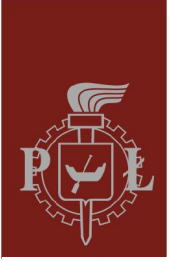


Politechnika Łódzka
Instytut Elektroniki

Programowanie telefonów z Windows Phone 7, cz. 5

Piotr M. Szczypiński
Instytut Elektroniki Politechniki Łódzkiej
http://www.eletel.p.lodz.pl/pms/
piotr.szczypinski@p.lodz.pl
Budynek B9, II piętro, pokój 217A

Materiał na dziś

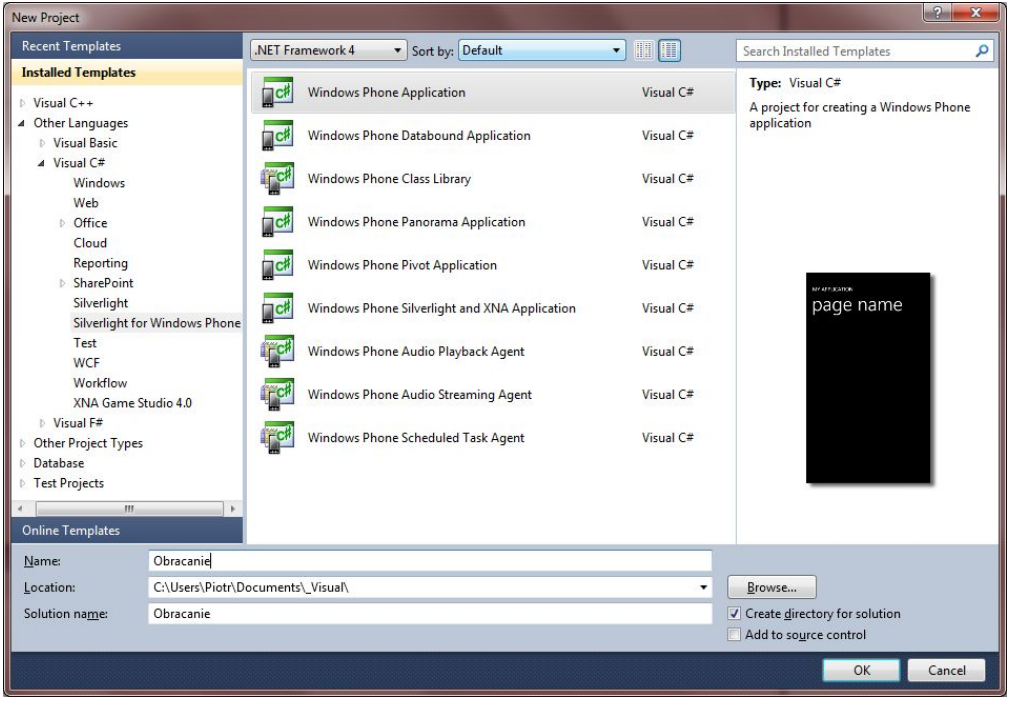
Silverlight:

