Cross Platform Mobile Development on Windows 7, Android and iOS

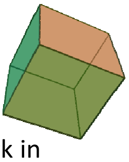
Jeroen Pluimers - better office benelux
jpluimers@better-office.com

27.09.2011 | 11:15 - 12:30

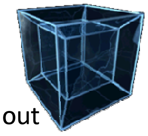
Developing for Mobile

is developing on the bleeding edge

Be prepared



- Mobile devices form a complex matrix (or worse: cube, and hybercube)
- Things that work in one cell won't work in another
- Or work in one version of your dev environment ant not in others
- Just stop working after N times without reason
- Simple things can take days to sort out



Why .NET on Mobile devices?

- Otherwise you will have to learn
  - The device APIs
  - A new language
  - A new platform
    - Example: manual memory management
- Now you only need to learn the device APIs
  - If you want you can learn the other parts later on

Outside the Windows Development world…

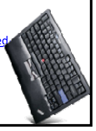
… Macs are popular.
Very, very, very popular.
Some things getting used to…

Moving from Windows to Mac 1

- The Mac Keyboard/Touchpad s*cks^H^H^H^H^H takes getting used to
  - Many Ctrl combinations on Windows are Command combinations on Mac
  - Keys are in different places
    - Ctrl-Win-Alt
      - versus
    - Control-Alt/Option-Command
      - You can set this in the "System Preferences" -> "Keyboard" settings http://codyhanson.com/blog/2007/11/13/os-x-leopard-and-usb-keyboards/

  - or not there at all on MacBook machines:
    - Insert Home PageUp Home == Ctrl-A
    - Delete End PageDown End == Ctrl-E

  - You either love or hate the Touchpad (Mac/PC) or TrackPoint (PC)
  - I use a ThinkPad USB keyboard instead
    - http://www.lenovoblogs.com/designmatters/2009/09/the-keyboard-you-helped



Moving from Windows to Mac 2

- Mac OS X is based on a XNU/Mach kernel (which is a successor to BSD)
  - so most things (especially command-line) feels Unix-Like (Linux, Unix, BSD, etc)
- Screen resolutions are all non-4x3
  - Make sure you get at least WXGA+ (1440x900) so you can display it on an XGA (1024x768) beamer
- External display uses Mini DisplayPort connector
  - http://en.wikipedia.org/wiki/Mini_DisplayPort
  - Requiring adapters http://support.apple.com/kb/ht3235 http://support.apple.com/kb/HT3382#1 for
    - DisplayPort (3rd party only, for instance http://www.sendstation.com/us/products/mdpadapter/)
    - VGA, DVI, Dual Link DVI

Developing for Mobile and PCs

Many platforms, many choices

Platforms, Choices

- Windows PC
  - .NET
  - Native (C++/Delphi/...)
- Mac OS X
  - Xcode + Objective-C
  - Mono with MonoDevelop and MonoMac
  - Delphi XE2
- Windows CE/Phone/..
  - **Windows Phone and beyond**
    - **.NET only**
  - CE/before
    - .NET
    - C++
- **iOS**
  - Xcode + Objective-C
  - **Mono with MonoDevelop+MonoTouch**
  - Delphi XE2
- **Android**
  - Oracle/Sun Java
  - **Mono with MonoDevelop+MonoDroid**
  - RemObjects Cooper
  - MainSoft Grasshopper
- Blackberry
  - Oracle/Sun Java
  - RemObjects Cooper
  - MainSoft Grasshopper

PhoneGap is special

- Works only well for on-line apps
  - Html5, CSS3, jQuery, etc.
- Require no client development
  - Only server
- See Christian Weyer’s session “Nicht nur Web: Android und iOS Apps mit HTML5 und Co.” http://entwickler.com/konferenzen/ext_scripts/v2/php/sessions-popup.php?module=basta2011&id=18466 27.09.2011 | 14:00 - 15:15

If you keep UI + Business separate …

… then .NET / Mono is a viable possibility

Outside the Windows Development world…

… .NET is called Mono

But wait, wasn’t

- Mono developed by Novell
- and Novell acquired by Attachmate?

