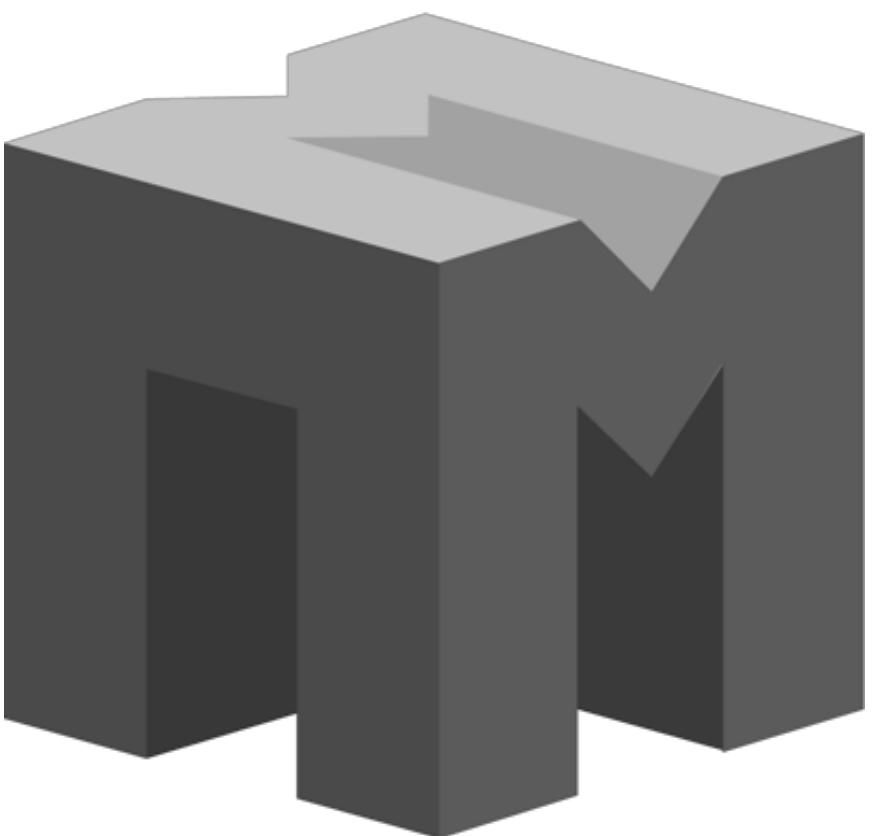


E D U C
S M I F
N O I T A

iOS Школа



E D U C A T I O N

Surf

Владислав Крупенько

User Interface

User Interface

Autolayout

Коллекции данных

Состояния экранов

User Interface

Autolayout

Коллекции данных

Состояния экранов

UIKit и с чем его едят

UIKit framework предоставляет необходимую инфраструктуру для ваших iOS приложений

Стилизация контроллов

My text that didn't get validated

Стилизация контроллов



Name

Your given name



Email

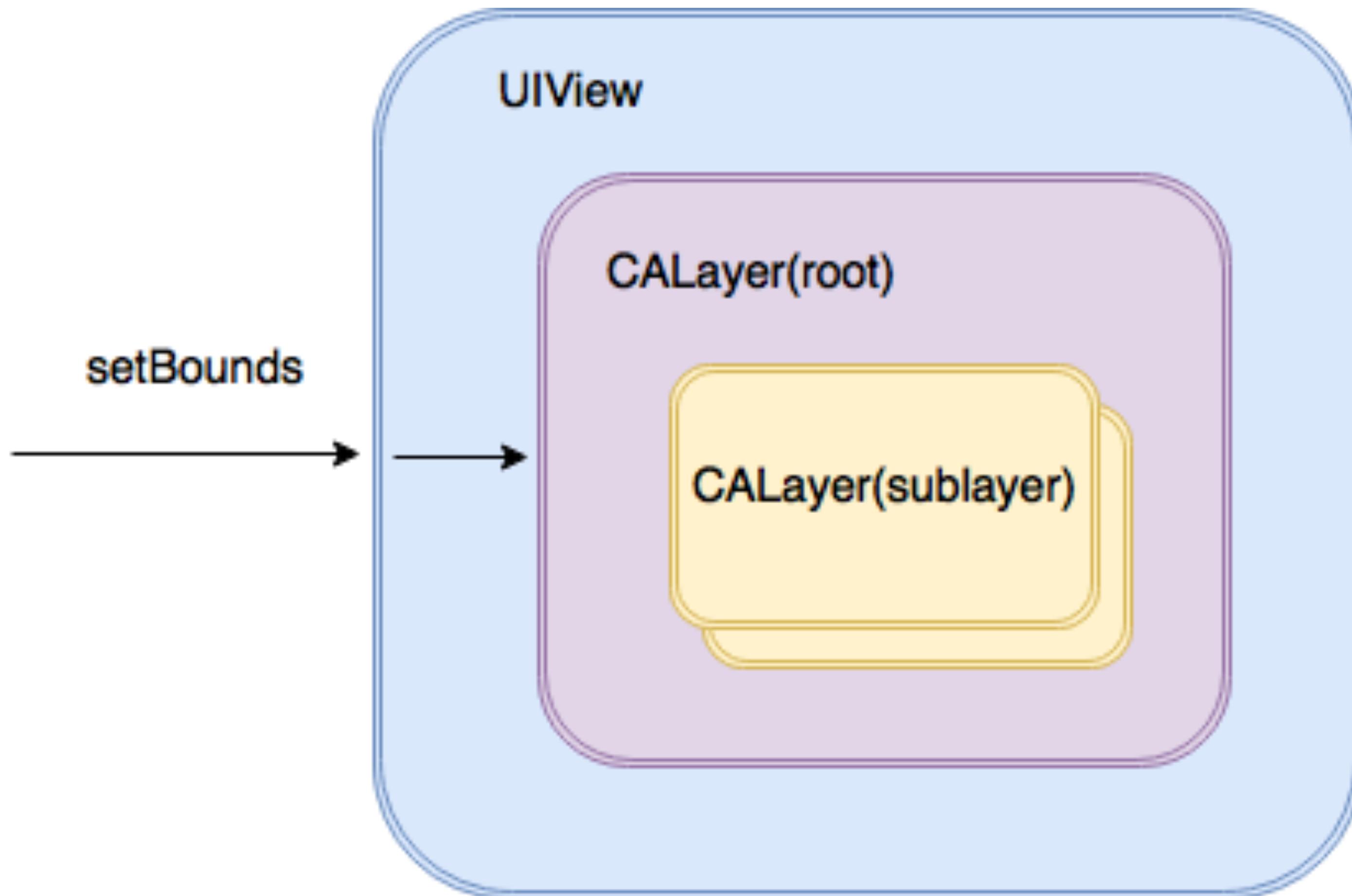
Стилизация контроллов

- Не изобретать велосипед
- Делать нативные приложения
- Использовать компоненты из SDK
- Не наследуйте
- Используйте контроллы от Apple

Категории элементов интерфейса

- Bars
- Views
- Controls

UIView



Обязанности UIView

- Отрисовка и анимация
- Layout и управление Subviews
- Обработка событий

Running MyPlayground

Live View > MyPlayground.playground (Live View)

```
1 //: A UIKit based Playground for presenting user interface
2
3 import UIKit
4 import PlaygroundSupport
5
6 class MyViewController : UIViewController {
7     override func loadView() {
8         let view = UIView()
9         view.backgroundColor = .white
10
11     let rectangle = UIView(frame: CGRect(x: 50, y: 50, width: 150,
12                                         height: 150))
13     rectangle.backgroundColor = .red
14     rectangle.clipsToBounds = false
15     view.addSubview(rectangle)
16     self.view = view
17
18     let childRectangle = UIView(frame: CGRect(x: -10, y: -10,
19                                             width: 50, height: 50))
20     childRectangle.backgroundColor = .yellow
21     rectangle.addSubview(childRectangle)
22 }
23 // Present the view controller in the Live View window
24 PlaygroundPage.current.liveView = MyViewController()
25
```

The screenshot shows the Xcode playground interface. On the left, the code editor displays the playground code. In the center, the live view window shows a red rectangular view with a smaller yellow square overlapping its top-left corner. To the right of the live view, a sidebar lists the views created in the code, each with a disclosure triangle icon.

