УДК 004

Федотов В. А.

Студент, 4 курс, факультет «Информационные системы и технологии»

Северный Арктический Федеральный Университет, Высшая школа информационных технологий и автоматизированных систем

Россия, г Архангельск

РАЗРАБОТКА МОБИЛЬНОГО-ПРИЛОЖЕНИЯ НА ПЛАТФОРМЕ ANDROID

Аннотация: В статье описывается разработка мобильного приложения, используя платформу Android для разработки.

Ключевые слова: Разработка, мобильное приложение, платформа Android.

Fedotov V. A.

Student, 4 year, faculty "Information Systems and Technology"

Northern Arctic Federal University, Graduate School of Information Technology and Automated Systems

Russia, Arkhangelsk

DEVELOPMENT OF A MOBILE APPLICATION ON THE ANDROID PLATFORM

Annotation: This article describes how to develop a mobile app using the Android platform for development.

Keywords: Development, mobile application, Android platform.

1 АНАЛИЗ ТРЕБОВАНИЙ ПРИЛОЖЕНИЯ

Необходимо разработать игровое приложение «Пятнашки» на платформу Android. Пятнашки - известная всему миру головоломка. Игроку доступно поле размером 3x3, состоящее из 9 клеток. Все клетки кроме одной заняты костяшками с номерами от 1 до 8, которые перемешаны между собой. Цель игры - упорядочить костяшки по порядку используя свободное поле. Игра поможет развивать память и логическое мышление.

Разрабатывать мобильное приложение будем в Visual Studio, используя Xamarin. Xamarin – это фреймворк для кроссплатформенной разработки мобильных приложений (iOS, Android, Windows Phone) с использованием языка C#.

Для создания игры будем использовать интегрированную среду разработки Microsoft Visual Studio, потому что это оригинальная среда, которая позволяет редактировать, отлаживать и создавать код, а затем публиковать приложения. Интегрированная среда разработки (IDE) – это многофункциональная программа, которую можно использовать для различных аспектов разработки программного обеспечения. Помимо стандартного редактора и отладчика, которые существуют в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства выполнения кода, графические конструкторы и многие другие функции для упрощения процесса разработки программного обеспечения.

Для разработки игры будем использовать объектно-ориентированный язык программирования C#. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства,

обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Пользовательский интерфейс необходим для передачи информации между программно-аппаратными компонентами системы и пользователем. Следовательно, интерфейс должен быть максимально простым, удобным и интуитивно понятным для каждого пользователя.

Макет стартового окна игрового приложения изображен на рисунке 1.

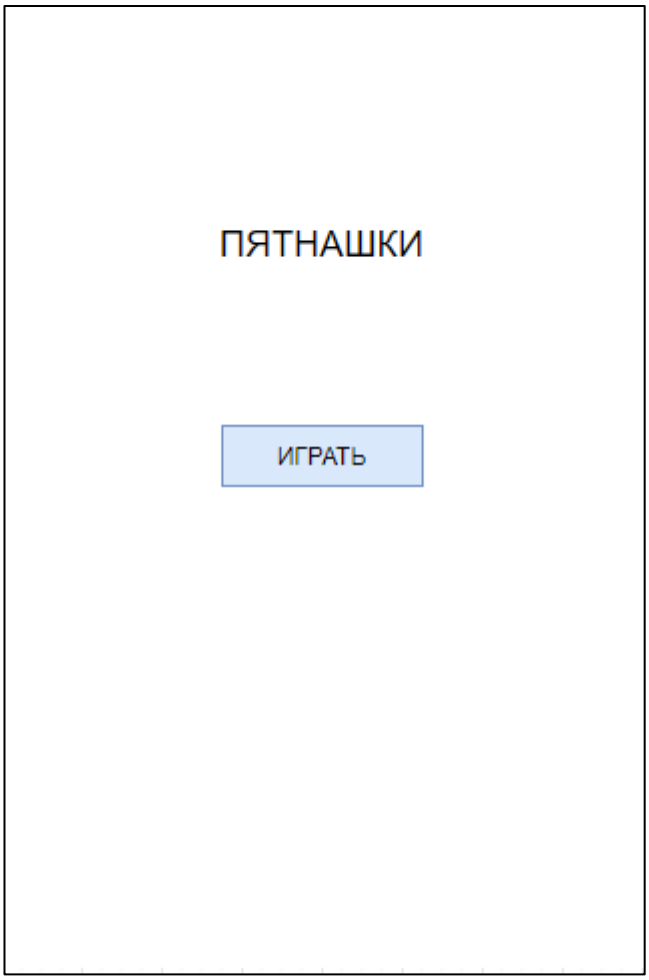

Рисунок 1 – Макет стартового окна игрового приложения

Макет главного окна игрового приложения изображен на рисунке 2.

