

Создание простой игры «Угадай слово» на C#

Ульянов Егор Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается и описывается разработка простой игры «Угадай слово» на языке программирования C#. Практическим результатом является разработанная и протестированная игра.

Ключевые слова: Угадай слово, игра, WPF, C#, Visual Studio

Creating a simple "Guess the Word" game in C#

Ulianov Egor Andreevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses and describes the development of a simple game "Guess the word" in the C# programming language. The practical result is a developed and tested game.

Keywords: Guess the Word, game, WPF, C#, Visual Studio

Самым популярным видом досуга многих людей на сегодняшний момент являются компьютерные игры. Виртуальные миры открывают нам огромные возможности: позволяют отыгрывать различные роли, играя за главного героя, многие игры способствуют развитию интеллекта, внимания, реакции, пространственной ориентации и логического мышления. Разработкой видеоигр может заниматься как один человек, так и команда. Создание игры - это продолжительный и трудоёмкий процесс, состоящий из самых разнообразных этапов, включающий в себя как технические, так и творческие моменты.

Целью данной статьи является создание простого игры 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 статье Н. А. Базеевой и Д. С. Лебедева описано

начало игровой индустрии, а также рассмотрены особенности языков программирования для разработки игр [3]. В своей работе И. Ю. Просвирнина создала приложение «Морской бой», обладающее игровым искусственным интеллектом, в котором предусмотрен режим игры «Игрок против компьютера» [4]. В своей работе Р.Ф. Гайнуллин, В.А. Захаров, Е.А. Аксенова изучили инструмент для разработки двух- и трёхмерных игр – Unity 3D [5]. В.И. Вахрушев в работе представил подробную инструкцию по разработке игры «Расстановка мебели» на движке Unity с использованием языка программирования C# [6].

Создаем проект «Windows Forms App» и называем произвольным именем см. рисунок 1.

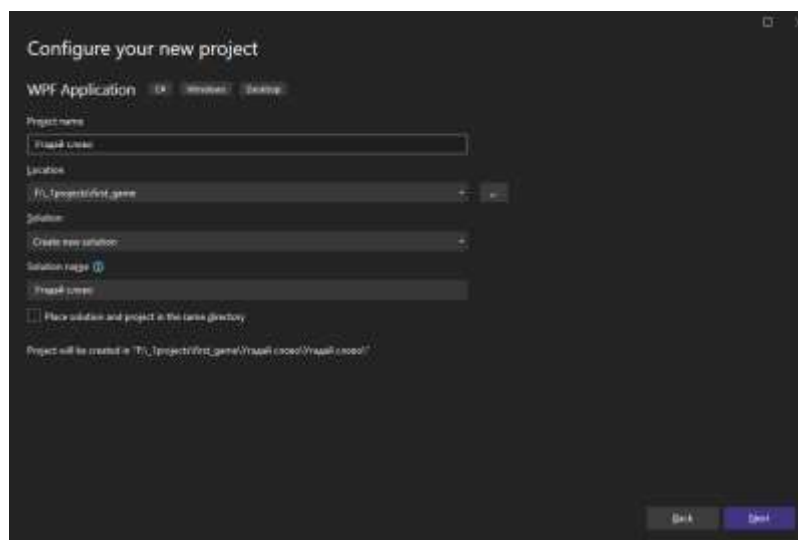


Рис. 1. Создание проекта

Далее необходимо создать холст пользовательского интерфейса и добавить нужные элементы, четыре этикетки: Объявление игры или название; этикетка, которая будет хранить загадываемое слово; этикетка которая, будет хранить счетчик попыток ввода слова; этикетка, которая будет хранить счетчик угаданных слов из общего числа слов, задаем различные цвета и параметры для элементов. Также для ввода пользователем слова понадобится элемент «TextBox» см. рисунке 2.

