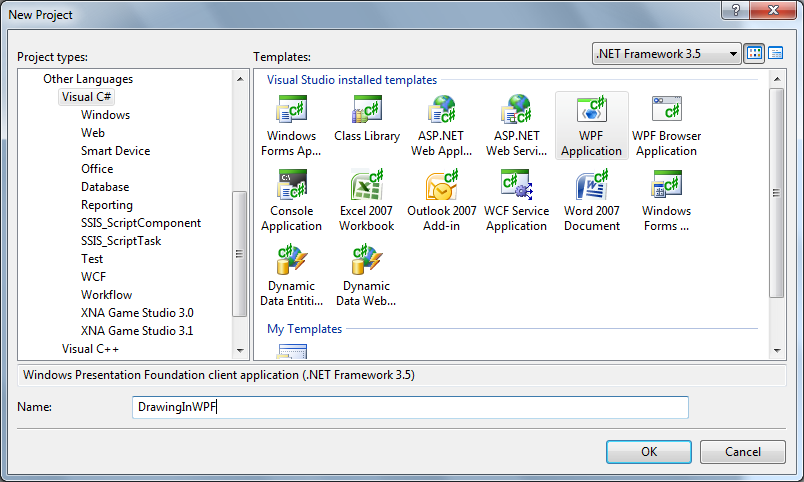
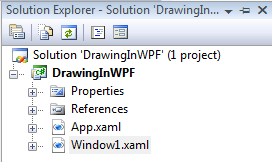
# Crtanje u WPF-u

### Novi WPF projekat





## DrawingVisual

DrawingVisual is a lightweight drawing class that is used to render shapes, images, or text. This class is considered lightweight because it does not provide layout, input, focus, or event handling, which improves its performance. For this reason, drawings are ideal for backgrounds and clip art.

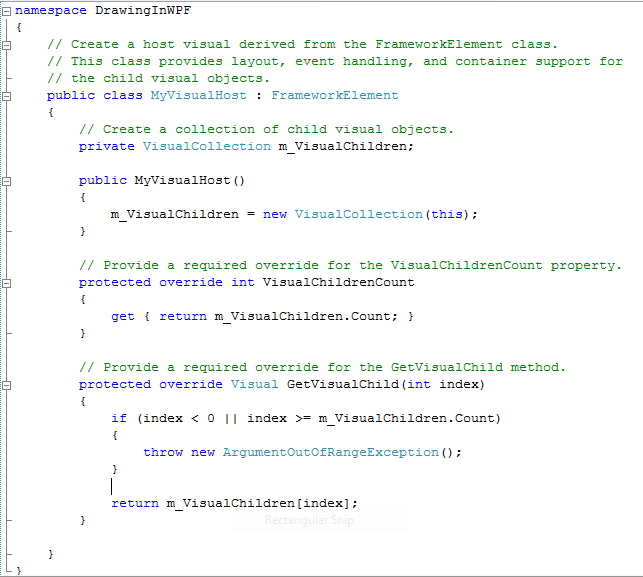
In order to use DrawingVisual objects, you need to create a host container for the objects. The host container object must be derived from the [FrameworkElement](http://msdn.microsoft.com/en-us/library/system.windows.frameworkelement.aspx) class, which provides the layout and event handling support that the DrawingVisual class does not support. The host container object does not display any visual properties, since its main purpose is to contain child objects.

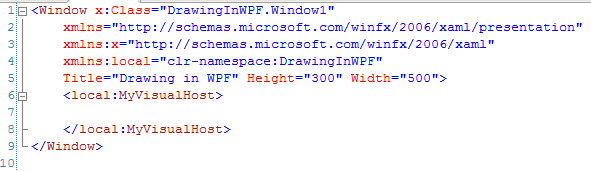
### FrameworkElement

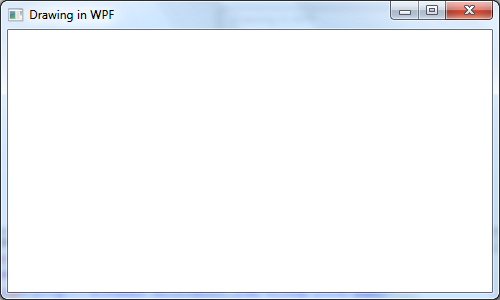
Provides a WPF framework-level set of properties, events, and methods for Windows Presentation Foundation (WPF) elements. This class represents the provided WPF framework-level implementation that is built on the WPF core-level APIs that are defined by [UIElement](http://msdn.microsoft.com/en-us/library/system.windows.uielement.aspx).

