Notes Romain Denis 335

Questions

> 1. Que veut dire l'acronyme MAUI ?
>   - Multi
>   - App
>   - User
>   - Interface
> 2. Comment ça se fait que le C# fonctionne sur Android ?
>   Le projet MAUI transforme (avec le runtime Android) pour Android
> 3. Comment tester une application MAUI sans smartphone et comme cela fonctionne-t-il donc ?
>   Avec un émulateur (machine virtuelle), qui fait croire qu'on est sur un téléphone
> 4. Citez 3 alternatives à MAUI pour faire du dev multi-platforme ?
>   1. Web
>   2. React
>   3. CodeClean
> 5. Citez le type d'application qui permetent deux types de navigation
>   Tout navigue avec app.shell
>   1. Flyout - Une page qui apparait au dessus de une autre
>   2. Tabulation - Petit tabs en bas
> 6. Avec une navigation non shell, comment est il possible de naviguer entre une page et une autre
>   Push (descendre dans le stack), pop (monter dans le stack)
> 7. Citez les 4 layouts de base et leurs contenus
>   1. Grid - Des carres
>   2. Flex - Reponsive
>   3. Stack - Met des elements a la suite verticallement ou horizontal (vertical de base)
>   4. Absolute - Toutes les positions sont mis en dur
> 8. Que veux dire MVVM?
>   Model view view model
> 9. A quoi sert l'annotation [RelayCommand] et [ObvservableProperty] ?
>   Ca vient du Community ToolKit, ca lie une variable a un type C#
> 10. Comment faire en sorte que un label affiche une valeur dynamique?
>   Sur l'attribut texte il faut metre un binding et sur le view model [ObservableProperty]
> 11. Comment faire pour que le contenu d'une liste soit savuegardé entre 2 utilisations de l'application
>   Avec une base de donnes, par example sqlite
>   En utilisant une gestion de fichier texte
> 12. A quelle frequence les donnes accelerometre sont transmises?
>   Avec des presets predefinis DEFAULT (200ms) UI(60ms) GAME(20ms) FASTEST (5ms)
> 13. Quelles axes et ce que l'accelerometre peux detecter des changements
>   X,Y,Z
> 14. En quoi et ce que les capteurs peuvement impacter negativement l'appareil
>   Utilise des resources du telephone (Batterie RAM ect)
>   Depend du nombre de capteurs et le presision d'un capteur

Sequence 1 (21/03/25)

Theory

Difference between mobile and non mobile applications

Interface: Mobile has a different interface with a smaller area to work with and it's oriented differently

Resources: Mobile needs to have a more precise way of managing resources, due to the weaker specs compared to a computer

Mobile Specific Function: Most phones have more media to work with than a computer like shake detection, camera and more

Controlled distribution: Mobile apps are typically controlled and need to go through stores to install like Apple Store, Google Play, ect

UX: There are different ways to communicate and navigate through an app

Technologies, advantages and disadvantages

Web

Specific native

- Using native language on each platform
  - Android → Java
  - iOS → Objective-C, Swift

Linked native

- Development in another language, but aiming a specific platform
  - Python (Kivy)
  - Kotlin

Natif with runtime

- Development in a language translated into the native language via runtime
  - .NET MAUI (Mono)
  - React (JavaScript)
  - Flutter

Avantages and disadvantages

- Development with runtime is less efficient than native specifique development
- Time is gained avoiding to learn multiple languages

Accelerometer in HTML

- The accelerometer can be used in HTML using Javascript APIs

Project

Start with the code or the storyboard

- A good schema decreases the risk of producing bad code
- Investing time into a schema allows for a clear vision of the frontend and

backend functionalities

Visual Studio

Setup of the environment

- Use Visual Studio 2022.
- Modify Visual Studio, tick `.NET Multi-platform App UI Development`.
- Open VS2022 and create a MAUI project.
- Accept the licence
- Execute the script that runs the emulator
- Execute the app from VS

Debug on phone

- Go to `Settings > Developer Settings > Activate Debug (Developer mode)`.
- Execute from VS2022.

Discussions

- Do not stock on external disks or on the cloud (too slow and too big)
- Deployment is possible on emulator and physical phone
- Physical phone is needed to test phyiscal features (shaking the phone, ect)

Project structure

- **MainProgram.cs** : Technical, don't touch
- **App.xaml / App.xaml.cs** :
  - `.xaml` graphical section (vue)
  - `.xaml.cs` code assosiated with the view
  - Calls `AppShell` don't touch
- **AppShell.xaml** : Manages the navigation
- **MainPage.xaml** : Main page showing `Hello world` originally

XAML

Définition

XML with beacons definied in XML namespaces

- **X**tensible
- **A**pplication
- **M**arkup

- **L**anguage

XAML basics

- Opening and closing beacons (like HTML)

