# Меню

CS

public Meny()

{

InitializeComponent();

WindowStartupLocation = System.Windows.WindowStartupLocation.CenterScreen;

}

private void Button1\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Choosing\_the\_car choosing\_The\_Car = new Choosing\_the\_car();

choosing\_The\_Car.ShowDialog();

}

private void Button2\_Click(object sender, RoutedEventArgs e)

{

this.Hide();

Window2 window2 = new Window2();

window2.ShowDialog();

}

XAML

Title="Menu" Height="517" Width="525">

<Window.Background>

<ImageBrush ImageSource="images/fon.png"/>

</Window.Background>

<Window.Resources>

<Style x:Key="FocusVisual">

<Setter Property="Control.Template">

<Setter.Value>

<ControlTemplate>

<Rectangle Margin="2" StrokeDashArray="1 2" Stroke="{DynamicResource {x:Static SystemColors.ControlTextBrushKey}}" SnapsToDevicePixels="true" StrokeThickness="1"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<SolidColorBrush x:Key="Button.Static.Background" Color="#FFDDDDDD"/>

<SolidColorBrush x:Key="Button.Static.Border" Color="#FF707070"/>

<SolidColorBrush x:Key="Button.MouseOver.Background" Color="#FFBEE6FD"/>

<SolidColorBrush x:Key="Button.MouseOver.Border" Color="#FF3C7FB1"/>

<SolidColorBrush x:Key="Button.Pressed.Background" Color="#FFC4E5F6"/>

<SolidColorBrush x:Key="Button.Pressed.Border" Color="#FF2C628B"/>

<SolidColorBrush x:Key="Button.Disabled.Background" Color="#FFF4F4F4"/>

<SolidColorBrush x:Key="Button.Disabled.Border" Color="#FFADB2B5"/>

<SolidColorBrush x:Key="Button.Disabled.Foreground" Color="#FF838383"/>

<Style x:Key="RoundedStyle" TargetType="{x:Type Button}">

<Setter Property="FocusVisualStyle" Value="{StaticResource FocusVisual}"/>

<Setter Property="Background" Value="{StaticResource Button.Static.Background}"/>

<Setter Property="BorderBrush" Value="{StaticResource Button.Static.Border}"/>

<Setter Property="Foreground" Value="{DynamicResource {x:Static SystemColors.ControlTextBrushKey}}"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="HorizontalContentAlignment" Value="Center"/>

<Setter Property="VerticalContentAlignment" Value="Center"/>

<Setter Property="Padding" Value="1"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type Button}">

<Border x:Name="border" CornerRadius="20" Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" SnapsToDevicePixels="true">

<ContentPresenter x:Name="contentPresenter" Focusable="False" HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}" Margin="{TemplateBinding Padding}" RecognizesAccessKey="True" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" VerticalAlignment="{TemplateBinding VerticalContentAlignment}"/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsDefaulted" Value="true">

<Setter Property="BorderBrush" TargetName="border" Value="{DynamicResource {x:Static SystemColors.HighlightBrushKey}}"/>

</Trigger>

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

<Setter Property="Background" TargetName="border" Value="{StaticResource Button.MouseOver.Background}"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.MouseOver.Border}"/>

</Trigger>

<Trigger Property="IsPressed" Value="true">

<Setter Property="Background" TargetName="border" Value="{StaticResource Button.Pressed.Background}"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.Pressed.Border}"/>

</Trigger>

<Trigger Property="IsEnabled" Value="false">

<Setter Property="Background" TargetName="border" Value="{StaticResource Button.Disabled.Background}"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.Disabled.Border}"/>

<Setter Property="TextElement.Foreground" TargetName="contentPresenter" Value="{StaticResource Button.Disabled.Foreground}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="FocusVisual1">

<Setter Property="Control.Template">

<Setter.Value>

<ControlTemplate>

<Rectangle Margin="2" StrokeDashArray="1 2" Stroke="{DynamicResource {x:Static SystemColors.ControlTextBrushKey}}" SnapsToDevicePixels="true" StrokeThickness="1"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<SolidColorBrush x:Key="Button.Static.Background1" Color="#FFDDDDDD"/>

