SCIENCES SORBONNE UNIVERSITÉ

Views, the parallax effect

Fabrice.Kordon@lip6.fr

LIP6

As an introduction…

Introduced with iOS7

- Provides a «deep space» impression
- Nice for games

Small impact on the autonomy

- Movement co-processor involved
  - **Since iPhone 5s/iPad air/Ipad mini retina**

How to access such mechanisms

- UIDynamicItem (et UIDynamicItem protocole)
- UIMotionEffect
  - **Customizable 3D effects**
- UIInterpolationMotionEffect
  - **predefined 3D effects (e.g. parallax effect)**

Enrichment of UIView

Major update

- Property : motionEffects
  - **Array of UIMotionEffect**
- Adding and suppressing effects

Effects can be reused

- You must associate an effect to all the involved views

Observing the parallax effect

Principles

I — elaborate an effect

- Use of UIInterpolatingMotionEffect
  - **Several possible effects**
- For the parallax one
  - **UIInterpolatingMotionEffectTypeTiltAlongHorizontalAxis**
    TiltAlongHorizontalAxis
  - **UIInterpolatingMotionEffectTypeTiltAlongVerticalAxis**
    TiltAlongVerticalAxis
- Do not forget its amplitude
  - **20 pts recommended**

II — add these effects to the view

- addMotionEffect: / addMotionEffect

iOS does it for you then

Setting up the parallax effect


```
private let label = UILabel()

convenience init(frame : CGRect) {
    self.init()
    let img = UIImage(named: "Encelade")
    let imgv = UIImageView(image: img)
    let scale = frame.height / img!.size.height
    imgv.frame = CGRect(x: 0.0, y: 0.0,
                        width: img!.size.width * scale + 50.0,
                        height: img!.size.height * scale + 50.0)
    self.view = UIView()
    self.view.addSubview(imgv)
    let effectH = UIInterpolatingMotionEffect(keyPath: "center.x",
                                              type: .tiltAlongHorizontalAxis)
    effectH.minimumRelativeValue = -50.0 // 20 recommended by Apple
    effectH.maximumRelativeValue = 50.0
    let effectV = UIInterpolatingMotionEffect(keyPath: "center.y",
                                              type: .tiltAlongVerticalAxis)
    effectV.minimumRelativeValue = -50.0 // 20 recommended by Apple
    effectV.maximumRelativeValue = 50.0
    imgv.addMotionEffect(effectH)
    imgv.addMotionEffect(effectV)
    label.text = "Encelade"
    label.textColor = UIColor.black
    label.textAlignment = .center
    label.font = UIFont.boldSystemFont(ofSize: 28.0)
    self.view.addSubview(label)
    self.displayLabel(frame.size)
}
```

Setting up the parallax effect


```
private let label = UILabel()

convenience init(frame : CGRect) {
    self.init()
    let img = UIImage(named: "Encelade")
    let imgv = UIImageView(image: img)
    let scale = frame.height / img!.size.height
    imgv.frame
    self.view
    self.view
    let effec

    effectH.m
    effectH.m
    let effec

    effectV.minimumRelativeValue = -50.0 // 20 recommended by Apple
    effectV.maximumRelativeValue = 50.0
    imgv.addMotionEffect(effectH)
    imgv.addMotionEffect(effectV)
    label.text = "Encelade"
    label.textColor = UIColor.black
    label.textAlignment = .center
    label.font = UIFont.boldSystemFont(ofSize: 28.0)
    self.view.addSubview(label)
    self.displayLabel(frame.size)
}
```

In Objective-C

```
UIInterpolatingMotionEffect *vMotionEffect =
[[UIInterpolatingMotionEffect alloc]
 initWithKeyPath:@"center.y"
 type:UIInterpolatingMotionEffectTypeTiltAlongVerticalAxis];
[vMotionEffect setMinimumRelativeValue:@(-50)];
[vMotionEffect setMaximumRelativeValue:@(+50)];
...
[myView addMotionEffect:vMotionEffect];
```

Groupping effects

E.g. effects on two dimensions?

- UIMotionEffectGroup
- Array of UIMotionEffect
  - **Same behavior**

```
let img = UIImage(named: "Encelade")
let imgv = UIImageView(image: img)
...
self.view.addSubview(imgv)
let effectH = UIInterpolatingMotionEffect(keyPath: "center.x",
                                          type: .tiltAlongHorizontalAxis)
effectH.minimumRelativeValue = -50.0 // 20 recommended by Apple
effectH.maximumRelativeValue = 50.0
let effectV = UIInterpolatingMotionEffect(keyPath: "center.y",
                                          type: .tiltAlongVerticalAxis)
effectV.minimumRelativeValue = -50.0 // 20 recommended by Apple
effectV.maximumRelativeValue = 50.0
let group = UIMotionEffectGroup()
group.motionEffects = [effectH, effectV]
imgv.addMotionEffect(group)
```

Handling some perspective

Possible at low cost

- Useful in games
- Perspective effect
  - **Move of planes with different amplitudes**

As a conclusion…

- You can have your apps look smarter
- It's easy
- But  ..
  - «vomit effect»
  - Increase (a bit) power consumption