Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

ПСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Передовая инженерная школа гибридных технологий в станкостроении Союзного государства

Отделение информационно-коммуникационных технологий

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

Мультимедиа технологии

Лабораторная работа №1

Разработка анимированной интерактивной презентации по технологии WPF

Выполнил студент

группы 0434-05

Гладков А.А.

Проверил:

Антонов И.В.

Псков

2025

**Оглавление**

[Задание 3](#_Toc191938668)

[Файл MainWindow.xaml.cs 5](#_Toc191938669)

[Файл Bomb.xaml 10](#_Toc191938670)

[Файл Bomb.xaml.cs 10](#_Toc191938671)

[Файл GameOver.xaml 11](#_Toc191938672)

[Файл GameOver.xaml.cs 11](#_Toc191938673)

[Файл ShipDto.cs 12](#_Toc191938674)

[Файл ShotDto.cs 12](#_Toc191938675)

[Файл SpaceShip.xaml 12](#_Toc191938676)

[Файл SpaceShip.xaml.cs 13](#_Toc191938677)

[Результат работы 15](#_Toc191938678)

# Задание

Разработать интерактивное приложение, использующее анимированные элементы, программно-управляемые на основе обработки действий пользователя программы. Индивидуальное задание согласовать с преподавателем.

Вариант задания:

Спроектировать игровую программу "Звездные войны".   
Внизу окна программы горизонтально перемещается платформа с вертикальным орудием, которое может вести огонь по вражеским кораблям. В верхней части окна горизонтально пролетают вражеские корабли, сбрасывая бомбы. За пораженный корабль начисляются очки. После пяти попаданий бомб в платформу игрок погибает. С течением времени сложность игры должна возрастать (частота появления кораблей, частота их бомбометания, число попаданий, необходимых для поражения кораблей).

Файл MainWindow.xaml

<Window x:Class="StarWars.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:StarWars"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800" KeyDown="Window\_KeyDown">

<Window.Resources>

<Storyboard x:Key="ShotInGunStoryboard" Completed="Storyboard\_Completed">

<DoubleAnimationUsingKeyFrames Storyboard.TargetName="Gun" Storyboard.TargetProperty="(UIElement.Opacity)">

<EasingDoubleKeyFrame KeyTime="00:00:00" Value="1"/>

<EasingDoubleKeyFrame KeyTime="00:00:00.5000000" Value="0"/>

<EasingDoubleKeyFrame KeyTime="00:00:01" Value="1"/>

<EasingDoubleKeyFrame KeyTime="00:00:01.5000000" Value="0"/>

<EasingDoubleKeyFrame KeyTime="00:00:02" Value="1"/>

<EasingDoubleKeyFrame KeyTime="00:00:02.5000000" Value="0"/>

<EasingDoubleKeyFrame KeyTime="00:00:03" Value="1"/>

</DoubleAnimationUsingKeyFrames>

</Storyboard>

</Window.Resources>

<Grid Height="450" Width="800">

<Button Margin="661,358,30,44" Click="Button\_Click" Foreground="Black" BorderBrush="Black" Background="Black">

<Button.Template>

<ControlTemplate>

<Grid>

<Rectangle x:Name="btn1" Stroke="Black" Grid.ColumnSpan="2" RadiusX="8" RadiusY="10"/>

<Label x:Name="buttonCaption" VerticalAlignment="Center"

HorizontalAlignment="Center" Content="Старт" FontSize="20" Background="White" Foreground="Black"></Label>

</Grid>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter TargetName="btn1" Property="Fill" Value="#FF4200FF" />

<Setter TargetName="buttonCaption" Property="Background" Value="#FF4200FF" />

<Setter TargetName="buttonCaption" Property="Foreground" Value="White" />

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Button.Template>

</Button>

<TextBlock x:Name="Timer" Text="00:00" Margin="661,10,30,403" Foreground="Black" FontSize="24" TextAlignment="Center"/>

<TextBlock x:Name="Size" Margin="661,170,30,244" Foreground="Black" FontSize="24" TextAlignment="Center"><Run Language="ru-ru" Text="Очки "/><Run Text=" 0"/></TextBlock>