Рисунок 2 – Макет главного окна игрового приложения

2 СРАВНИТЕЛЬНЫЙ АНАЛИЗ С СУЩЕСТВУЮЩИМИ АНАЛОГАМИ

Классические пятнашки представлены на рисунке 3. Классическая игра для тех, кто хочет скоротать время и потренировать свои мозги.

Рисунок 3 – Классические пятнашки

Пятнашки: собери картинку представлена на рисунке 4. Игра представляет собой набор квадратных костяшек с нанесёнными числами или изображением, заключённых в квадратную коробку, в коробке остаётся незаполненным одно квадратное поле.

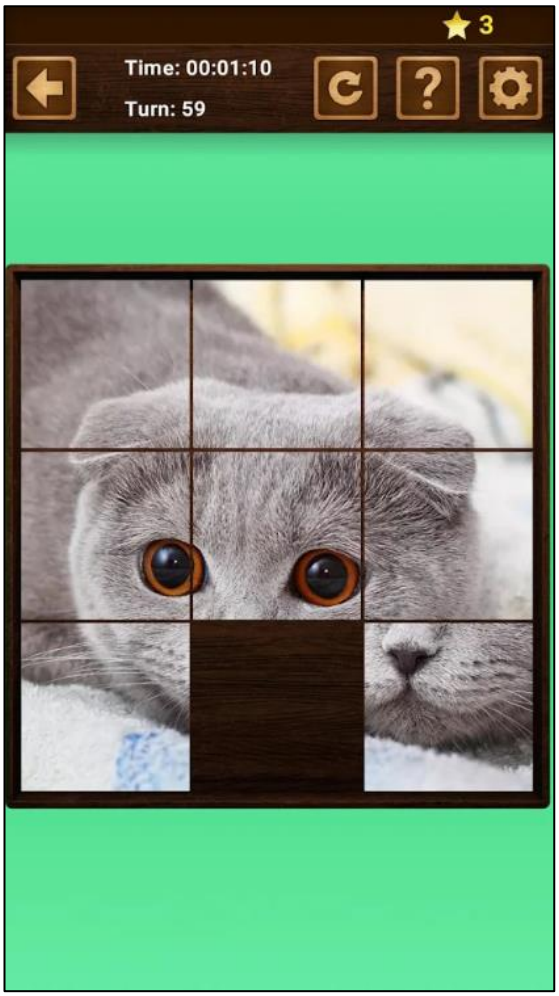

Рисунок 4 – Пятнашки: собери картинку

Пятнашки представлены на рисунке 5. Если Вам надоели скучные фишки, то эта игра для Вас. Вместо них здесь используются капли росы на зеленом листе. При этом они трясутся и булькают, когда до них дотрагиваешься. Необычный дизайн игры понравится как взрослым, так и детям.

Рисунок 5 – Пятнашки

Сравнение 3-х вышеописанных приложений представлено в таблице 1.

Таблица 1 – Сравнение существующих приложений

Мобильное приложение	Основной функционал	Отличительные особенности	Стоимость	Реклама	Интерфейс
Классические пятнашки	Выбор размера поля, лучший результат	На костяшках цифры	Бесплатно	Есть	Классический
Пятнашки: собери картинку	Выбор размера поля, время, ходы	На костяшках картинки и цифры	Бесплатно	Есть	Стандартный

Продолжение таблицы 1

Пятнашки	Выбор размера поля, время, ходы, звук	На костяшках цифры	Бесплатно	Есть	Необычный интерфейс, вместо костяшек капли росы

3 РАЗРАБОТКА ПРИЛОЖЕНИЯ

3.1 Описание приложения с точки зрения разработчика

Игровое приложение будет состоять из двух окон. Стартовое окно при запуске приложения и игровое окно с отображением игрового поля.

Создание пользовательского интерфейса стартового окна представлено в листинге 1.