1. Transformacje – namiastka 3D
2. Animacje – samoczynna zmiana właściwości
3. Grafika wektorowa

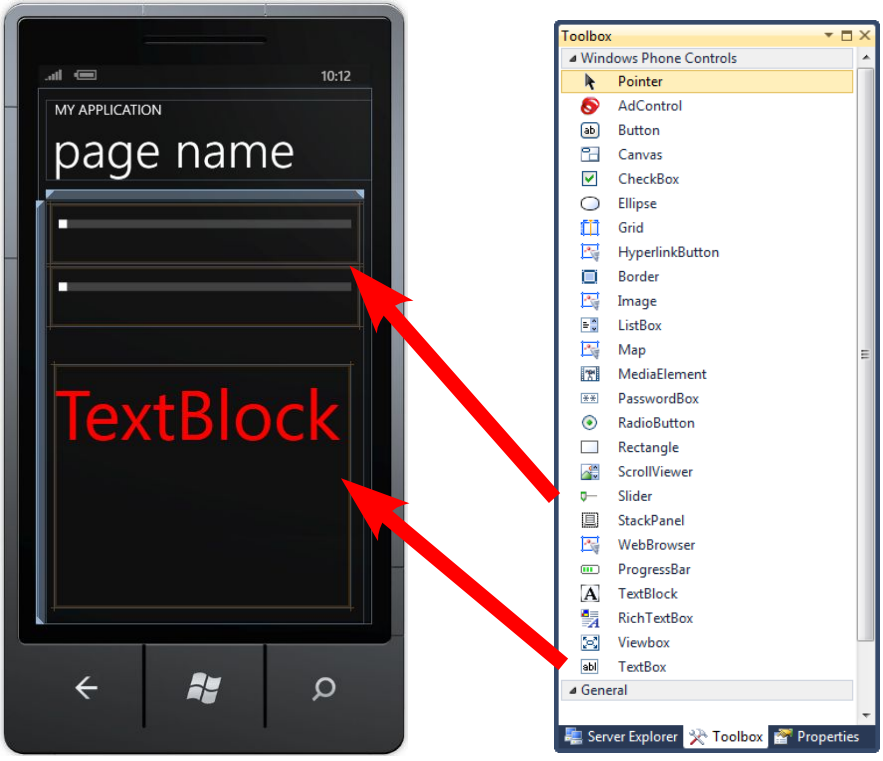
XNA:

4. Transformacje – jeszcze raz inaczej

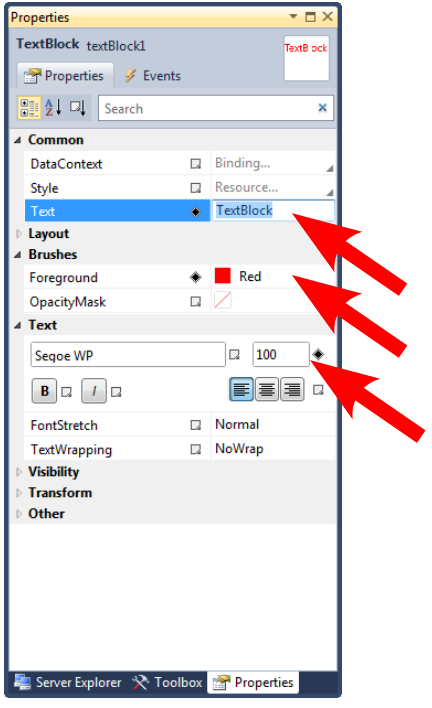
Projekt Obracanie

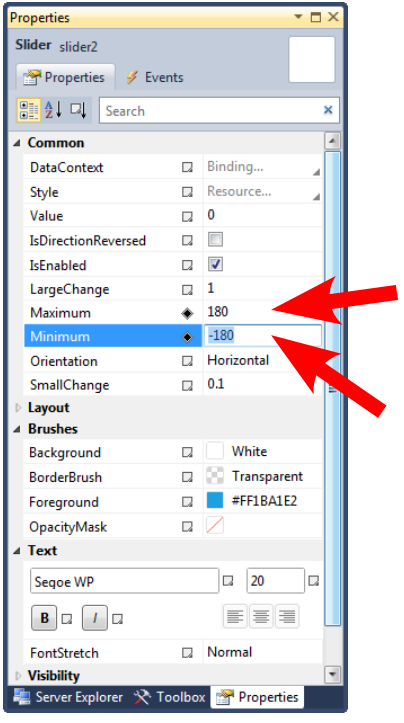


Obracanie – GUI



Obracanie – GUI





Obracanie – właściwość

W pliku MainPage.xaml definiujemy właściwość Projection.