<TextBlock x:Name="Lives" Text="5 жизней" Margin="661,225,30,188" Foreground="Black" FontSize="24" TextAlignment="Center"/>

<Canvas x:Name="GameZone" Margin="0,0,160,37" Height="400" Width="655">

<Canvas.Background>

<ImageBrush ImageSource="/background.png"/>

</Canvas.Background>

<Rectangle x:Name="Gun" Width="50" HorizontalAlignment="Left" Height="50" Canvas.Left="302.5" Canvas.Top="340" VerticalAlignment="Top">

<Rectangle.Fill>

<ImageBrush ImageSource="/pig.png" Stretch="Fill"/>

</Rectangle.Fill>

</Rectangle>

</Canvas>

</Grid>

</Window>

# Файл MainWindow.xaml.cs

using System;

using System.Collections.Generic;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Animation;

using System.Windows.Shapes;

using System.Windows.Threading;

namespace StarWars

{

public partial class MainWindow : Window

{

private byte lives;

byte ticks = 0;

private int points;

private bool start = false;

private double spawnShipTick = 5;

private DispatcherTimer timer;

private DispatcherTimer fastTimer;

private DispatcherTimer shotTimer;

private DispatcherTimer spawnShipTimer;

private TimeSpan elapsedTime;

private List<ShotDto> shots;

private List<ShipDto> ships;

public static List<Bomb> bombs;

public MainWindow()

{

InitializeComponent();

InitializeTimer();

}

private void InitializeTimer()

{

timer = new DispatcherTimer();

timer.Interval = TimeSpan.FromSeconds(1);

timer.Tick += Timer\_Tick;

fastTimer = new DispatcherTimer();

fastTimer.Interval = TimeSpan.FromMilliseconds(18);

fastTimer.Tick += Fast\_Tick;

shotTimer = new DispatcherTimer();

shotTimer.Interval = TimeSpan.FromSeconds(1.2);

shotTimer.Tick += Shot;

spawnShipTimer = new DispatcherTimer();

spawnShipTimer.Interval = TimeSpan.FromSeconds(spawnShipTick);

spawnShipTimer.Tick += spawn\_tick;

elapsedTime = TimeSpan.Zero;

}

private void Timer\_Tick(object sender, EventArgs e)

{

ticks++;

if (ticks == 2) {

ticks = 0;

if(BoomParameters.chanceBam < 8)

{

BoomParameters.chanceBam++;

}

if(BoomParameters.bomdDropTime > 1)

{

BoomParameters.bomdDropTime -= 0.1;

}

if(spawnShipTick > 0.5)

{

spawnShipTick -= 0.1;

}

}

elapsedTime = elapsedTime.Add(TimeSpan.FromSeconds(1));

Timer.Text = elapsedTime.ToString(@"mm\:ss");

}

private void Fast\_Tick(object sender, EventArgs e)

{

for (int i = 0; i < shots.Count; ++i)

{

var shot = shots[i];

for (int j = 0; j < ships.Count; ++j)

{

var ship = ships[j];

if (ship.getShip().IsHit(shot.getShot()))

{

points += 1;

Size.Text = "Очки " + points;

Storyboard storyboard = ship.getShip().FindResource("DestroyStoryboard") as Storyboard;

storyboard.Completed += (s, args) => DestroyStoryboard\_Completed(s, args, ship);

ship.getShip().destroyed = true;

storyboard.Begin();

GameZone.Children.Remove(shots[0].getShot());

shots.Remove(shots[0]);

ships.Remove(ships[j]);

i--;

j--;

}

}

}

for (int i = 0; i < shots.Count; i++) {

if (Canvas.GetBottom(shots[i].getShot()) < GameZone.Height)

{

Canvas.SetBottom(shots[i].getShot(), Canvas.GetBottom(shots[i].getShot()) + 2.5);

} else

{

GameZone.Children.Remove(shots[i].getShot());

shots.Remove(shots[i]);

i--;

}

}

for (int i = 0; i < ships.Count; i++)

{

if (ships[i].getPlaceOnRight())

{

if (Canvas.GetLeft(ships[i].getShip()) > 0)

