

# «PlaySound»


Fabrice.Kordon@lip6.fr



# Goal of the example

2

## Illustrate AVAudioPlayer

-  Show several situations
  - ▶ Correct file
  - ▶ File with problems
  - ▶ Bad file (not music)
  - ▶ Several sounds in parallel

## Minimalist example

-  To let you understand...

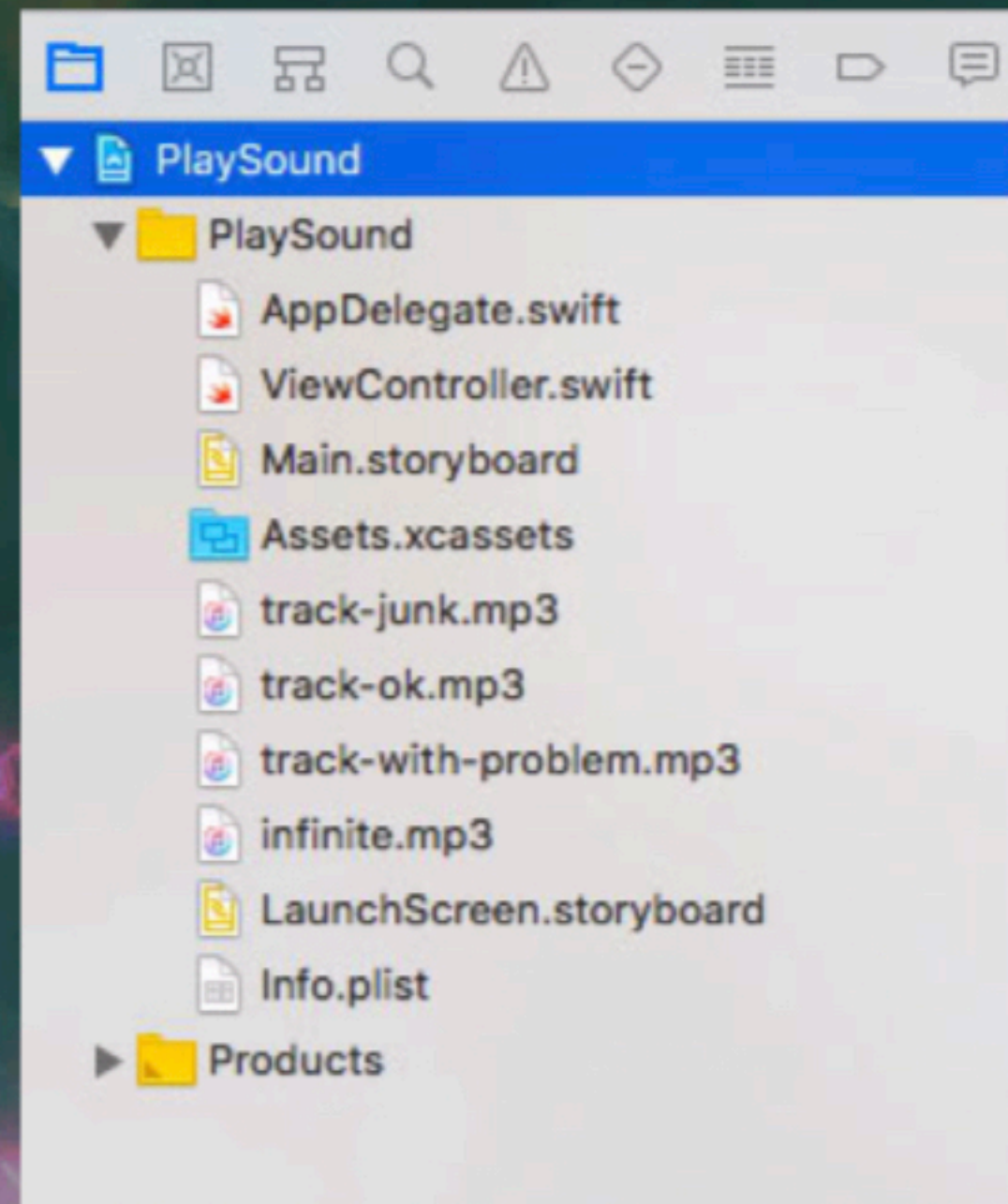
# Demo



# Inside the project

## Music files of course...

- Remember to copy these items when dragging in



# ViewController



**Sake of simplicity...**

Code located in a  
ViewController

# ViewController

```
import UIKit
import AVFoundation // do not forget it

class ViewController: UIViewController, AVAudioPlayerDelegate {

    private let t = UITextView()
    private let b1 = UIButton(type: .system)
    private let b2 = UIButton(type: .system)
    private let b3 = UIButton(type: .system)
    private let b4 = UIButton(type: .system)

    private var myPlayer1 : AVAudioPlayer?
    private var myPlayer2 : AVAudioPlayer?
```