clipsToBounds

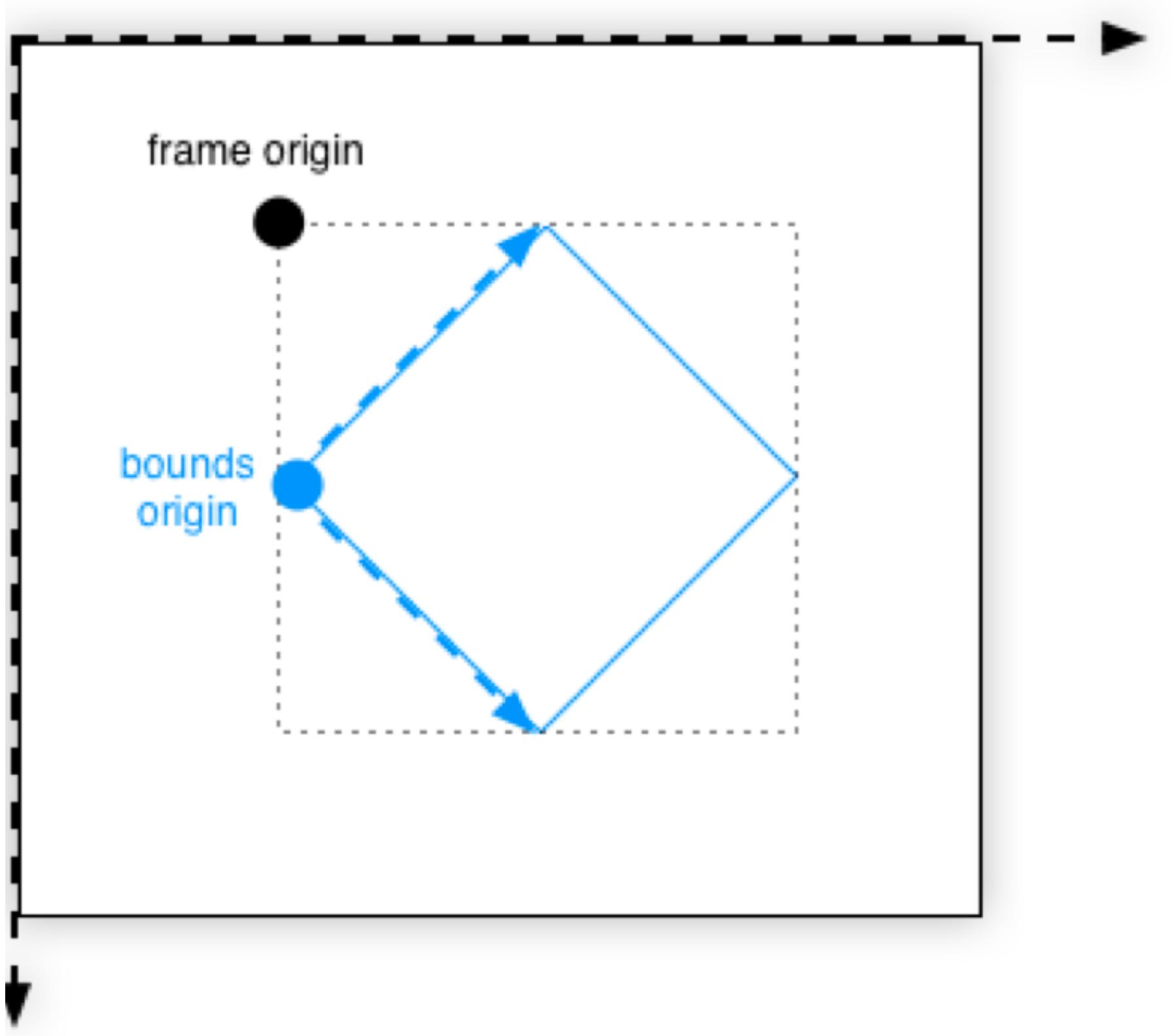


False



True

Геометрия UIView



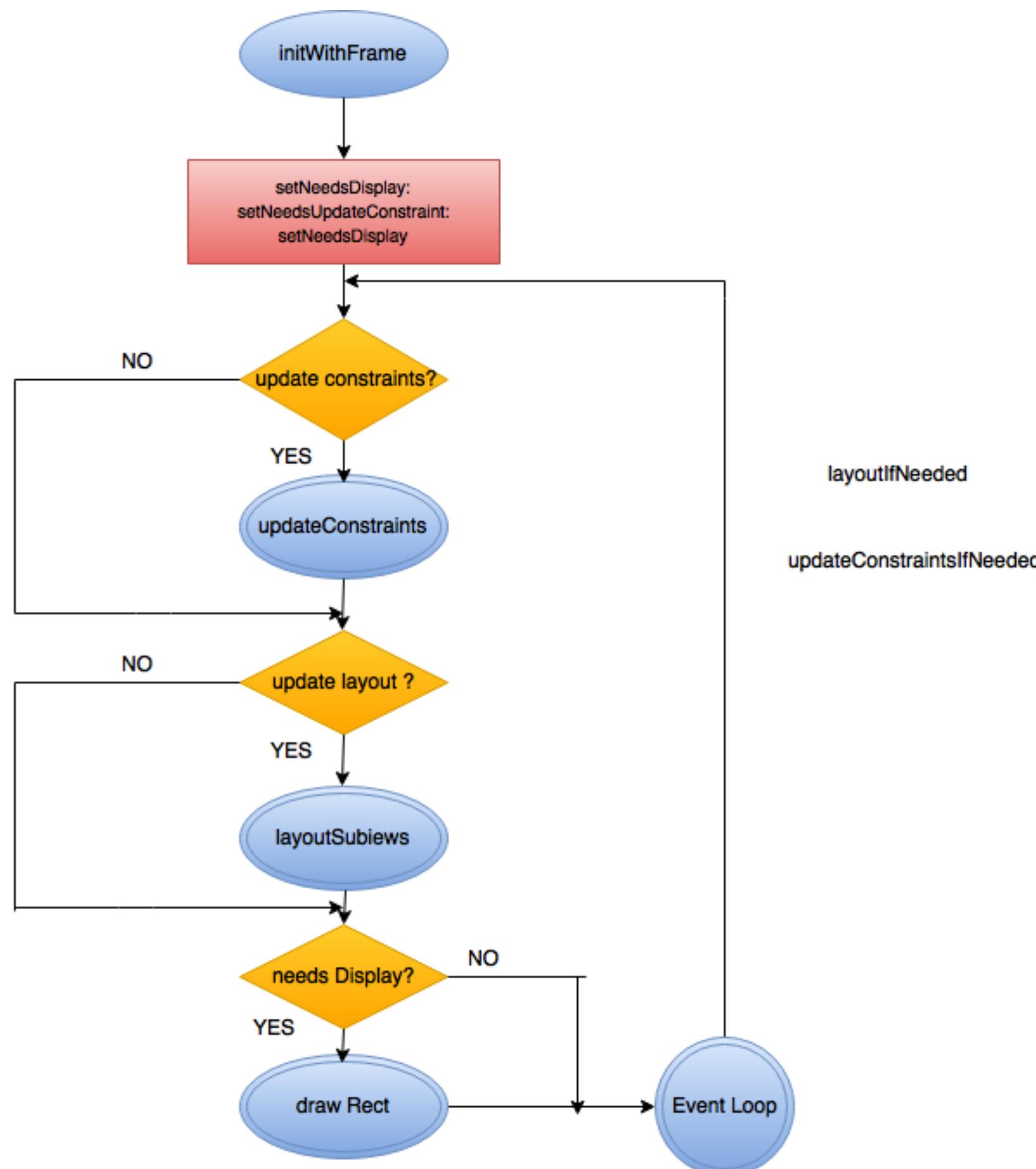
Координаты UIView

- **CGFloat** – это просто float значение, которые **всегда** будет использоваться для работы с графикой
- **CGPoint** – структура с двумя float значениями: X и Y
- **CGSize** – структура с двумя float значениями: width и height
- **CGRect** – структура с 4мя float значениями. CGPoint позиция и CGSize размер