{

Canvas.SetLeft(ships[i].getShip(), Canvas.GetLeft(ships[i].getShip()) - 1.7);

}

else

{

GameZone.Children.Remove(ships[i].getShip());

ships.Remove(ships[i]);

i--;

}

}

else {

if (Canvas.GetLeft(ships[i].getShip()) < GameZone.Width - ships[i].getShip().Width)

{

Canvas.SetLeft(ships[i].getShip(), Canvas.GetLeft(ships[i].getShip()) + 1.7);

}

else

{

GameZone.Children.Remove(ships[i].getShip());

ships.Remove(ships[i]);

i--;

}

}

}

for (int i = 0; i < bombs.Count; i++)

{

if (bombs[i].IsHit(Gun)){

bombs[i].IsBam = true;

timer.Stop();

shotTimer.Stop();

fastTimer.Stop();

spawnShipTimer.Stop();

lives -= 1;

Lives.Text = lives + " жизней";

if (lives == 0)

{

GameOver gameOver = new GameOver();

gameOver.ShowDialog();

InitializeTimer();

foreach (var shot in shots)

{

GameZone.Children.Remove(shot.getShot());

}

foreach (var ship in ships)

{

GameZone.Children.Remove(ship.getShip());

}

foreach (var bomb in bombs)

{

GameZone.Children.Remove(bomb);

}

ships = new List<ShipDto>();

shots = new List<ShotDto>();

bombs = new List<Bomb>();

start = false;

} else

{

Storyboard storyboard = bombs[i].FindResource("BamStoryboard") as Storyboard;

storyboard.Completed += (s, args) => BamStoryboard\_Completed(s, args, bombs[0]);

storyboard.Begin();

Storyboard GunStoryboard = FindResource("ShotInGunStoryboard") as Storyboard;

GunStoryboard.Begin();

}

}

if (bombs.Count != 0)

{

if (!bombs[i].IsBam)

{

if (Canvas.GetTop(bombs[i]) > 0)

{

Canvas.SetTop(bombs[i], Canvas.GetTop(bombs[i]) + 2);

}

else

{

GameZone.Children.Remove(bombs[i]);

bombs.Remove(bombs[i]);

i--;

}

}

}

}

}

private void BamStoryboard\_Completed(object s, EventArgs args, Bomb bomb){

GameZone.Children.Remove(bomb);

bombs.Remove(bomb);

}

private void DestroyStoryboard\_Completed(object s, EventArgs args, ShipDto ship)

{

GameZone.Children.Remove(ship.getShip());

}

private void Shot(object sender, EventArgs e)

{

Rectangle shot = new Rectangle

{

Stroke = Brushes.White,

StrokeThickness = 2,

Width = 2,

Height = 5,

};

Canvas.SetLeft(shot, Canvas.GetLeft(Gun) + Gun.ActualWidth / 2);

Canvas.SetBottom(shot, 15);

GameZone.Children.Add(shot);

Storyboard storyboard = new Storyboard();

ShotDto dto = new ShotDto(shot, storyboard);

shots.Add(dto);

}

private void spawn\_tick(object sender, EventArgs e)

{

var ship = new SpаceShip();

Random random = new Random();

bool placeOnRight = random.NextDouble() >= 0.5;

Storyboard storyboard = new Storyboard();

Canvas.SetTop(ship, 3);

if (placeOnRight)

{

Canvas.SetLeft(ship, GameZone.ActualWidth - ship.Width);

}

else

{

Canvas.SetLeft(ship, 0);

SkewTransform skewTransform = new SkewTransform(0, 180);

ship.RenderTransform = skewTransform;

ship.RenderTransformOrigin = new Point(0.5, 0.5);

}

GameZone.Children.Add(ship);

ShipDto dto = new ShipDto(ship, storyboard, placeOnRight);

ships.Add(dto);

}

private void Window\_KeyDown(object sender, KeyEventArgs e)

{

double currentLeft = Canvas.GetLeft(Gun);

switch (e.Key)

{

case Key.Left:

case Key.A:

if (Canvas.GetLeft(Gun) - 20 >= 0)

{

Canvas.SetLeft(Gun, currentLeft - 20);

