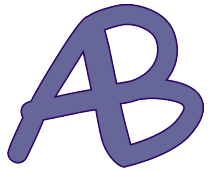
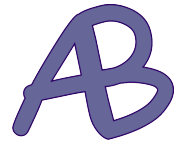


Wykład Canvas



Przykład 1 – pozycja kursora



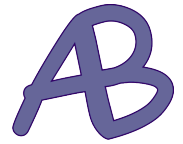


Canvas

Layout aplikacji:

```
<Canvas x:Name="canvas" Background="White"
    MouseMove="canvas_MouseMove">
    <Border x:Name="opis_kursora"
        BorderBrush="Black" BorderThickness="1">
        <Label Name="pozycja" Content="0 ; 0"/>
    </Border>
</Canvas>
```

Uwaga: tło obiektu canvas jest istotne, aby mógł on przechwytywać zdarzenia



Canvas

Oprogramowujemy zdarzenie MouseMove dla obiektu canvas:

```
private void canvas_MouseMove(object sender, MouseEventArgs e)
{
    Point p = e.GetPosition(canvas);
    //Pobieramy pozycję kursora względem elementu canvas
    Canvas.SetLeft(opis_kursora, p.X+10);
    // aktualizujemy położenie kontrolki
    // "opis_kursora" na canvas
    Canvas.SetTop(opis_kursora, p.Y+5);
    pozycja.Content = Math.Round(p.X) + " ; " + Math.Round(p.Y);
    // wypisujemy aktualną pozycję kursora (po zaokrągleniu)
}
```

The screenshot shows a standard Windows application window. The title bar at the top reads 'MainWindow'. Below the title bar is a menu bar with two visible menu items: 'Czyść canvas' and 'Usuń najstarszą kropkę'. The main area of the window is a white canvas filled with numerous overlapping circles. These circles vary in size and color, including shades of purple, blue, green, yellow, orange, red, pink, and brown. Some circles are semi-transparent, allowing others behind them to be visible. The circles are scattered across the canvas, with some appearing in clusters and others isolated.

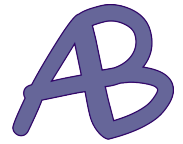


Canvas

Layout aplikacji:

```
<DockPanel>
  <ToolBar DockPanel.Dock="Top">
    <Button Padding="10,2" Click="Button_Click">
      Czyść canvas
    </Button>
    <Button Padding="10,2" Click="Button_Click_1">
      Usuń najstarszą kropkę
    </Button>
  </ToolBar>
  <Canvas x:Name="canvas" Background="White"
    MouseDown="canvas_MouseDown" />
</DockPanel>
```

Uwaga: tło obiektu canvas jest istotne, aby mógł on przechwytywać zdarzenia



Canvas

Przygotowujemy listę przechowującą obiekty klasy `Ellipse` oraz generator liczb pseudolowowych

```
List<Ellipse> ellipseList = new List<Ellipse>();
```

```
Random random = new Random();
```

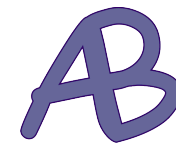


Canvas

Dodawanie elips po kliknięciu myszą

```
private void canvas_MouseDown(object sender, MouseButtonEventArgs e)
{
    Point p = e.GetPosition(canvas);
    //Pobieramy pozycję kursora względem elementu canvas
    Ellipse e1 = new Ellipse();
    //tworzymy obiekt klasy Ellipse
    e1.Width = random.Next(50) + 10;
    e1.Height = random.Next(50) + 10;
    //losujemy wymiary elipsy
    e1.Fill = new SolidColorBrush(Color.FromRgb((Byte)random.Next(255),
                                                    (Byte)random.Next(255),
                                                    (Byte)random.Next(255)));

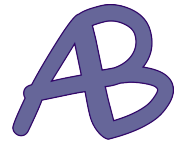
    //losujemy kolor wypełnienia
    ellipseList.Add(e1);
    //dodajemy elipsę do listy
    Canvas.SetLeft(e1, p.X);
    Canvas.SetTop(e1, p.Y);
    //ustawiamy położenie elipsy na podstawie pozycji kursora
    canvas.Children.Add(e1);
    //dodajemy elipsę do obiektu canvas
}
```