Создание UIView

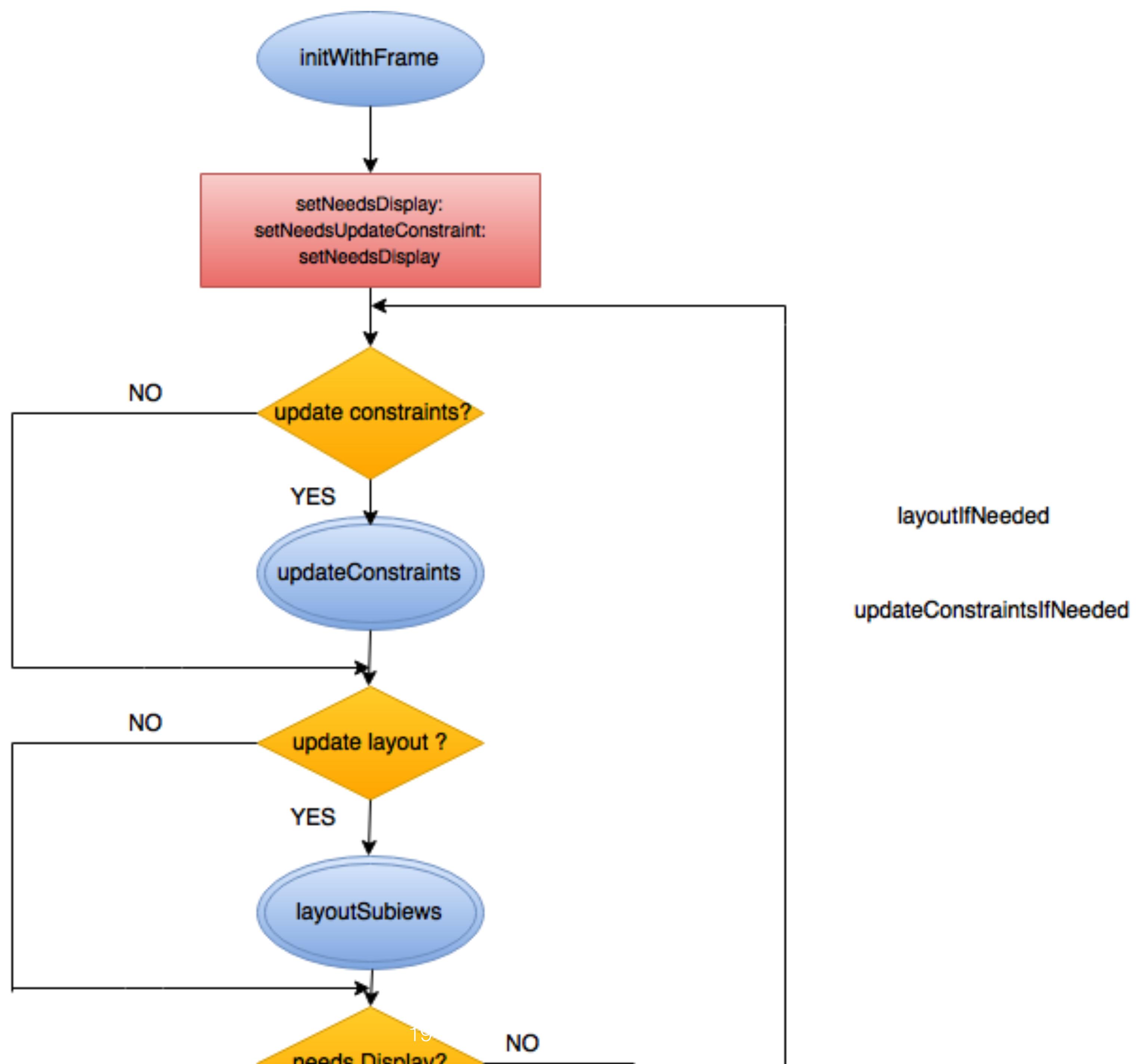
```
let rect = CGRect(x: 10, y: 10, width: 100, height: 100)  
let myView = UIView(frame: rect)
```

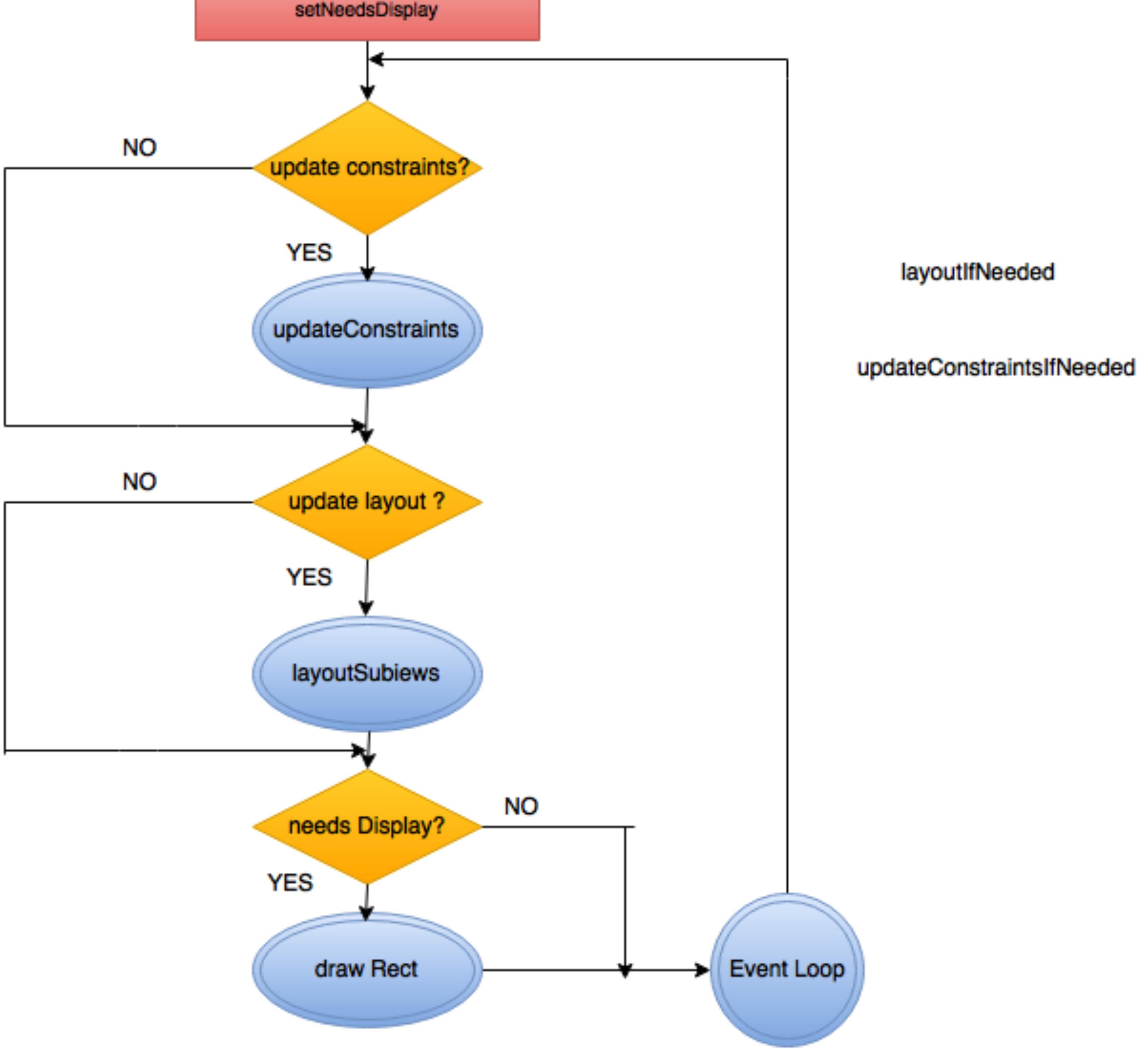
View drawing cycle



layoutIfNeeded

updateConstraintsIfNeeded





UILabel



Создание UILabel

```
let label = UILabel(frame: CGRect(x: 20, y: 20, width: 130,  
height: 100))  
label.font = UIFont.systemFont(ofSize: 20.0, weight: .bold)  
label.textColor = .white  
label.text = "Привет!"  
rectangle.addSubview(label)
```

AttributedString

The `attributedText` property allows **you** to control the appearance of individual cha...

lineBreakMode

The `attributedText` property allows **you** to control the appearance of individual cha...

textColor

textAlignment

The `attributedText` property allows you to control the a...edString API.

lineBreakMode

25 500,00 ₽

UIImage и UIImageView

UIImage – это объект, который управляет данными картинки в вашем приложение.

UIImageView - это объект который отображает одну картинку или последовательность анимированных картинок в вашем интерфейсе.

Варианты использования

- Присваивать картинку к UIImageView объекту, для отображения картинок в вашем интерфейсе
- Использовать картинки для кастомизации различных контроллов, таких как кнопки, слайдеры и сегмент контролы.
- Рисовать картинку напрямую внутри view или в других graphics context (графических контекстах)
- Передавать картинку в другие APIs, которые могут требовать image data.

Форматы

- .png
- .tiff or .tif
- .jpeg or .jpg
- .gif
- .bmp or .BMPf
- .ico
- .cur
- .xbm

Создание UIImage

- `init(named:in:compatibleWith:)` или `init(named:)`
- `imageWithContentsOfFile:` или `init(contentsOfFile:)`

Сравнение объектов картинок

```
let image1 = UIImage(named: "MyImage")
let image2 = UIImage(named: "MyImage")
if image1 != nil && image1!.isEqual(image2) {
    // Correct. This technique compares the image data
    correctly.
}
if image1 == image2 {
    // Incorrect! Direct object comparisons may not work.
}
```