- Beacons with attributes and content (without content = autoclosing beacon `/>`)

- Structured hierachy (russian dolls)

- Direct translation possible from french to XAML

Sequence 2 (28/03/25)

Theorie

Shell

Types de page:

Content Page

- Some content

Flyout Page

- Page that pops out of the side (like a menu)

Tabbed Page

- Little icons at the bottom
- What happens when they are no more space for tabs ?
  - Three little dots with "more"

Navigation Page

- With a bar at the top

Types of MAUI APPS

1. Basic

- No navigation menu

2. Shell (default on a new app)

- Has a navigation or flyout system

Adding a new page

While in a MAUI project, go to project > add a new class > MAUI pages > content page (XAML)

Navigation

- We will be stacking pages
- Every page has a back button to go down the proverbial stack
- All pages before the current page will be generated, so it will be much more

resource intensive
- Going up a page in the stack is called a push, and the opposite is a pop
  - Example: a back button's code would be await `Navigation.PopAsync();`

Layouts

There will be a header main left right and footer sections

Layout templates

Stack Layout

- To stack different layouts
- Can put a absolute inside a grid inside the header ect.

Absolute Layout

- Using absolute values
- Tell every element exaclty where to be
- Can have mutliple elements overlapping
- Not responsive
- Takes a long time

Grid Layout

- Define the number of rows and columns
- Self explantory

Flex Layout

- Like CSS (In the manner that it is responsive)

Sequence 3 (04/04/25)

Theory

CRUD

How to program user interactions in MAUI?

1. Code behind

- Spaghetti code
- Hard to test
- Completly manual

2. MVVM (Model view view model)

- Understanble
- Extendable
- Easy to test

Model : C#
View: XAML

Quirk:
Model library imposes that methods have a capital letter while declaring it is in lowercase

For example :

`Not in method: private int counter = 0;
In method: Counter+= 1;`

This is really bad practice but it is what it is

Project

Nothing this week

Sequence 4 (11/04/25)

Debug

Using Trace.Command() traces something in the output file

Sequence 5 (25/04/25)

Continuation de projet

Sequence 6 (02/05/25)

Continuation de projet

Sequence 7 (09/05/25)

Continuation de projet

Sequence 8 (16/05/25)

Animations

- Purpose: make app feel dynamic
- Types:
  - Fade → FadeTo(opacity)
  - Rotate → RotateTo(angle)
  - Scale → ScaleTo(size)
  - Translate → TranslateTo(position)
- All return Task, use with async/await
- Only View (code-behind) touches UI

MVVM method:

- ViewModel:
  - Has Action<int> → RotateBoxUIAction, MoveUIAction
  - Calls: RotateBoxUIAction?.Invoke(angle)

Example in exersice:

- Switch → starts/stops rotation
- Slider → controls speed
- Buttons << and >> → move box (left and right)
- Uses Easing.SpringOut
- Needs `if (box != null)` to avoid crash

Problems i had

- Animation doesn’t run

  - Check `if (box != null)`
  - Made the Action = `set (vm.RotateBoxUIAction = RotateUI;)`

- UI freezes during animation

  - Need to use await aync `await box.RotateTo(...)`

- Nothing happens on button press

  - Bind the command properly `{Binding MoveBoxCommand}`

- Animation jumps instead of moving smoothly
  - Reset _currentX correctly
  - Used easing (ex: `Easing.SpringOut`)

Sensors

- Namespace: Microsoft.Maui.Devices.Sensors
- Measures acceleration on:
  - X: left/right
  - Y: up/down
  - Z: front/back
- Values in G (m/s^2)

SensorSpeed:

- Default: 200ms
- UI: 60ms
- Game: 20ms
- Fastest: 5ms

Events:

- ReadingChanged → live data
- ShakeDetected → detects shake

MVVM usage:

- Properties: XValue, YValue, ZValue, Status
- Commands: StartMonitoring, StopMonitoring
- Values updated on OnReadingChanged
- Bound to UI labels in XAML

Shake-only version:

- Uses ShakeDetected
- Triggers animation (ex: ScaleTo() label)

Best practices:

- Stop sensor in OnDisappearing()
- Request permission (AndroidManifest)

Problems i had

- No data showing

  - Checked _Accelerometer.Default.IsSupported_

  - Ensured _StartMonitoring()_ was called

- App crashes on close

  - Stop sensor in _OnDisappearing()_

- Shake not detected

  - Used _SensorSpeed.Game_ for higher sensitivity

- Permission errors on Android

  - Add to AndroidManifest.xml:

  ```bash
    <uses-feature android:name="android.hardware.sensor.accelerometer"
android:required="true" />
  ```

- Lag in value display
  - Use _MainThread.BeginInvokeOnMainThread()_ to update UI