

Views, the blur effect

Fabrice.Kordon@lip6.fr



As an production...

2



Blur effect

- Often used in iOS
- Various types of blur
 - ▶ Light
 - ▶ ExtraLight
 - ▶ Dark



Relies on UIVisualEffectView

- Variant of «classical» views

Principles

I — Build a `UIVisualEffectView`

- Associate the desired effect
 - ▶ `UIBlurEffect`
 - ▶ `UIVibrancyEffect`

II — add the view in front of them to blur

- Hidden property is useful

Observing the blur effect



Programming a blur

```
private let label = UILabel()
private let button = UIButton(type : .system)
private var blurV : UIVisualEffectView?
private var blurVL : UIVisualEffectView?
private var blurVD : UIVisualEffectView?
```

Programming a blur

```
convenience init(frame : CGRect) {
    self.init()
    let img = UIImage(named: "Encelade")
    button.setTitle("blur", for: .normal)
    button.addTarget(self, action: #selector(ViewController4.blur),
                    for: .touchDown)

    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,
                       height: img!.size.height * scale)

    self.view = UIView()
    let blur = UIBlurEffect(style: .light)
    let blurL = UIBlurEffect(style: .extraLight)
    let blurD = UIBlurEffect(style: .dark)
    blurV = UIVisualEffectView(effect: blur)
    blurVL = UIVisualEffectView(effect: blurL)
    blurVD = UIVisualEffectView(effect: blurD)
    blurV!.isHidden = true
    blurVL!.isHidden = true
    blurVD!.isHidden = true
    label.text = "Encelade"
    label.textColor = UIColor.white
    label.textAlignment = .center
    label.font = UIFont.boldSystemFont(ofSize: 28.0)
    self.view.addSubview(imgv)
    self.view.addSubview(blurV!)
    self.view.addSubview(blurVL!)
    self.view.addSubview(blurVD!)
    self.view.addSubview(label)
    self.view.addSubview(button)
    self.displayLabel(frame.size) // set-up positions
}
```

Programming a blur

```
@objc func blur() {  
    if blurV!.isHidden {  
        blurV!.isHidden = false  
        blurVL!.isHidden = false  
        blurVD!.isHidden = false  
        label.text = "blured Encelade"  
        button.setTitle("Unblur", for: .normal)  
    } else {  
        blurV!.isHidden = true  
        blurVL!.isHidden = true  
        blurVD!.isHidden = true  
        label.text = "Encelade"  
        button.setTitle("Blur", for: .normal)  
    }  
}
```

Programming a blur

```
@objc func blur() {  
    if blurV!.isHidden {  
        blurV!.isHidden = false  
        blurVL!.isHidden = false  
        blurVD!.isHidden = false  
        label.text = "blured Encelade"  
        button.setTitle("Unblur", for: .normal)  
    } else {  
        blurV!.isHidden = true  
        blurVL!.isHidden = true  
        blurVD!.isHidden = true  
        label.text = "  
        button.setTitl  
    }  
}
```




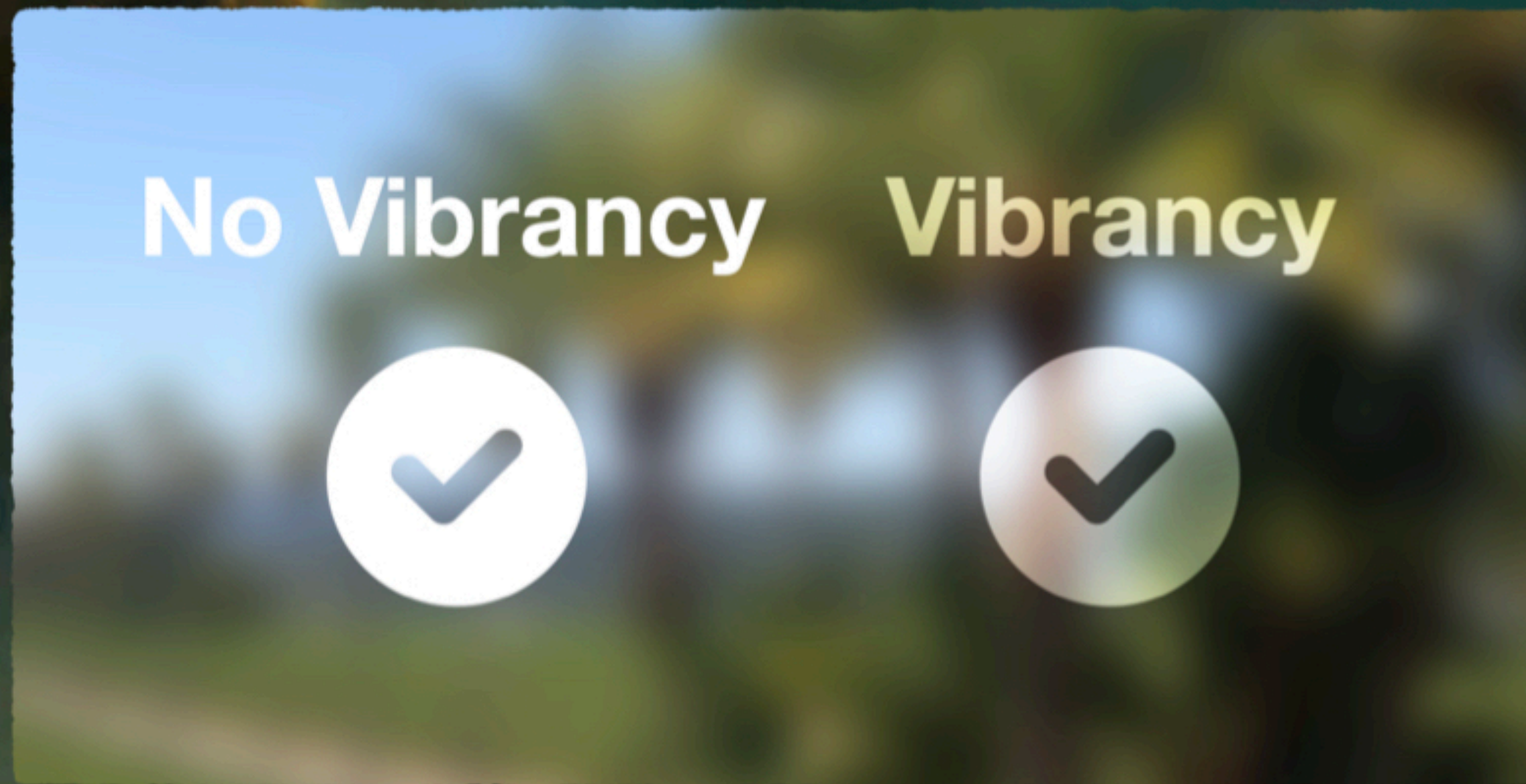
Exercise...

Translate it in Objective-C

UI Vibrancy Effect

Adjustment of text and images

-  Increase contrast (useful when background is unknown)



This is a dedicated effect

-  Works exactly like the blur effect

As a conclusion...

You have it «for free»

- Part of the graphical chart since iOS7
- You must use it appropriately