- Yes, and yes.
- But now Xamarin is developing Mono, MonoTouch and MonoDroid
  - http://en.wikipedia.org/wiki/Xamarin
  - They have a perpetual license from Novell/Attachmate for this

- So: Miguel de Icaza is still very much involved
  - Without Miguel the whole Mono thing was quite worthless for Novell/Attachmate

Mono is similar to .NET

- http://www.mono-project.com
- Current Mono 2.10.x is very compatible with .NET 4.x
  http://www.mono-project.com/Compatibility
  - But lacks
    - WPF, EF, WF and
    - most of WCF (only the SilverLight 2.0 bits are present)
  - Supports both
    - VB.NET and
    - C# (including 4.0 syntax)
  - Not all documentation is up to date
    (some docs still talk about 2.8 as being “current”)
  - Zillion classes outside the .NET scope (OpenGL, Cairo, etc)
  - Runs in many platforms
    - Including PC, Mac, Linux, Mobile, Wii, PS3, iSeries (AS400), zSeries (S390), etc

MonoDevelop is similar to Visual Studio

- Visual Studio wants to upgrade .sln, .csproj files generated by MonoDevelop
- MonoDevelop understands .sln, .csproj files from Visual Studio just fine

Platforms

- Windows Phone
  - Runs on ARM processors
- Apple iOS
  - Runs on ARM processors
- Android
  - Runs on ARM (phones) and x86 processors (Google TV)
  - Supports Java Bytecode
- All support multi-touch gestures

- Note:
  - most phone “emulators” are simulators and run in x86 mode
  - Always test on real hardware!

Other platforms

- PlayStation: http://www.engadget.com/2011/09/15/playstation-suite-sdk-beta-coming-in-november-offering-games-in
- NETDuino http://netduino.com/

Develop for Windows Phone 7.0 or 7.1

- Make sure you get the
  - May 2011 “Windows Phone Developer Tools 7.1 Beta” installer
    - http://www.microsoft.com/download/en/details.aspx?id=11321
    - Compatible with VS 2010 SP1
  - Or the Oct 2010 “Windows Phone Developer Tools RTW”
    - http://www.microsoft.com/download/en/details.aspx?id=13890
    - Compatible with VS 2010 RTM (not SP1)

When developing for iOS

- Apple is far more hostile than Microsoft:
  - New iOS and Xcode versions break lots of things that worked in older versions
    - Including MonoTouch
  - The Apple developer community really dislikes anything other than Xcode / Objective-C
- You deliver separate apps for any combination of:
  - iOS hardware
    (different iPhone/iPad/iPod-touch/Apple-TV versions)
  - iOS software
    (different iOS versions can run on the same hardware)
- The iOS hardware/software matrix is far less complicated than the Microsoft combination on
  CE/Phone/… * hardware is
- Apple comes with iOS updates/patches often
- Most users do apply those updates

When developing for Android or Windows Phone

- The matrix/cube of combinations is overwhelming:
  - Hardware devices
  - Android/Windows OS versions
  - Phone Provider Skins
- You can simplify this by limiting the support for:
  - Phone like sizes/screen resolutions
  - Tablet like sizes/screen resolutions
- Hardware vendors hardly come with OS updates
  - This means that you will have to workaround bugs often
  - There are tons of security vulnerabilities
    (the Dutch DigiNotar vulnerability only being one)

MonoTouch/MonoDroid getting started

Prepare for at least a day to do your installs

Some of them are
huge (Xcode 4: 4+GB, Visual Studio 2010: ditto)
or slow (Android post SDK actions: 30 minutes)