}

break;

case Key.Right:

case Key.D:

if (currentLeft + Gun.Width + 20 <= GameZone.ActualWidth)

{

Canvas.SetLeft(Gun, currentLeft + 20);

}

break;

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

if (!start) {

start = true;

shots = new List<ShotDto>();

ships = new List<ShipDto>();

bombs = new List<Bomb>();

lives = 5;

points = 0;

ticks = 0;

BoomParameters.bomdDropTime = 3;

BoomParameters.chanceBam = 2;

Size.Text = "Очки " + points;

Lives.Text = lives + " жизней";

timer.Start();

shotTimer.Start();

fastTimer.Start();

spawnShipTimer.Start();

}

}

private void Storyboard\_Completed(object sender, EventArgs e)

{

timer.Start();

fastTimer.Start();

shotTimer.Start();

spawnShipTimer.Start();

}

}

}

# Файл Bomb.xaml

<UserControl x:Class="StarWars.Bomb"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:StarWars"

mc:Ignorable="d"

Width="10" Height="12">

<UserControl.Resources>

<Storyboard x:Key="BamStoryboard">

<DoubleAnimationUsingKeyFrames Storyboard.TargetName="image" Storyboard.TargetProperty="(UIElement.Opacity)">

<EasingDoubleKeyFrame KeyTime="00:00:00" Value="1"/>

<EasingDoubleKeyFrame KeyTime="00:00:01.5000000" Value="0"/>

</DoubleAnimationUsingKeyFrames>

</Storyboard>

</UserControl.Resources>

<Grid>

<Image x:Name="image" Source="/egg.png" Stretch="Fill" RenderTransformOrigin="0.5,0.5">

<Image.RenderTransform>

<TransformGroup>

<ScaleTransform ScaleX="2" ScaleY="2"/>

<SkewTransform/>

<RotateTransform/>

<TranslateTransform/>

</TransformGroup>

</Image.RenderTransform>

</Image>

</Grid>

</UserControl>

# Файл Bomb.xaml.cs

public partial class Bomb : UserControl

{

public bool IsBam = false;

public Bomb()

{

InitializeComponent();

}

public bool IsHit(Rectangle Gun)

{

double bombLeft = Canvas.GetLeft(this);

double bombTop = Canvas.GetTop(this);

double gunLeft = Canvas.GetLeft(Gun);

double gunTop = Canvas.GetTop(Gun);

bool overlapX = bombLeft < gunLeft + Gun.Width && bombLeft + Width > gunLeft;

bool overlapY = bombTop < gunTop + Gun.Height && bombTop + Height > gunTop;

return overlapX && overlapY;

}

}

# Файл GameOver.xaml

<Window x:Class="StarWars.GameOver"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:StarWars"

mc:Ignorable="d"

Title="GameOver" Height="147" Width="204">

<Grid Height="130" VerticalAlignment="Top">

<Button Margin="64,65,64,25" Click="Button\_Click">

<Button.Template>

<ControlTemplate>

<Grid>

<Rectangle x:Name="btn1" Stroke="Red" RadiusX="8" RadiusY="10" Margin="5,2,5,2"/>

<Label x:Name="buttonCaption" VerticalAlignment="Center"

HorizontalAlignment="Center" Content="ОК" FontSize="15" Background="White" Foreground="Red" Height="30" Width="30"/>

</Grid>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter TargetName="btn1" Property="Fill" Value="Red" />

<Setter TargetName="buttonCaption" Property="Background" Value="Red" />

<Setter TargetName="buttonCaption" Property="Foreground" Value="White" />

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Button.Template>

</Button>

<TextBlock TextWrapping="Wrap" FontSize="24" TextAlignment="Center" Foreground="Red" Margin="0,15,0,0" VerticalAlignment="Top"><Run Language="ru-ru" Text="Вы проиграли"/></TextBlock>

</Grid>

</Window>

# Файл GameOver.xaml.cs

public partial class GameOver : Window

{

public GameOver()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

this.DialogResult = true;

}

}

# Файл ShipDto.cs

public class ShipDto

{

private SpаceShip ship;

private Storyboard storyboard;

private bool placeOnRight;

public ShipDto(SpаceShip ship, Storyboard storyboard, bool placeOnRight)