Листинг 1 – Пользовательский интерфейс стартового окна

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:local="clr-namespace:PuzzleGame"
             x:Class="PuzzleGame.MainPage"
             >

    <StackLayout BackgroundColor="#ffffff">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition />
            </Grid.ColumnDefinitions>
            <Grid.RowDefinitions>
                <RowDefinition Height="200" />
                <RowDefinition />
            </Grid.RowDefinitions>
            <Label Text="ПЯТНАШКИ" FontSize="Large"
HorizontalOptions="Center" VerticalOptions="Center" Grid.Column="0"
Grid.Row="0" TextColor="#043A6B"/>
            <Button Text="ИГРАТЬ" x:Name="PlayBtn" FontSize="Medium"
Clicked="PlayBtn_Clicked" WidthRequest="200"
HorizontalOptions="Center" VerticalOptions="Start" Grid.Column="0"
Grid.Row="1" BackgroundColor="#408DD2"/>
        </Grid>
    </StackLayout>

</ContentPage>
```

Код для загрузки страницы и обработки нажатия на кнопку «Играть» представлен в листинге 2.

Листинг 2 – Загрузка и обработка нажатия

```
namespace PuzzleGame
{
    public partial class MainPage : ContentPage
    {
        public MainPage()
        {
            InitializeComponent();

        }
```

```
        async void PlayBtn_Clicked(object sender, EventArgs e)
        {
            var game = new Game();
            await Navigation.PushAsync(game);
        }
    }
}
```

Создание пользовательского интерфейса игрового окна представлено в листинге 3.

Листинг 3 - Пользовательский интерфейс игрового окна

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="PuzzleGame.Game">
    <ContentPage.Content>
        <StackLayout BackgroundColor="#ffffff">
            <Grid>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition />
                </Grid.ColumnDefinitions>
                <Grid.RowDefinitions>
                    <RowDefinition Height="200" />
                    <RowDefinition Height="230" />
                    <RowDefinition />
                    <RowDefinition />
                </Grid.RowDefinitions>
                <Label Text="ПЯТНАШКИ" FontSize="Large"
FontFamily="Times New Roman Bolds"  HorizontalOptions="Center"
VerticalOptions="Center" Grid.Column="0" Grid.Row="0"
TextColor="#043A6B"/>

                <!-- puzzles -->
                <Grid Grid.Column="0" Grid.Row="1"
HorizontalOptions="Center" VerticalOptions="Start" Margin="10">
                    <Grid.ColumnDefinitions>
                        <ColumnDefinition />
                        <ColumnDefinition />
                        <ColumnDefinition />
                    </Grid.ColumnDefinitions>
                    <Grid.RowDefinitions>
                        <RowDefinition Height="75" />
                        <RowDefinition Height="75" />
                        <RowDefinition Height="75" />
                    </Grid.RowDefinitions>
                    <Button x:Name="puz1" Clicked="Puz1_Clicked"
WidthRequest="75" Grid.Column="0" Grid.Row="0"
BackgroundColor="#0C5DA5" FontSize="Large"/>
                    <Button x:Name="puz2" Clicked="Puz2_Clicked"
WidthRequest="75" Grid.Column="1" Grid.Row="0"
BackgroundColor="#0C5DA5" FontSize="Large"/>
                    <Button x:Name="puz3" Clicked="Puz3_Clicked"
WidthRequest="75" Grid.Column="2" Grid.Row="0"
BackgroundColor="#0C5DA5" FontSize="Large"/>
```

```
                    <Button x:Name="puz4" Clicked="Puz4_Clicked"
WidthRequest="75" Grid.Column="0" Grid.Row="1"
BackgroundColor="#0C5DA5" FontSize="Large"/>
                    <Button x:Name="puz5" Clicked="Puz5_Clicked"
WidthRequest="75" Grid.Column="1" Grid.Row="1"
BackgroundColor="#0C5DA5" FontSize="Large"/>
                    <Button x:Name="puz6" Clicked="Puz6_Clicked"
WidthRequest="75" Grid.Column="2" Grid.Row="1"
BackgroundColor="#0C5DA5" FontSize="Large"/>
                    <Button x:Name="puz7" Clicked="Puz7_Clicked"
WidthRequest="75" Grid.Column="0" Grid.Row="2"
BackgroundColor="#0C5DA5" FontSize="Large"/>
                    <Button x:Name="puz8" Clicked="Puz8_Clicked"
WidthRequest="75" Grid.Column="1" Grid.Row="2"
BackgroundColor="#0C5DA5" FontSize="Large"/>
                    <Button x:Name="puz9" Clicked="Puz9_Clicked"
WidthRequest="75" Grid.Column="2" Grid.Row="2"
BackgroundColor="#0C5DA5" FontSize="Large"/>
                </Grid>

                <Button Text="НАЗАД" FontSize="Medium"
x:Name="backToMenuBtn" Clicked="BackToMenuBtn_Clicked"
WidthRequest="150" Grid.Column="0" Grid.Row="3"
HorizontalOptions="Start" VerticalOptions="Start"
BackgroundColor="#408DD2" Margin="25,0,0,0"/>
                <Button Text="ОБНОВИТЬ" FontSize="Medium"
x:Name="PlayBtn"       Clicked="PlayBtn_Clicked"
WidthRequest="150" Grid.Column="0" Grid.Row="3"
HorizontalOptions="End"   VerticalOptions="Start"
BackgroundColor="#408DD2" Margin="25,0,20,0"/>
                <Label Text="Поздравляем!" FontSize="Large"
x:Name="grats" Grid.Column="0" Grid.Row="4"
HorizontalOptions="Center" VerticalOptions="Start" IsVisible="False"
TextColor="#043A6B"/>
            </Grid>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

Далее опишем логику работы приложения. Для начала необходимо создать массив, который будем рандомно располагать значения в полях. Создание массива представлено в листинге 4.

Листинг 4 – Создание массива