UIImageView

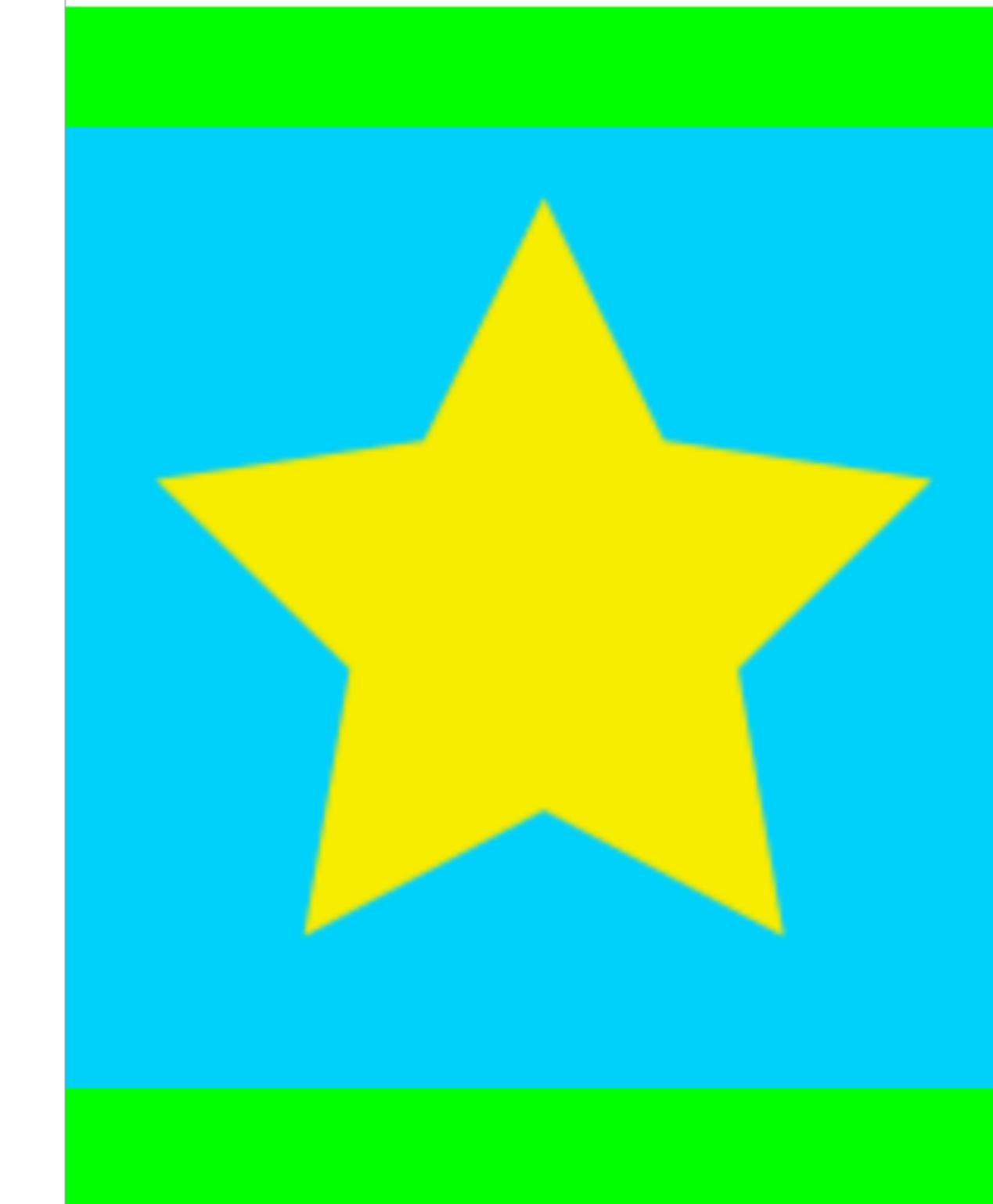


© Instagram/Taylor Swift

ScaleToFit и ScaleAspectFit



.ScaleToFit



.ScaleAspectFit

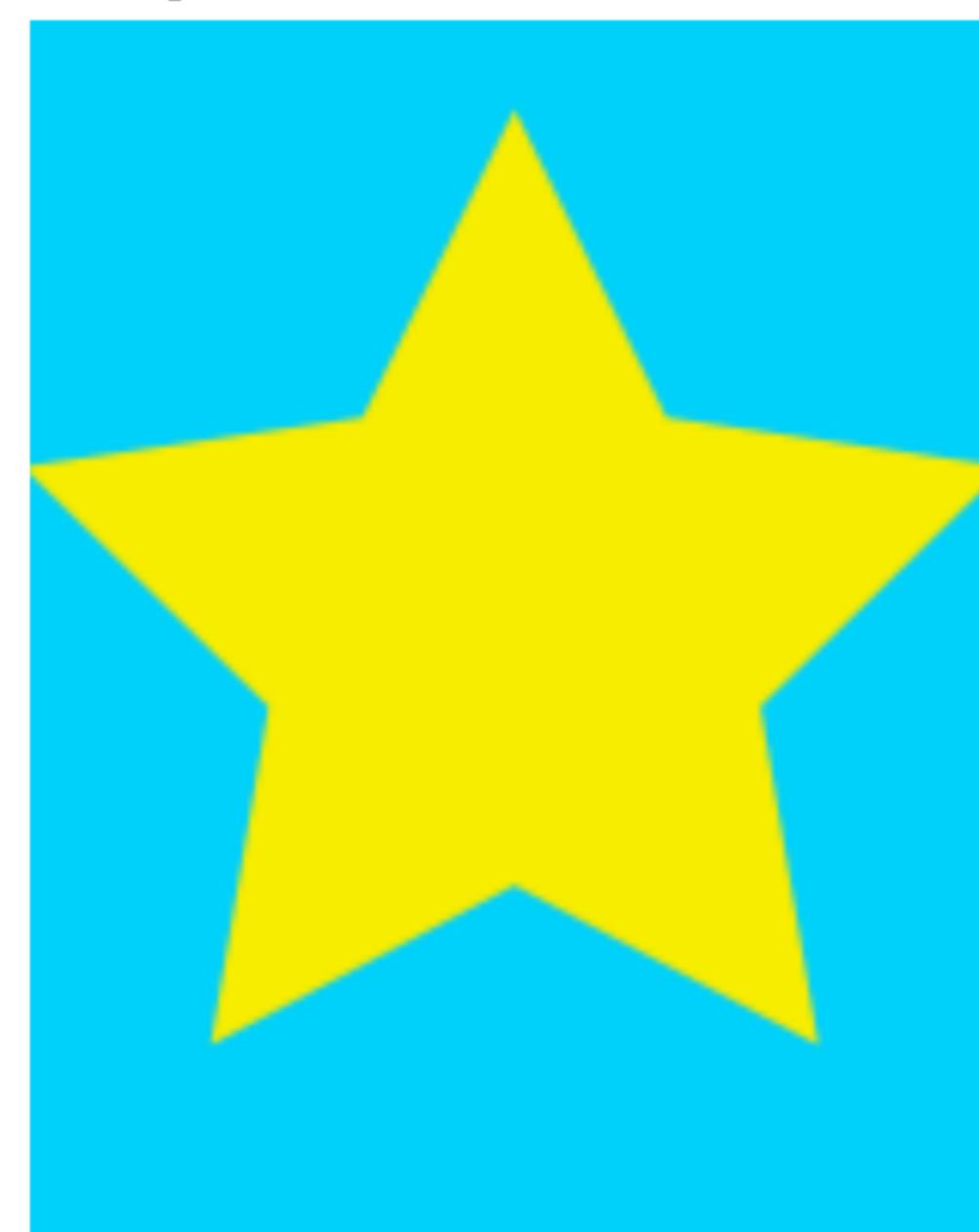
ScaleAspectFill

`clipsToBounds = false`



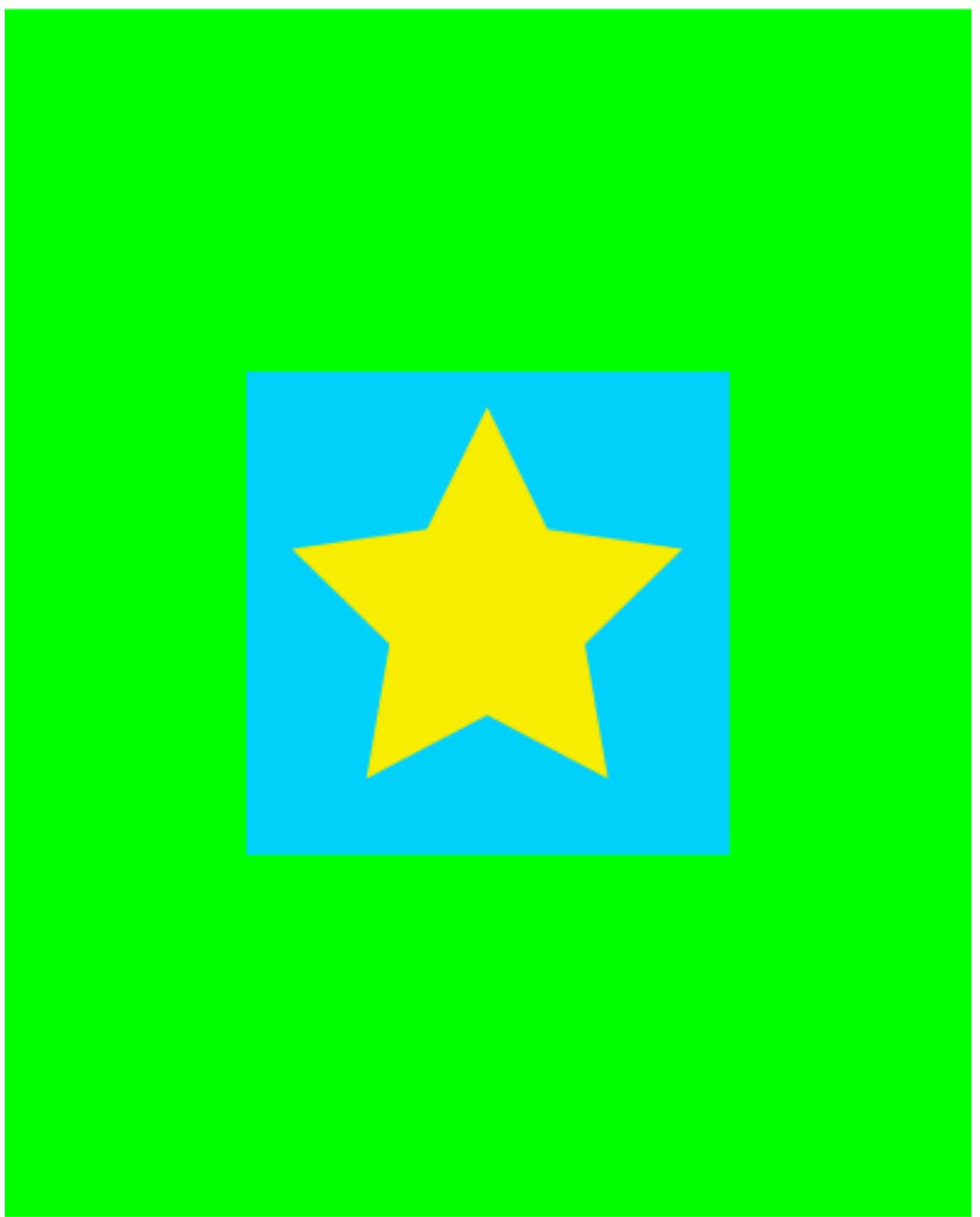
`.ScaleAspectFill`

`clipsToBounds = true`



`.ScaleAspectFill`

Center

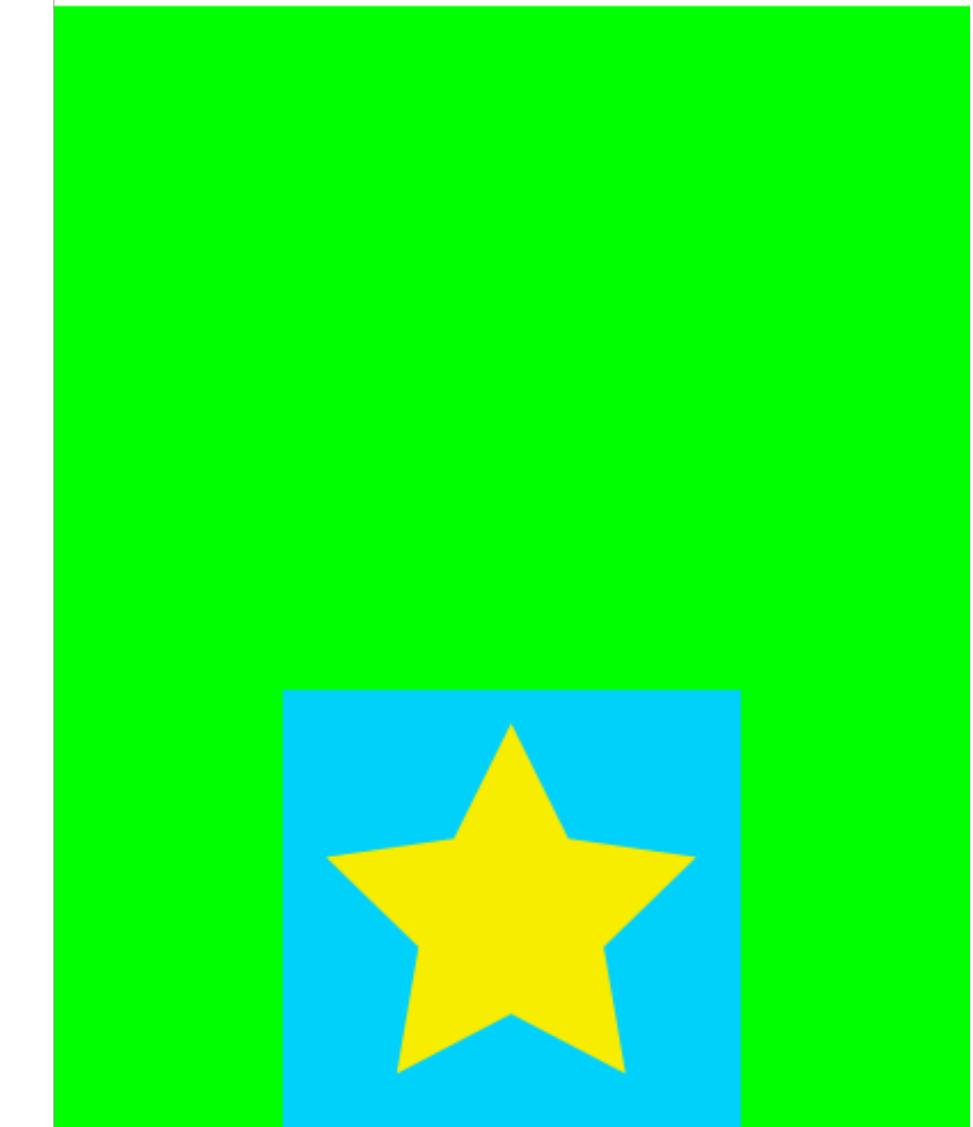


