

Grafiske brugergrænseflader i F#

Programmering og problemløsning

Jon Sporring

Kommandolinjen

```
[Jons-mac:winforms sporring$ pwd  
/Users/sporring/repositories/fsharpNotes/src/winforms  
[Jons-mac:winforms sporring$ ls *.fsx  
Window3.fsx          openWindowFullScreen.fsx  
analogClock.fsx      panel.fsx  
bounds.fsx           pixels.fsx  
buttonControl.fsx    pointNClick.fsx  
buttonControlAdv.fsx progressBar.fsx  
clock.fsx            refresh.fsx  
clockAdv.fsx         rotationalSymmetry.fsx  
controls.fsx         simpleFlowLayoutPanel.fsx  
dateTimePicker.fsx   simpleTableLayoutPanel.fsx  
fSharpForever.fsx    tabControl.fsx  
flowLayoutPanel.fsx  trackBar.fsx  
flowLayoutPanelAdvanced.fsx transformWindows.fsx  
flowLayoutPanelAdvanced2.fsx triangle.fsx  
hilbert.fsx          triangleClientSize.fsx  
hilbert2.fsx         triangleOrganized.fsx  
imageProcessing.fsx  triangleOrganizedAdv.fsx  
imageProcessing2.fsx window2.fsx  
messageBox.fsx        windowEvents.fsx  
openFileDialog.fsx   windowProperty.fsx  
openWindow.fsx  
[Jons-mac:winforms sporring$ ]
```

