

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования
«Гродненский государственный университет имени Янки Купалы»

Факультет математики и информатики
Кафедра современных технологий программирования

ЗАХАРОВА АЛИНА АЛЕКСАНДРОВНА

Игра «Однорукий бандит»

Курсовая работа
по дисциплине «Алгоритмизация и программирование»
студентки 2 курса специальности
1-26 03 01 «Управление информационными ресурсами»
дневной формы получения образования

Научный руководитель
Урбан Ольга Ивановна
старший преподаватель
кафедры современных
технологий программирования

Гродно 2021

РЕЗЮМЕ

Тема курсовой работы **Игра «Однорукий бандит»**

Работа содержит: 30 страниц, 6 рисунков, 5 использованных источников литературы, 1 таблица, 12 листингов.

Ключевые слова: игра, компьютерная игра, игра «Однорукий бандит», правила игры «Однорукий бандит», C#, объектно-ориентированное программирование, Windows Form, приложение, классы, методы.

Цель курсовой работы – разработка компьютерной игры «Однорукий бандит», основанной на генерации случайных чисел и правиле подбора комбинаций.

Для достижения указанной цели необходимо решить следующие задачи:

1. Изучить правила и особенности игрового автомата «Однорукий бандит».
2. Изучить необходимые библиотеки.
3. Реализовать генерацию изображений.
4. Разработать интерфейс приложения.
5. Запрограммировать игру с учётом всех требований.
6. Протестировать разработанную игру.

Объектом исследования выступает игра «Однорукий бандит».

Предметом исследования является процесс разработки компьютерной игры «Однорукий бандит».

Программа служит для проведения досуга детей и взрослых. При работе с данной у пользователя не должно возникать вопросов касемо правил и последовательности действий интерфейс должен быть прост и удобен в использовании.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ГЛАВА 1. АНАЛИЗ ИГРЫ «ОДНОРУКИЙ БАНДИТ»	6
1.1 Правила игры «Однорукий бандит»	6
1.2 Описание требований к приложению	7
1.3 Описание средств реализации, использованных классов и библиотек....	8
1.4 Вывод по главе 1	10
ГЛАВА 2. ОПИСАНИЕ РЕАЛИЗАЦИИ ПРИЛОЖЕНИЯ.....	11
2.1 Структура проекта	11
2.2 Описание форм C#.....	11
2.3 Функция кнопки «Спин»	22
2.4 Вывод по главе 2.....	23
ГЛАВА 3. РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРИЛОЖЕНИЯ	24
3.1 Руководство пользователю.....	24
3.2 Вывод по главе 3	28
ЗАКЛЮЧЕНИЕ	29
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	30

ВВЕДЕНИЕ

«Однорукий бандит» – это название механических слот-машин старого образца, тех, с которых началась история игровых автоматов. Их в конце XIX века придумал американец Чарльз Фей, который в 1896 году представил первый подобный автомат – с максимальным выигрышем в 50 центов при ставке 5 центов и символами, нанесёнными на три вращающихся барабана, приводящиеся в движение рычагом. Именно благодаря этому рычагу машины и получили своё название.

С течением времени классические слоты были усовершенствованы, появились видео слоты, в которых барабаны не вращаются механически, а изображаются на экране. В действие они приводятся уже не рычагом, а кнопкой, но название по-прежнему сохранилось за этими автоматами, считающимися классикой игровой индустрии [3].

Любая слот-машина – это компьютер с экраном и несколькими кнопками. В сердце этого компьютера чип, который решает, выиграете вы сейчас или нет. Чип сделан так, чтобы предсказать его поведение конкретно сейчас было невозможно, но чтобы в долгосрочной перспективе на большом количестве игр он давал статистически предсказуемый результат.

Все современные игры проектируются с использованием генераторов случайных чисел, которые постоянно генерируют последовательность моделируемых чисел, на скорости сотен или, возможно, тысяч в секунду. Случайное число визуализируется на привычные игровые картинки не только карт, но и героев из детских игр, комиксов или персонажей известных кинолент.

Игровые автоматы, как правило, запрограммированы на выплату выигрышей в определённом количестве от тех средств, которые отыграны игроками. Например, минимальная сумма выплаты в Неваде составляет 75%, в Нью-Джерси 83%, а в Миссисипи 80% [1].

В слот-индустрии есть показатель RTP – return to player. Это усредненный процент денег, которые игровой автомат теоретически возвращает игрокам, если смотреть на длинном отрезке времени. Казино зарабатывает на разнице между тем, сколько клиенты внесли, и тем, сколько забрали.

Минимальный RTP определён законом и должен составлять не менее 80 - 95%. Теоретически, если 1000 человек поставит по 1000 рублей, то автомат получит 1 млн рублей и вернёт в качестве выигрыша 900 тысяч.

На коротких промежутках возможны большие колебания выплат: сейчас тебе выплатили 1000 рублей, потом долго ничего не выплачивали,

а потом сразу 30 000 рублей. Но с течением времени средняя выплата приближается к RTP.

Длительность вращения барабанов и возможность их ручной остановки никак не влияет на результат. Результаты определяются при нажатии кнопки запуска.

В большинстве случаев идёт генерация не полностью случайных чисел, а псевдослучайных. Но для обычного игрока эта разница не будет заметна. Об этом позаботится разработчик, которому совсем не выгодно, чтобы кто-то смог определить используемые алгоритмы. Более того, каждый игровой автомат проходит сертификацию в тестовой лаборатории, которая также должна удостовериться, что никто не сможет догадаться, каким образом формируются случайные числа.