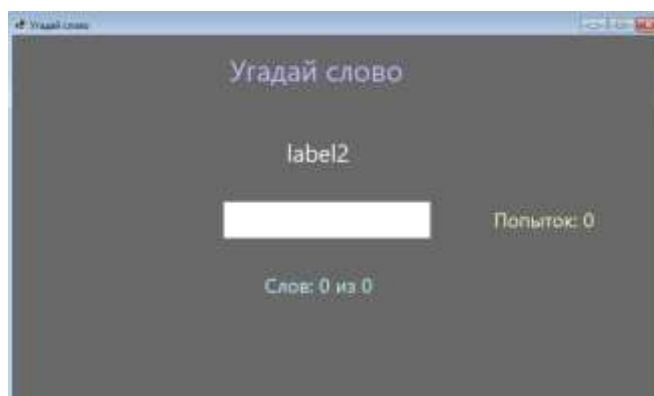


Рис. 2. Расположение элементов

Переходим к написанию кода, объявляем список слов «words» для хранения в памяти компьютера загаданных слов, а также строковую переменную «newText», и два счетчика «i» и «guessed». Создаем обработчик событий «KeyIsPressed», которые срабатывает при нажатии клавиши «Enter». Данный обработчик проверяет, есть ли введенное слово в списке «words». Создаем условные выражение, которое проверяет совпадает ли введенное слово с заданным, если да то, выводится сообщение «Правильно», индекс «i» увеличивается, и отображается новое слово. Также обновляется количество попыток и текст этикетки. Если больше нет слов в списке, на метке отображается "Победа" см. рисунок 3.

```
public partial class Form1 : Form
{
    List<string> words = new List<string>();
    string newText;
    int i = 0;
    int guessed = 0;

    //reference
    public Form1()
    {
        InitializeComponent();
        Setup();
    }

    //reference
    private void KeyIsPressed(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == (char)Keys.Enter)
        {
            if (words[i].ToLower() == textBox1.Text.ToLower())
            {
                if (i < words.Count - 1)
                {
                    MessageBox.Show("Правильно!", "Ответ: ");
                    textBox1.Text = "";
                    i += 1;
                    newText = Scramble(words[i]);
                    lblWord.Text = newText;
                    lblInfo.Text = "Слова: " + (i + 1) + " из " + words.Count;
                    guessed = 0;
                    lblGuessed.Text = "Попыток: " + guessed + " шт.";
                }
                else
                {
                    lblWord.Text = "Победа";
                    return;
                }
            }
        }
    }
}
```

Рис. 3. Метод «KeyIsPressed»

Создаем функцию «Setup()» выполняющую начальную настройку игры, метод загружает список слов из файла "words.txt", перемешивает первое слово и отображает его в текстовой метке «lblWord», также обновляется информация о текущем слове в текстовой метке «lblInfo». Функция «Scramble()» будет принимать строку текста и возвращает перемешанную версию. Метод использует «OrderBy» для перемешивания символов строки по случайному GUID. Создаем еще один обработчик событий «KeyPress» который будет отлавливать ввод пользователя и, проверять, является ли введенная буква частью загаданного слова. Если да, то буквы отображается в текстовой метке «lblWord». Если нет, то увеличивается счетчик попыток и обновляется соответствующая текстовая метка см. рисунок 4.

```
    }
    else
    {
        guessed += 1;
        lblGussed.Text = "Попыток: " + guessed + " шт.";
    }
    e.Handled = true;
}

1 reference
private void Setup()
{
    words = File.ReadLines("words.txt").ToList();
    newText = Scramble(words[1]);
    lblWord.Text = newText;
    lblInfo.Text = "Words: " + (1 + 1) + " of " + words.Count;
}

2 references
private string Scramble(string text)
{
    return new string(text.ToCharArray().OrderBy(x => Guid.NewGuid()).ToArray());
}
}
```

Рис. 4. Методы «Setup()» и «Scramble()»

Далее создаем в директории программы файл с названием, заданным в коде - «words.txt», который будет хранить список загаданных слов. И загадываем слова см. рисунок 5.

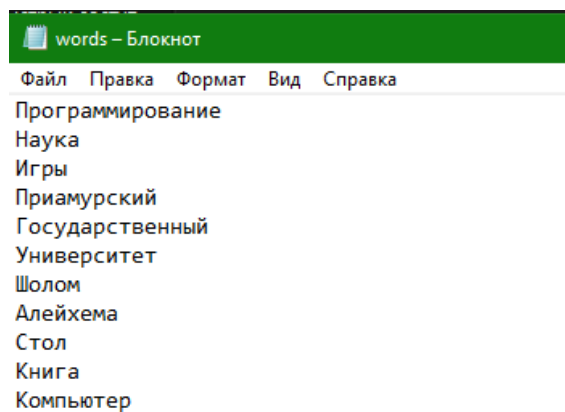


Рис. 5. Файл с загаданными словами

Приступаем к тестированию программы, нажимаем клавишу «f5» см. рисунок 6-10.