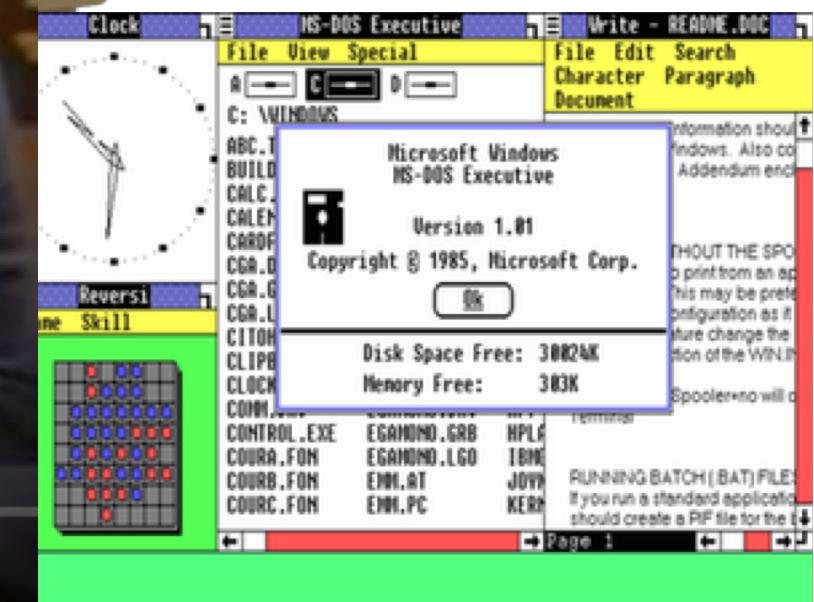


1:31 / 8:56

HP Touch computer, 1983, youtube.com

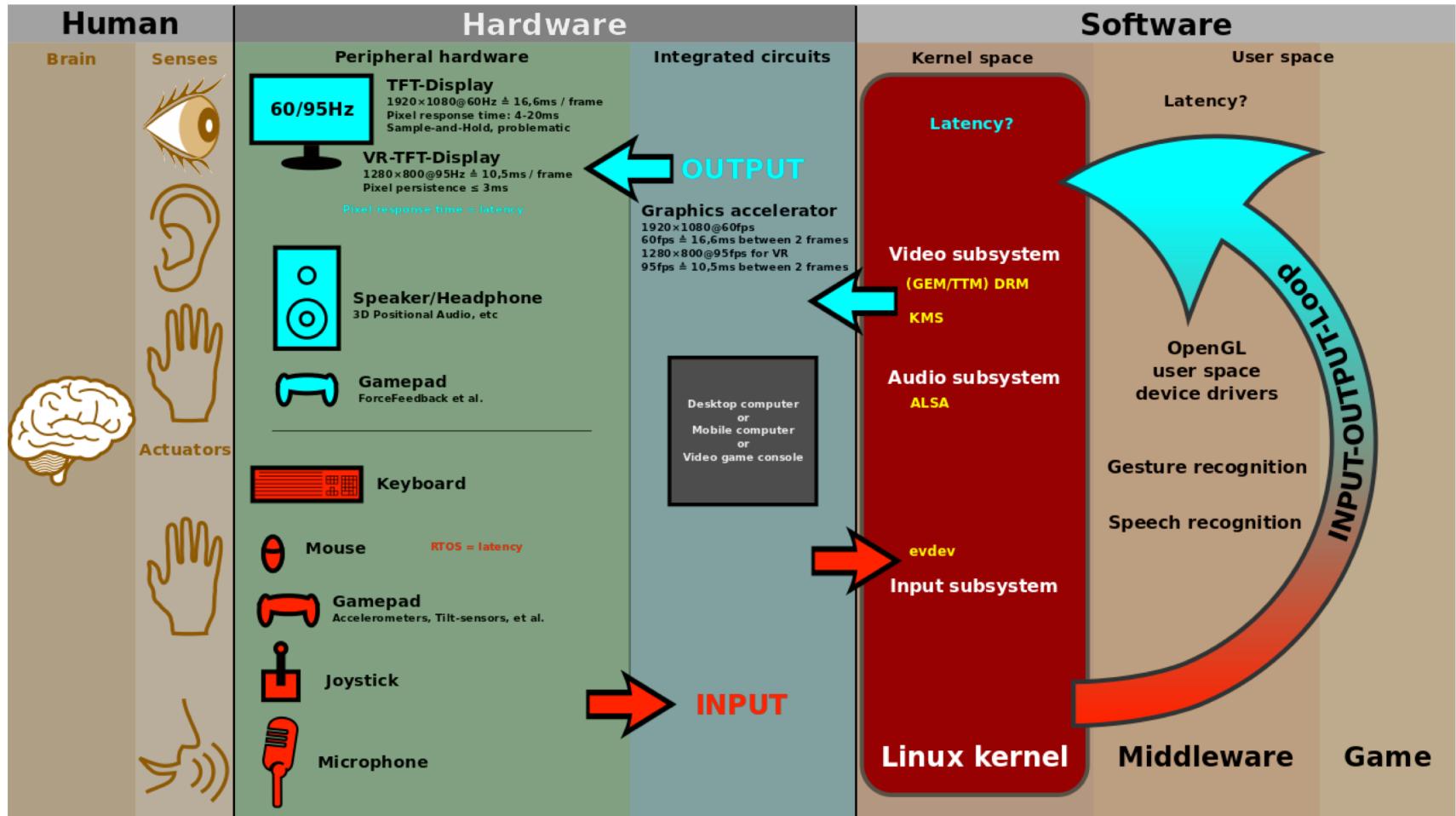


Douglas Engelbart's mouse, 1963, wikipedia



Windows 1.0, 1985, wikipedia

Interaktion mellem bruger, software og hardware

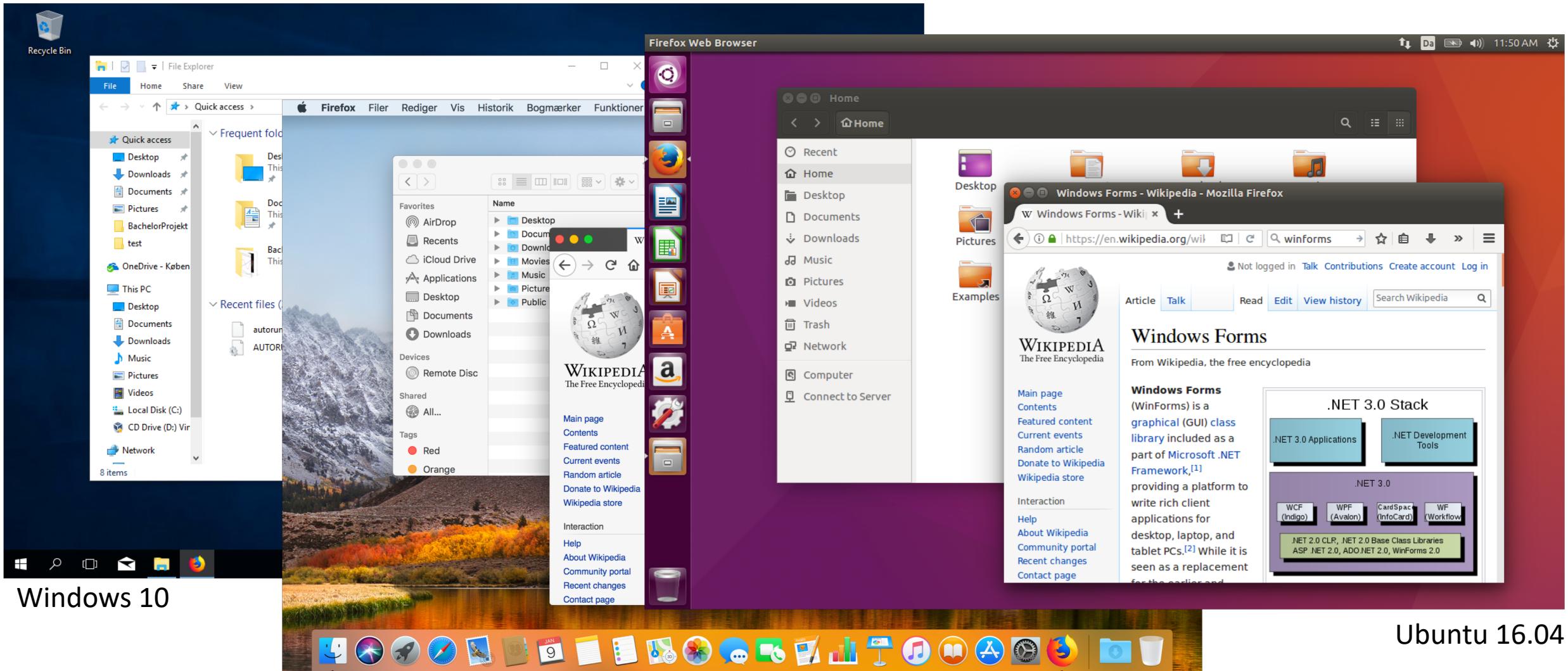


GNU Free Documentation License, author: Shmuel Csaba Otto Traian, en.wikipedia.com, retrieved: 2018/1/10.

Design guidelines

- Foretræk simple interfaces
- Gå efter konsistens og med elementer, brugeren kender
- Vælg en opsætning, som fremhæver de vigtigste elementer
- Brug farve og teksturer strategisk
- Vær konservativ mht. font, størrelse og form
- Gør brugeren opmærksom på, hvad der sker
- Tænk på default værdier

Grafiske brugergrænseflader



Windows 10

MacOs 10.13

