**РУКОВОДСТВО ПРОГРАММИСТА**

**Игра**

**«Flappy Bird»**

2022

**АННОТАЦИЯ**

В данном программном документе приведено руководство программиста по работе с игрой «Flappy Bird».

В разделе «Назначение и условия применения программы» указаны назначение и функции, условия, необходимые для её работы.

В разделе «Характеристика игры» приведено описание основных характеристик и особенностей.

1. Назначение и условия применения программы

С каждым днём игры приобретают всё большую популярность в связи со своей доступностью с множества устройств: телефонов, компьютеров, ноутбуков, планшетов.

Flappy Bird имеет игровой процесс с использованием 2D-графики. Цель игры состоит в управлении полётом птицы, которая непрерывно передвигается между рядами зелёных труб. При столкновении с ними происходит завершение игры. Управление производится нажатием на пробел, при котором птица совершает небольшой рывок вверх. При отсутствии рывков птица падает из-за силы тяжести, и игра также завершается. Очки набираются при каждом успешном перелёте между двумя трубами. Геймплей не имеет изменений на протяжении всей игры.

1. Характеристики и работа программы

Требования к логике игры, которые были выполнены в ходе разработки:

1. Возможность поиграть есть у любого человека, у которого есть доступ в интернет для возможности скачать игру.

На рисунке 1 представлена файловая архитектура «Flappy Bird»

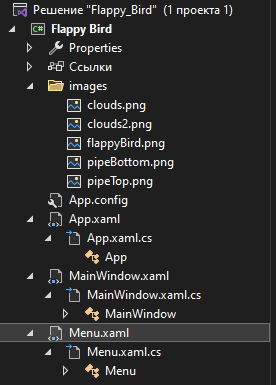


Рисунок 1 – Файловая архитектура игры

Для работы игры необходимы все файлы, представленные на рисунке 1, их можно разделить на следующие группы:

1. Форма меню
2. Форма игры
3. Код игровой формы
4. Коды формы с меню
5. Картинки

При создании игры использовался язык C#, бля создания форм использовался Xaml.

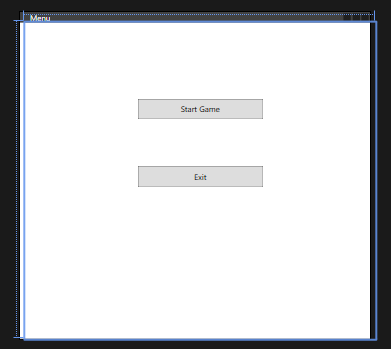


Рисунок 2 - Меню



Рисунок 3 – Форма с игрой

ЛИСТИНГ ПРОГРАММЫ

Форма с игрой

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Windows.Threading;

namespace Flappy\_Bird

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

DispatcherTimer gameTimer = new DispatcherTimer();

double score;

int gravity = 8;

bool gameOver;

Rect flappyBirdHitBox;

public MainWindow()

{

InitializeComponent();

gameTimer.Tick += MainEventTimer;

gameTimer.Interval = TimeSpan.FromMilliseconds(20);

StartGame();

}

private void MainEventTimer(object sender, EventArgs e)

{

txtScore.Content = "Score: " + score;

flappyBirdHitBox = new Rect(Canvas.GetLeft(flappyBird), Canvas.GetTop(flappyBird), flappyBird.Width - 12, flappyBird.Height);

Canvas.SetTop(flappyBird, Canvas.GetTop(flappyBird) + gravity);

if (Canvas.GetTop(flappyBird) < -30 || Canvas.GetTop(flappyBird) + flappyBird.Height > 460)

{

EndGame();

}

foreach (var x in MyCanvas.Children.OfType<Image>())

{

if ((string)x.Tag == "obs1" || (string)x.Tag == "obs2" || (string)x.Tag == "obs3")

{

Canvas.SetLeft(x, Canvas.GetLeft(x) - 5);

if (Canvas.GetLeft(x) < -100)

{

Canvas.SetLeft(x, 800);

score += .5;

}

Rect PillarHitBox = new Rect(Canvas.GetLeft(x), Canvas.GetTop(x), x.Width, x.Height);

if (flappyBirdHitBox.IntersectsWith(PillarHitBox))

{

EndGame();

}

}

if ((string)x.Tag == "clouds")

{

Canvas.SetLeft(x, Canvas.GetLeft(x) - 1);

if (Canvas.GetLeft(x) < -250)

{

Canvas.SetLeft(x, 550);

score += .5;

}

}

}

}

private void KeyIsDown(object sender, KeyEventArgs e)

{

if (e.Key == Key.Space)

{

flappyBird.RenderTransform = new RotateTransform(-20, flappyBird.Width / 2, flappyBird.Height / 2);

gravity = -8;

}

if (e.Key == Key.R && gameOver == true)

{

StartGame();

}

if (e.Key == Key.Escape && gameOver == true)

{

Menu dd = new Menu();

dd.Show();

Close();

}

}

private void KeyIsUp(object sender, KeyEventArgs e)

{

flappyBird.RenderTransform = new RotateTransform(5, flappyBird.Width /2, flappyBird.Height /2);

gravity = 8;

}

private void StartGame()

{

MyCanvas.Focus();

int temp = 300;

score = 0;

gameOver = false;

Canvas.SetTop(flappyBird, 190);

foreach (var x in MyCanvas.Children.OfType<Image>())

{

if ((string)x.Tag == "obs1")

{

Canvas.SetLeft(x, 500);

}

if ((string)x.Tag == "obs2")

{

Canvas.SetLeft(x, 800);

}

if ((string)x.Tag == "obs3")

{

Canvas.SetLeft(x, 1100);

}

if ((string)x.Tag == "clouds")

{

Canvas.SetLeft(x, 300 + temp);

temp = 800;

}

}

gameTimer.Start();

}

private void EndGame()

{

gameTimer.Stop();

gameOver = true;

txtScore.Content += " Game Over!!! " + "Press R to restart.";

}

}

}