<SolidColorBrush x:Key="Button.Static.Border1" Color="#FF707070"/>

<SolidColorBrush x:Key="Button.MouseOver.Background1" Color="#FFBEE6FD"/>

<SolidColorBrush x:Key="Button.MouseOver.Border1" Color="#FF3C7FB1"/>

<SolidColorBrush x:Key="Button.Pressed.Background1" Color="#FFC4E5F6"/>

<SolidColorBrush x:Key="Button.Pressed.Border1" Color="#FF2C628B"/>

<SolidColorBrush x:Key="Button.Disabled.Background1" Color="#FFF4F4F4"/>

<SolidColorBrush x:Key="Button.Disabled.Border1" Color="#FFADB2B5"/>

<SolidColorBrush x:Key="Button.Disabled.Foreground1" Color="#FF838383"/>

<Style x:Key="ButtonStyle1" TargetType="{x:Type Button}">

<Setter Property="FocusVisualStyle" Value="{StaticResource FocusVisual1}"/>

<Setter Property="Background" Value="{StaticResource Button.Static.Background1}"/>

<Setter Property="BorderBrush" Value="{StaticResource Button.Static.Border1}"/>

<Setter Property="Foreground" Value="{DynamicResource {x:Static SystemColors.ControlTextBrushKey}}"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="HorizontalContentAlignment" Value="Center"/>

<Setter Property="VerticalContentAlignment" Value="Center"/>

<Setter Property="Padding" Value="1"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type Button}">

<Border x:Name="border" CornerRadius="20" Background="{TemplateBinding Background}" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" SnapsToDevicePixels="true">

<ContentPresenter x:Name="contentPresenter" Focusable="False" HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}" Margin="{TemplateBinding Padding}" RecognizesAccessKey="True" SnapsToDevicePixels="{TemplateBinding SnapsToDevicePixels}" VerticalAlignment="{TemplateBinding VerticalContentAlignment}"/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsDefaulted" Value="true">

<Setter Property="BorderBrush" TargetName="border" Value="{DynamicResource {x:Static SystemColors.HighlightBrushKey}}"/>

</Trigger>

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

<Setter Property="Background" TargetName="border" Value="{StaticResource Button.MouseOver.Background1}"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.MouseOver.Border1}"/>

</Trigger>

<Trigger Property="IsPressed" Value="true">

<Setter Property="Background" TargetName="border" Value="{StaticResource Button.Pressed.Background1}"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.Pressed.Border1}"/>

</Trigger>

<Trigger Property="IsEnabled" Value="false">

<Setter Property="Background" TargetName="border" Value="{StaticResource Button.Disabled.Background1}"/>

<Setter Property="BorderBrush" TargetName="border" Value="{StaticResource Button.Disabled.Border1}"/>

<Setter Property="TextElement.Foreground" TargetName="contentPresenter" Value="{StaticResource Button.Disabled.Foreground1}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</Window.Resources>

<Grid Margin="0,0,10,-6">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="163\*"/>

<ColumnDefinition Width="627\*"/>

<ColumnDefinition Width="0\*"/>

</Grid.ColumnDefinitions>

<Button x:Name="Button1" Style="{DynamicResource RoundedStyle}" Content="Single Game" Height="78" HorizontalAlignment="Left" Margin="52,176,0,0" VerticalAlignment="Top" Background="Lime" Foreground="Black" FontWeight="Bold" FontFamily="Arial" Width="198" FontSize="24" Grid.Column="1" Click="Button1\_Click"/>

<Button x:Name="Button2" Style="{DynamicResource ButtonStyle1}" Grid.Column="1" Content="Pair Game" Height="78" HorizontalAlignment="Left" Margin="52,298,0,0" VerticalAlignment="Top" Width="198" Background="Lime" Foreground="Black" FontWeight="Bold" FontFamily="Arial" FontSize="24" Click="Button2\_Click"/>

<Label Grid.Column="1" Content="Race Game" HorizontalAlignment="Left" VerticalAlignment="Top" Height="77" Width="300" FontSize="50" FontFamily="Arial" FontWeight="Bold" Foreground="Black" Margin="19,57,0,0"/>

