Lab 1

Marchenko Anton, AS-36

Controll two metro cars using Speed, Distance, and Brake

In this work we are design a virtual loop with 1000 pixels on which two metro cars.

The source code:

Code below recalculate speed and coordinates each cars.

MainViewModel.cs

```
using System;
using System.Threading.Tasks;
using GalaSoft.MvvmLight;
using GalaSoft.MvvmLight.Command;

namespace WpfApplication1.ViewModel
{
    public class MainViewModel : ViewModelBase
    {
           private double _x0 = 237;
           private double _y0 = 202;
           private double _radius = 200;


              private readonly Random _random = new Random();
              public MainViewModel()
        {
                     RunCarsCommand = new RelayCommand(RunCarsMethod);
                     RedCar = new Car {Angle = 0, CoordX = _x0 + _radius*Math.Cos(0),
CoordY = _y0 + _radius*Math.Sin(0), Speed = 0.01};
                     GreenCar = new Car {Angle = Math.PI, CoordX = _x0 +
_radius*Math.Cos(Math.PI), CoordY = _y0 + _radius*Math.Sin(Math.PI), Speed = 0.01};
              }

           public Car RedCar { get; set; }
           public Car GreenCar { get; set; }

              public RelayCommand RunCarsCommand
              {
                     get;
                     private set;
              }

              private void RunCarsMethod()
              {

                     Task.Run(async () =>
                     {
                            while (true)
                            {
                                   await Task.Delay(1);
                                   RecalculateCarsState(RedCar);
                                   RecalculateCarsState(GreenCar);
                                   RecalculateDistance();

                     if ( (Math.Abs(RedCar.Angle  - GreenCar.Angle ) % (2 * Math.PI)) <
0.05)
                                   {
                                          break;
                                   }
                            }
```

```
                });
            }

           private void RecalculateCarsState(Car car)
           {
                  car.CoordX = _x0 + _radius*Math.Cos(car.Angle);
                  car.CoordY = _y0 + _radius*Math.Sin(car.Angle);

                    var r = _random.Next(3);

                    switch (r)
                    {
                           case 0:
                                  car.Speed -= 0.001;
                                  break;
                           case 2:
                                  car.Speed += 0.001;
                                  break;
                    }

             if (car.Speed <= 0)
                 car.Speed = 0;
                     car.Angle += car.Speed;
              }

        private void RecalculateDistance()
        {
            RedCar.Distance += (int)((RedCar.Speed) / (2 * Math.PI) * 1000 -
(GreenCar.Speed) / (2 * Math.PI) * 1000);
            GreenCar.Distance += (int)((GreenCar.Speed) / (2 * Math.PI) * 1000 -
(RedCar.Speed) / (2 * Math.PI) * 1000);
        }

       }
//Class car,  that store information about metro cars.
       public class Car : ViewModelBase
       {
              private double _coordX;
              private double _coordY;
         private double _speed;
         private int _distance = 500;

              public double Speed
         {
             get { return _speed; }
             set
             {
                 _speed = value;
                 RaisePropertyChanged();
             }
         }

         public int Distance
         {
             get { return _distance; }
             set
             {
                 _distance = value;
                 RaisePropertyChanged();
             }
         }

         public double Angle { get; set; }

              public double CoordX
```

```
		{
			get { return _coordX; }
			set
			{
				_coordX = value;
				RaisePropertyChanged();
			}
		}

		public double CoordY
		{
			get { return _coordY; }
			set
			{
				_coordY = value;
				RaisePropertyChanged();
			}
		}
	}
}
```

Code below drowing the animation with moving cars by ellipse and displaying Distance and speed each of cars.

MainWindow.xaml

```
<Window x:Class="WpfApplication1.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:WpfApplication1"
        mc:Ignorable="d"
        Title="MainWindow" Height="350" Width="525"
             DataContext="{Binding Main, Source={StaticResource Locator}}">
    <Grid>
             <Canvas>
                    <Rectangle Width="30"
                                    Height="30"
                                    Canvas.Left="{Binding GreenCar.CoordX, Mode=OneWay}"
                                    Canvas.Top="{Binding GreenCar.CoordY, Mode=OneWay}"
                                    Fill="Green"></Rectangle>

                    <Rectangle Width="30"
                                    Height="30"
                                    Canvas.Left="{Binding RedCar.CoordX, Mode=OneWay}"
                                    Canvas.Top="{Binding RedCar.CoordY, Mode=OneWay}"
                                    Fill="Red"></Rectangle>

                    <Ellipse Width="400"
                                  Height="400"
                                  Canvas.Top="15"
                                  Canvas.Left="50"
                                  Stroke="Aqua"
                                  StrokeThickness="4"></Ellipse>
                    <Button Content="tap"
                           Width="50"
                           Height="50"
                                 Canvas.Top="202"
                                 Canvas.Left="237"
                           Command="{Binding RunCarsCommand, Mode=OneWay}"></Button>
             </Canvas>
         <StackPanel HorizontalAlignment="Right">
             <TextBlock Text="Green Train:"
```

```
                    Foreground="Green">
                <Run Text="Distance: " Foreground="Black"/><Run Text="{Binding
GreenCar.Distance}" Foreground="Black"/>
                <Run Text="Speed: " Foreground="Black"/><Run Text="{Binding
GreenCar.Speed, Converter={StaticResource SpeedConverter}}" Foreground="Black"/>
            </TextBlock>
            <TextBlock Text="Red Train:"
                    Foreground="Red">
                <Run Text="Distance" Foreground="Black"/><Run Text="{Binding
RedCar.Distance}" Foreground="Black"/>
                <Run Text="Speed: " Foreground="Black"/><Run Text="{Binding RedCar.Speed,
Converter={StaticResource SpeedConverter}}" Foreground="Black"/>
            </TextBlock>
        </StackPanel>

    </Grid>
</Window>
```