Zamieniamy:

```
<TextBlock (. . .) />
```

na:

```
<TextBlock (. . .) >
    <TextBlock.Projection>
        <PlaneProjection x:Name="planeProjection" />
    </TextBlock.Projection>
</TextBlock>
```

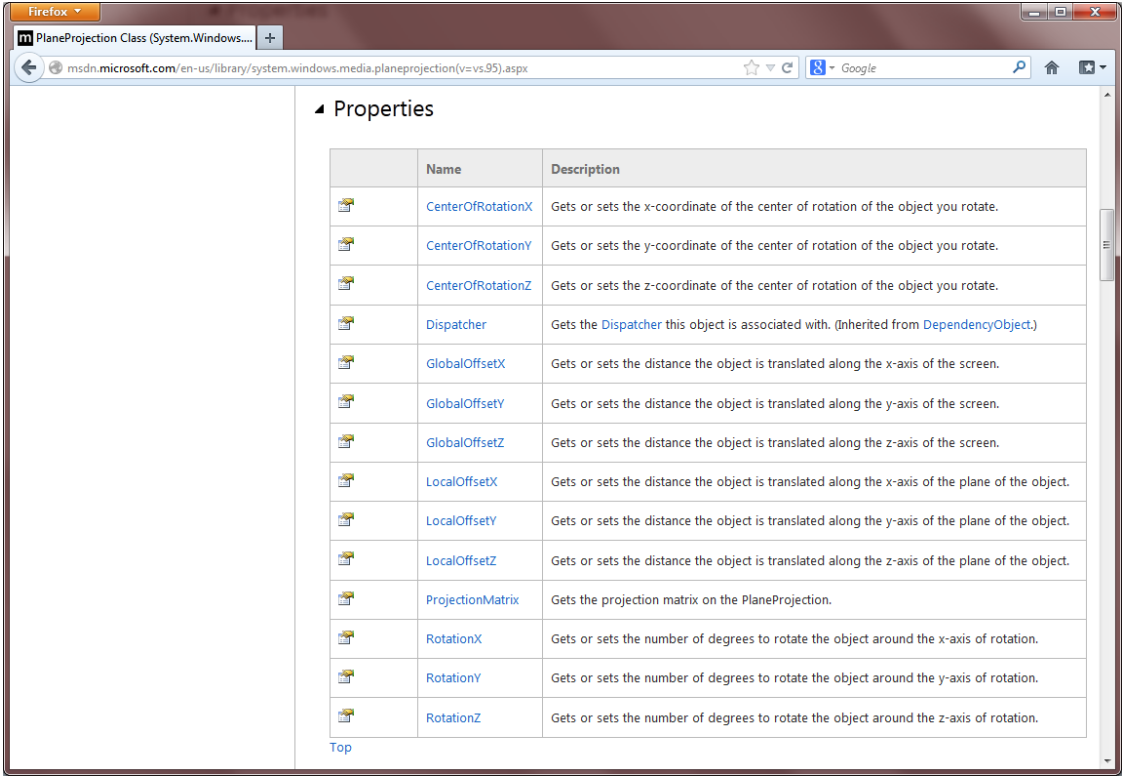
Obracanie – obsługa suwaków

W pliku MainPage.xaml.cs, po dwukliku na suwakach, definiujemy funkcje:

```
private void slider1_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e)
{
    textBlock1.Projection.SetValue(PlaneProjection.RotationXProperty, slider1.Value);
}
```

```
private void slider2_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e)
{
    textBlock1.Projection.SetValue(PlaneProjection.RotationYProperty, slider2.Value);
}
```