</Grid>

# Выбор машины

CS

public Choosing\_the\_car()

{

InitializeComponent();

WindowStartupLocation = System.Windows.WindowStartupLocation.CenterScreen;

}

private void Car1\_Click(object sender, RoutedEventArgs e)

{

var image1 = "C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car1.png";

this.Hide();

MainWindow mainWindow = new MainWindow(image1);

mainWindow.ShowDialog();

}

private void Car2\_Click(object sender, RoutedEventArgs e)

{

var image2 = "C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car2.png";

this.Hide();

MainWindow mainWindow = new MainWindow(image2);

mainWindow.ShowDialog();

}

private void Car4\_Click(object sender, RoutedEventArgs e)

{

var image3 = "C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car4.png";

this.Hide();

MainWindow mainWindow = new MainWindow(image3);

mainWindow.ShowDialog();

}

private void Car5\_Click(object sender, RoutedEventArgs e)

{

var image4 = "C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car5.png";

this.Hide();

MainWindow mainWindow = new MainWindow(image4);

mainWindow.ShowDialog();

}

private void Car6\_Click(object sender, RoutedEventArgs e)

{

var image5 = "C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car6.png";

this.Hide();

MainWindow mainWindow = new MainWindow(image5);

mainWindow.ShowDialog();

}

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

{

var image6 = "C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\playerImage.png";

this.Hide();

MainWindow mainWindow = new MainWindow(image6);

mainWindow.ShowDialog();

}

XAML

Title="Choosing\_the\_car" Height="450" Width="800">

<Window.Resources>

<Style TargetType="{x:Type Button}" x:Key="ImageButtonStyle">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type Button}">

<ContentPresenter/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

</Window.Resources>

<Grid>

<Button x:Name="Car1" Style="{StaticResource ImageButtonStyle}" Click="Car1\_Click" HorizontalAlignment="Left" Height="Auto" Margin="50,100,0,0" VerticalAlignment="Top" Width="100" BorderBrush="Transparent" Background="White">

<Image Source="/images/car1.png"/>

</Button>

<Button x:Name="Car2" Style="{StaticResource ImageButtonStyle}" Click="Car2\_Click" HorizontalAlignment="Center" Height="Auto" Margin="0,100,0,0" VerticalAlignment="Top" Width="100" BorderBrush="Transparent" Background="White">

<Image Source="/images/car2.png"/>

</Button>

<Button x:Name="Car4" Style="{StaticResource ImageButtonStyle}" Click="Car4\_Click" HorizontalAlignment="Right" Height="Auto" Margin="0,100,50,0" VerticalAlignment="Top" Width="100" BorderBrush="Transparent" Background="White">

<Image Source="/images/car4.png"/>

</Button>

<Button x:Name="Car5" Style="{StaticResource ImageButtonStyle}" Click="Car5\_Click" HorizontalAlignment="Left" Height="Auto" Margin="50,100,0,20" VerticalAlignment="Bottom" Width="100" BorderBrush="Transparent" Background="White">

<Image Source="/images/car5.png"/>

</Button>

<Button x:Name="Car6" Style="{StaticResource ImageButtonStyle}" Click="Car6\_Click" HorizontalAlignment="Center" Height="Auto" Margin="0,100,0,20" VerticalAlignment="Bottom" Width="100" BorderBrush="Transparent" Background="White">

<Image Source="/images/car6.png"/>

</Button>

<Button x:Name="PlayerImage" Style="{StaticResource ImageButtonStyle}" Click="PlayerImage\_Click" HorizontalAlignment="Right" Height="Auto" Margin="0,100,50,20" VerticalAlignment="Bottom" Width="100" BorderBrush="Transparent" Background="White">

<Image Source="/images/playerImage.png"/>

</Button>

<Label Content="Choose the car you want to play for" HorizontalAlignment="Center" Margin="0,50,0,0" VerticalAlignment="Top" FontSize="20" FontFamily="Cascadia Mono"/>

</Grid>

# Одиночная игра

CS

readonly DispatcherTimer gameTimer = new DispatcherTimer();

readonly List<Rectangle> itemRemover = new List<Rectangle>();

readonly Random rand = new Random();

readonly ImageBrush playerImage = new ImageBrush();