```
static Random rnd = new Random();

        static void Shuffle<T>(T[] array)
        {
            int n = array.Length;
            for (int i = 0; i < n; i++)
            {
                int r = i + rnd.Next(n - i);
                T t = array[r];
                array[r] = array[i];
                array[i] = t;
```

```
            }
        }

        public Game()
        {
            InitializeComponent();

            Button[] puzzles = { puz1, puz2, puz3, puz4, puz5, puz6,
puz7, puz8, puz9 };

            string[] puz = { "1", "2", "3", "4", "5", "6", "7", "8",
"" };

            Shuffle(puz);
            for (int i = 0; i < puzzles.Length; i++)
            {
                puzzles[i].Text = puz[i];
            }
        }
```

После расстановки значений необходимо осуществить замену значений в ячейках, т. е. будет двигаться не ячейка, а меняться значение в ней. При нажатии на ячейку 1, код, представленный в листинге 5, будет перемещать значение, находящееся в ячейке 1 в соседнюю (пустую), т.е. либо в ячейку 2, либо в ячейку 4.

Листинг 5 – Нажатие на ячейку 1

```
private void Puz1_Clicked(object sender, EventArgs e)
        {
            if (puz2.Text == "")
            {
                puz2.Text = puz1.Text;
                puz1.Text = "";
            }
            if (puz4.Text == "")
            {
                puz4.Text = puz1.Text;
                puz1.Text = "";
            }

            EndCheck();
        }
```

При нажатии на ячейку 2, код, представленный в листинге 6, будет перемещать значение, находящееся в ячейке 2 в соседнюю (пустую), т. е. либо в ячейку 1, либо в ячейку 3, либо в ячейку 5.

Листинг 6 – Нажатие на ячейку 2

```
private void Puz2_Clicked(object sender, EventArgs e)
        {
            if (puz1.Text == "")
```

```
            {
                puz1.Text = puz2.Text;
                puz2.Text = "";
            }
            if (puz3.Text == "")
            {
                puz3.Text = puz2.Text;
                puz2.Text = "";
            }
            if (puz5.Text == "")
            {
                puz5.Text = puz2.Text;
                puz2.Text = "";
            }

            EndCheck();
        }
```

При нажатии на ячейку 3, код, представленный в листинге 7, будет перемещать значение, находящееся в ячейке 3 в соседнюю (пустую), т.е. либо в ячейку 2, либо в ячейку 6.

Листинг 7 – Нажатие на ячейку 3

```
private void Puz3_Clicked(object sender, EventArgs e)
        {
            if (puz2.Text == "")
            {
                puz2.Text = puz3.Text;
                puz3.Text = "";
            }
            if (puz6.Text == "")
            {
                puz6.Text = puz3.Text;
                puz3.Text = "";
            }