{

this.ship = ship;

this.storyboard = storyboard;

this.placeOnRight = placeOnRight;

}

public SpаceShip getShip()

{

return ship;

}

public Storyboard getStoryboard()

{

return storyboard;

}

public bool getPlaceOnRight()

{

return placeOnRight;

}

}

# Файл ShotDto.cs

public class ShotDto

{

private Rectangle shot;

private Storyboard storyboard;

public ShotDto(Rectangle shot, Storyboard storyboard)

{

this.shot = shot;

this.storyboard = storyboard;

}

public Rectangle getShot()

{

return shot;

}

public Storyboard getStoryboard() {

return storyboard;

}

}

# Файл SpaceShip.xaml

<UserControl x:Class="StarWars.SpаceShip"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:StarWars"

mc:Ignorable="d" Height="40" Width="40">

<UserControl.Resources>

<Storyboard x:Key="DestroyStoryboard">

<DoubleAnimationUsingKeyFrames Storyboard.TargetName="image" Storyboard.TargetProperty="(UIElement.Opacity)">

<EasingDoubleKeyFrame KeyTime="00:00:00" Value="1"/>

<EasingDoubleKeyFrame KeyTime="00:00:00.5000000" Value="0.5"/>

<EasingDoubleKeyFrame KeyTime="00:00:01" Value="0.25">

<EasingDoubleKeyFrame.EasingFunction>

<BounceEase EasingMode="EaseOut"/>

</EasingDoubleKeyFrame.EasingFunction>

</EasingDoubleKeyFrame>

<EasingDoubleKeyFrame KeyTime="00:00:01.5000000" Value="0"/>

</DoubleAnimationUsingKeyFrames>

<ObjectAnimationUsingKeyFrames Storyboard.TargetName="image" Storyboard.TargetProperty="(Image.Stretch)">

<DiscreteObjectKeyFrame KeyTime="00:00:00" Value="{x:Static Stretch.Fill}"/>

<DiscreteObjectKeyFrame KeyTime="00:00:00.2000000" Value="{x:Static Stretch.Uniform}"/>

<DiscreteObjectKeyFrame KeyTime="00:00:00.5000000" Value="{x:Static Stretch.Fill}"/>

</ObjectAnimationUsingKeyFrames>

</Storyboard>

<Style x:Key="ImageStyle1" TargetType="{x:Type Image}"/>

</UserControl.Resources>

<Grid>

<Image x:Name="image" Canvas.Left="583.5" Source="/bird.png" HorizontalAlignment="Left" VerticalAlignment="Top" RenderTransformOrigin="0.5,0.5" Width="67" Height="76" Margin="-15,-10,0,0">

<Image.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform/>

<TranslateTransform/>

</TransformGroup>

</Image.RenderTransform>

</Image>

</Grid>

</UserControl>

# Файл SpaceShip.xaml.cs

using System;

using System.Windows.Controls;

using System.Windows.Shapes;

using System.Windows.Threading;

namespace StarWars

{

public partial class SpаceShip : UserControl

{

public double speed = 6;

public bool destroyed = false;

private DispatcherTimer bomdDropTimer;

public SpаceShip()

{

InitializeComponent();

bomdDropTimer = new DispatcherTimer();

bomdDropTimer.Interval = TimeSpan.FromSeconds(BoomParameters.bomdDropTime);

bomdDropTimer.Tick += drop\_tick;

bomdDropTimer.Start();

}

public bool IsHit(Rectangle shot)

{

if ((Canvas.GetLeft(this) <= Canvas.GetLeft(shot) + shot.Width && Canvas.GetLeft(this) + Width >= Canvas.GetLeft(shot)) &&

(Canvas.GetTop(this) <= Canvas.GetBottom(shot) && Canvas.GetTop(this) + Height >= Canvas.GetBottom(shot) - shot.Height))

{

return true;

} else {

return false;

}

}

public bool Cabum()

{

Random r = new Random();

if (BoomParameters.chanceBam > r.Next(11)){

return true;

} else

{

return false;

}

}

private void drop\_tick(object sender, EventArgs e)