.Center

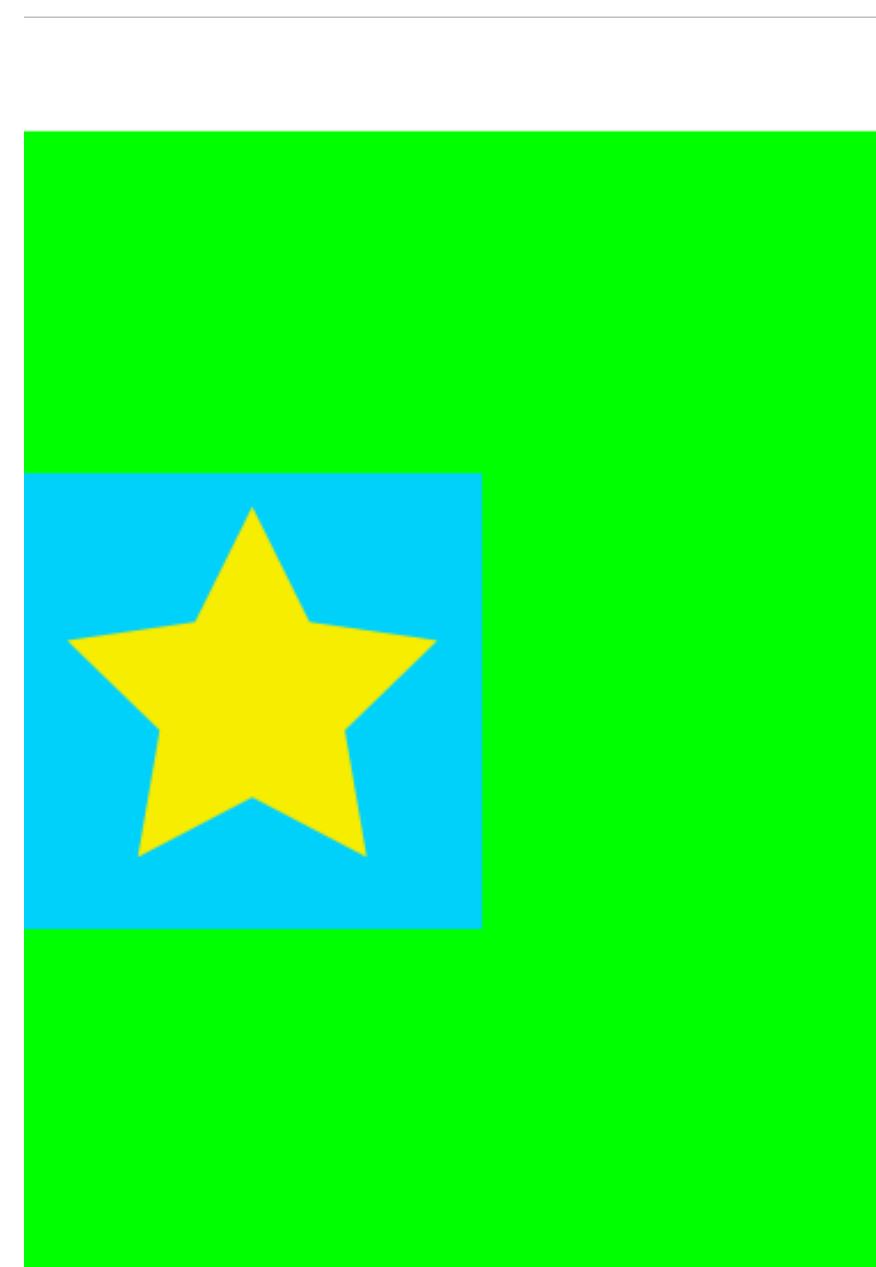
Top, Bottom, Left, Right



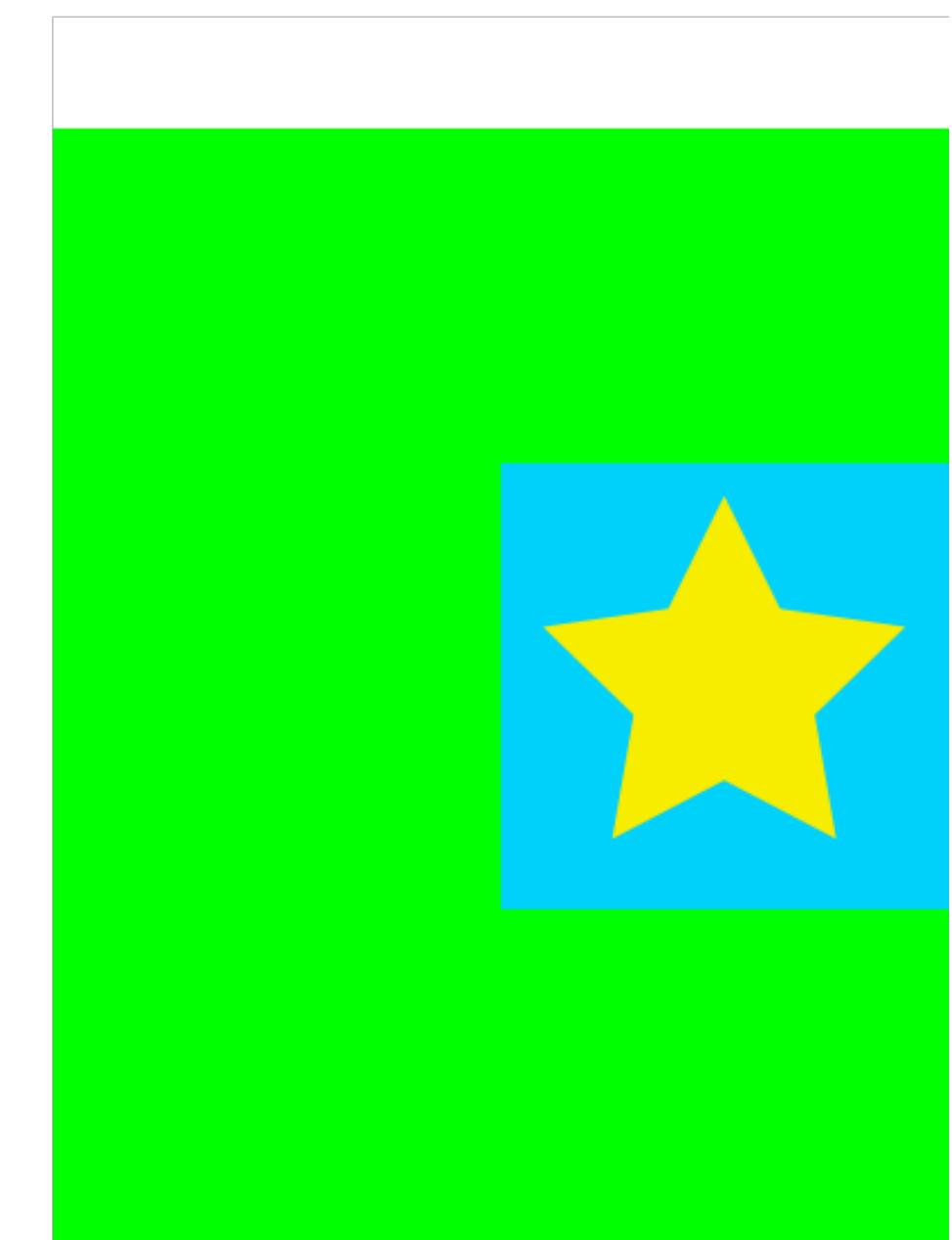
.Top



.Bottom

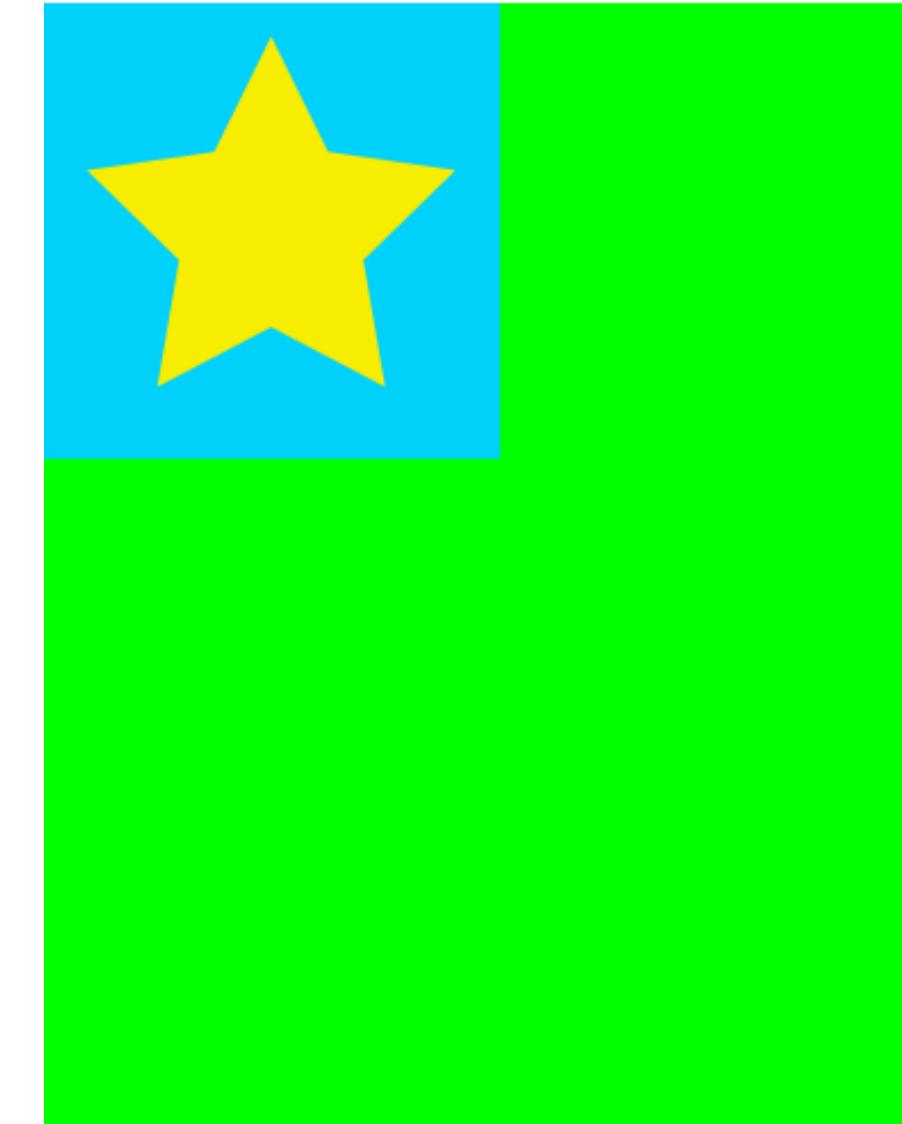


.Left

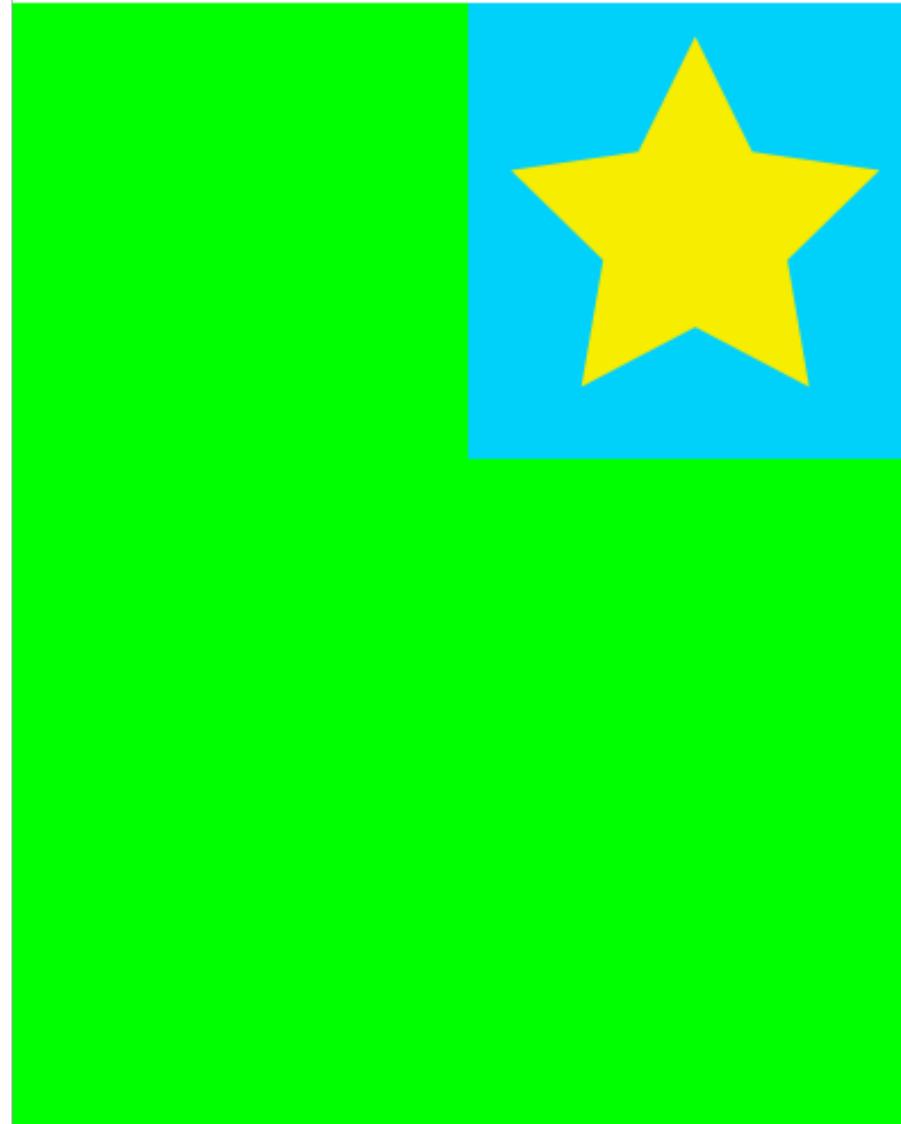


.Right

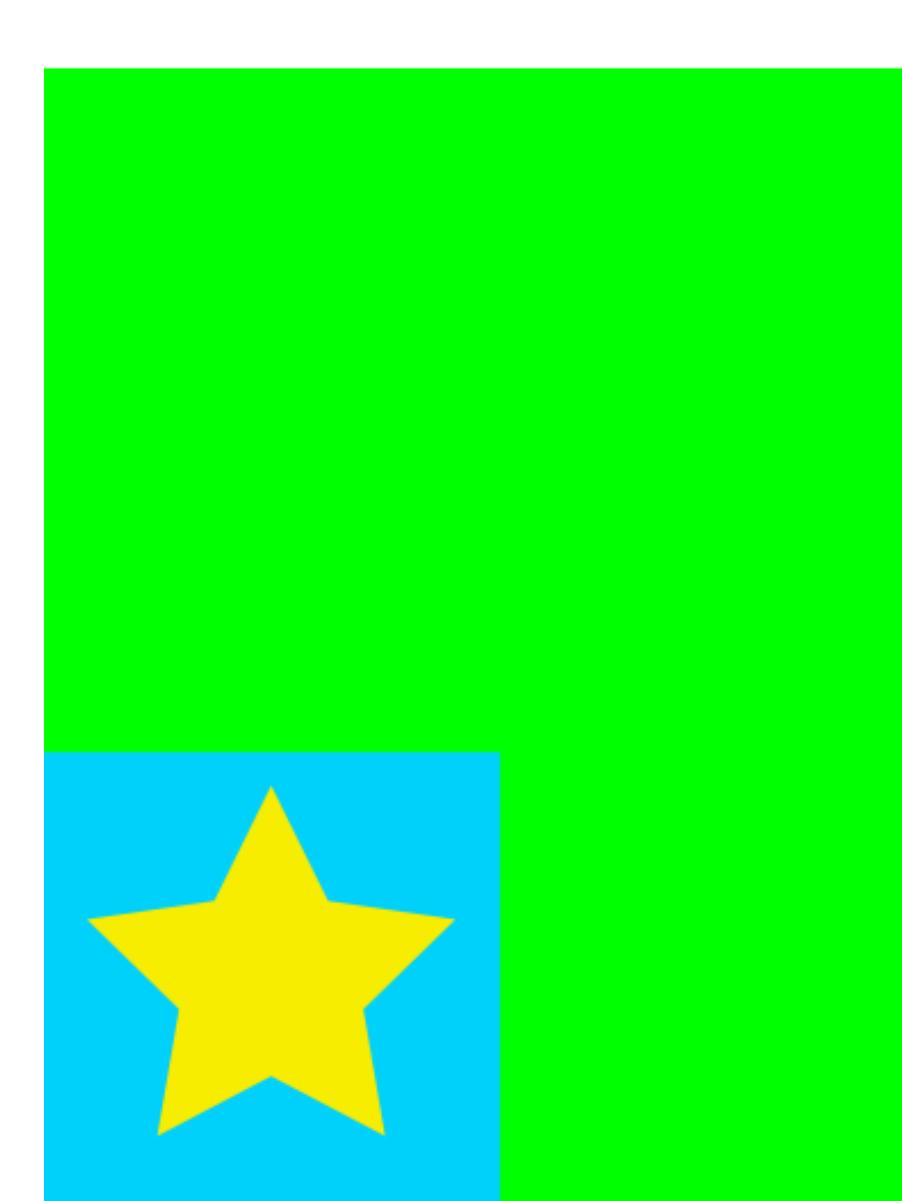
TopLeft
TopRight
BottomLeft
BottomRight



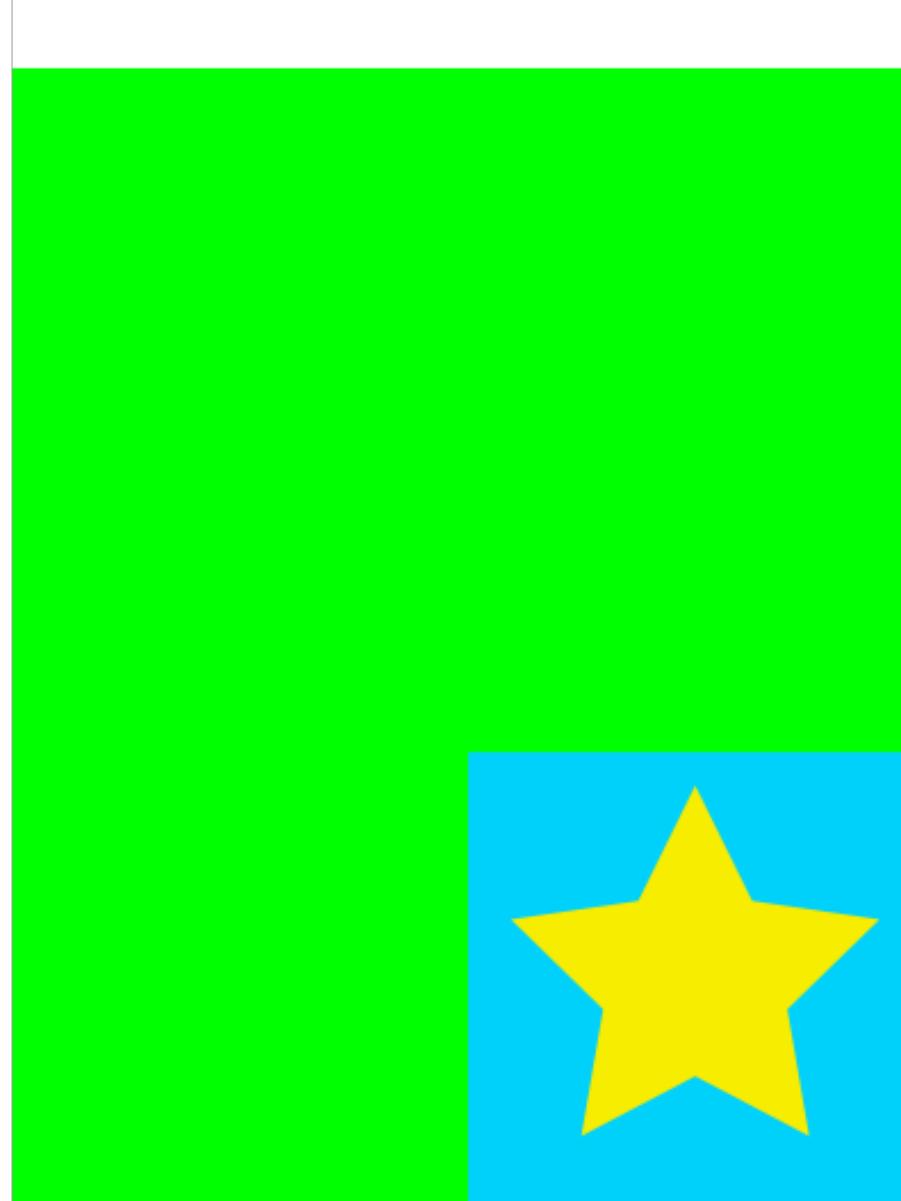