Однако есть как минимум один подтверждённый случай, когда способ генерации результатов слота был определён. Россиянин, известный как Алекс, купил себе реальные игровые автоматы и смог определить используемый алгоритм. Таким образом он всегда знал, когда будут выпадать призовые комбинации.

Комментируя случай, разработчики Novomatic признались, что игровые автоматы устроены так, что при продолжительном и целенаправленном наблюдении можно определить закономерность в их работе. Так что, псевдослучайность оставляет возможности для подобных сценариев [2].

ГЛАВА 1. АНАЛИЗ ИГРЫ «ОДНОРУКИЙ БАНДИТ»

1.1 Правила игры «Однорукий бандит»

Игра «Однорукий бандит» – это игровой автомат, основой которого являются три барабана, в которых при помощи генератора случайных чисел появляются изображения, и кнопка «Спин», которая запускает данные барабаны.

Когда останавливается последний барабан, три картинки образуют комбинации, которым заранее прописаны коэффициенты, представленные в таблице 1.1.

Таблица 1.1 – Коэффициенты комбинаций

Комбинация картинок			Коэффициент
Первый барабан	Второй барабан	Третий барабан	
Семёрка	Семёрка	Семёрка	200
Алмаз	Алмаз	Алмаз	150
Слиток	Слиток	Слиток	100
Слиток	Слиток	Алмаз	90
Слива	Слива	Слива	85
Слива	Слива	Алмаз	70
Арбуз	Арбуз	Арбуз	55
Арбуз	Арбуз	Алмаз	50
Банан	Банан	Банан	35
Банан	Банан	Алмаз	30
Семёрка	Семёрка	Любая	180
Семёрка	Любая	Семёрка	
Любая	Семёрка	Семёрка	
Алмаз	Алмаз	Любая	130
Алмаз	Любая	Алмаз	
Любая	Алмаз	Алмаз	
Слиток	Слиток	Любая	80
Слиток	Любая	Слиток	
Любая	Слиток	Слиток	
Слива	Слива	Любая	45
Слива	Любая	Слива	

Комбинация картинок			Коэффициент
Первый барабан	Второй барабан	Третий барабан	
Любая	Слива	Слива	25
Арбуз	Арбуз	Любая	
Арбуз	Любая	Арбуз	
Любая	Арбуз	Арбуз	
Банан	Банан	Любая	15
Банан	Любая	Банан	
Любая	Банан	Банан	
Вишня	Вишня	Любая	5
Вишня	Любая	Вишня	
Любая	Вишня	Вишня	

Примечание – Источник: собственная разработка

Размер выигрыша – это коэффициент, умноженный на ставку. В начале игры дан стартовый баланс, равный 100\$. Количество попыток в раунде – 5. Если в одном из барабанов выпадет клевер, игроку даётся ещё одна попытка. По истечении всех попыток игроку предлагается сделать ещё одну ставку, чтобы продолжить игру.

Чтобы приступить к игре, необходимо авторизироваться. Если пользователь играет в первый раз, ему нужно пройти краткую регистрацию. Первый спин можно сделать только после указания размера ставки, которую нельзя изменить в ходе одного раунда.

1.2 Описание требований к приложению

Название разрабатываемого приложения – «Однорукий бандит».

Игра «Однорукий бандит» – это игровой автомат, основой которого являются три барабана, в которых при помощи генератора случайных чисел появляются изображения, и кнопка «Спин», которая запускает данные барабаны.

Предполагаемая область применения данной компьютерной игры – это проведение досуга для пользователей любого возраста. При работе с данной у пользователя не должно возникать вопросов касемо правил и

последовательности действий интерфейс должен быть прост и удобен в использовании.

Приложение должно быть реализовано на языке C#, объектно-ориентированный язык программирования. Программа предназначена для игры одного пользователя.

Игра должна иметь:

- три барабана, в которых будут появляться изображения;
- поля для отображения баланса, сделанной ставки, выигрыша и количества оставшихся попыток;
- кнопки «Спин», «Сделать ставку», «Завершить игру», «Вернуться на начальный экран»;
- возможность нажатия на кнопку «Спин», чтобы начать игру;
- несколько таймеров, запускающих генерацию картинок на каждом из трёх барабанов и останавливающих их через определённый промежуток времени;
- ограничение на размер ставки (5, 10, 15, 20);
- коллекцию картинок.

Кроме того, приложение должно обладать простым пользовательским интерфейсом и иметь мягкую цветовую гамму.

При следующем запуске игры пользователю предоставляются все игровые формы в первоначальном состоянии.

1.3 Описание средств реализации, использованных классов и библиотек

C# – объектно-ориентированный язык программирования, который появился на свет в июне 2000 г., в результате кропотливой работы большой группы разработчиков компании Microsoft, возглавляемой Андерсом Хейлсбергом [4].

Язык C# как средство обучения программированию обладает рядом несомненных достоинств. Он хорошо организован, строг, большинство его конструкций логичны и удобны. Развитые средства диагностики и редактирования кода делают процесс программирования приятным и эффективным. Мощная библиотека классов платформы .NET берет на себя массу рутинных операций, что дает возможность решать более сложные задачи, используя готовые «строительные блоки». Все это позволяет расценивать C# как перспективную замену языков Pascal, BASIC и C++ при обучении программированию [5].

