    misses++;  
}  
  
if (score > 15)  
{  
    timer1.Stop();  
    Mole.Enabled = false;  
    MessageBox.Show("You Win");  
}  
else if (misses > 10)  
{  
    timer1.Stop();  
    Mole.Enabled = false;  
    MessageBox.Show("You Lose");  
}
```

Рис. 13. Программирование

Осталось только проверить игру. Ждем кнопку «Play» см. рисунок 14-18.

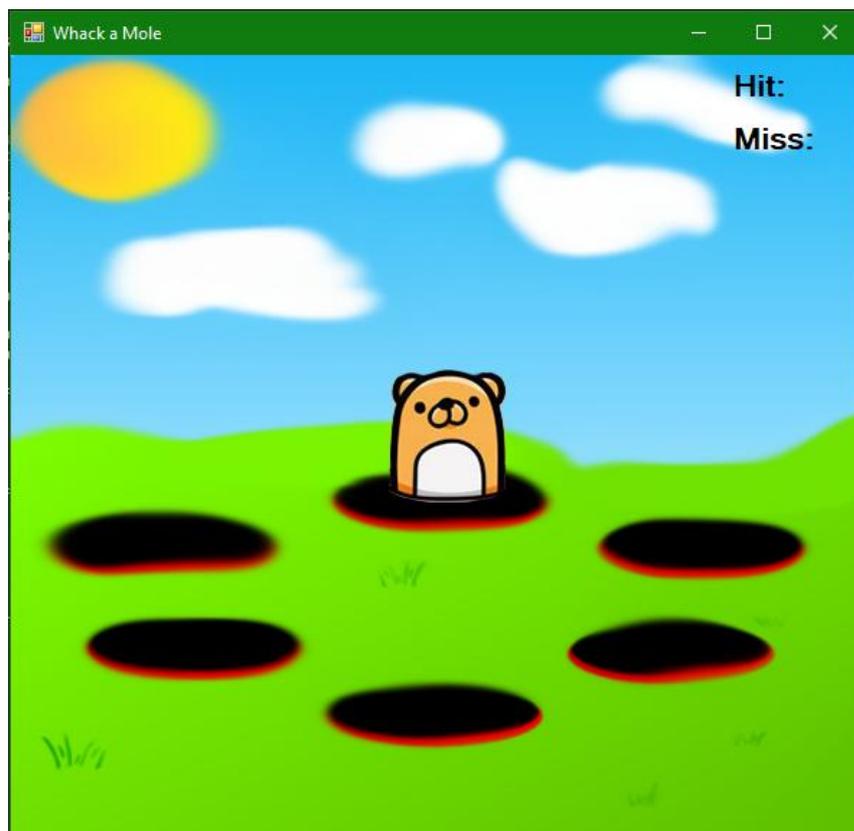


Рис. 14. Начальный экран

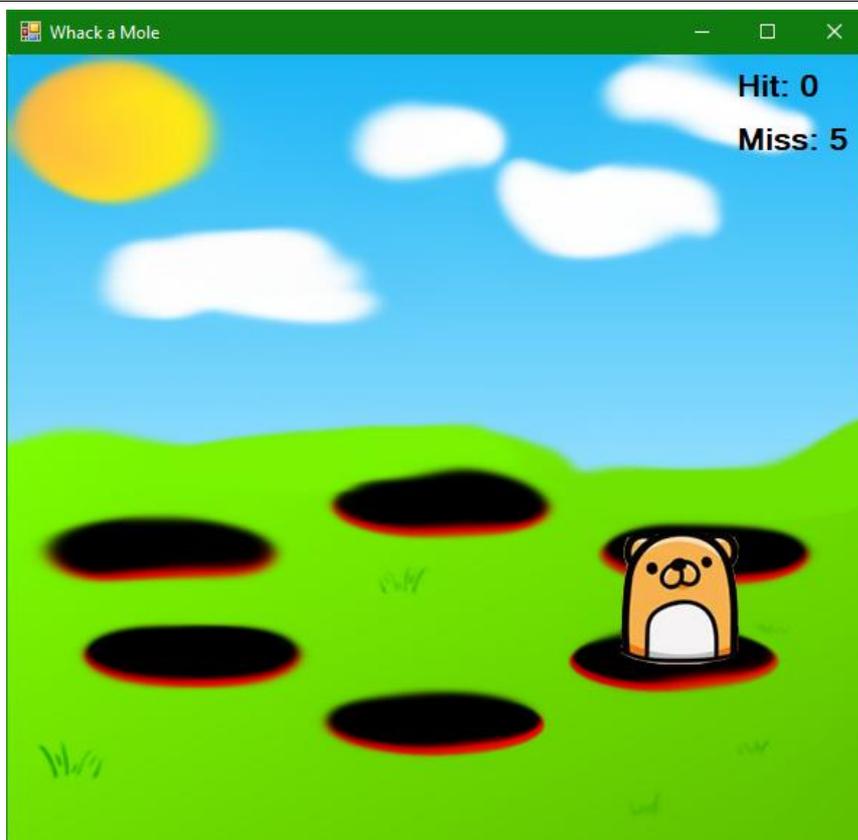


Рис. 15. Работа кода перемещения и промахов