### MyDrawingHost

Derived from FrameworkElement. Overrides **VisualChilderenCount** property and **GetVisualChild** method.

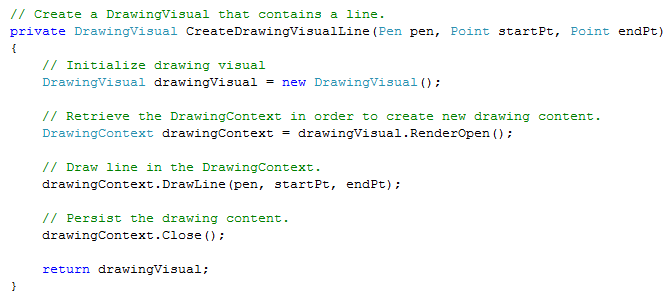


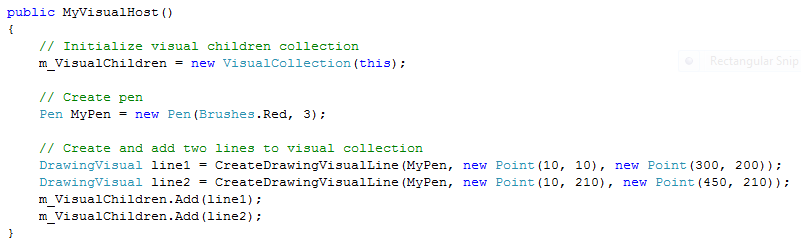


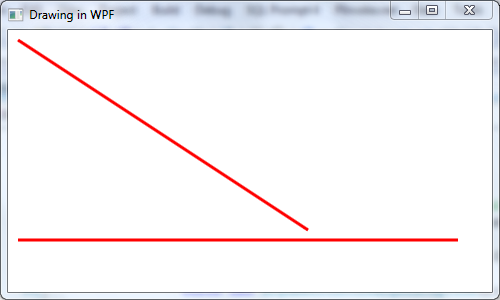


### Draw Line

Prvi način: DIREKTNO





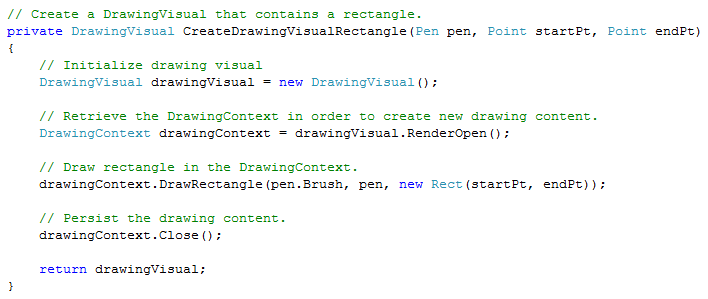


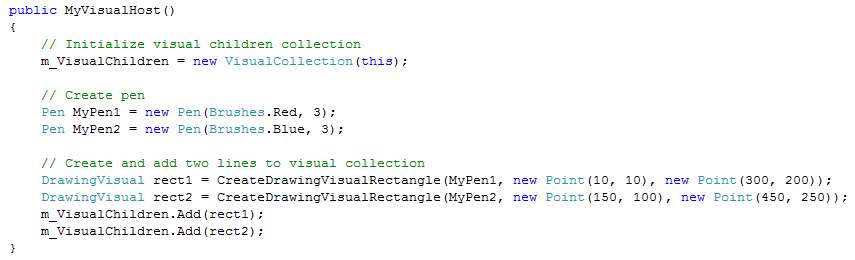
Drugi način: GEOMETRIJA

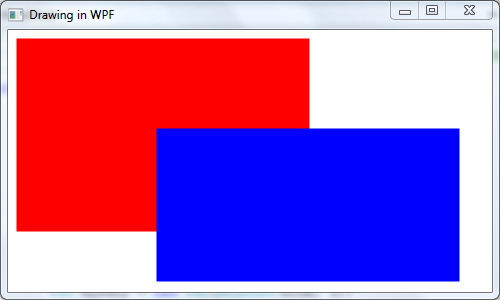
### 

### Draw Rectangle

Prvi način: DIREKTNO





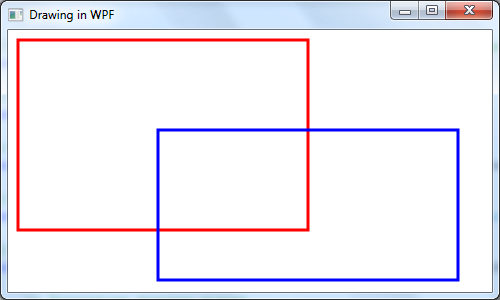


Drugi način: Geometrija

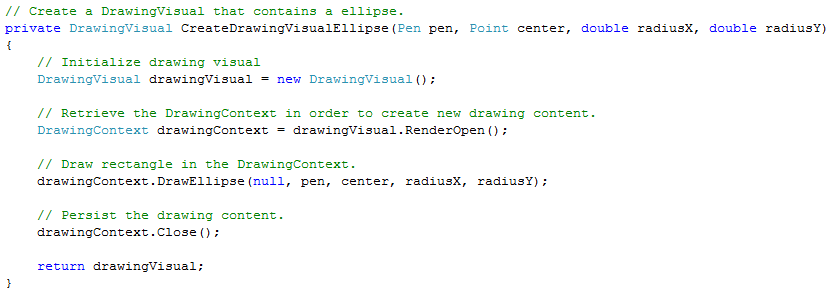


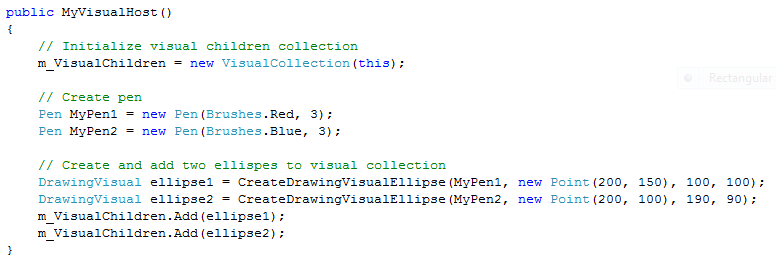
Obojene samo ivice

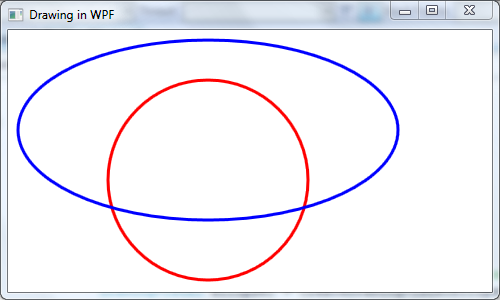