C# является не учебным, а профессиональным языком, предназначенным для решения широкого спектра задач, и в первую очередь – в быстро развивающейся области создания распределенных приложений. Поэтому базовый курс программирования, построенный на основе языка C#, позволяет студентам быстрее стать востребованными специалистами-профессионалами.

В результате проведенного анализа и того факта, что некоторые навыки по написанию программ на языке программирования C# у нас уже имеются, можно сделать вывод о том, что наиболее подходящим языком программирования является C#.

Подходящей средой программирования для написания программы игры «Танграм» является среда программирования Microsoft Visual Studio 2010 Express.

Среда разработки Microsoft Visual Studio 2010 Express является инструментом разработки net-приложений для Windows. В ней интегрированы удобный дизайнер форм, специализированный редактор кода, отладчик, мастер публикаций и другие компоненты, необходимые для написания программы.

Для работы с файлами используются классы пространства имён System.IO. Пространство имен System.IO содержит типы, позволяющие осуществлять чтение и запись в файлы и потоки данных, а также типы для базовой поддержки файлов и папок [6].

В данной работе использовались следующие классы пространства имён System.IO:

- класс Stream является абстрактным базовым классом всех потоков. Поток – это абстракция последовательности байтов, например файл, устройство ввода-вывода, канал взаимодействия процессов. Класс Stream и его производные классы предоставляют универсальное представление различных типов ввода и вывода, изолируя программиста от отдельных сведений операционной системы и базовых устройств;

- класс StreamReader служит для чтения строк данных из стандартного текстового файла;

- класс StreamWriter служит для записи вносимых данных в текстовый файл.

Для того, чтобы сгенерировать изображения, использовался класс Random. Он представляет генератор псевдослучайных чисел, то есть алгоритм, который генерирует последовательность чисел, отвечающую определенным статистическим критериям случайности. Псевдослучайные числа выбираются с равной вероятностью из конечного набора чисел. Выбранные числа не являются полностью случайными, поскольку для

их выбора используется математический алгоритм, но они достаточно случайные для практических целей.

1.4 Вывод по главе 1

В 1 главе был проведен анализ предметной области. Сформулировано техническое задание, на основании которого будет проводится дальнейшая работа по разработке приложения. Также были описаны средства разработки приложения, выявлены их достоинства и недостатки, и использованные в ходе разработки классы и библиотеки.

ГЛАВА 2. ОПИСАНИЕ РЕАЛИЗАЦИИ ПРИЛОЖЕНИЯ

2.1 Структура проекта

При реализации проекта было разработано пять игровых форм: Welcome (Form1.cs), Game (Form2.cs), PictureForm (Form3.cs), Registration (Form4.cs), Authorization (Form5.cs). Данные формы отличаются алгоритмом работы и шаблоном.

Проект содержит папку «Pectures», в которую загружены все изображения, используемые во время игры.

Также в проекте находится файл «Users», в котором хранятся имена и пароли зарегистрированных пользователей.

2.2 Описание форм C#

Элементы форм не имеют привязки к границам формы, что значит, что при изменении размера формы они распределяются равномерно, не теряя исходного положения. Все формы при загрузке программы имеют центральное расположение и минимальный размер.

Описание формы Welcome

Данная форма содержится в файле «Form1.cs» и осуществляет переход на форму авторизации либо завершает игру.

```
public partial class Welcome : Form
{
    public Welcome()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender,
EventArgs e)
    {
        Authorization frm5 = new Authorization();
        frm5.Show();
        Hide();
    }
}
```

```

    }

    private void button2_Click(object sender,
EventArgs e)
    {
        PictureForm frm3 = new PictureForm();
        frm3.Show();
        Hide();
    }
}

```

Листинг 2.1 – Функции перехода на форму авторизации и завершения игры

Примечание – Источник: Собственная разработка

Описание формы Game

Форма Game находится в файле «Form2.cs» и является основной в проекте. Она осуществляет реализацию всей игры, содержит метод загрузки картинок и их генерации, обработку таймеров, все необходимые расчёты, а также комбинации картинок с коэффициентами.

```

int attempt = 5;    //количество попыток
int bet = 0;        //начальная ставка
int balance = 100;  //стартовый баланс
int win = 0;        //выигрыш
int sum = 0;

```

Листинг 2.2 – Объявление переменных

Примечание – Источник: Собственная разработка

```

public string[] GetPicture()
{
    string[] picture = new string[8];

    picture[0] = "D:\\My
papers\\CourseWork\\pictures\\banana.png";
    picture[1] = "D:\\My
papers\\CourseWork\\pictures\\cherry.png";
    picture[2] = "D:\\My
papers\\CourseWork\\pictures\\clover.png";
    picture[3] = "D:\\My
papers\\CourseWork\\pictures\\diamond.png";
    picture[4] = "D:\\My
papers\\CourseWork\\pictures\\plum.png";
    picture[5] = "D:\\My
papers\\CourseWork\\pictures\\seven.png";

```

```

        picture[6] = "D:\\My
papers\\CourseWork\\pictures\\watermelon.png";
        picture[7] = "D:\\My
papers\\CourseWork\\pictures\\gold.png";

        return picture;
    }