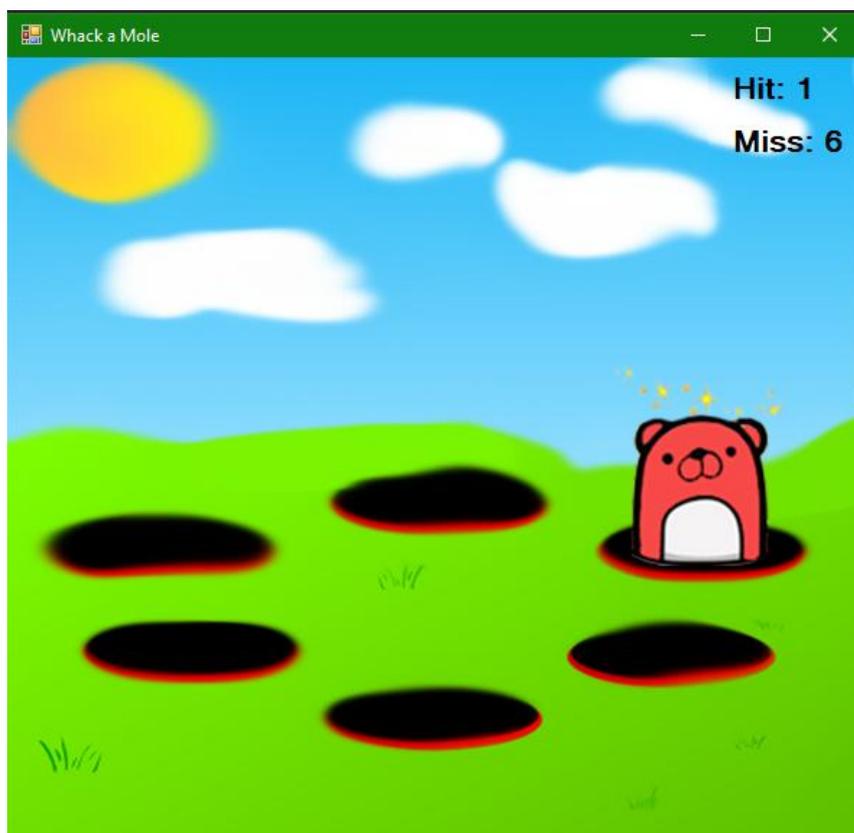


Рис. 16. Работа кода попадания

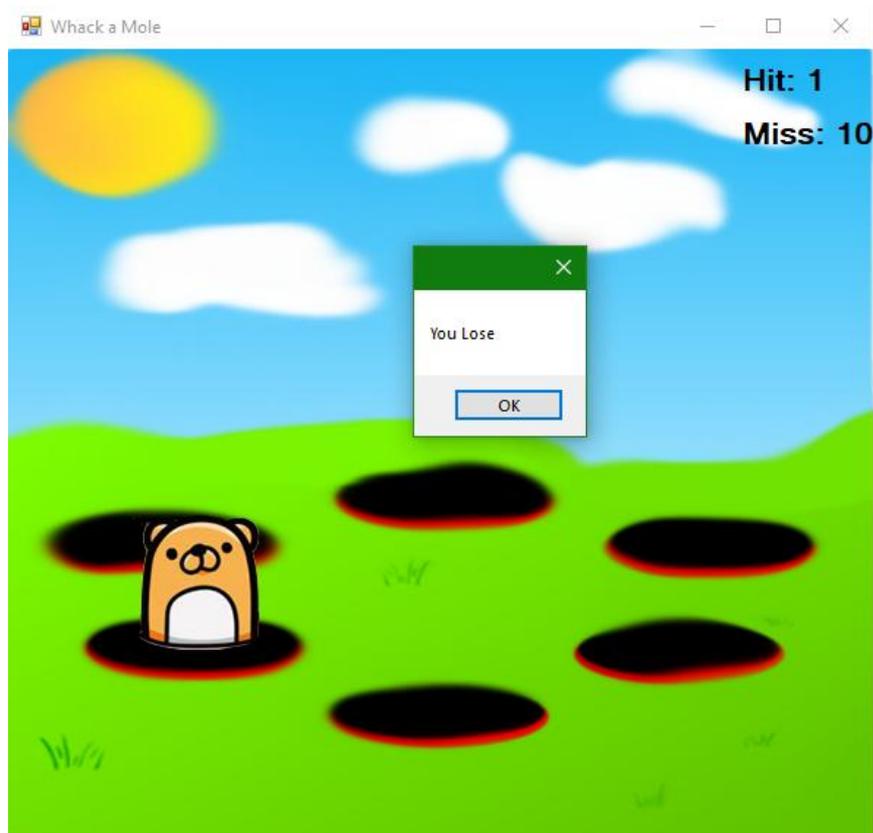


Рис. 17. Поражение игрока

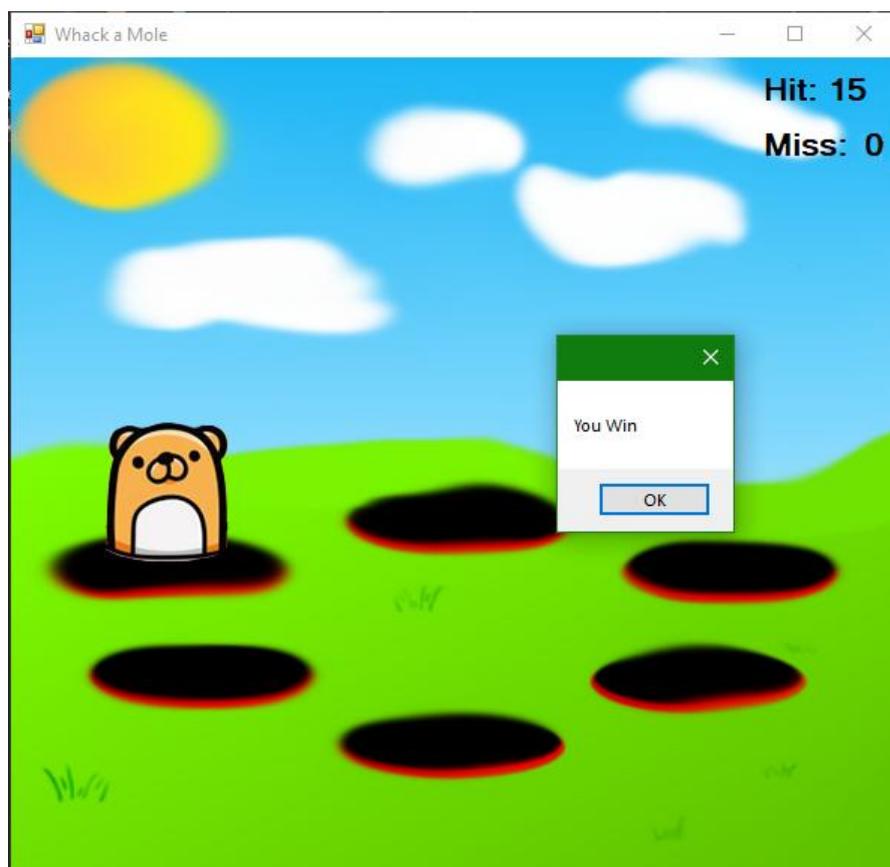


Рис. 18. Победа игрока

Таким образом, была написана игра «Ударь крота» в среде разработки Visual Studio на языке программирования C#.

### **Библиографический список**

1. Додобоев Н. Н., Кукарцева О. И., Тынченко Я. А. Современные языки программирования // Современные технологии: актуальные вопросы, достижения и инновации. 2014. №5. С. 81-85.
2. Магомадова З. С. Языки программирования высокого уровня // Разработка и применение наукоёмких технологий в эпоху глобальных трансформаций. 2020. №8. С. 94-96.
3. Просвирнина И. Ю., Егунова А. И., Аббакумов А. А. Среда разработки Microsoft Visual Studio на примере создания игры "Морской бой" //Интеграционные процессы в науке в современных условиях. 2017. С. 123-125.
4. Базеева Н. А., Лебедев Д. С. Языки программирования для создания игр //E-Scio. 2019. №4. С. 31-39.