Canvas

Czyszczenie obiektu canvas (oraz listy elips)

```
private void Button_Click(object sender, RoutedEventArgs e)
{
    canvas.Children.Clear();
    ellipseList.Clear();
}
```

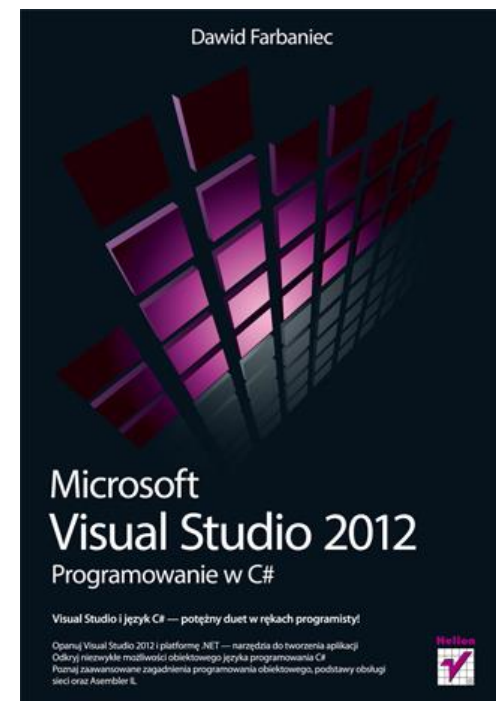
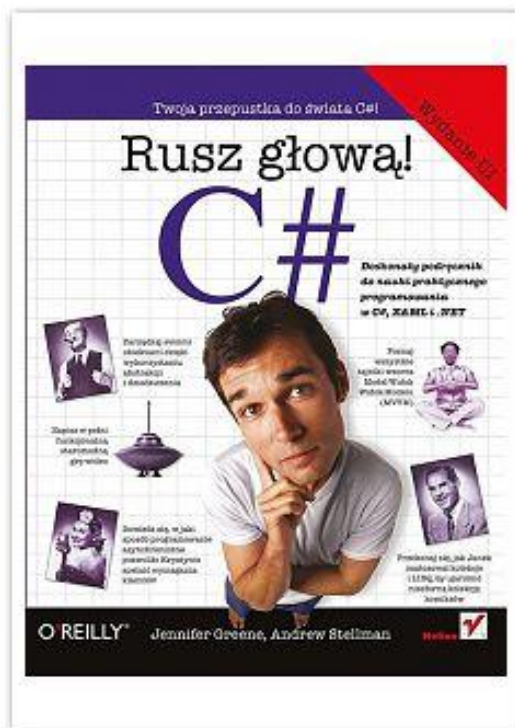
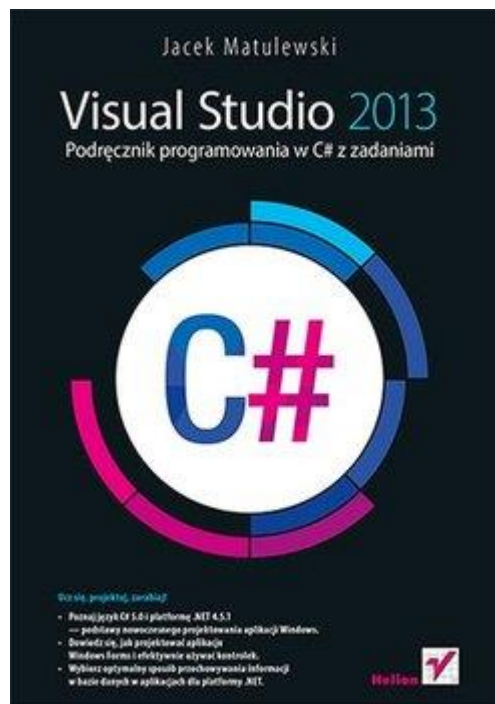
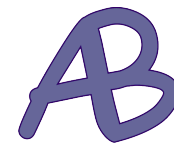


Canvas

Usunięcie najstarszej elipsy (pierwszej na liście)

```
private void Button_Click_1(object sender, RoutedEventArgs e)
{
    canvas.Children.Remove(ellipseList[0]);
    ellipseList.RemoveAt(0);
}
```

Literatura:



Użyte w tej prezentacji tabelki pochodzą z książki: Visual Studio 2013. Podręcznik programowania w C# z zadaniami
Autor: Matulewski Jęcek, Helion