{

Canvas gameZone = (Canvas)Parent;

if (!destroyed && gameZone != null)

{

if (Cabum())

{

var bomb = new Bomb();

Canvas.SetTop(bomb, Canvas.GetTop(this) + Height);

Canvas.SetLeft(bomb, Canvas.GetLeft(this) + Width / 2);

Canvas.SetZIndex(bomb, 1);

gameZone.Children.Add(bomb);

MainWindow.bombs.Add(bomb);

}

}

}

}

}

# Результат работы

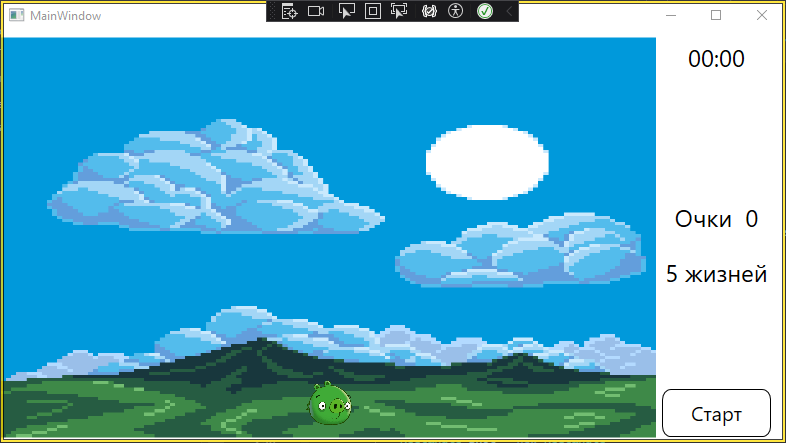


Рис. 1 – скриншот результата

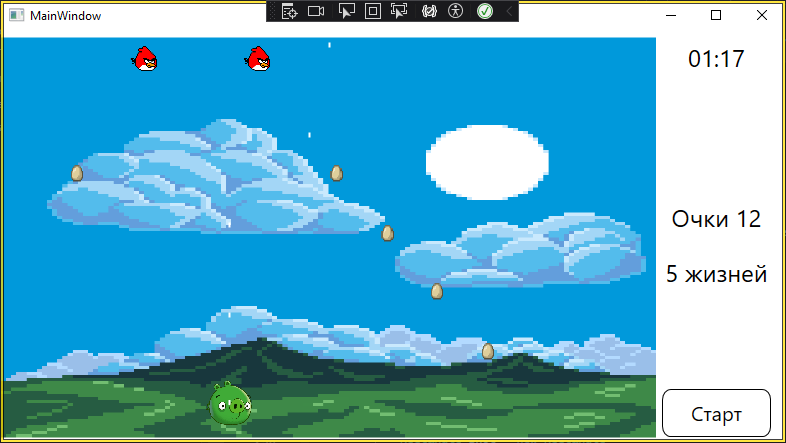


Рис. 2 – скриншот результата

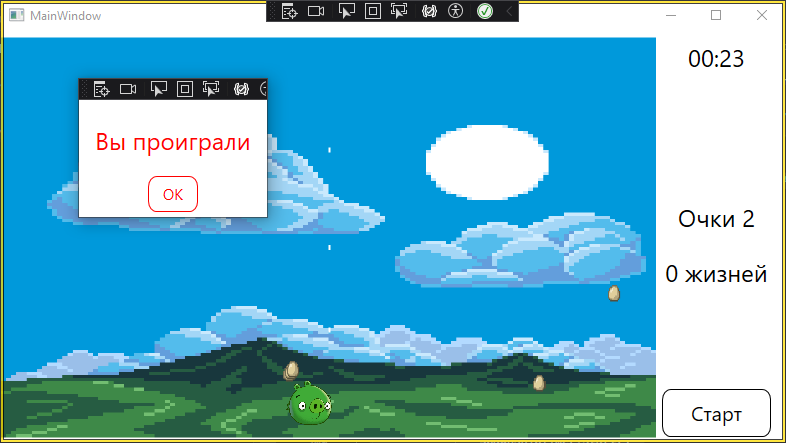


Рис. 3 – скриншот результата