Mono development: what you need 1

MonoDevelop http://monodevelop.com/Download

- Supports
  - Mac OSX
    - Mono Develop Beta 2.8
    - Mono Develop Stable 2.6
    - Both require
      - » Mono 2.10.5 + GTK#
        - Runtime
          http://download.mono-project.com/archive/2.10.5/macos-10-x86/0/MonoFramework-MRE-2.10.5_0.macos10.xamarin.x86.dmg
        - SDK
          http://download.mono-project.com/archive/2.10.5/macos-10-x86/0/MonoFramework-MDK-2.10.5_0.macos10.xamarin.x86.dmg
  - Windows
    - Mono Develop Stable 2.6, no beta 2.8
    - Requires
      - » .NET 3.5 from Microsoft
      - » GTK# from Novell
        http://download.mono-project.com/gtk-sharp/gtk-sharp-2.12.10.win32.msi
  - Linux flavours: openSUSE, SLE, Debia, Ubuntu

- MonoDroid
- MonoTouch

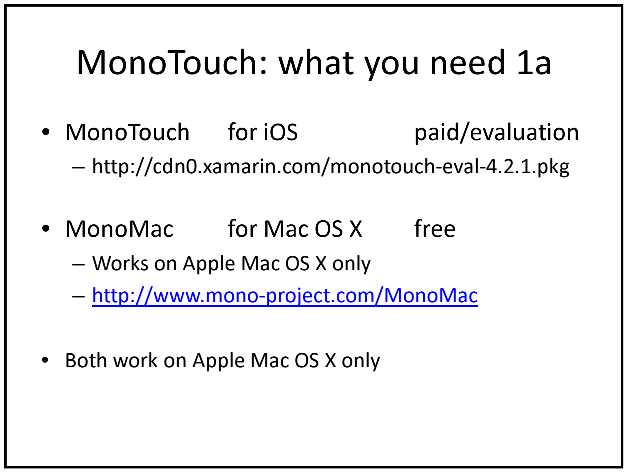
MonoDroid: what you need 1a

- MonoDroid for Android (paid or evaluation)
  - Works on both Windows and Apple Mac OS X
  - Requires
    - Oracle/Sun x86 Java 1.6 JDK aka Java SE 6 (not x64, not JRE, not 1.7 aka Java SE 7)
      - Current: Java SE 6 Update 27
      - Windows:
        http://www.oracle.com/technetwork/java/javase/downloads/jdk-6u27-download-440405.html
        - http://download.oracle.com/otn-pub/java/jdk/6u27-b07/jdk-6u27-windows-i586.exe
      - Mac: Use System Preferences
    - Android SDK
      - http://developer.android.com/sdk/index.html
      - Windows http://dl.google.com/android/installer_r13-windows.exe
        http://dl.google.com/android/android-sdk_r13-windows.zip
        - » Do not install in a directory with spaces (not C:\Program Files, **but C:\android-sdk**)
      - Mac http://dl.google.com/android/android-sdk_r13-mac_x86.zip
      - Both
        - » After installing the SDK, run the SDK Manager, install all things that have a chekmark
- DroidDraw
  - http://www.droiddraw.org/