```

Листинг 2.3 – Метод для загрузки изображений

Примечание – Источник: Собственная разработка

```

e)    private void engine1_Tick(object sender, EventArgs
    {
        string[] allpicture = GetPicture();
        Random rnd = new Random();
        int i = rnd.Next(8);
        pictureBox1.ImageLocation = allpicture[i];
    }

e)    private void engine2_Tick(object sender, EventArgs
    {
        string[] allpicture = GetPicture();
        Random rnd = new Random();
        int i = rnd.Next(8);
        pictureBox2.ImageLocation = allpicture[i];
    }

e)    private void engine3_Tick(object sender, EventArgs
    {
        string[] allpicture = GetPicture();
        Random rnd = new Random();
        int i = rnd.Next(8);
        pictureBox3.ImageLocation = allpicture[i];
    }

```

Листинг 2.4 – Таймеры для запуска барабанов и генерации изображений

Примечание – Источник: Собственная разработка

```

private void spin_Click(object sender, EventArgs e)
{
    attempt--;
    attemptlabel.Text = "Осталось попыток: " + attempt;

    engine1.Enabled = true;
    engine2.Enabled = true;
    engine3.Enabled = true;
    stop1.Enabled = true;
    stop2.Enabled = true;
    stop3.Enabled = true;
}

```

```

        spin.Enabled = false;
        setbet.Enabled = false;
        numericUpDown1.Enabled = false;
    }

```

Листинг 2.5 – Активация работы барабанов и стоп-таймеров при нажатии на кнопку «Спин»

Примечание – Источник: Собственная разработка

```

private void stop1_Tick(object sender, EventArgs e)
{
    engine1.Enabled = false;
    stop1.Enabled = false;
}

private void stop2_Tick(object sender, EventArgs e)
{
    engine2.Enabled = false;
    stop2.Enabled = false;
}

private void stop3_Tick(object sender, EventArgs e)
{
    engine3.Enabled = false;
    stop3.Enabled = false;
    spin.Enabled = true;
    Combinations();
    Attempt();
}

```

Листинг 2.6 – Таймеры для остановки барабанов

Примечание – Источник: Собственная разработка

```

private void numericUpDown1_ValueChanged(object sender,
EventArgs e)
{
    numericUpDown1.Increment = 5;
    numericUpDown1.Maximum = 20;
    numericUpDown1.Minimum = 5;
    numericUpDown1.ReadOnly = true;
}

private void Bet(decimal count)
{
    bet = Convert.ToInt32(count);
    balance -= bet;

    if (balance < 0)
    {
        MessageBox.Show("Игра завершена!", "Турурум...  
На балансе недостаточно средств", MessageBoxButtons.OK,
        MessageBoxIcon.Stop);
    }
}

```

```

        Welcome frm1 = new Welcome();
        frm1.Show();
        Hide();
    }
    else
    {
        balancelabel.Text = "Баланс: $" + balance;
        attempt = 5;
        attemptlabel.Text = "Осталось попыток: " +
attempt;
    }
}

private void setbet_Click(object sender, EventArgs e)
{
    Bet(numericUpDown1.Value);
    setbet.Enabled = false;
    numericUpDown1.Enabled = false;
    spin.Enabled = true;
}
private void Attempt()
{
    if (attempt != 0)
    {
        spin.Enabled = true;
    }
    else
    {
        attemptlabel.Text = "Осталось попыток: " +
attempt;
        MessageBox.Show("Сделайте новую ставку!",
"Попытки закончились...", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
        balance += sum;
        balancelabel.Text = "Баланс: $" + balance;
        sum = 0;
        winlabel.Text = "Выигрыш: $0";

        setbet.Enabled = true;
        numericUpDown1.Enabled = true;
    }
}
}

```

Листинг 2.7 – Расчёт ставки, условия нулевого баланса и количества оставшихся попыток

Примечание – Источник: Собственная разработка

```

private void Combinations()
{
    string[] allpicture = GetPicture();

```

```

        //один любой клевер
        if ((pictureBox1.ImageLocation == allpicture[2] &&
pictureBox2.ImageLocation != allpicture[2] &&
pictureBox3.ImageLocation != allpicture[2]) ||
        (pictureBox2.ImageLocation == allpicture[2] &&
pictureBox1.ImageLocation != allpicture[2] &&
pictureBox3.ImageLocation != allpicture[2]) ||
        (pictureBox3.ImageLocation == allpicture[2] &&
pictureBox1.ImageLocation != allpicture[2] &&
pictureBox2.ImageLocation != allpicture[2]))
        {
            Chance();
        }
        //два любых банана
        if ((pictureBox1.ImageLocation == allpicture[0] &&
pictureBox2.ImageLocation == allpicture[0] &&
pictureBox3.ImageLocation != allpicture[0]) ||
        (pictureBox2.ImageLocation == allpicture[0] &&
pictureBox3.ImageLocation == allpicture[0] &&
pictureBox1.ImageLocation != allpicture[0]) ||
        (pictureBox1.ImageLocation == allpicture[0] &&
pictureBox3.ImageLocation == allpicture[0] &&
pictureBox2.ImageLocation != allpicture[0]))
        {
            Win(15);
        }
        //две любые вишни
        if ((pictureBox1.ImageLocation == allpicture[1] &&
pictureBox2.ImageLocation == allpicture[1] &&
pictureBox3.ImageLocation != allpicture[1]) ||
        (pictureBox2.ImageLocation == allpicture[1] &&
pictureBox3.ImageLocation == allpicture[1] &&
pictureBox1.ImageLocation != allpicture[1]) ||
        (pictureBox1.ImageLocation == allpicture[1] &&
pictureBox3.ImageLocation == allpicture[1] &&
pictureBox2.ImageLocation != allpicture[1]))
        {
            Win(5);
        }
        //два любых алмаза
        if ((pictureBox1.ImageLocation == allpicture[3] &&
pictureBox2.ImageLocation == allpicture[3] &&
pictureBox3.ImageLocation != allpicture[3]) ||
        (pictureBox2.ImageLocation == allpicture[3] &&
pictureBox3.ImageLocation == allpicture[3] &&
pictureBox1.ImageLocation != allpicture[3]) ||
        (pictureBox1.ImageLocation == allpicture[3] &&
pictureBox3.ImageLocation == allpicture[3] &&
pictureBox2.ImageLocation != allpicture[3]))
        {
            Win(130);
        }
        //две любые сливы