.TopLeft



.TopRight

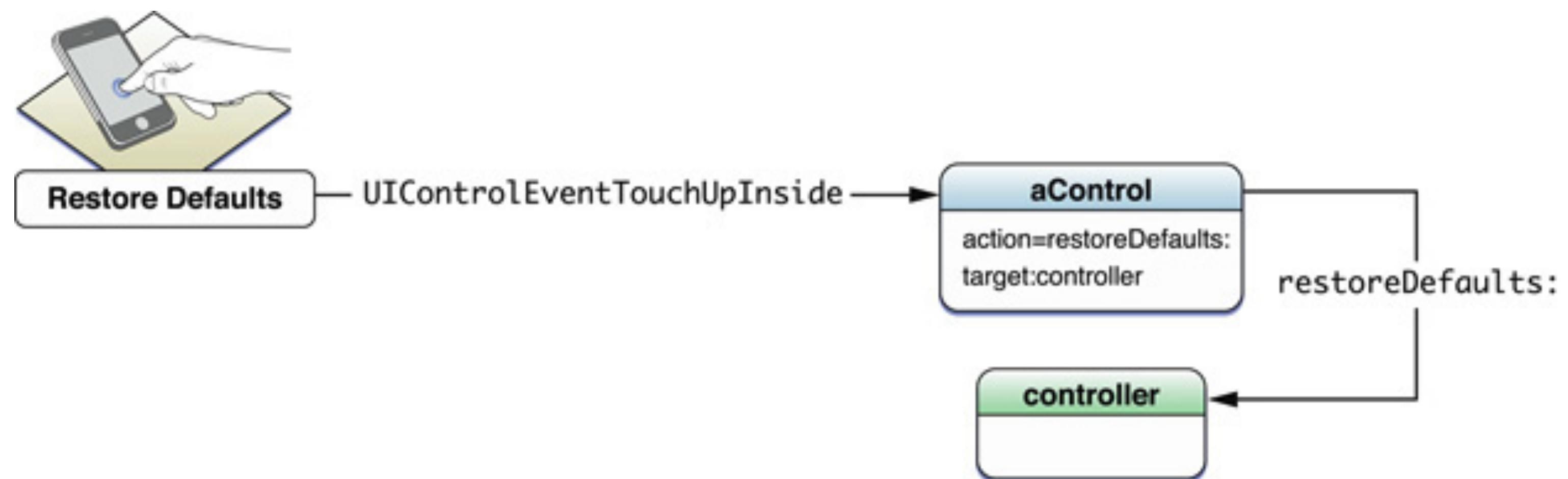


.BottomLeft



.BottomRight

UIButton Action



Добавление Target

```
@IBAction func doSomething()  
@IBAction func doSomething(sender: UIButton)  
@IBAction func doSomething(sender: UIButton, forEvent event: UIEvent)
```

Создание Button

```
let button = UIButton(| ▲ Initialization of immutable value...