Ubuntu 16.04

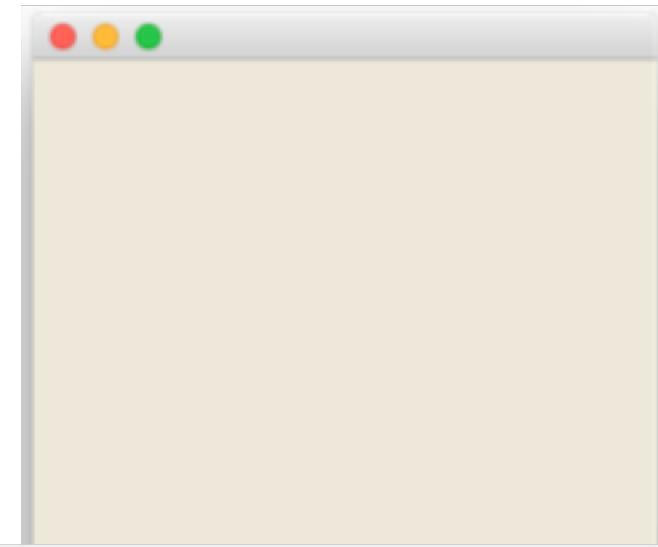
Windows and libraries for Mono

- **WinForms 2.0** (.Net, MS Windows, DirectX vs. GDI+, Windows Presentation Foundation)
- Gtk# (Gtk+, Linux/Gnome)
- Cocoa (MacOS)

My first window

openWindow.fsx

```
// Create a window
let win = new System.Windows.Forms.Form ()
// Start the event-loop.
System.Windows.Forms.Application.Run win
```

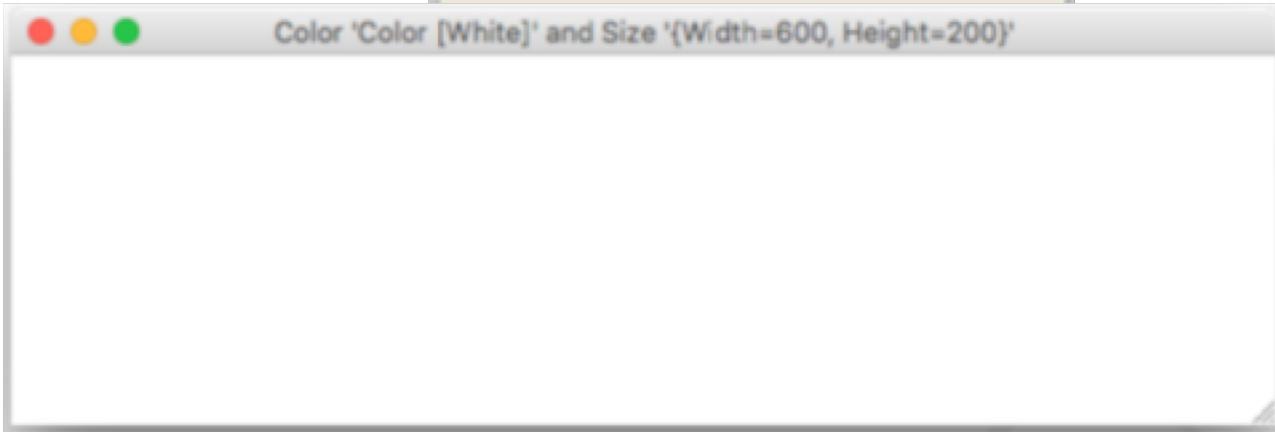


windowProperty.fsx

```
// Prepare window form
let win = new System.Windows.Forms.Form ()

// Set some properties
win.BackColor <- System.Drawing.Color.White
win.Size <- System.Drawing.Size (600, 200)
win.Text <- sprintf "Color '%A' and Size '%A'" win.BackColor win.Size

// Start the event-loop.
System.Windows.Forms.Application.Run win
```



