SCIENCES SORBONNE UNIVERSITÉ

Timers

Fabrice.Kordon@lip6.fr

LIP6

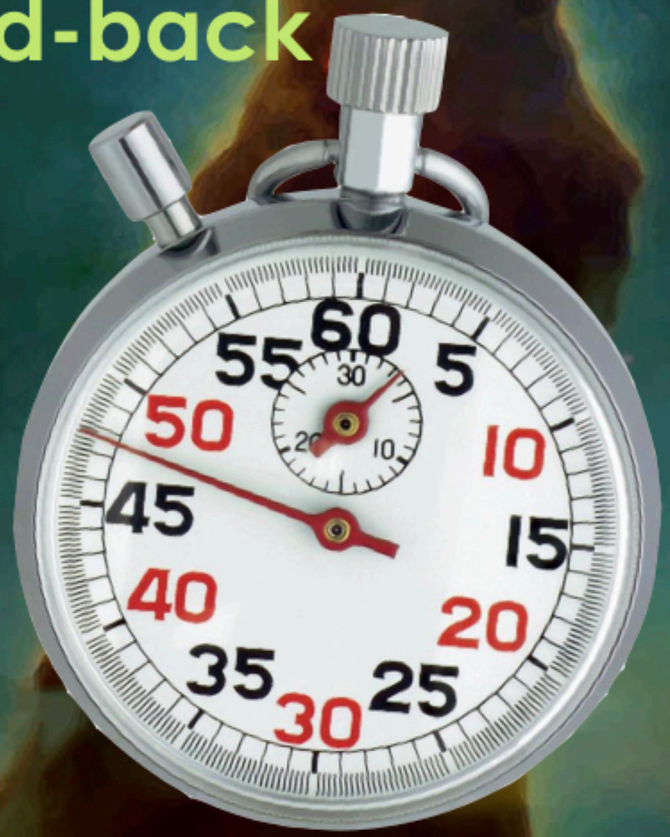
As an introduction…

Timers are very useful

- Clock interrupt in OS
- When one needs to take-over the execution
  - **Like in a game**
  - **To update something regularly**
  - **etc.**

Need for a method to be called-back

- Within some time interval
- At a given date & time (alarm-clock)
- Etc.



As an introduction…

Timers are very useful

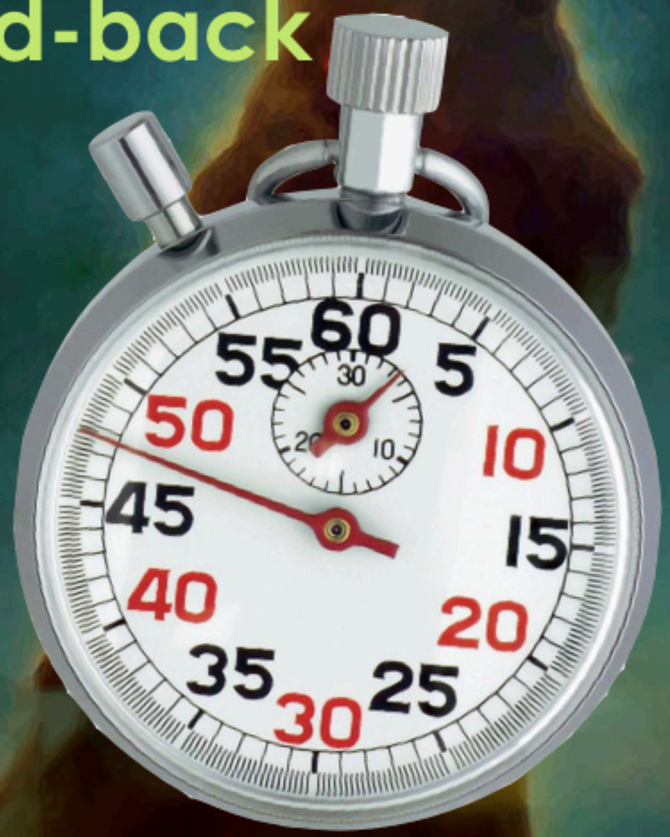
- Clock interrupt in OS
- When one needs to take-over the execution
  - **Like in a game**
  - **To update something**
  - **etc.**

In iOS

Timer / NSTimer

Need for a method to be called-back

- Within some time interval
- At a given date & time (alarm-clock)
- Etc.



Principle

Trigger of a method at a given date

- Unique
  - Activation + trigger and automated invalidation
- Repeated
  - Activation + periodic trigger
  - Explicit invalidation

What to do when the event triggers?

- Invoke a method having the following format
  - (Timer) -> Void
- Invoke a target
  - No format required

Caution

- Timers stop when the App is in background
  - Might be a few events left in background

Timer, how does it work?

Start a timer

- Insertion in the «run loop» of the current thread

```
class func scheduledTimer(timeInterval ti: TimeInterval,
                          target aTarget: Any,
                          selector aSelector: Selector,
                          userInfo: Any?,
                          repeats yesOrNo: Bool) -> Timer

class func scheduledTimer(withTimeInterval interval: TimeInterval,
                          repeats: Bool,
                          block: @escaping (Timer) -> Void) -> Timer
```

Stop a timer

- Mandatory for repeated events
  - **invalidate()**
  - **Usually followed by an assign to nil**
- Remind, automated for unique events
  - **Once they fired**

Timer, how does it work?

Start a timer

- Insertion in the «run loop» of the current thread

```
class func scheduledTimer(timeInterval ti: TimeInterval,
                          target aTarget: Any,
                          selector aSelector: Selector,
                          userInfo: Any?,
                          repeats yesOrNo: Bool) -> Timer

class func scheduledTi…withTimeInterval interval…val,
                                                  Timer
```

Objective-C?

Exactly the same
(name of functions change a bit)

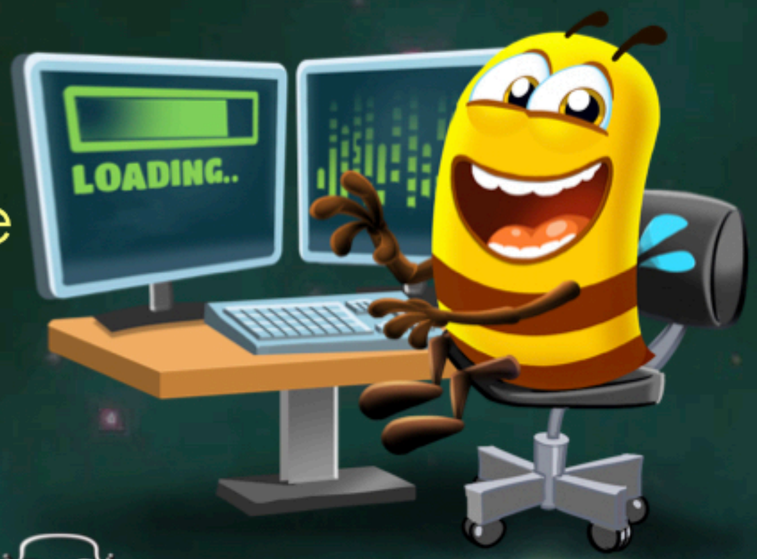
Stop a tim…

- Mandatory for repeated events
  - **invalidate()**
  - **Usually followed by an assign to nil**
- Remind, automated for unique events
  - **Once they fired**

As a conclusion…

Many more usages

- Only the more useful mentioned here
  - **You will need it for your next exercice**



But…

- You have the