    M UIButton? (coder: NSCoder)
    M UIButton (frame: CGRect)
    M UIButton (type: UIButtonType)
```

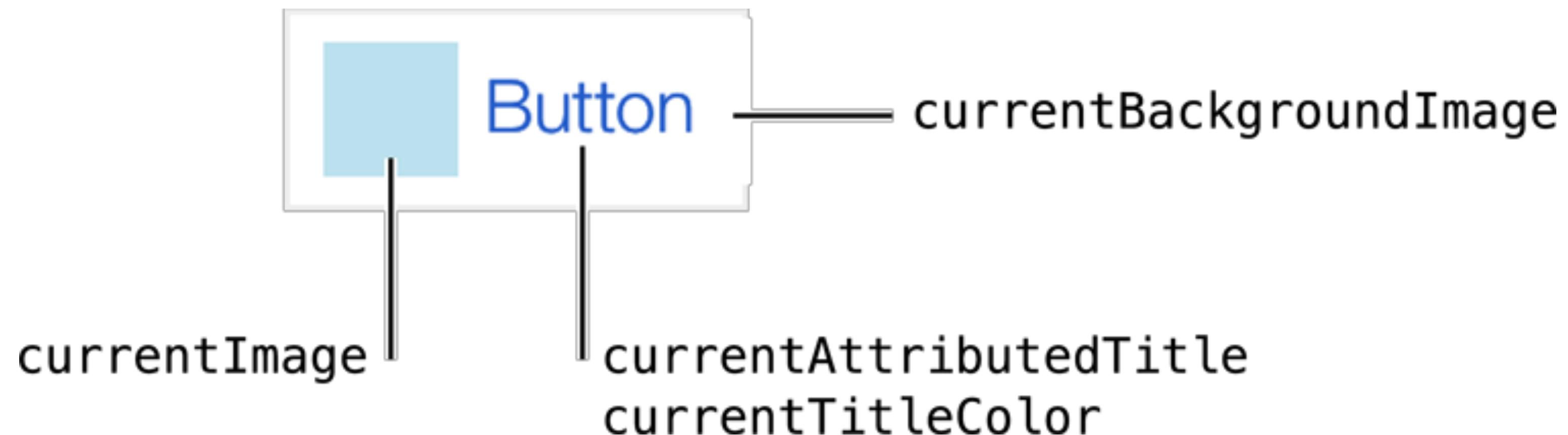
Типы Button

- Custom
- System
- detailDisclosure
- infoLight
- infoDark
- contactAdd
- plain

UIButton States

- default
- highlighted
- focused
- selected
- disabled

UIButton Content



UITextField и UITextView



User Interface