```



```

        if ((pictureBox1.ImageLocation == allpicture[4] &&
pictureBox2.ImageLocation == allpicture[4] &&
pictureBox3.ImageLocation != allpicture[4]) ||
        (pictureBox2.ImageLocation == allpicture[4] &&
pictureBox3.ImageLocation == allpicture[4] &&
pictureBox1.ImageLocation != allpicture[4]) ||
        (pictureBox1.ImageLocation == allpicture[4] &&
pictureBox3.ImageLocation == allpicture[4] &&
pictureBox2.ImageLocation != allpicture[4]))
        {
            Win(45);
        }
        //две любые семёрки
        if ((pictureBox1.ImageLocation == allpicture[5] &&
pictureBox2.ImageLocation == allpicture[5] &&
pictureBox3.ImageLocation != allpicture[5]) ||
        (pictureBox2.ImageLocation == allpicture[5] &&
pictureBox3.ImageLocation == allpicture[5] &&
pictureBox1.ImageLocation != allpicture[5]) ||
        (pictureBox1.ImageLocation == allpicture[5] &&
pictureBox3.ImageLocation == allpicture[5] &&
pictureBox2.ImageLocation != allpicture[5]))
        {
            Win(180);
        }
        //два любых арбуза
        if ((pictureBox1.ImageLocation == allpicture[6] &&
pictureBox2.ImageLocation == allpicture[6] &&
pictureBox3.ImageLocation != allpicture[6]) ||
        (pictureBox2.ImageLocation == allpicture[6] &&
pictureBox3.ImageLocation == allpicture[6] &&
pictureBox1.ImageLocation != allpicture[6]) ||
        (pictureBox1.ImageLocation == allpicture[6] &&
pictureBox3.ImageLocation == allpicture[6] &&
pictureBox2.ImageLocation != allpicture[6]))
        {
            Win(25);
        }
        //два любых слитка
        if ((pictureBox1.ImageLocation == allpicture[7] &&
pictureBox2.ImageLocation == allpicture[7] &&
pictureBox3.ImageLocation != allpicture[7]) ||
        (pictureBox2.ImageLocation == allpicture[7] &&
pictureBox3.ImageLocation == allpicture[7] &&
pictureBox1.ImageLocation != allpicture[7]) ||
        (pictureBox1.ImageLocation == allpicture[7] &&
pictureBox3.ImageLocation == allpicture[7] &&
pictureBox2.ImageLocation != allpicture[7]))
        {
            Win(80);
        }
        //три семёрки
        if (pictureBox1.ImageLocation == allpicture[5] &&

```

```

pictureBox2.ImageLocation == allpicture[5] &&
pictureBox3.ImageLocation == allpicture[5])
{
    Win(200);
}
//три алмаза
if (pictureBox1.ImageLocation == allpicture[3] &&
pictureBox2.ImageLocation == allpicture[3] &&
pictureBox3.ImageLocation == allpicture[3])
{
    Win(150);
}
//три слитка
if (pictureBox1.ImageLocation == allpicture[7] &&
pictureBox2.ImageLocation == allpicture[7] &&
pictureBox3.ImageLocation == allpicture[7])
{
    Win(100);
}
//три сливы
if (pictureBox1.ImageLocation == allpicture[4] &&
pictureBox2.ImageLocation == allpicture[4] &&
pictureBox3.ImageLocation == allpicture[4])
{
    Win(75);
}
//три арбуза
if (pictureBox1.ImageLocation == allpicture[6] &&
pictureBox2.ImageLocation == allpicture[6] &&
pictureBox3.ImageLocation == allpicture[6])
{
    Win(55);
}
//три банана
if (pictureBox1.ImageLocation == allpicture[0] &&
pictureBox2.ImageLocation == allpicture[0] &&
pictureBox3.ImageLocation == allpicture[0])
{
    Win(35);
}
//два слитка и один алмаз
if (pictureBox1.ImageLocation == allpicture[7] &&
pictureBox2.ImageLocation == allpicture[7] &&
pictureBox3.ImageLocation == allpicture[3])
{
    Win(90);
}
//две сливы и один алмаз
if (pictureBox1.ImageLocation == allpicture[4] &&
pictureBox2.ImageLocation == allpicture[4] &&
pictureBox3.ImageLocation == allpicture[3])
{
    Win(70);
}

```

```

    }
    //два арбуза и один алмаз
    if (pictureBox1.ImageLocation == allpicture[6] &&
pictureBox2.ImageLocation == allpicture[6] &&
pictureBox3.ImageLocation == allpicture[3])
    {
        Win(50);
    }
    //два банана и один алмаз
    if (pictureBox1.ImageLocation == allpicture[0] &&
pictureBox2.ImageLocation == allpicture[0] &&
pictureBox3.ImageLocation == allpicture[3])
    {
        Win(30);
    }
}