Rect playerHitBox;

int speed = 0;

readonly int playerSpeed = 10;

int carNum;

double score = 0;

bool moveLeft, moveRight, gameOver;

public MainWindow(object sender12)

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterScreen;

myCanvas.Focus();

gameTimer.Tick += GameLoop;

gameTimer.Interval = TimeSpan.FromMilliseconds(20);

StartGame(sender12);

}

private void GameLoop(object sender, EventArgs e)

{

score += .05;

scoreText.Content = "Score " + score.ToString("#.#\nSpeed ") + speed;

playerHitBox = new Rect(Canvas.GetLeft(player), Canvas.GetTop(player), player.Width, player.Height);

if (moveLeft == true && Canvas.GetLeft(player) > 0)

Canvas.SetLeft(player, Canvas.GetLeft(player) - playerSpeed);

else if (moveRight == true && Canvas.GetLeft(player) + 90 < Application.Current.MainWindow.Width)

Canvas.SetLeft(player, Canvas.GetLeft(player) + playerSpeed);

foreach (var x in myCanvas.Children.OfType<Rectangle>())

{

switch ((string)x.Tag)

{

case "roadMarks":

Canvas.SetTop(x, Canvas.GetTop(x) + speed);

if (Canvas.GetTop(x) > 500)

Canvas.SetTop(x, -250);

break;

case "Car":

Canvas.SetTop(x, Canvas.GetTop(x) + speed);

if (Canvas.GetTop(x) > 500)

ChangeCars(x);

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

if (playerHitBox.IntersectsWith(carHitBox))

{

gameTimer.Stop();

scoreText.Content += "\nPress Enter to go Meny";

gameOver = true;

}

break;

default:

break;

}

}

foreach (Rectangle y in itemRemover)

myCanvas.Children.Remove(y);

}

private void OnKeyDown(object sender, KeyEventArgs e)

{

switch (e.Key)

{

case Key.Left:

moveLeft = true;

break;

case Key.Right:

moveRight = true;

break;

case Key.Up:

{

if (speed < 22)

speed++;

}

break;

case Key.Down:

{

if (speed > 2)

speed--;

}

break;

case Key.Enter:

{

if (gameOver)

{

this.Hide();

Meny mainWindow = new Meny();

mainWindow.ShowDialog();

}

}

break;

default:

break;

}

}

private void OnKeyUP(object sender, KeyEventArgs e)

{

switch (e.Key)

{

case Key.Left:

moveLeft = false;

break;

case Key.Right:

moveRight = false;

break;

default:

break;

}

}

private void StartGame(object sender12)

{

speed = 8;

gameTimer.Start();

moveLeft = false;

moveRight = false;

gameOver = false;

score = 0;

scoreText.Content = "Score: 0\nSpeed: 0";

playerImage.ImageSource = new BitmapImage(new Uri((string)sender12));

player.Fill = playerImage;

myCanvas.Background = Brushes.Gray;

foreach (var x in myCanvas.Children.OfType<Rectangle>())

{

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

{

Canvas.SetTop(x, (rand.Next(100, 400) \* -1));

Canvas.SetLeft(x, rand.Next(0, 430));

ChangeCars(x);

}

}

itemRemover.Clear();

}

private void ChangeCars(Rectangle car)

{

carNum = rand.Next(1, 5);

ImageBrush carImage = new ImageBrush();

switch (carNum)

{

case 1:

carImage.ImageSource = new BitmapImage(new Uri("C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car1.png"));

break;

case 2:

carImage.ImageSource = new BitmapImage(new Uri("C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car2.png"));

break;

case 3:

carImage.ImageSource = new BitmapImage(new Uri("C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car4.png"));

break;

case 4:

carImage.ImageSource = new BitmapImage(new Uri("C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car5.png"));

break;

case 5:

carImage.ImageSource = new BitmapImage(new Uri("C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car6.png"));

break;

}

car.Fill = carImage;

Canvas.SetTop(car, (rand.Next(100, 400) \* -1));

Canvas.SetLeft(car, rand.Next(0, 430));