MonoDroid: what you need 1b

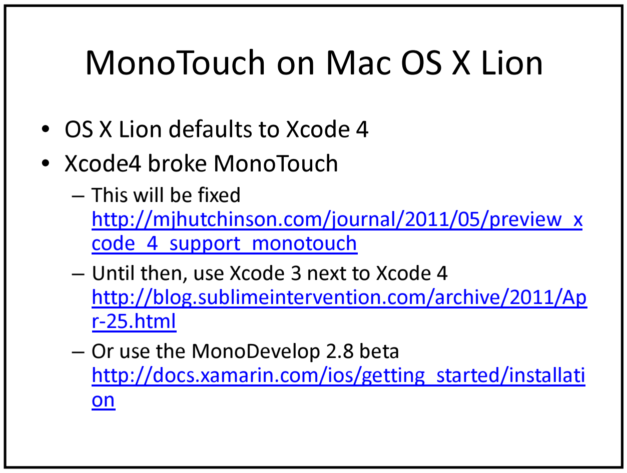
- MonoDroid installation steps
  - http://android.xamarin.com/Installation
    - Mac
      - http://cdn0.xamarin.com/monoandroid-eval-1.2.0.dmg
      - MonoDevelop
        - http://android.xamarin.com/Installation/Installation_for_Mac
    - Windows
      - http://cdn0.xamarin.com/monoandroid-1.2.0.msi
      - Visual Studio
        - http://android.xamarin.com/Installation/Visual_Studio
      - MonoDevelop
        - http://android.xamarin.com/Installation/Windows
        - MonoDevelop works better than Visual Studio

- Note the odd endings of the URLs :)



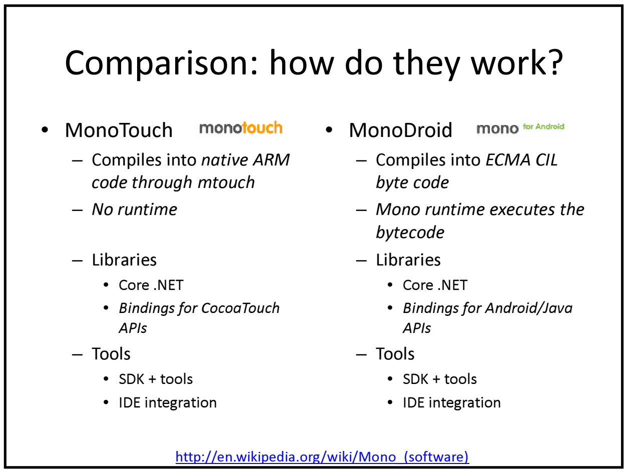
MonoTouch: what you need 1a

- MonoTouch for iOS paid/evaluation
  - http://cdn0.xamarin.com/monotouch-eval-4.2.1.pkg
- MonoMac for Mac OS X free
  - Works on Apple Mac OS X only
  - http://www.mono-project.com/MonoMac
- Both work on Apple Mac OS X only



MonoTouch on Mac OS X Lion

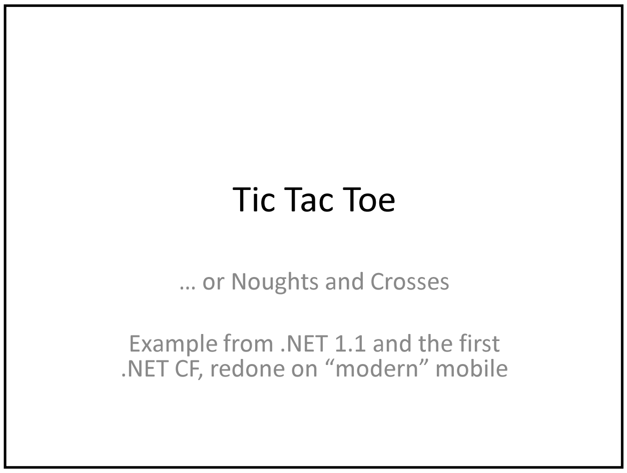
- OS X Lion defaults to Xcode 4
- Xcode4 broke MonoTouch
  - This will be fixed http://mjhutchinson.com/journal/2011/05/preview_xcode_4_support_monotouch
  - Until then, use Xcode 3 next to Xcode 4 http://blog.sublimeintervention.com/archive/2011/Apr-25.html
  - Or use the MonoDevelop 2.8 beta http://docs.xamarin.com/ios/getting_started/installation



Comparison: how do they work?