            EndCheck();
        }
```

При нажатии на ячейку 4, код, представленный в листинге 8, будет перемещать значение, находящееся в ячейке 4 в соседнюю (пустую), т.е. либо в ячейку 1, либо в ячейку 5, либо в ячейку 7.

Листинг 8 – Нажатие на ячейку 4

```
private void Puz4_Clicked(object sender, EventArgs e)
        {
            if (puz1.Text == "")
            {
                puz1.Text = puz4.Text;
                puz4.Text = "";
            }
            if (puz5.Text == "")
            {
```

```
                puz5.Text = puz4.Text;
                puz4.Text = "";
            }
            if (puz7.Text == "")
            {
                puz7.Text = puz4.Text;
                puz4.Text = "";
            }

            EndCheck();
        }
```

При нажатии на ячейку 5, код, представленный в листинге 9, будет перемещать значение, находящееся в ячейке 5 в соседнюю (пустую), т.е. либо в ячейку 2, либо в ячейку 4, либо в ячейку 6, либо в ячейку 8.

Листинг 9 – Нажатие на ячейку 5

```
private void Puz5_Clicked(object sender, EventArgs e)
        {
            if (puz2.Text == "")
            {
                puz2.Text = puz5.Text;
                puz5.Text = "";
            }
            if (puz4.Text == "")
            {
                puz4.Text = puz5.Text;
                puz5.Text = "";
            }
            if (puz6.Text == "")
            {
                puz6.Text = puz5.Text;
                puz5.Text = "";
            }
            if (puz8.Text == "")
            {
                puz8.Text = puz5.Text;
                puz5.Text = "";
            }

            EndCheck();
        }
```

При нажатии на ячейку 6, код, представленный в листинге 10, будет перемещать значение, находящееся в ячейке 6 в соседнюю (пустую), т.е. либо в ячейку 3, либо в ячейку 5, либо в ячейку 9.

Листинг 10 – Нажатие на ячейку 6

```
private void Puz6_Clicked(object sender, EventArgs e)
        {
            if (puz3.Text == "")
            {
                puz3.Text = puz6.Text;
```

```
                puz6.Text = "";
            }
            if (puz5.Text == "")
            {
                puz5.Text = puz6.Text;
                puz6.Text = "";
            }
            if (puz9.Text == "")
            {
                puz9.Text = puz6.Text;
                puz6.Text = "";
            }

            EndCheck();
        }
```

При нажатии на ячейку 7, код, представленный в листинге 11, будет перемещать значение, находящееся в ячейке 7 в соседнюю (пустую), т.е. либо в ячейку 4, либо в ячейку 8.

Листинг 11 – Нажатие на ячейку 7

```
private void Puz7_Clicked(object sender, EventArgs e)
        {
            if (puz4.Text == "")
            {
                puz4.Text = puz7.Text;
                puz7.Text = "";
            }
            if (puz8.Text == "")
            {
                puz8.Text = puz7.Text;
                puz7.Text = "";
            }

            EndCheck();
        }
```

При нажатии на ячейку 8, код, представленный в листинге 12, будет перемещать значение, находящееся в ячейке 8 в соседнюю (пустую), т.е. либо в ячейку 7, либо в ячейку 6, либо в ячейку 9.

Листинг 12 – Нажатие на ячейку 8

```
private void Puz8_Clicked(object sender, EventArgs e)
        {
            if (puz5.Text == "")
            {
                puz5.Text = puz8.Text;
                puz8.Text = "";
            }
            if (puz7.Text == "")
            {
                puz7.Text = puz8.Text;
                puz8.Text = "";
```

```
            }
            if (puz9.Text == "")
            {
                puz9.Text = puz8.Text;
                puz8.Text = "";
            }

            EndCheck();
        }
```

При нажатии на ячейку 9, код, представленный в листинге 13, будет перемещать значение, находящееся в ячейке 9 в соседнюю (пустую), т.е. либо в ячейку 8, либо в ячейку 6.

Листинг 13 – Нажатие на ячейку 9

```
private void Puz9_Clicked(object sender, EventArgs e)
        {
            if (puz6.Text == "")
            {
                puz6.Text = puz9.Text;
                puz9.Text = "";
            }
            if (puz8.Text == "")
            {
                puz8.Text = puz9.Text;
                puz9.Text = "";
            }