Inne właściwości klasy *PlaneProjection*

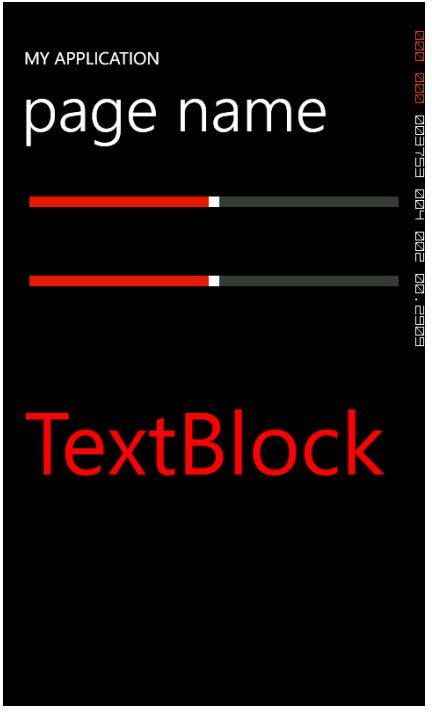


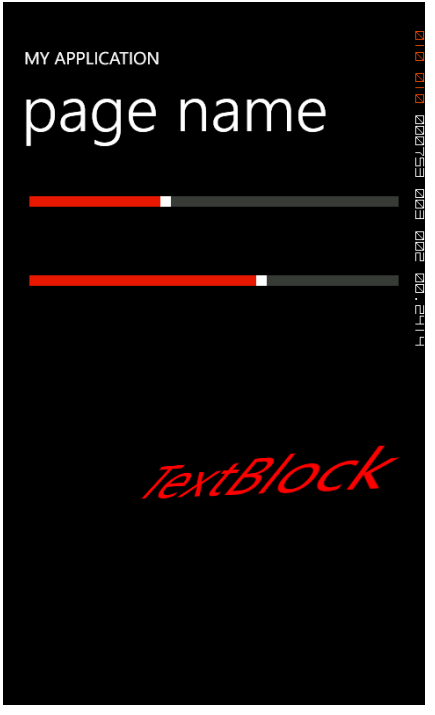
▲ Properties

	Name	Description
	CenterOfRotationX	Gets or sets the x-coordinate of the center of rotation of the object you rotate.
	CenterOfRotationY	Gets or sets the y-coordinate of the center of rotation of the object you rotate.
	CenterOfRotationZ	Gets or sets the z-coordinate of the center of rotation of the object you rotate.
	Dispatcher	Gets the Dispatcher this object is associated with. (Inherited from DependencyObject.)
	GlobalOffsetX	Gets or sets the distance the object is translated along the x-axis of the screen.
	GlobalOffsetY	Gets or sets the distance the object is translated along the y-axis of the screen.
	GlobalOffsetZ	Gets or sets the distance the object is translated along the z-axis of the screen.
	LocalOffsetX	Gets or sets the distance the object is translated along the x-axis of the plane of the object.
	LocalOffsetY	Gets or sets the distance the object is translated along the y-axis of the plane of the object.
	LocalOffsetZ	Gets or sets the distance the object is translated along the z-axis of the plane of the object.
	ProjectionMatrix	Gets the projection matrix on the PlaneProjection.
	RotationX	Gets or sets the number of degrees to rotate the object around the x-axis of rotation.
	RotationY	Gets or sets the number of degrees to rotate the object around the y-axis of rotation.
	RotationZ	Gets or sets the number of degrees to rotate the object around the z-axis of rotation.

Top

Obracanie


MY APPLICATION
page name
TextBlock

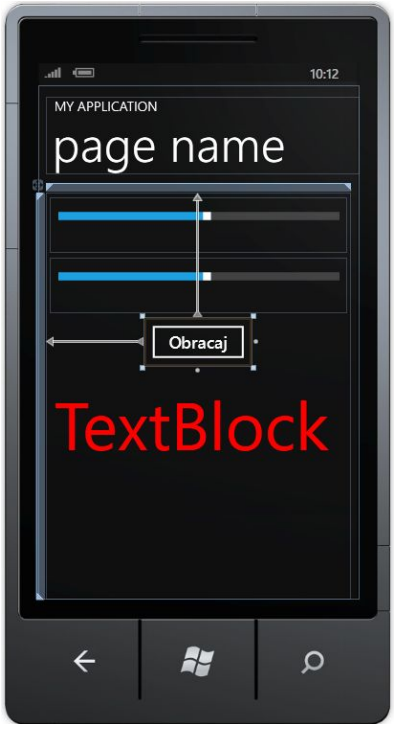

MY APPLICATION
page name
TextBlock

Obracanie – animacja, automatyzacja

Nad definicją LayoutRoot w MainPage.xaml dodajemy zasób Storyboard:

```
<phone:PhoneApplicationPage.Resources>
    <Storyboard x:Name="ObracajX">
        <DoubleAnimation Storyboard.TargetName="planeProjection"
                         Storyboard.TargetProperty="RotationX"
                         From="0" To="360" Duration="0:0:5" />
    </Storyboard>
</phone:PhoneApplicationPage.Resources>
```