- MonoTouch monotouch
  - *Compiles into native ARM code through mtouch*
  - *No runtime*
  - Libraries
    - Core .NET
    - *Bindings for CocoaTouch APIs*
  - Tools
    - SDK + tools
    - IDE integration
- MonoDroid mono for Android
  - Compiles into *ECMA CIL byte code*
  - *Mono runtime executes the bytecode*
  - Libraries
    - Core .NET
    - *Bindings for Android/Java APIs*
  - Tools
    - SDK + tools
    - IDE integration

http://en.wikipedia.org/wiki/Mono_(software)



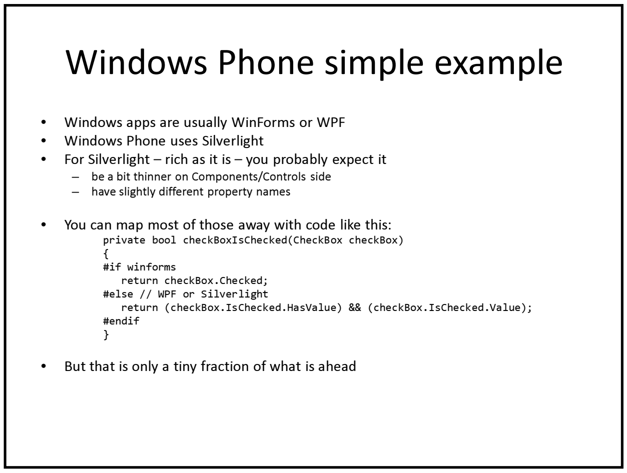
Tic Tac Toe

… or Noughts and Crosses

Example from .NET 1.1 and the first .NET CF, redone on "modern" mobile

Windows Phone

Silverlight based



Windows Phone simple example

- Windows apps are usually WinForms or WPF
- Windows Phone uses Silverlight
- For Silverlight – rich as it is – you probably expect it
  - be a bit thinner on Components/Controls side
  - have slightly different property names
- You can map most of those away with code like this:

```
private bool checkBoxIsChecked(CheckBox checkBox)
{
#if winforms
   return checkBox.Checked;
#else // WPF or Silverlight
   return (checkBox.IsChecked.HasValue) && (checkBox.IsChecked.Value);
#endif
}
```

- But that is only a tiny fraction of what is ahead

Windows Phone simple example

- Windows Phone is based on the .NET Compact Framework
  - It still lacks a lot of basic things that the regular Framework has
    - Enum.GetValues: http://stackoverflow.com/questions/5878895/enum-getvalues-in-wp7
      Workaround: http://stackoverflow.com/questions/105372/c-how-to-enumerate-an-enum/1375234#1375234

```
public static IEnumerable<T> GetValues<T>(T enumeration)
    where T : struct, IComparable, IFormattable, IConvertible
{
    List<T> enumerations = new List<T>();
    foreach (FieldInfo fieldInfo in enumeration.GetType().
       GetFields(BindingFlags.Static | BindingFlags.Public))
    {
        enumerations.Add((T)fieldInfo.GetValue(enumeration));
    }
    return enumerations;
}
```

- Time for the next thing lacking: Timers

Silverlight lacks a Timer component

- For a rich multi-media kind of framework, you'd expect the timer component to be there
- Not! :)
- Alternative

```
System.Windows.Threading.DispatcherTimer dpt = new
  System.Windows.Threading.DispatcherTimer();
     dpt.Interval = new TimeSpan(0, 0, 1);
     dpt.Tick += new EventHandler(dpt_Tick);
     dpt.Start();
```

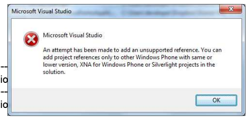
- It does not interoperate well with assemblies outside the Silverlight framework
- Now you are up for the next one step: Sharing assemblies

Windows Phone: no DoEvents

- DoEvents is dangerous
  (all sort of reentrancy problems)
  - http://blogs.msdn.com/b/jfoscoding/archive/2005/08/06/448560.aspx
- But the way the Windows Phone UI is processed, you need to be careful with combinations like this:
    - MyControl.Text = "some text";
    - MessageBox.Show("some other text");
  - The first will not display.

