SCIENCES SORBONNE UNIVERSITÉ

WatchOS 5, the basics

Fabrice.Kordon@lip6.fr

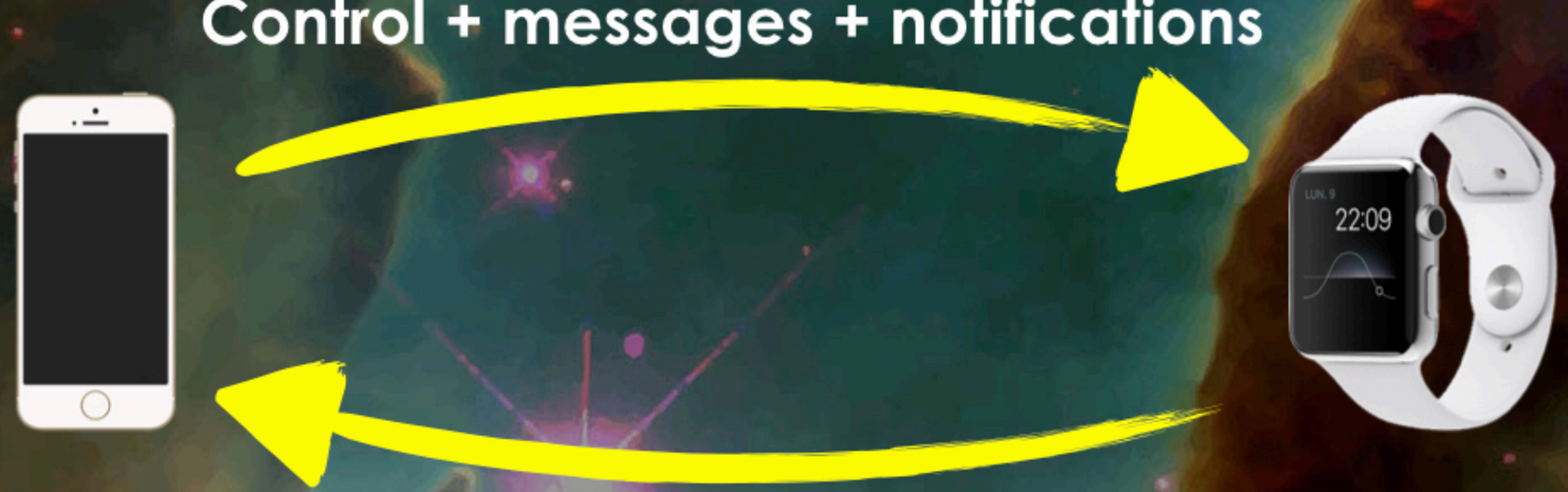
LIP6

The watch

Companion of a phone

- WatchOS 1 ➠ exécution on the device + display on the watch
- WatchOS 2+ ➠ execution on the watch itself
  (image/sounds can be located in the watch)

Interactions



About the Watch

Watch 1 to 3

@2x

38mm
340x272px
170x136pts

42mm
390x312px
195x156pts

Watch 4

@2x

40mm
394x324px
197x162pts

44mm
448x368px
224x184pts

About the Watch

Watch 1 to 3

@2x

38mm
340x272px
170x136pts

42mm
390x312px
195x156pts

Watch 4

watch 4 brings a trap!

Safe area (round corners)

394x324px
197x162pts

44mm
448x368px
224x184pts

Consequences

Difficult to escape Storyboard

- Apparently
  - **You must set everything relatively**

Interaction inspired from iOS

- The screen is rather small
  - **Simplified widgets**
  - **Be careful when designing the interface**
- Dedicated interactivity
  - **No way to reuse the on from your iPhone**
- Power consumption is even more a concern
  - **Numerous restrictions to iOS mechanisms**

    e.g. for geolocation, use requestLocation()

Principles

Entry points for a «watch app»

- main, notification, complications

in Xcode

- Create a new target
  - **As for extensions**
- Use of similar tools
  - **Storyboard**
  - **Simulator**
  - **etc.**

API & frameworks

- Derived from this of iOS
  - **Prefix = WK**
  - **Restrictions apply**

Creating a watch app in Xcode

Structure of a watch app

The embedded part

- Storyboard files
- Images (embedded)
- info.plist
- etc.

The extension part

- Interface controller (WKInterfaceController)
  - **Possibly several (if views are pushed)**
- Extension delegate (WKExtensionDelegate protocol)
  - **Corrresponds to the AppDelegate**
- Notification controller (WKUserNotificationInterfaceController)
  - **Optional**
- Complication controller (CLKComplicationDataSource protocol)
  - **Optional**

WatchKit

WKInterfaceController

- Main class to be implemented (for the main controller)

WKInterfaceDevice

- currentDevice
- screenBounds
- screenScale
- currentLocale
- Etc.