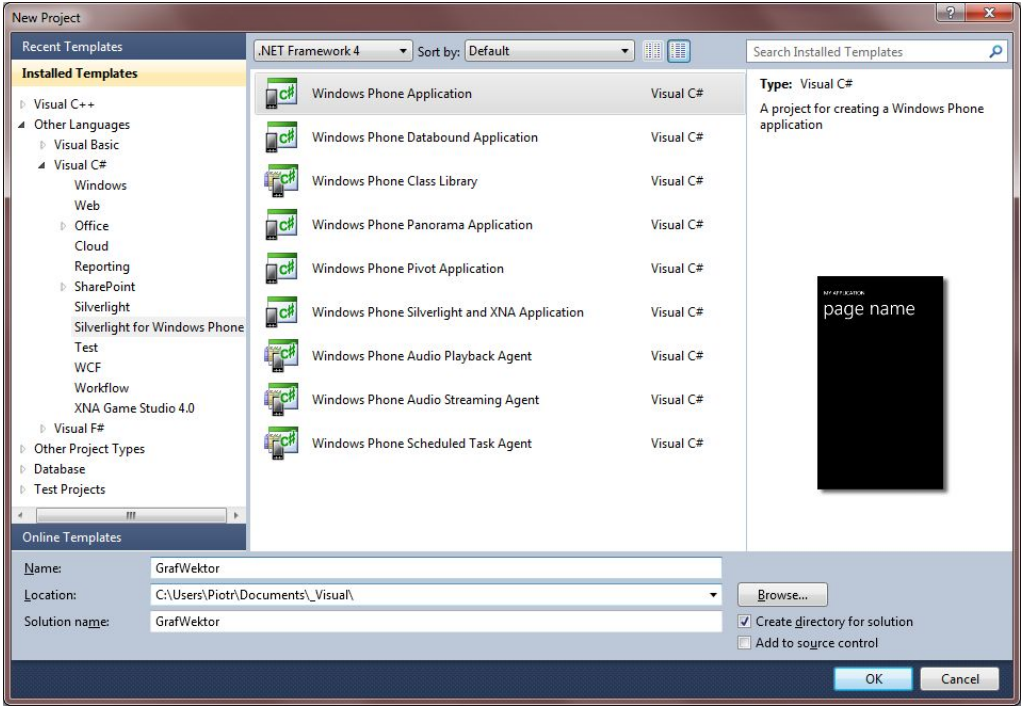
Obracanie – animacja, automatyzacja



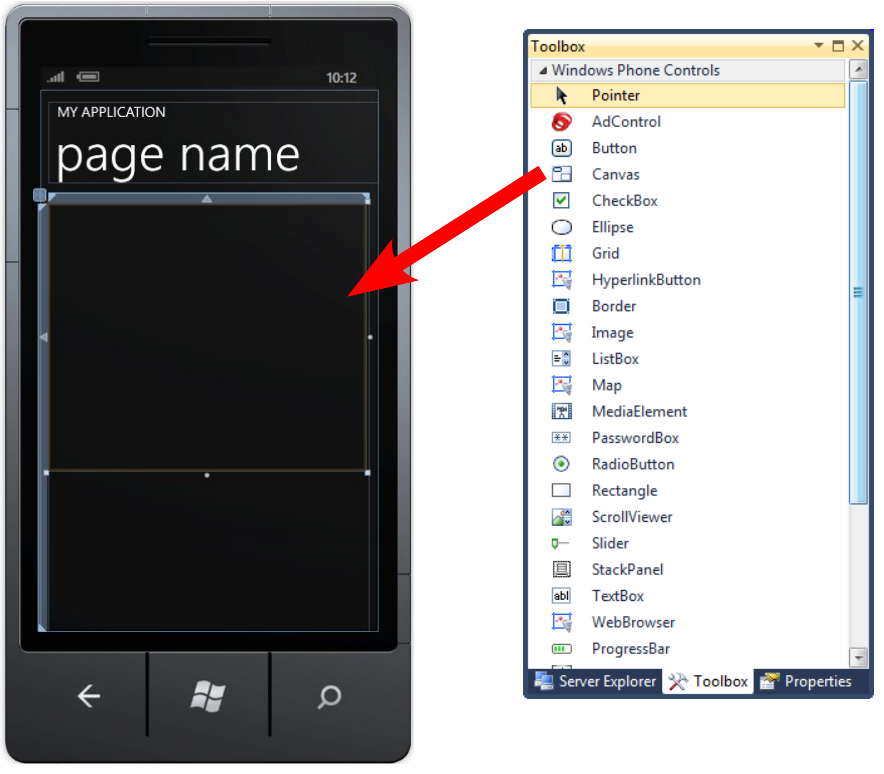

```
private void button1_Click(object sender,
                           RoutedEventArgs e)
{
    ObracajX.Begin();
}
```