Windows Phone simple example

- Funny error/warning messages
  - Reason: Windows Phone targets only accept PhoneClassLibrary projects as a reference
- You think the solution will be to move all code into a PhoneClassLibrary



Microsoft Visual Studio

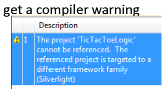
Microsoft Visual Studio

An attempt has been made to add an unsupported reference. You can add project references only to other Windows Phone with same or lower version, XNA for Windows Phone or Silverlight projects in the solution.

```
---------------------------
Microsoft Visual Studio
---------------------------
Microsoft Visual Studio

An attempt has been made to add an unsupported reference. You can add project references only to other Windows Phone with same or lower version, XNA for Windows Phone or Silverlight projects in the solution.
---------------------------
OK
---------------------------
```

- Wrong:
  - now you get a compiler warning



		Description
⚠	1	The project 'TicTacToeLogic' cannot be referenced. The referenced project is targeted to a different framework family (Silverlight)

- Worse:
  - at run-time the TicTacToeLogic assembly will not load.

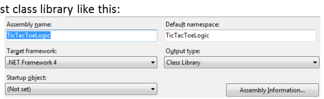
```
Warning	1		The project 'TicTacToeLogic' cannot be referenced.  The referenced project is targeted to a different framework family (Silverlight)
```

Basically, re-using code is hard

- I'm not sure why, as this kind of code re-use should be simple and painless
- Solution 1:
  - Create 2 assemblies, one of them links to the source code from the other, then manually keep them in sync
    - http://stackoverflow.com/questions/3637378/silverlight-class-library-to-windows-phone-7-class-library/3639458#3639458
  - Be prepared that sometimes (even though a C# file is changed and saved) the change will not be picked up
    - Close and reopen your solution
- Solution 2:
  - Create 2 assemblies, then use the Project Linker add-in to keep them in sync
    - Perform Solution 1
    - Link the two assemblies
    - Download: http://visualstudiogallery.msdn.microsoft.com/en-us/5e730577-d11c-4f2e-8e2b-cbb87f76c044
    - Howto: http://msdn.microsoft.com/en-us/library/ff921108(v=pandp.20).aspx
  - Still not a nice solution

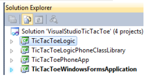
Re-using code between platforms

- Solution 3:
  - Use the "Portable Libraries Tools" add-in
    - http://visualstudiogallery.msdn.microsoft.com/b0e0b5e9-e138-410b-ad10-00cb3caf4981
    - http://msdn.microsoft.com/en-us/library/gg597391.aspx
    - http://blogs.msdn.com/b/sburke/archive/2011/02/03/using-observablecollection-with-the-portable-library-tools-ctp.aspx
  - Is not recognized by MonoDevelop (yet?)
- Solution 4:
  - Like Solution 2, but with the first class library like this:
    - .NET Framework 4
    - Class Library



Assembly name: TicTacToeLogic
Default namespace: TicTacToeLogic
Target framework: .NET Framework 4
Output type: Class Library
Startup object: (Not set)

  - This allows
    - Windows Phone to use the "Project Linker" linked library
    - MonoDevelop still refuses to use the first class library
      - It compiles fine, but referencing from a MonoTouch app is impossible
      - http://stackoverflow.com/questions/5137017/create-assembly-compatible-with-desktop-wp7-and-monotouch



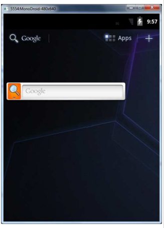
Solution 'VisualStudioTicTacToe' (4 projects)
TicTacToeLogic
TicTacToeLogicPhoneClassLibrary
TicTacToePhoneApp
TicTacToeWindowsFormsApplication