```

Листинг 2.8 – Комбинации и коэффициенты

Примечание – Источник: Собственная разработка

Описание формы PictureForm

Данная форма находится в файле «Form3.cs». Она появляется при завершении игры и автоматически закрывается через две секунды.

```

public partial class PictureForm : Form
{
    public PictureForm()
    {
        InitializeComponent();

        Timer disappear = new Timer();
        disappear.Interval = 2000;
        disappear.Start();
        disappear.Tick += new EventHandler(disappear_Tick);
    }

    private void disappear_Tick(object sender, EventArgs e)
    {
        Close();
        Application.Exit();
    }
}

```

Листинг 2.9 – Таймер для автоматического закрытия формы

Примечание – Источник: Собственная разработка

Описание формы Registration

Форма осуществляет регистрацию пользователя. Данные, которые игрок вводит в поля, через пробел записываются в файл «Users». Форма содержится в файле «Form4.cs».

```
private void WriteToFile()
{
    string writerfileusers = @"D:\\My
papers\\CourseWork\\users.txt";
    using (StreamWriter writer = new
StreamWriter(writerfileusers, true))
    {
        writer.Write(nametextBox.Text + " ");
        writer.Write(passwordtextBox.Text + " ");
        writer.Close();
    }
}
```

Листинг 2.10 – Запись в файл

Примечание – Источник: Собственная разработка

```
private void Check()
{
    if (nametextBox.TextLength == 0 &&
passwordtextBox.TextLength == 0 &&
repeatpasswordtextBox.TextLength == 0)
    {
        MessageBox.Show("Поля не должны быть пустыми!",
"Проверьте введённые данные", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
    }
    else if (passwordtextBox.TextLength < 5)
    {
        MessageBox.Show("Пароль должен содержать не
менее 5 символов", "Проверьте введённые данные",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    else if (passwordtextBox.Text !=
repeatpasswordtextBox.Text)
    {
        MessageBox.Show("Пароли не совпадают!",
"Проверьте введённые данные", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
        repeatpasswordtextBox.Clear();
    }

    string readtext = File.ReadAllText(@"D:\\My
papers\\CourseWork\\users.txt");
}
```

```

        if (readtext.Contains(nametextBox.Text))
        {
            MessageBox.Show("Такое имя уже есть :(",
"Проверьте введенные данные", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
        }
        else if (readtext.Contains(passwordtextBox.Text))
        {
            MessageBox.Show("Такой пароль уже есть :(",
"Проверьте введенные данные", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
        }

        if ((nametextBox.TextLength != 0 &&
passwordtextBox.TextLength != 0 &&
repeatpasswordtextBox.TextLength != 0) &&
(passwordtextBox.TextLength > 5)
&& (passwordtextBox.Text ==
repeatpasswordtextBox.Text) &&
(! (readtext.Contains(nametextBox.Text)) &&
(! (readtext.Contains(passwordtextBox.Text)))))
        {
            WriteToFile();
            MessageBox.Show("Ваши данные записаны", "Наши
поздравления", MessageBoxButtons.OK);
            Authorization frm5 = new Authorization();
            frm5.Show();
            Hide();
        }
    }
}

```

Листинг 2.11 – Проверка полей

Примечание – Источник: Собственная разработка

Описание формы Authorization

В данной форме происходит авторизация пользователя. Данный, которые игрок вносит в поля, сравниваются с содержимым файла «Users» и, если имя и пароль совпадают, происходит переход на форму Game.

```

private void Check()
{
    string[] readtext = File.ReadAllText(@"D:\My
papers\CourseWork\users.txt").Split(' ');

    if (! (readtext.Contains(nametextBox1.Text)))
    {

```

```

        MessageBox.Show("Пользователя с таким именем
нет...\nВам нужно зарегистрироваться! ;)", "Что-то не так...",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    if (!(readtext.Contains(passwordtextBox1.Text)))
    {
        MessageBox.Show("Такого пароля нет...", "Что-то
не так...", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
    if (nametextBox1.TextLength == 0 &&
passwordtextBox1.TextLength == 0)
    {
        MessageBox.Show("Поля не должны быть пустыми!",
"Проверьте введенные данные", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
    }

    //чтение из файла
    for (int i = 0; i < readtext.Count() - 1; i += 2)
    {
        if (nametextBox1.Text == readtext[i] &&
passwordtextBox1.Text == readtext[i + 1])
        {
            MessageBox.Show("Данные подтверждены", "Вы
есть в наших списках!", MessageBoxButtons.OK);
            Game frm2 = new Game();
            frm2.Show();
            Hide();
        }
        if (!(nametextBox1.Text == readtext[i]) &&
(passwordtextBox1.Text == readtext[i + 1]))
        {
            MessageBox.Show("Имя пользователя и пароль
не совпадают", "Хмм...", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
        }
    }
}

```

Листинг 2.12 – Проверка полей и чтение из файла

Примечание – Источник: Собственная разработка

2.3 Функция кнопки «Спин»

Как было отмечено ранее, главная деталь в работе игрового автомата – это чип, который определяет, какие изображения появятся на экране барабанов. В данном проекте роль чипа осуществляет генератор

псевдослучайных чисел и таймер, который останавливает генерацию на определённом значении.