}

XAML

Title="Racing Game " Height="517" Width="525">

<Canvas Background="Gray" Name="myCanvas" Focusable="True" KeyDown="OnKeyDown" KeyUp="OnKeyUP">

<Rectangle x:Name="Track1" Height="500" Width="20" Fill="White" Tag="roadMarks" Canvas.Left="237" Canvas.Top="-250"/>

<Rectangle x:Name="Track2" Height="500" Width="20" Fill="White" Tag="roadMarks" Canvas.Left="237" Canvas.Top="0"/>

<Rectangle x:Name="Track3" Height="500" Width="20" Fill="White" Tag="roadMarks" Canvas.Left="237" Canvas.Top="250"/>

<Rectangle x:Name="Track4" Height="500" Width="20" Fill="White" Tag="roadMarks" Canvas.Left="237" Canvas.Top="500" HorizontalAlignment="Center" VerticalAlignment="Top"/>

<Rectangle Tag="Car" Height="80" Width="55" Fill="Blue" Canvas.Left="90" Canvas.Top="56"/>

<Rectangle Tag="Car" Height="80" Width="55" Fill="Purple" Canvas.Left="381" Canvas.Top="286"/>

<Rectangle Name="player" Height="80" Width="55" Fill="Yellow" Canvas.Left="222" Canvas.Top="374"/>

<Label Name="scoreText" Content="Score: 00" FontSize="18" FontWeight="Bold"/>

</Canvas>

# Парная игра

CS

readonly DispatcherTimer gameTimer = new DispatcherTimer();

readonly List<Rectangle> itemRemover = new List<Rectangle>();

readonly Random rand = new Random();

readonly ImageBrush playerImage = new ImageBrush();

readonly ImageBrush playerImage2 = new ImageBrush();

Rect playerHitBox, playerHitBox2;

int speed = 0;

readonly int playerSpeed = 10;

int carNum;

double score = 0, score2 = 0;

bool moveLeft, moveLeft2, moveRight, moveRight2, moveUP, moveUP2, moveDown, moveDown2, gameOver;

private void Continue\_game\_Click(object sender, RoutedEventArgs e)

{

gameTimer.Start();

Continue\_game.Visibility = Visibility.Hidden;

}

public Window2()

{

InitializeComponent();

WindowStartupLocation = WindowStartupLocation.CenterOwner;

Continue\_game.Visibility = Visibility.Hidden;

myCanvas.Focus();

gameTimer.Tick += GameLoop;

gameTimer.Interval = TimeSpan.FromMilliseconds(20);

StartGame();

}

private void GameLoop(object sender, EventArgs e)

{

score += .05;

score2 += .05;

scoreText.Content = "Score " + score.ToString("#.#\nSpeed ") + speed;

resultText2.Content = "Score " + score2.ToString("#.#\nSpeed ") + speed;

playerHitBox = new Rect(Canvas.GetLeft(player), Canvas.GetTop(player), player.Width, player.Height);

if (moveLeft == true && Canvas.GetLeft(player) > 10)

Canvas.SetLeft(player, Canvas.GetLeft(player) - playerSpeed);

else if (moveRight == true && Canvas.GetLeft(player) + 60 < 540)

Canvas.SetLeft(player, Canvas.GetLeft(player) + playerSpeed);

else if (moveUP == true && Canvas.GetTop(player) > 10)

Canvas.SetTop(player, Canvas.GetTop(player) - playerSpeed);

else if (moveDown == true && Canvas.GetTop(player) + 130 < Application.Current.MainWindow.Height)

Canvas.SetTop(player, Canvas.GetTop(player) + playerSpeed);

playerHitBox2 = new Rect(Canvas.GetLeft(player2), Canvas.GetTop(player2), player2.Width, player2.Height);

if (moveLeft2 == true && Canvas.GetLeft(player2) > 560)

Canvas.SetLeft(player2, Canvas.GetLeft(player2) - playerSpeed);

else if (moveRight2 == true && Canvas.GetLeft(player2) + 80 < myCanvas.Width)