Po wciśnięciu guzika animowany jest obrót tekstu wokół osi X

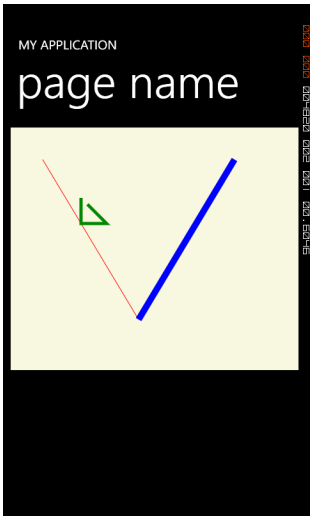
Grafika wektorowa w Silverlight



Grafika wektorowa w Silverlight



Grafika statyczna



W pliku MainPage.xaml dodajemy do płótna elementy graficzne.

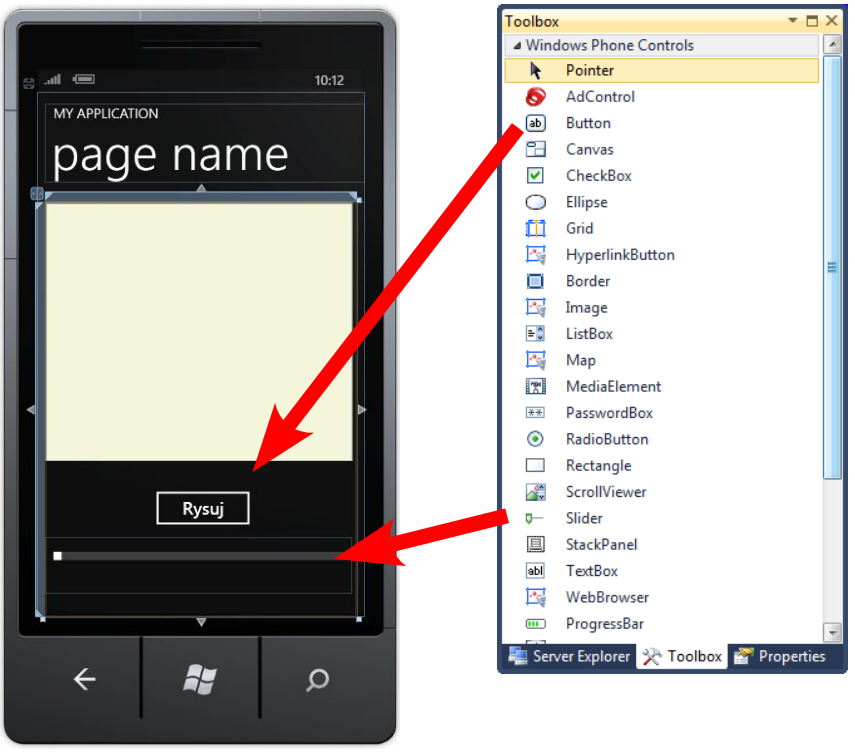
Zamieniamy:

```
<Canvas (. . .) />
```

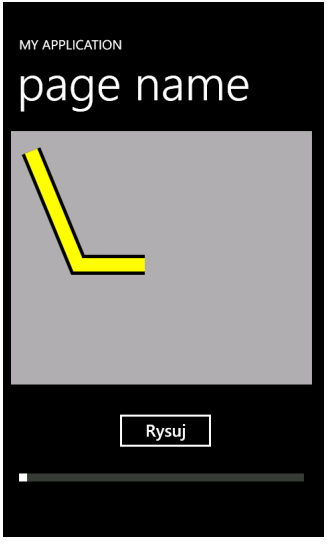
na:

```
<Canvas Background="Beige" (. . .) >
    <Line X1="50" Y1="50" X2="200" Y2="300" Stroke="Red" />
    <Line X1="200" Y1="300" X2="350" Y2="50"
          Stroke="Blue" StrokeThickness="10" />
    <Polyline Points="110 110, 110 150, 150 150, 120 120"
              Stroke="Green" StrokeThickness="5" />
    <Polyline Points=" 50 100, 300 200, 100 300" Stroke="HotPink"
              StrokeThickness="30" StrokeStartLineCap="Round"
              StrokeEndLineCap="Round" StrokeLineJoin="Round" />
</Canvas>
```

Grafika dynamiczna



Grafika dynamiczna – linie