WKInterfaceObject

- For the interface mechanisms

Some protocols

- WKExtensionDelegate, WKUserNotificationInterfaceController, CLKComplicationDataSource

WatchKit interface elements

- **Inherited from WKInterfaceObject**
- WKInterfaceButton
- WKInterfaceLabel / WKInterfaceDate
- WKInterfaceGroup
- WKInterfaceImage
- WKInterfaceMap
  - **interaction contraints + limit on annotations**
- WKInterfaceSlider
- WKInterfaceSwitch
- WKInterfacePicker (predefined variants)
- WKInterfaceTable (simplified)
- WKInterfaceTimer (new)
  - **also includes a countdown**
- WKInterfaceSeparator
- WKInterfaceActivityRing

WatchKit interface elements

Inherited from WKInterfaceObject

Common attributes

height, width, alpha, hidden…

te

WKInterfaceImage

KInterfaceMap

teraction contraints + limit on annotations

terfaceSlider

Main menu

Up to four items

ed variants)

H)

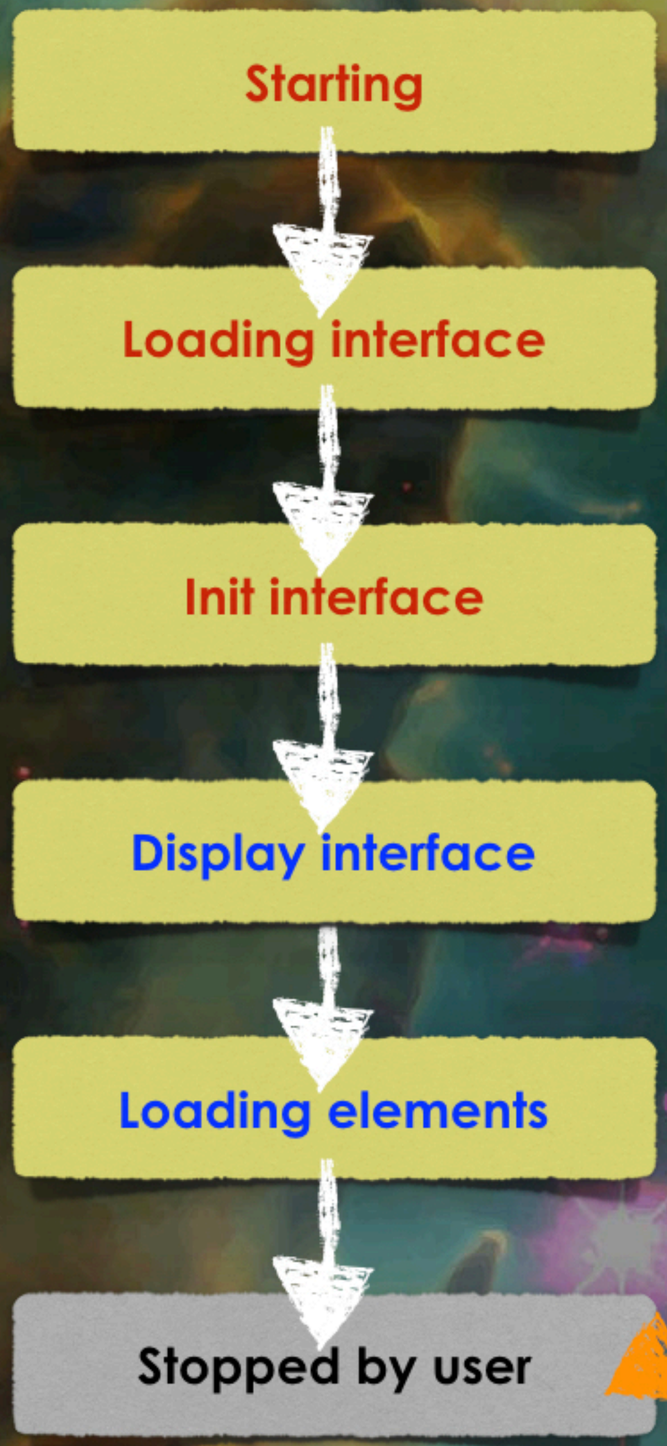
terfaceTimer (new)

lso includes a countdown

KInterfaceSeparator

WKInterfaceActivityRing

The watch app lifecycle



As a conclusion…

Process rather similar to the one of iOS apps

- Despite numerous changes
  - **Widget behave differently**
  - **Reduced interactivity**
  - **Low power CPU & interface**
- WKGestureRecognizer exists
  - **LongPress, Pan, Swipe, Tap**
  - **New events Crown handling**

Ready for an example?

As a conclusion…

Process rather similar to the one of iOS apps

- Despite numerous changes
  - **Widget behave differently**
  - **Reduced interactivity**
  - **Low power CPU & interface**
- WKGestureRecognizer exists
  - **LongPress, Pan, Swipe, Tap**
  - **New events Crown handling**

Ready for an example?