Как только игрок нажимает на кнопку «Спин», активируются первые три таймера, которые на каждом барабане запускают генерацию картинок. Далее работают три других таймера, имеющие возможность останавливать первые три таймера, то есть генератор изображений. Остановка происходит не одновременно, первый стоп-таймер включается через 1,5 секунды после нажатия на кнопку «Спин», второй стоп-таймер – через 2,5 секунды, а третий – через 3 секунды.

После того, как на последнем барабане появляется изображение, метод комбинаций, сравнивающий все три картинки, умножает раннее присвоенный данной комбинации коэффициент на ставку, установленную игроком. Размер выигрыша отображается в соответствующем поле.

2.4 Вывод по главе 2

Задачи, поставленные техническим заданием, были реализованы при помощи знаний языка C#. В ходе выполнения работы было реализовано приложение «Однорукий бандит». В данной главе была подробно описана структура проекта, а именно содержимое файлов и их назначение.

ГЛАВА 3. РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРИЛОЖЕНИЯ

3.1 Руководство пользователю

После запуска приложения загружается главная форма программы (Рисунок 3.1), где расположены две кнопки: «Старт» и «Завершить игру». Здесь также находится приветствие пользователя.



Рисунок 3.1 – Главная форма приложения

Примечание – Источник: Собственная разработка

Кнопка «Завершить игру» вызывает появление формы, представленной на Рисунке 3.2 и которая автоматически закрывается через две секунды. После этого программа закрывается.

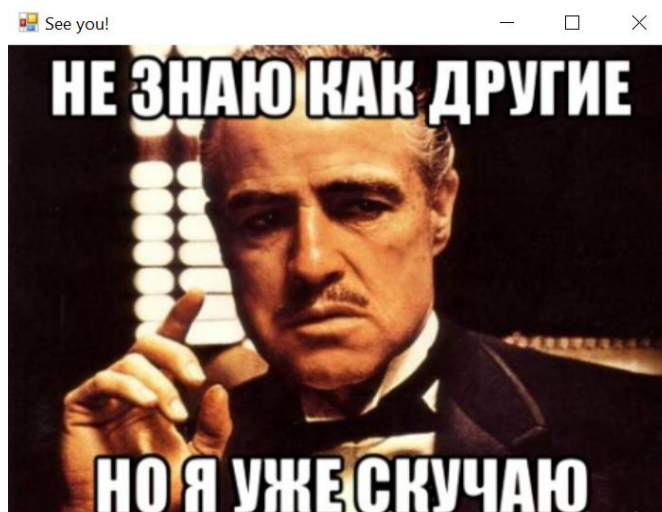


Рисунок 3.2 – Форма выхода из приложения

Примечание – Источник: Собственная разработка

При нажатии на кнопку «Старт» пользователь переходит на форму авторизации (Рисунок 3.3), где ему необходимо ввести имя и пароль, если он уже зарегистрирован.

A screenshot of a Windows-style window titled "Authorization". The window has a light blue background. At the top, in red italicized text, it says "Чтобы начать игру, вам необходимо авторизироваться" (To start the game, you must authorize). Below this, the word "Авторизация" (Authorization) is written in large, bold, blue letters. There are two input fields: "Имя:" (Name) with the text "alina" entered, and "Пароль:" (Password) with ten black dots. To the right of the password field is a small blue square button. At the bottom, there are two dark blue buttons with white text: "Зарегистрироваться" (Register) on the left and "Подтвердить данные" (Confirm data) on the right.

Рисунок 3.3 – Форма авторизации

Примечание – Источник: Собственная разработка

Синяя кнопка рядом с полем «Пароль» позволяет просмотреть введённый пароль, если пользователь не уверен, что ввёл его правильно. Однако система в любом случае даст понять, подтверждены ли данные или же произошла какая-то ошибка, например, введённое имя пользователя не

было найдено.

Каждое поле имеет ограничение на количество вводимых символов (до 15) и запрещение на ввод пробела.

Нажав на кнопку «Зарегистрироваться», пользователь перейдёт на форму регистрации (Рисунок 3.4). Здесь ему предлагают ввести имя, пароль и повторить введённый пароль.

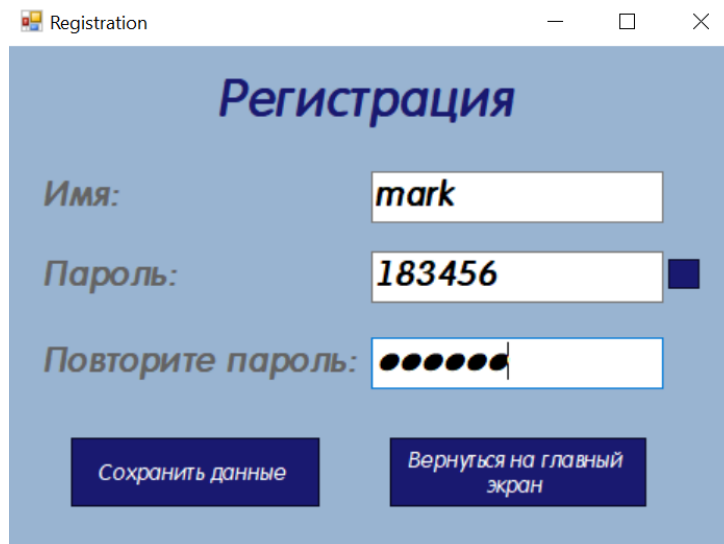


Рисунок 3.5 – Форма регистрации

Примечание – Источник: Собственная разработка

После нажатия на кнопку «Сохранить данные» система проверяет, есть ли пользователь с таким же именем и паролем в файле «Users». Если повторение есть, появится соответствующее уведомление. В противном случае данные запишутся в файл, и произойдёт автоматический переход на форму авторизации.