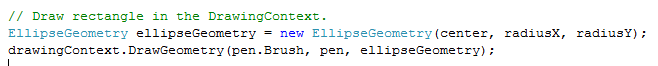
### Draw Ellipse

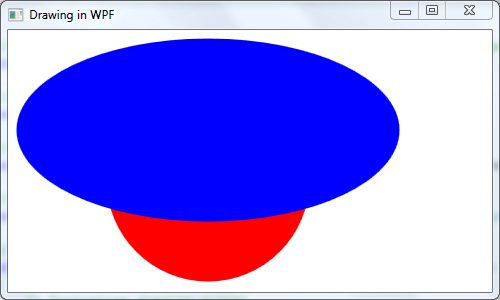




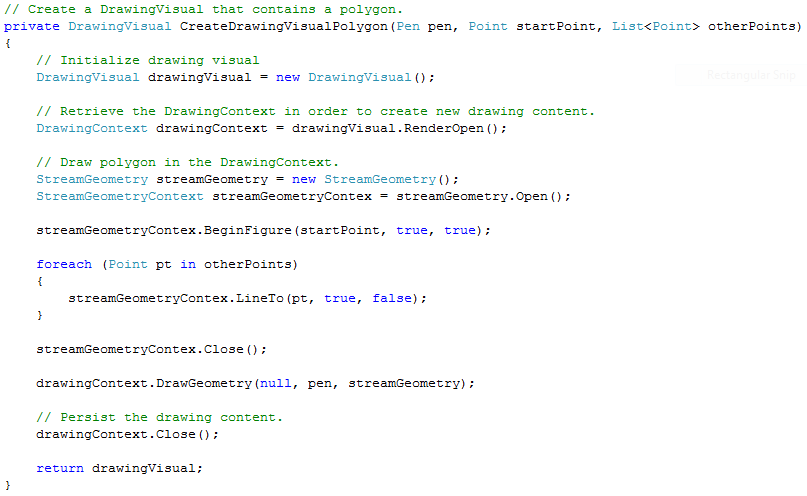


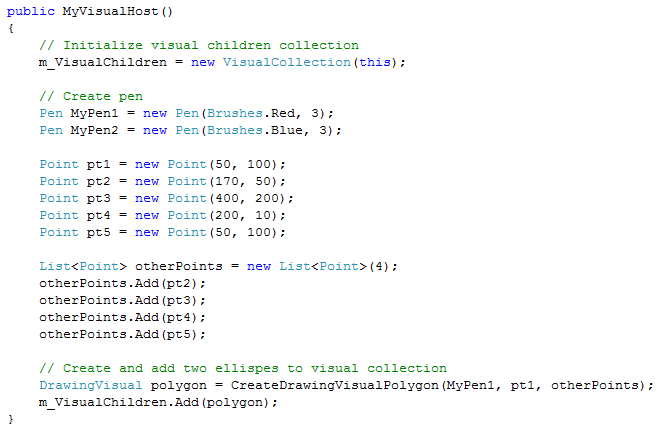
Drugi način: Geometrija

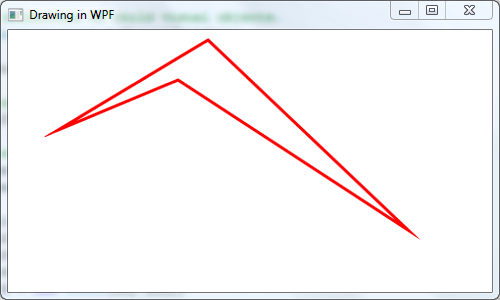




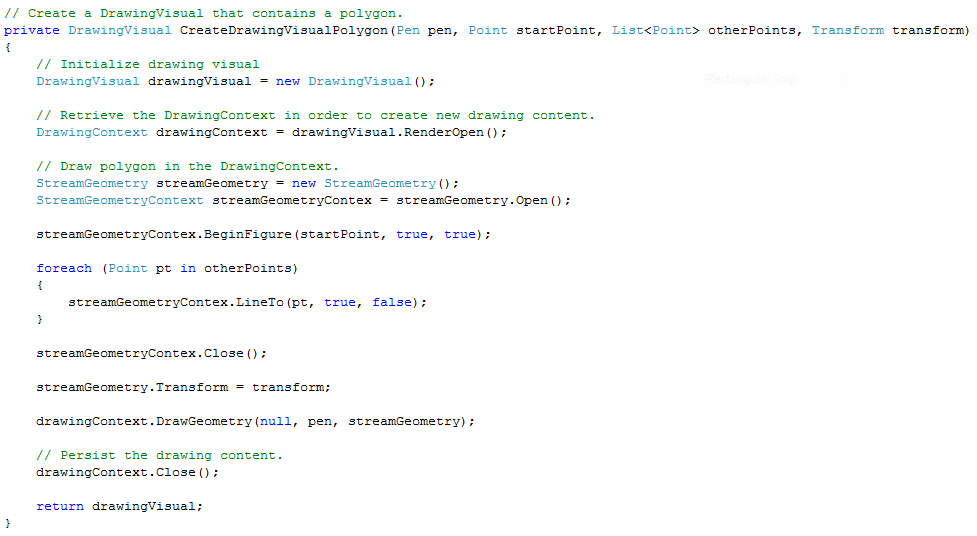
### Draw Polygon

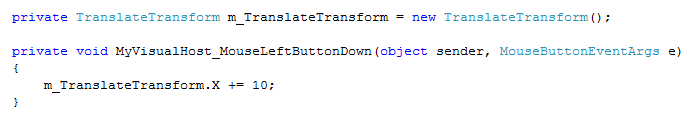


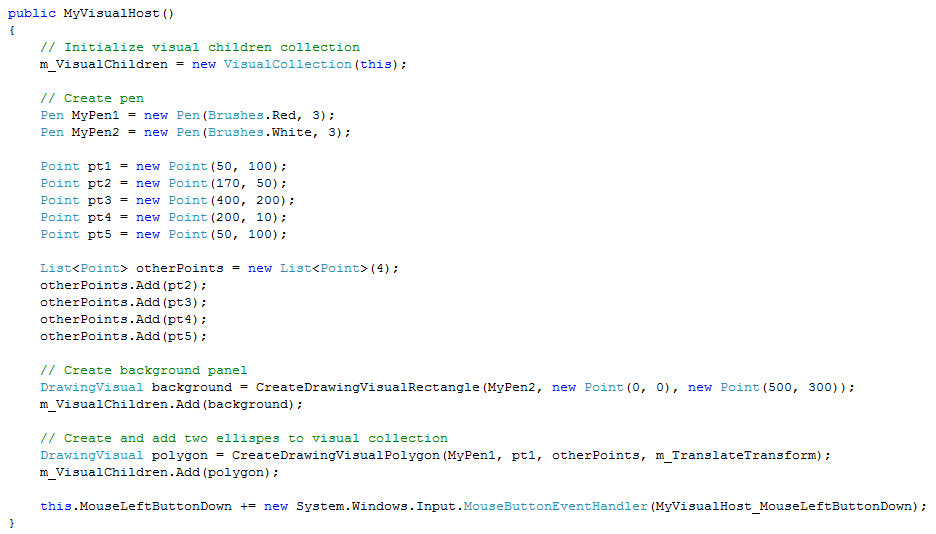


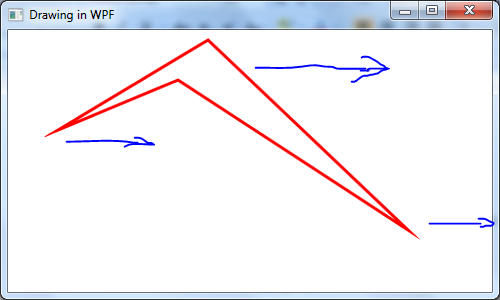


Transformacije

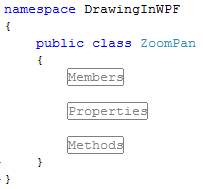


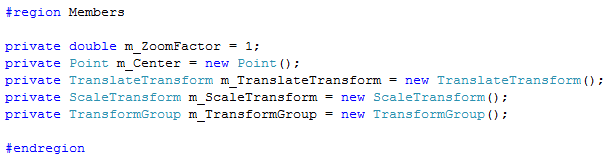


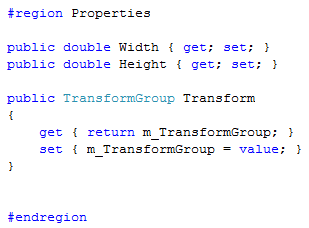


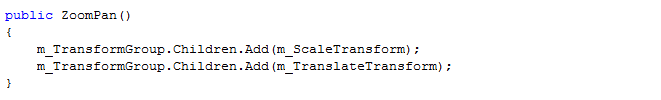


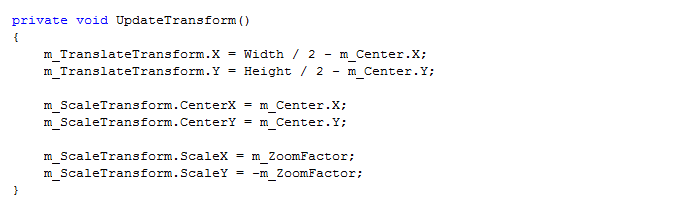