            EndCheck();
        }
```

После каждого клика по ячейке запускается метод «EndCheck()», который представлен в листинге 14. Данный метод необходим для проверки правильного расположения костяшек. При победе будет выведено соответствующее сообщение.

Листинг 14 – Метод «EndCheck()»

```
private void EndCheck()
        {
            if (puz1.Text == "1" && puz2.Text == "2" && puz3.Text ==
"3" && puz4.Text == "4" && puz5.Text == "5" && puz6.Text == "6" &&
puz7.Text == "7" && puz8.Text == "8" && puz9.Text == "")
            {
                grats.Text = "Поздравляем! Вы выиграли!";
                grats.IsVisible = true;
            }
            else
                grats.IsVisible = false;
        }
```

Методы обработки событий нажатий на кнопки «Назад» и «Обновить» представлены в листинге 15.

Листинг 15 – Обработка нажатий на кнопки «Назад» и «Обновить»

```
async void PlayBtn_Clicked(object sender, EventArgs e)
        {
            var game = new Game();
            await Navigation.PushAsync(game);
        }
        async void BackToMenuBtn_Clicked(object sender, EventArgs e)
        {
            await Navigation.PopAsync();
        }
```

3.2 Описание приложения с точки зрения пользователя

При запуске приложения запускается стартовое окно, представленное на рисунке 6.

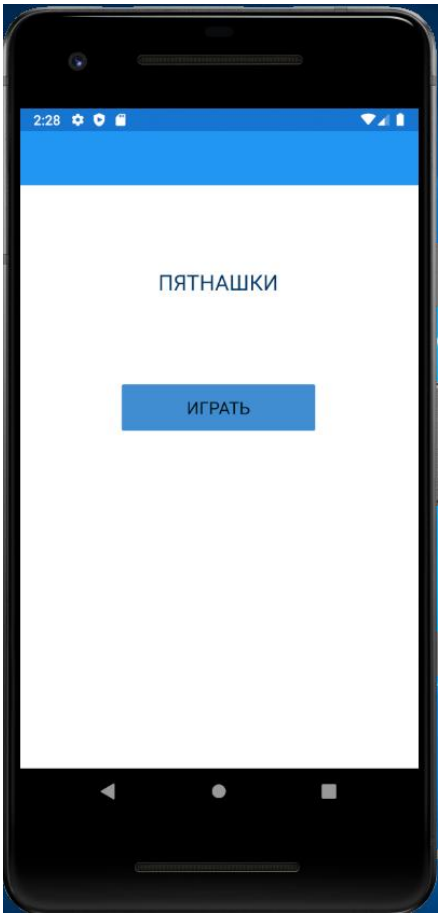

Рисунок 6 – Интерфейс стартового окна

После нажатия на кнопку «Играть» открывается игровое окно, представленное на рисунке 7.

Рисунок 7 – Игровое окно

ЗАКЛЮЧЕНИЕ

В итоге мы имеем работающее мобильное приложение на платформе Android, имеющее простой и понятный интерфейс. Приложение можно доработать, добавив в него выбор уровня, например, выбор поля 4 на 4 или 5 на 5 и так далее. Приложение имеет не сложную логику работы, поэтому в нем разберется разработчик, который не «писал» данное приложение.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Разработка приложений под мобильную платформу Android : учебное пособие / Д. В. Кравцов, М. А. Лосева, Е. А. Леонов [и др.]. — Москва : ФЛИНТА, 2018. — 72 с. — ISBN 978-5-9765-4014-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/113495 (дата обращения: 17.05.2020). — Режим доступа: для авториз. пользователей.

2. Черников, В. Разработка мобильных приложений на C# для iOS и Android : учебное пособие / В. Черников. — Москва : ДМК Пресс, 2020. — 188 с. — ISBN 978-5-97060-805-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/140592 (дата обращения: 17.05.2020). — Режим доступа: для авториз. пользователей.

3. Сильвен, Р. Android NDK. Разработка приложений под Android на C/C++ / Р. Сильвен ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2012. — 496 с. — ISBN 978-5-94074-657-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/9126 (дата обращения: 17.05.2020). — Режим доступа: для авториз. пользователей.