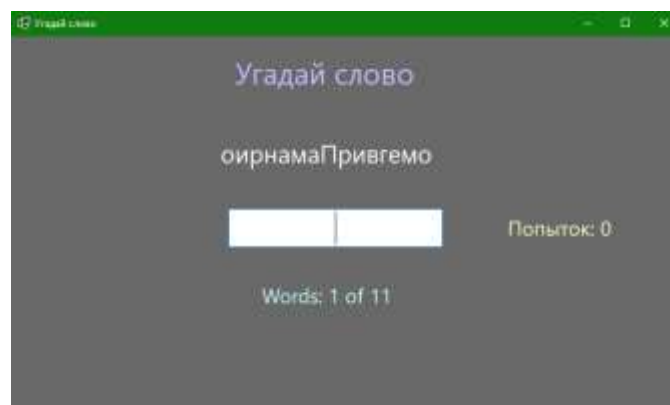


Рис. 6. Запуск программы

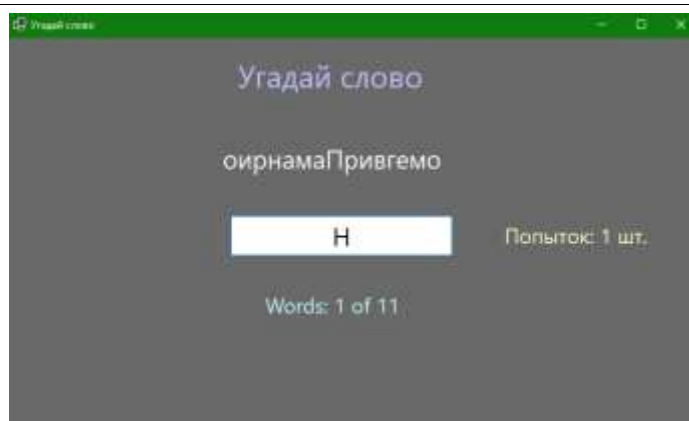


Рис. 7. Работа счетчиков

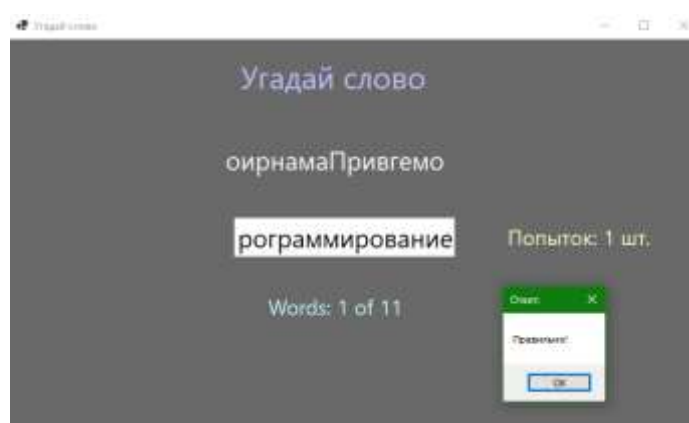


Рис. 8. Обработка правильного ввода слова

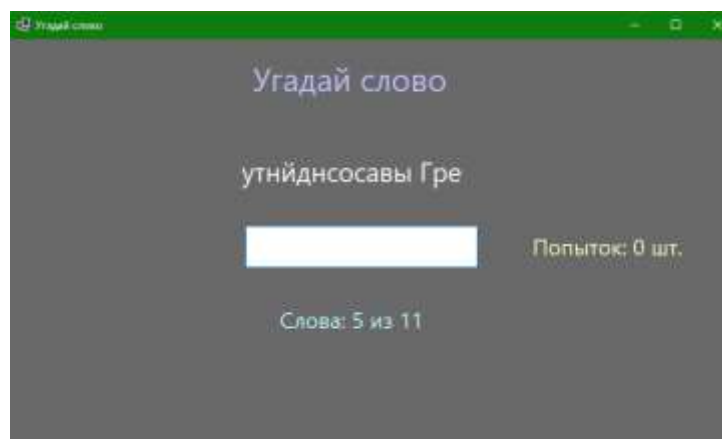


Рис. 9. Продолжение игры

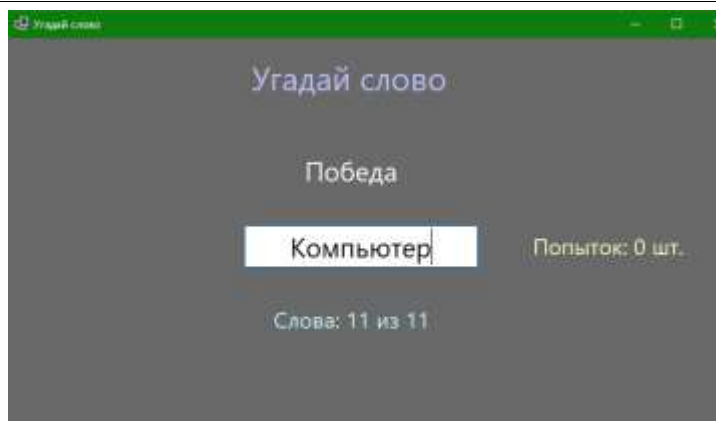


Рис. 10. Сообщение о победе

Таким образом, была написана игра «Угадай слово» 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. Базеева Н. А., Лебедев Д. С. Языки программирования для создания игр //E-Scio. 2019. №4. С. 31-39.
4. 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.
5. Гайнуллин Р.Ф., Захаров В.А., Аксенова Е.А. Создание 2d игры на Unity 3D 5.4 // Вестник современных исследований. 2018. №4. С. 78-82.
6. Вахрушев В.И. Реализация игры-головоломки "Расстановка мебели" с использованием unity и языка C#// Развитие современной науки: теоретические и прикладные аспекты. 2016. С.23-25.