

Seminar 1

Jörn Loviscach

Versionsstand: 22. März 2011, 20:55



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Die ersten Schritte mit C# und WPF: ein Programm, das eine kleine Kreisscheibe wie einem Ball durch das Fenster springen lässt. Die Farbe und die Geschwindigkeit sind dabei einstellbar. Eine Schaltfläche startet und stoppt die Bewegung. Ein Zähler gibt an, wie viele Mausklicks im Flug auf der Kreisscheibe gelandet wurden.

Erzeugen Sie in Microsoft Visual Studio 2010 ein Projekt vom Typ „Visual C#: WPF-Anwendung“. Gehen Sie zu dem .xaml-Fenster. Ziehen Sie aus der Toolbox in das Fenster: einen Button, eine Checkbox und drei Radiobuttons. Beschriften Sie diese (Eigenschaft `Content`) mit „Start/Stop“, „schnell“, „rot“, „grün“ beziehungsweise „blau“. Setzen Sie den Name jedes dieser fünf Elemente auf einen aussagekräftigen Bezeichner. Setzen Sie die Eigenschaft `IsChecked` des Radiobuttons für rot auf `true`.

Ersetzen Sie das Element `Grid` durch `Canvas`, sowohl im öffnenden wie im schließenden Tag. Geben Sie ihm den Namen `theCanvas`. Tragen Sie per Hand am Ende des `Canvas`-Elements das Element ein, das der Ball werden wird:

```
<Ellipse Width="20" Height="20" Canvas.Left="150"
Canvas.Top="100" Fill="Red" Name="ball"/>
```

Fügen Sie allen drei Radiobuttons einen neuen Ereignishandler für das Ereignis `Click` hinzu. Für rot schreiben sie als Code in den Ereignishandler: `ball.Fill = Brushes.Red`; Entsprechend bei den beiden anderen. Lassen Sie das Programm laufen und prüfen Sie, ob der Wechsel der Farbe funktioniert.

Fügen Sie in die den Programmcode der Klasse ein:

```
System.Windows.Threading.DispatcherTimer timer =
new System.Windows.Threading.DispatcherTimer();
```

Dies ist der Zeitgeber, der für das Fortschreiten der Animation sorgen wird. Beobachten Sie, wie Visual Studio bei der Eingabe der Namen hilft.

Ergänzen Sie eine einfache Animationsfunktion:

```
void physics(object sender, EventArgs e)
{
    double x = Canvas.GetLeft(ball);
    x += 5.0;
    Canvas.SetLeft(ball, x);

    double y = Canvas.GetTop(ball);
    y += 5.0;
    Canvas.SetTop(ball, y);
}
```

Sorgen Sie in dafür, dass der Timer diese Animationsfunktion alle 0,05 Sekunden aufruft. Ergänzen Sie dazu in public MainWindow() hinter InitializeComponent(); diese Zeilen:

```
timer.Tick += physics;
timer.Interval = TimeSpan.FromSeconds(0.05);
timer.IsEnabled = true;
```

Hinweise für die weiteren Funktionen: `theCanvas.ActualWidth` liefert die Breite der Zeichenfläche namens `theCanvas`. Der Zählerstand kann in ein Label-Element ausgegeben werden.

Mögliche Erweiterungen: Schwerkraft und Reibung; Sounds beim Aufprall mit `System.Media.SystemSounds.Beep.Play();` usw.