Canvas.SetLeft(player2, Canvas.GetLeft(player2) + playerSpeed);

else if (moveUP2 == true && Canvas.GetTop(player2) > 10)

Canvas.SetTop(player2, Canvas.GetTop(player2) - playerSpeed);

else if (moveDown2 == true && Canvas.GetTop(player2) + 130 < Application.Current.MainWindow.Height)

Canvas.SetTop(player2, Canvas.GetTop(player2) + playerSpeed);

foreach (var x in myCanvas.Children.OfType<Rectangle>())

{

switch ((string)x.Tag)

{

case "roadMarks":

Canvas.SetTop(x, Canvas.GetTop(x) + speed);

if (Canvas.GetTop(x) > 500)

Canvas.SetTop(x, -250);

break;

case "Car":

Canvas.SetTop(x, Canvas.GetTop(x) + speed);

if (Canvas.GetTop(x) > 500)

ChangeCars(x);

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

if (playerHitBox.IntersectsWith(carHitBox) && playerHitBox2.IntersectsWith(carHitBox))

{

gameTimer.Stop();

scoreText.Content += "\n\t\tDRAW GAME.\nPress Enter to go Meny";

resultText2.Content += "\n\t\tDRAW GAME.\nPress Enter to go Meny";

gameOver = true;

}

else if (playerHitBox.IntersectsWith(carHitBox))

{

gameTimer.Stop();

scoreText.Content += "\n\t\tYOU LOSE.\nPress Enter to go Meny";

resultText2.Content += "\n\t\tYOU WIN.\nPress Enter to go Meny";

gameOver = true;

}

else if (playerHitBox2.IntersectsWith(carHitBox))

{

gameTimer.Stop();

scoreText.Content += "\n\t\tYOU WIN.\nPress Enter to go Meny";

resultText2.Content += "\n\t\tYOU LOSE.\nPress Enter to go Meny";

gameOver = true;

}

break;

default:

break;

}

}

foreach (Rectangle y in itemRemover)

myCanvas.Children.Remove(y);

if ((score >= 10) && (score < 20) && (score2 >= 10) && (score2 < 20))

speed = 10;

else if ((score >= 20) && (score < 30) && (score2 >= 20) && (score2 < 30))

speed = 12;

else if ((score >= 30) && (score < 40) && (score2 >= 30) && (score2 < 40))

speed = 14;

else if ((score >= 40) && (score < 50) && (score2 >= 40) && (score2 < 50))

speed = 16;

else if ((score >= 50) && (score < 60) && (score2 >= 50) && (score2 < 60))

speed = 18;

else if ((score >= 60) && (score < 70) && (score2 >= 60) && (score2 < 70))

speed = 20;

else if ((score >= 70) && (score < 80) && (score2 >= 70) && (score2 < 80))

speed = 22;

}

private void OnKeyDown(object sender, KeyEventArgs e)

{

switch (e.Key)

{

case Key.A:

moveLeft = true;

break;

case Key.D:

moveRight = true;

break;

case Key.W:

moveUP = true;

break;

case Key.S:

moveDown = true;

break;

case Key.Left:

moveLeft2 = true;

break;

case Key.Right:

moveRight2 = true;

break;

case Key.Up:

moveUP2 = true;

break;

case Key.Down:

moveDown2 = true;

break;

case Key.Enter:

{

if (gameOver)

{

this.Hide();

Meny mainWindow = new Meny();

mainWindow.ShowDialog();

}

}

break;

case Key.Escape:

{

gameTimer.Stop();

Continue\_game.Visibility = Visibility.Visible;

}

break;

default:

break;

}

}

private void OnKeyUP(object sender, KeyEventArgs e)

{

switch (e.Key)

{

case Key.A:

moveLeft = false;

break;

case Key.D:

moveRight = false;

break;

case Key.W:

moveUP = false;

break;

case Key.S:

moveDown = false;

break;

case Key.Left:

moveLeft2 = false;

break;

case Key.Right:

moveRight2 = false;

break;

case Key.Up:

moveUP2 = false;

break;

case Key.Down:

moveDown2 = false;

break;

default:

break;

}

}

private void StartGame()