Как только данные подтверждены и пользователь вошёл в игру, открывается форма «Game» (Рисунок 3.6).



Рисунок 3.6 – Игра

Примечание – Источник: Собственная разработка

На форме представлены три барабана, которые не активированы, как и кнопка «Спин», но ровно до того момента, пока пользователь не сделает ставку. После этого с баланса спишется размер ставки, станет доступна кнопка «Спин» и можно начинать игру (Рисунок 3.7).

Изначально выигранные деньги отображаются в поле «Выигрыш», а после того, как закончатся все пять попыток, они суммируются с балансом.

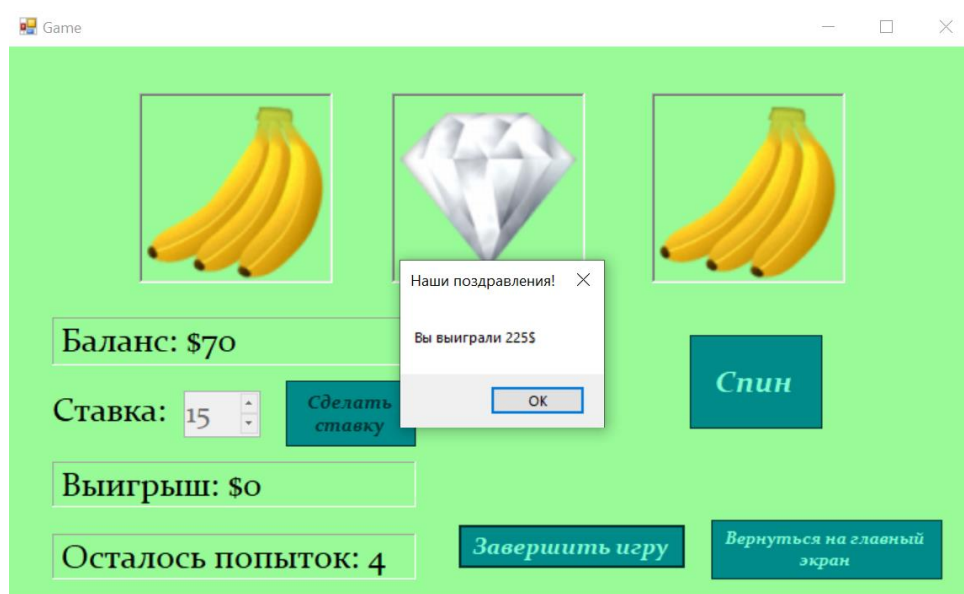


Рисунок 3.7 – Процесс игры

Примечание – Источник: Собственная разработка

При следующем запуске игры пользователю предоставляются все игровые формы в первоначальном состоянии.

3.2 Вывод по главе 3

В данной главе было подробно приведено руководство по использованию данного приложения. Рассмотрены все возможности и функции, а также приведены примеры для наглядного понимания процесса игры.

ЗАКЛЮЧЕНИЕ

Разработка приложений в среде разработки Microsoft Visual Studio очень популярна и актуальна среди как студентов, так и обычных пользователей.

В рамках курсовой работы была разработана игра «Однорукий бандит», которая предназначена для проведения досуга среди пользователей всех возрастов. Приложение интуитивно простое и понятное, для его использования не нужно специального обучения пользователя

Для создания данного игрового автомата был разработан интерфейс, проанализирован алгоритм генерации псевдослучайных чисел и изображений, реализован процесс записи данных во время регистрации и чтение данных из файла во время авторизации. Также был изучен принцип работы и математика игровых автоматов, подобраны коэффициенты для соответствующих комбинаций картинок.

Стоит отметить, что данная программа является приложением начального уровня и обладает возможностью дальнейшего усовершенствования. Например, можно составить большее количество комбинаций при помощи добавления новых изображений, в систему выхода из игры добавить предложение о записи результата и времени. Также можно добавить музыкальное сопровождение.

Таким образом, в процессе разработки приложения была достигнута цель курсовой работы и достигнуты поставленные задачи, а все требования соблюдены.

Программное приложение реализовано с использованием объектно-ориентированной технологии программирования. Тестирование, разрабатываемого приложения показало работоспособность компьютерной игры, целостность и структурированность.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Как работают игровые автоматы в казино [Электронный ресурс] / LatestCasinoBonuses. – 27.11.2020. – Режим доступа: Как работают игровые автоматы в казино и как считается возврат? . – Дата доступа: 23.05.2021.
2. Павловская, Т.А. С#. Программирование на языке высокого уровня. Учебник для вузов / Т.А. Павловская. – СПб: Питер, 2009. – 432 с.
3. Почему названо однорукий бандит [Электронный ресурс] / Развитие науки. – 2019. – Режим доступа: Почему названо однорукий бандит (v-pauke.ru). – Дата доступа: 23.05.2021.
4. Справочник по С# [Электронный ресурс] / MSDN. – Режим доступа: <https://msdn.microsoft.com/ru-ru/>. – Дата доступа: 30.04.2015.
5. Фленов, М.Е. Библия С# / М.Е. Фленов. – 2-е издание. – СПб: Питер, 2011. – 560 с.