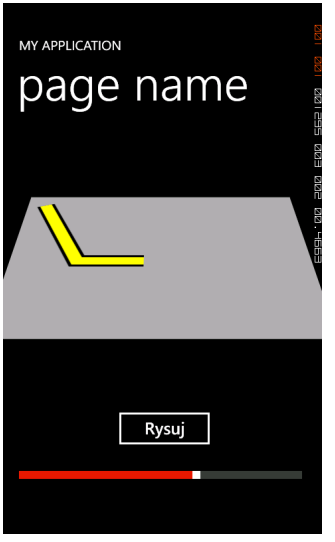

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    SolidColorBrush spod = new SolidColorBrush();
    spod.Color = Colors.Black;
    Polyline droga_spod = new Polyline();
    droga_spod.Points.Add(new Point(30, 30));
    droga_spod.Points.Add(new Point(100, 200));
    droga_spod.Points.Add(new Point(200, 200));
    droga_spod.Stroke = spod;
    droga_spod.StrokeThickness = 30.0;
    canvas1.Children.Add(droga_spod);

    SolidColorBrush wierzch = new SolidColorBrush();
    wierzch.Color = Colors.Yellow;
    Polyline droga_wierzch = new Polyline();
    droga_wierzch.Points.Add(new Point(30, 30));
    droga_wierzch.Points.Add(new Point(100, 200));
    droga_wierzch.Points.Add(new Point(200, 200));
    droga_wierzch.Stroke = wierzch;
    droga_wierzch.StrokeThickness = 20.0;
    canvas1.Children.Add(droga_wierzch);
}
```

Grafika dynamiczna

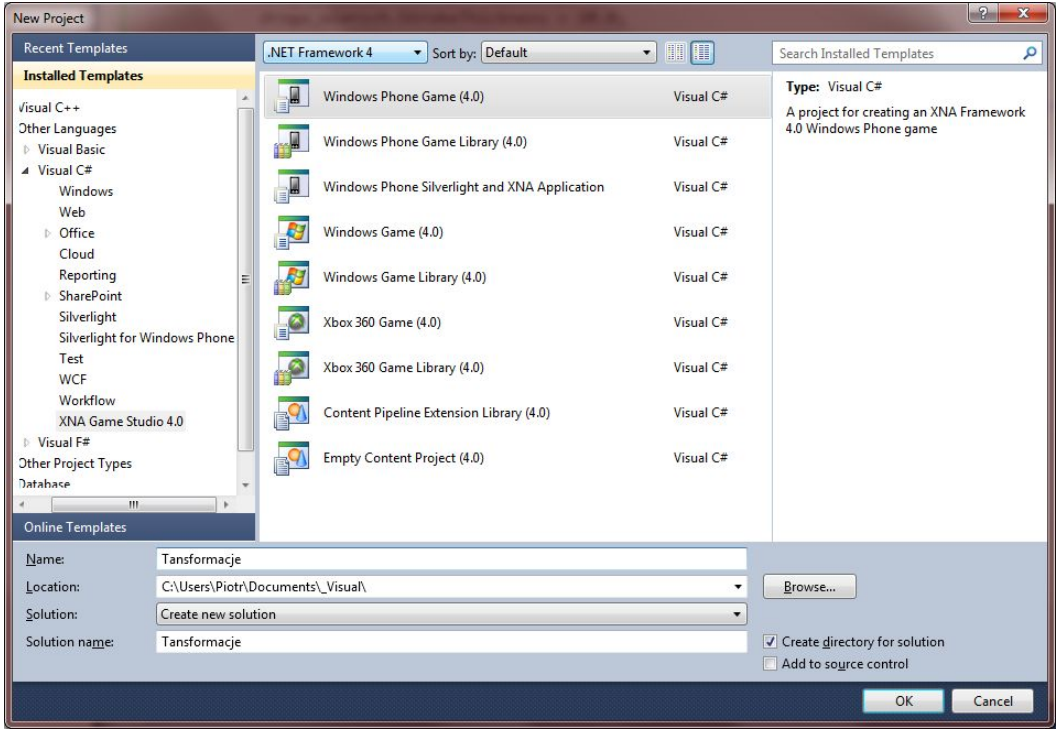
```
<Canvas (. . .) >
    <Canvas.Projection>