# ViewController

```
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.
    self.view = UIView()
    self.view.backgroundColor = UIColor.white
    let w = UIScreen.main.bounds.size.width

    t.isSelectable = false
    t.frame = CGRect(x: 10.0, y: 50.0, width: w - 20.0, height: 40.0)
    t.text = "Information to come"
    t.textAlignment = .center
    self.view.addSubview(t)

    b1.frame = CGRect(x: 100.0, y: 100.0, width: w - 200.0, height: 25.0)
    b1.setTitle("Track OK", for: .normal)
    b1.addTarget(self, action: #selector(trackOK), for: .touchDown)
    self.view.addSubview(b1)

    b2.frame = CGRect(x: 100.0, y: 140.0, width: w - 200.0, height: 25.0)
    b2.setTitle("Junk Track", for: .normal)
    b2.addTarget(self, action: #selector(trackPb1), for: .touchDown)
    self.view.addSubview(b2)

    b3.frame = CGRect(x: 50.0, y: 180.0, width: w - 100.0, height: 25.0)
    b3.setTitle("Track with problems", for: .normal)
    b3.addTarget(self, action: #selector(trackPb2), for: .touchDown)
    self.view.addSubview(b3)

    b4.frame = CGRect(x: 50.0, y: 220.0, width: w - 100.0, height: 25.0)
    b4.setTitle("Parallel Tracks", for: .normal)
    b4.addTarget(self, action: #selector(twoTracks), for: .touchDown)
    self.view.addSubview(b4)
}
```

# ViewController

```
func playSound (name : String, ext : String) { // Service function
    t.text = ""
    let url = Bundle.main.url(forResource: name,
                              withExtension: ext)
    if url != nil {
        do { // let's handle an exception there...
            myPlayer1 = try AVAudioPlayer(contentsOf: url!)
            t.text = "Decoding \"\"(name).mp3\"\""
            myPlayer1!.delegate = self
            myPlayer1!.play()
            b1.isHidden = true
            b2.isHidden = true
            b3.isHidden = true
            b4.isHidden = true
        } catch let error { // Exception of type Error
            t.text = "Problem : \"\"(error.localizedDescription)\"\""
        }
    } else {
        t.text = "missing file"
    }
}

@objc func trackOK () {
    self.playSound(name: "track-ok", ext: "mp3")
}

@objc func trackPb1 () {
    self.playSound(name: "track-junk", ext: ".mp3")
}
```



# ViewController

```
@objc func trackPb2 () {
    self.playSound(name: "track-with-problem", ext: ".mp3")
}

@objc func twoTracks () {
    self.playSound(name: "track-ok", ext: "mp3")
    // We use teh second player for two sounds in parallel
    let url = Bundle.main.url(forResource: "infinite",
                               withExtension: ".mp3")

    if url != nil {
        do { // let's handle an exception there...
            myPlayer2 = try AVAudioPlayer(contentsOf: url!)
            t.text = t.text+" and \"infinite.mp3\""
            myPlayer2!.delegate = self
            myPlayer2!.play()
            b1.isHidden = true
            b2.isHidden = true
            b3.isHidden = true
            b4.isHidden = true
        } catch let error { // Exception of type Error
            t.text = "Problem : \(error.localizedDescription)"
        }
    } else {
        t.text = "missing file"
    }
}
}
```

# ViewController

```
// AVAudioPlayerDelegate protocol

func audioPlayerDidFinishPlaying(_ player: AVAudioPlayer,
                                 successfully flag: Bool) {
    if flag {
        t.text = "End of track"
    } else {
        t.text = "Unexpected interruption"
    }
    b1.isHidden = false
    b2.isHidden = false
    b3.isHidden = false
    b4.isHidden = false
}

func audioPlayerDecodeErrorDidOccur(_ player: AVAudioPlayer,
                                     error: Error?) {
    t.text = "Read error \(error?.localizedDescription ?? "unknown error")"
}
}
```

# As a conclusion...

## This is a basic example

- You may control and check for data while reading
  - ▶ With a dedicated timer
- You may also play files from a distant URL
  - ▶ Many possibilities...



## Next step?

- Access the device's music library
- To be presented later