ZoomAndPan



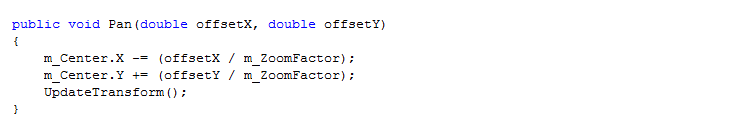


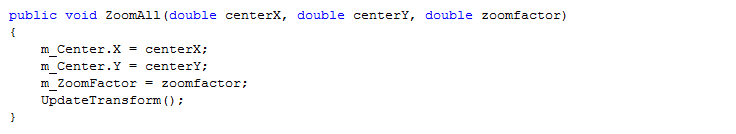


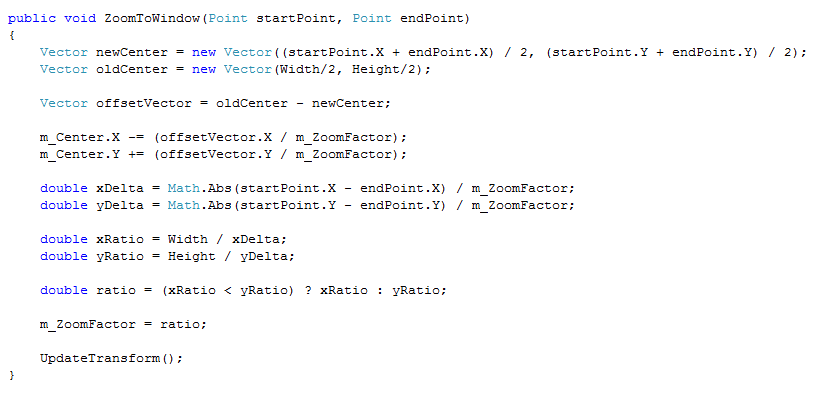






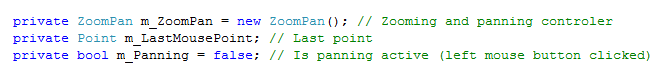




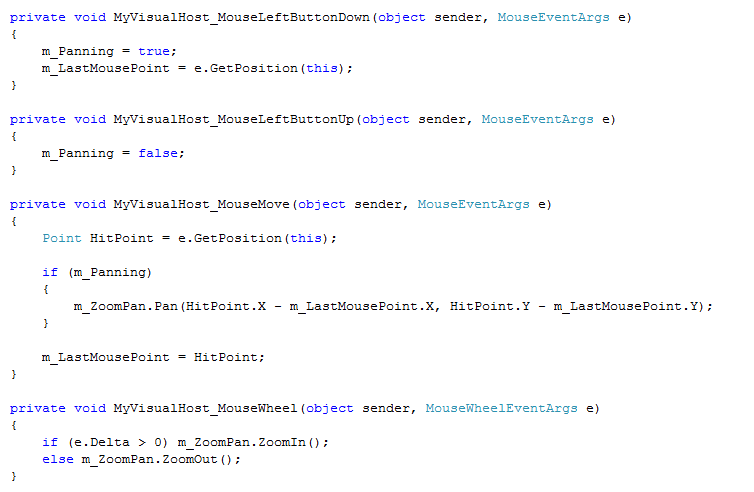


U Hostu:

* Promenljive



* Metode



* Konstruktor

public MyVisualHost()

{

// Initialize visual children collection

m\_VisualChildren = new VisualCollection(this);

// Create pen

Pen MyPen1 = new Pen(Brushes.Red, 3);

Pen MyPen2 = new Pen(Brushes.White, 3);

Point pt1 = new Point(50, -100);

Point pt2 = new Point(170, -50);

Point pt3 = new Point(400, -200);

Point pt4 = new Point(200, -10);

Point pt5 = new Point(50, -100);

List<Point> otherPoints = new List<Point>(4);