        <PlaneProjection x:Name="obracaj" />
    </Canvas.Projection>
</Canvas>
```




```
private void slider1_ValueChanged(object sender,
                                  RoutedPropertyChangedEventArgs<double> e)
{
    canvas1.Projection.SetValue(PlaneProjection.RotationXProperty, -slider1.Value);
}
```

XNA


New Project
Recent Templates
Installed Templates
Visual C++
Other Languages
Visual Basic
Visual C#
Windows
Web
Office
Cloud
Reporting
SharePoint
Silverlight
Silverlight for Windows Phone
Test
WCF
Workflow
XNA Game Studio 4.0
Visual F#
Other Project Types
Database
Online Templates
.NET Framework 4
Sort by: Default
Search Installed Templates
Windows Phone Game (4.0) Visual C#
Windows Phone Game Library (4.0) Visual C#
Windows Phone Silverlight and XNA Application Visual C#
Windows Game (4.0) Visual C#
Windows Game Library (4.0) Visual C#
Xbox 360 Game (4.0) Visual C#
Xbox 360 Game Library (4.0) Visual C#
Content Pipeline Extension Library (4.0) Visual C#
Empty Content Project (4.0) Visual C#
Type: Visual C#
A project for creating an XNA Framework 4.0 Windows Phone game
Name: Tansformacje
Location: C:\Users\Piotr\Documents_Visual\
Browse...
Solution: Create new solution
Solution name: Tansformacje
Create directory for solution
Add to source control
OK
Cancel

XNA – proste transformacje

Funkcja *spriteBatch.Draw* umożliwia podanie argumentów obrotu i przesunięcia:

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);
    spriteBatch.Begin();
    spriteBatch.Draw(texture, texturePosition, null, Color.White,
                     textureRotation, textureCenter, 1, SpriteEffects.None, 0);
    spriteBatch.End();
    base.Draw(gameTime);
}
```

XNA – macierz transformacji

Generowanie macierzy za pomocą funkcji:

```
Matrix matrix1 = Matrix.CreateTranslation(xOffset, yOffset, 0);
Matrix matrix2 = Matrix.CreateScale(xScale, yScale, 1);
Matrix matrix3 = Matrix.CreateRotationZ(radians);
```

Składanie transformacji:

```
Matrix matrix = matrix1 * matrix2 * matrix3;
```

Operowanie na elementach macierzy:

```
matrix.Mij
i = 0, 1, 2, 3
j = 0, 1, 2, 3
```

Renderowanie grafiki z wykorzystaniem transformacji:

```
spriteBatch.Begin(SpriteSortMode.Deferred, null, null, null, null, null, matrix);
...
spriteBatch.End();
```

...i na tym koniec
piątej części wykładu