Tegne call-back funktionen

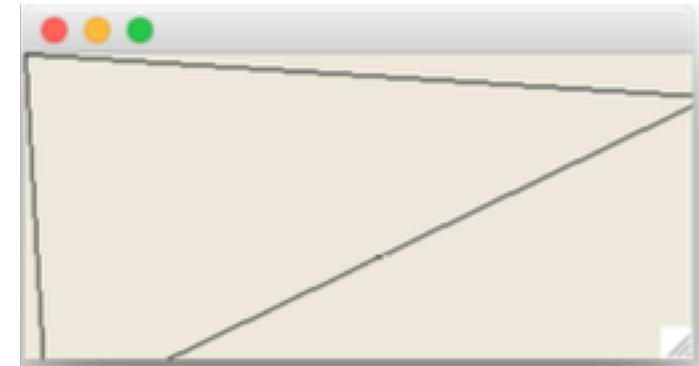
triangle.fsx

```
// Open often used libraries, beware of namespace polution!
open System.Windows.Forms
open System.Drawing

// Prepare window form
let win = new Form ()
win.Size <- Size (320, 170)

// Set paint call-back function
let paint (e : PaintEventArgs) : unit =
    printfn "redrawing"
    let pen = new Pen (Color.Black)
    let points =
        [|Point (0,0); Point (10,170); Point (320,20); Point (0,0)|]
    e.Graphics.DrawLines (pen, points)
    win.Paint.Add paint

// Start the event-loop.
Application.Run win
```



Skrive på skærmen

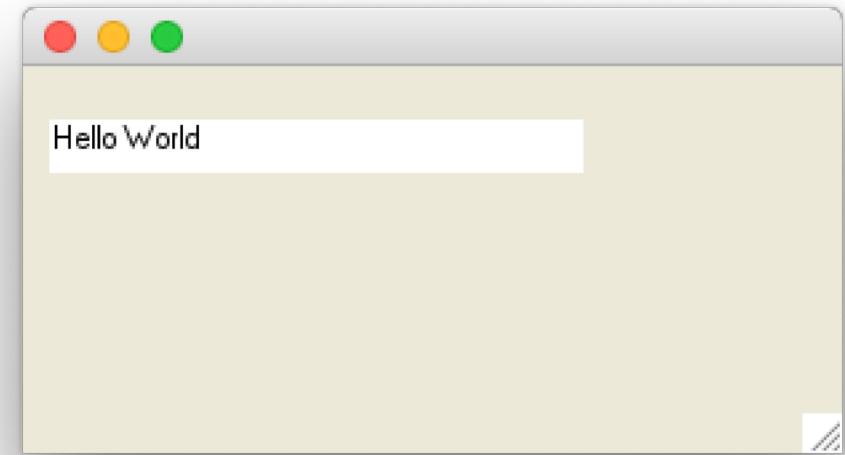
label.fsx

```
open System.Windows.Forms
open System.Drawing

let win = new Form () // make a window form
win.ClientSize <- Size (200, 100)

// make a label to show time
let label = new Label()
win.Controls.Add label
label.Width <- 200
label.Location <- new Point (10, 20)
label.Text <- "Hello World"
label.BackColor <- Color.White
label.Height <- 20

Application.Run win // start event-loop
```



System.Timers

fsharpi

```
let t = new System.Timers.Timer()
t.Interval <- 1000.0
t.Elapsed.Add (fun e -> printfn "%s" (string System.DateTime.Now));;
t.Start();;
t.Stop();;
```

System.Windows.Forms

clock.fsx

```
open System.Windows.Forms
open System.Drawing

let win = new Form () // make a window form
win.ClientSize <- Size (200, 50)

// make a label to show time
let label = new Label()
win.Controls.Add label
label.Width <- 200
label.Text <- string System.DateTime.Now // get present time and
date

// make a timer and link to label
let timer = new Timer()
timer.Interval <- 1000 // create an event every 1000 millisecond
timer.Enabled <- true // activiate the timer
timer.Tick.Add (fun e -> label.Text <- string System.DateTime.Now)

Application.Run win // start event-loop
```