MonoDroid

C# for Android

MonoDroid simple example

- Note:
  - To run, don't use F5 (run with debugger)
    use Ctrl-F5 (run without debugger)
- Deploying
  - Sometimes fails the first time; just retry
  - You can get weird errors, for instance:
    - Unhandled exception in button_ok_Click:Task.
      See exception for details.
    - Just retry when you get these
  - The emulator takes a long while to start (even on fast hard
    - Wait until it looks like the image on the right
  - Can take forever to install the framework
  - Then suddenly works
  - Or sometimes it doesn't



- Remember this is *bleeding* edge, even when using Java :)

MonoDroid tic tac toe

- Use DroidDraw for drawing your layout
  - It is often slowwwwwww
- Use an XML editor to set properties, then copy to the "Output" window, and press "Load"
- Load that into your Layout axml file
- Then regenerate your Resource .Designer.cs file
  - http://stackoverflow.com/questions/7016167/monodroid-getting-the-resource-designer-cs-to-refresh
- This allows you to use the controls from within your code
- Like in iOS, no MessageBox.Show
  - Use a Toast as an alternative
    http://weblogs.asp.net/wallym/archive/2011/01/24/displaying-error-messages-with-android-and-monodroid.aspx

MonoDroid tic tac toe

- Timers in MonoDroid run in a separate thread
  - http://stackoverflow.com/questions/7097729/monodroid-timer
- So you need to make a thread transition

- Oh, and this great book

MonoTouch

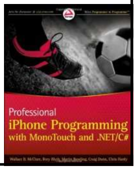
C# for iOS

MonoTouch tic tac toe

- MonoTouch uses the Xcode 4 designer:
  - very different from any of the Visual Studio 2010 designers
- iOS has no Modal dialogs, two possibilities
  - Create a new view (lot of work)
  - Abuse UIAlertView
    http://stackoverflow.com/questions/4613071/messagebox-show-and-dialogresult-equivalent-in-monotouch

MonoTouch tic tac toe

- MonoTouch wraps the iOS CocoaTouch library, platform specific controls and properties
  - UIButton
    - Title property http://stackoverflow.com/questions/5622442/programmatically-changing-button-label-in-monotouch
  - UISwitch
  - NSTimer http://monotouchexamples.com
  - NSString
  - et cetera
- Binding requires Outlets and Actions
  - Very different from Visual Studio


Professional
iPhone Programming
with MonoTouch and .NET/C#
Wallace B. McClure, Rory Blyth, Martin Bowling, Craig Dunn, Chris Hardy

Demoing on hardware

- This is always the hard part
  - It won’t connect when you want it to
  - Display capture is hard or impossible
    - iOS works on Jail Broken devices using Display Recorder
      - http://www.youtube.com/watch?v=wOc-tg0qlNA
    - Some people have iOS to work when holding down Home and pressing Sleep button
      - http://www.nkjlive.com/2011/02/capture-screen-in-iphone-without-any-app/
      - But not on my devices :(
    - Android
      - http://www.cuteandroid.com/open-source-android-apps-for-developers-android-screenshots-and-screen-capture
- Requires the non-evaluation versions of MonoTouch/MonoDroid
- Nokia Developers still have not received their Windows Phone 7 device

Some more useful links

- Read http://www.mono-droid.com often
- Learn about User Experience (UX), as it is far more than just a nice GUI
  - http://en.wikipedia.org/wiki/User_experience
- Study the Apple iOS Human Interface Guidelines, very useful outside the Apple realm too
  - http://www.google.com/search?q=MobileHIG
- Xcode 4 shortcuts
  - http://pragmaticstudio.com/media/Xcode4Shortcuts.pdf
- Delegates
  - http://www.codeproject.com/KB/cs/DelegatesOMy.aspx

Cross Platform Mobile Development on Windows 7, Android and iOS

Jeroen Pluimers - better office benelux
jpluimers@better-office.com

27.09.2011 | 11:15 - 12:30