{

speed = 4;

gameTimer.Start();

moveLeft = false;

moveLeft2 = false;

moveRight = false;

moveRight2 = false;

moveUP = false;

moveUP2 = false;

moveDown = false;

moveDown2 = false;

gameOver = false;

score = 0;

score2 = 0;

scoreText.Content = "Score: 0\nSpeed: 0";

resultText2.Content = "Score: 0\nSpeed: 0";

playerImage.ImageSource = new BitmapImage(new Uri("C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\playerImage.png"));

playerImage2.ImageSource = new BitmapImage(new Uri("C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car6.png"));

player.Fill = playerImage;

player2.Fill = playerImage2;

myCanvas.Background = Brushes.Gray;

foreach (var x in myCanvas.Children.OfType<Rectangle>())

{

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

{

Canvas.SetTop(x, (rand.Next(100, 970) \* -1));

Canvas.SetLeft(x, rand.Next(0, 1000));

ChangeCars(x);

}

}

itemRemover.Clear();

}

private void ChangeCars(Rectangle car)

{

carNum = rand.Next(1, 5);

ImageBrush carImage = new ImageBrush();

switch (carNum)

{

case 1:

carImage.ImageSource = new BitmapImage(new Uri("C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car1.png"));

break;

case 2:

carImage.ImageSource = new BitmapImage(new Uri("C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car2.png"));

break;

case 3:

carImage.ImageSource = new BitmapImage(new Uri("C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car4.png"));

break;

case 4:

carImage.ImageSource = new BitmapImage(new Uri("C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car5.png"));

break;

case 5:

carImage.ImageSource = new BitmapImage(new Uri("C:\\Users\\Бакыт\\OneDrive\\Документы\\ИСП-308\\Гонки\\race\\race\\images\\car6.png"));

break;

}

car.Fill = carImage;

Canvas.SetTop(car, (rand.Next(100, 970) \* -1));

Canvas.SetLeft(car, rand.Next(0, 1000));

}

XAML

Title="Window2" Height="520" Width="1100">

<Grid>

<Canvas Background="Gray" Name="myCanvas" Focusable="True" KeyDown="OnKeyDown" KeyUp="OnKeyUP" HorizontalAlignment="Left" Width="1100">

<Rectangle Height="500" Width="20" Fill="White" Tag="roadMarks" Canvas.Left="540" Canvas.Top="-250"/>

<Rectangle Height="500" Width="20" Fill="White" Tag="roadMarks" Canvas.Left="540" Canvas.Top="0"/>

<Rectangle Height="500" Width="20" Fill="White" Tag="roadMarks" Canvas.Left="540" Canvas.Top="250"/>

<Rectangle Height="500" Width="20" Fill="White" Tag="roadMarks" Canvas.Left="540" Canvas.Top="500"/>

<Rectangle Tag="Car" Height="80" Width="55" Fill="Blue" Canvas.Left="90" Canvas.Top="60"/>

<Rectangle Tag="Car" Height="80" Width="55" Fill="Purple" Canvas.Left="380" Canvas.Top="280"/>

<Rectangle Name="player" Height="80" Width="55" Fill="Yellow" Canvas.Left="244" Canvas.Top="380" HorizontalAlignment="Left" VerticalAlignment="Center"/>

<Label x:Name="scoreText" Content="Score: 00" FontSize="18" FontWeight="Bold"/>

<Rectangle Tag="Car" Height="80" Width="55" Fill="Blue" Canvas.Left="650" Canvas.Top="50"/>

<Rectangle Tag="Car" Height="80" Width="55" Fill="Purple" Canvas.Left="940" Canvas.Top="280"/>

<Rectangle Name="player2" Height="80" Width="55" Fill="Yellow" Canvas.Left="800" Canvas.Top="380"/>

<Label x:Name="resultText2" Content="Score: 00" FontSize="18" FontWeight="Bold" Margin="560,0,0,0"/>

<Button x:Name="Continue\_game" Content="Continue" Height="40" Width="200" FontSize="20" FontFamily="Cascadia Mono" Background="White" BorderBrush="Black" Canvas.Left="450" Canvas.Top="220" Click="Continue\_game\_Click"/>

</Canvas>

</Grid>