otherPoints.Add(pt2);

otherPoints.Add(pt3);

otherPoints.Add(pt4);

otherPoints.Add(pt5);

m\_ZoomPan.Width = 500;

m\_ZoomPan.Height = 300;

// Create background panel

DrawingVisual background = CreateDrawingVisualRectangle(MyPen2, new Point(0, 0),

new Point(System.Windows.SystemParameters.WorkArea.Width,

System.Windows.SystemParameters.WorkArea.Height));

m\_VisualChildren.Add(background);

// Create and add two ellispes to visual collection

DrawingVisual polygon = CreateDrawingVisualPolygon(MyPen1, pt1, otherPoints, m\_ZoomPan.Transform);

m\_VisualChildren.Add(polygon);

this.MouseLeftButtonDown += new MouseButtonEventHandler(MyVisualHost\_MouseLeftButtonDown);

this.MouseLeftButtonUp += new MouseButtonEventHandler(MyVisualHost\_MouseLeftButtonUp);

this.MouseWheel += new MouseWheelEventHandler(MyVisualHost\_MouseWheel);

this.MouseMove += new MouseEventHandler(MyVisualHost\_MouseMove);

double xMax = 400;

double xMin = 50;

double yMax = -10;

double yMin = -200;

double x = (xMax + xMin) / 2;

double y = (yMax + yMin) / 2;

double xDelta = Math.Abs(xMax - xMin);

double yDelta = Math.Abs(yMax - yMin);

double xRatio = 500 / xDelta;

double yRatio = 300 / yDelta;

double ratio = (xRatio < yRatio) ? xRatio : yRatio;

m\_ZoomPan.ZoomAll(x, y, ratio);

}

Sada se sa točkićem miša radi zumiranje, a klikom i prevlačenjem miša pomeranje objekata.