Autolayout

Коллекции данных

Состояния экранов

User Interface

Autolayout

Коллекции данных

Состояния экранов

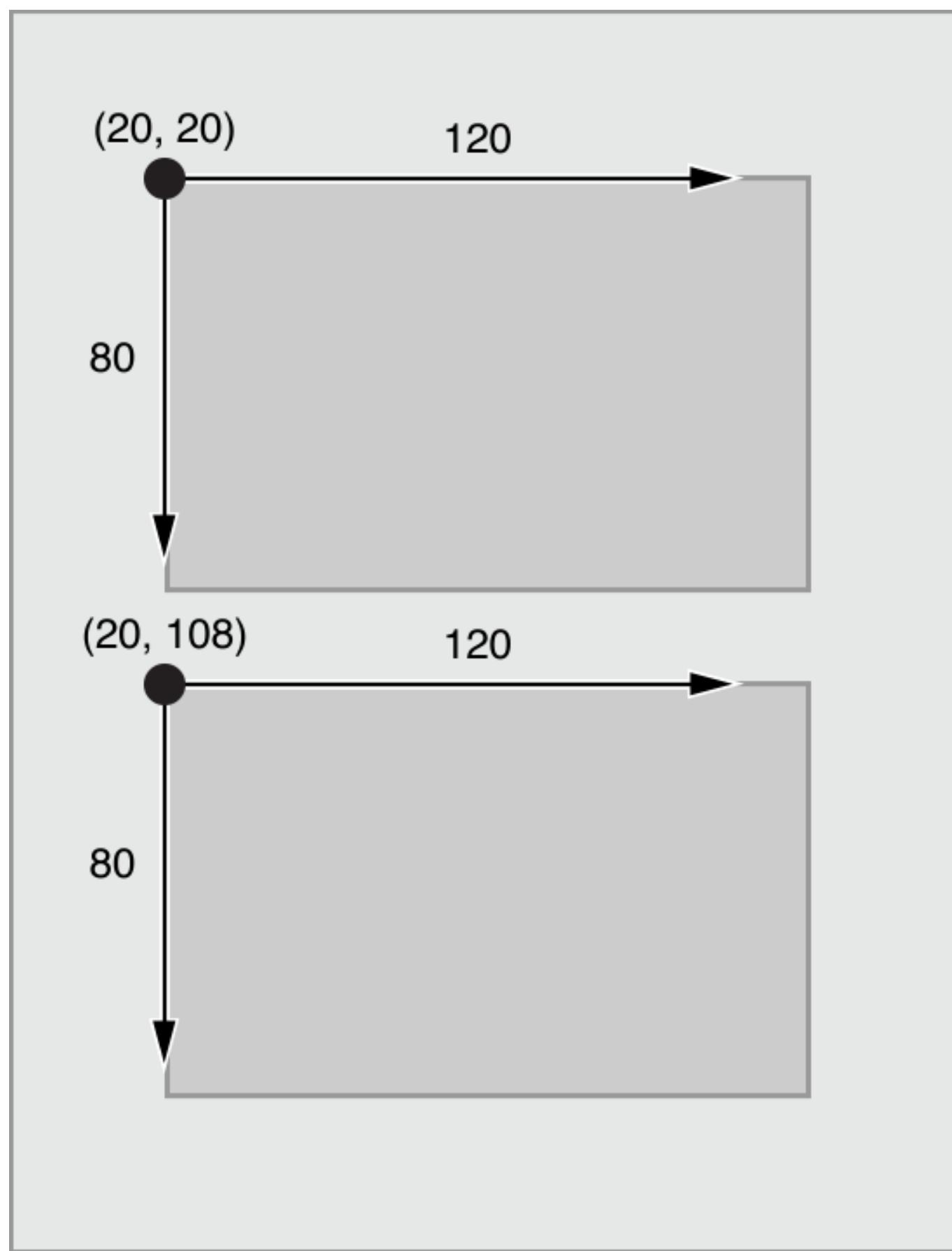
AutoLayout

Auto layout занимается динамическим вычислением позиций и размера всех view в view иерархии на основе constraints

Создание интерфейсов

- Программно (Frame layout)
- Autoresizing mask
- AutoLayout

Frame Layout



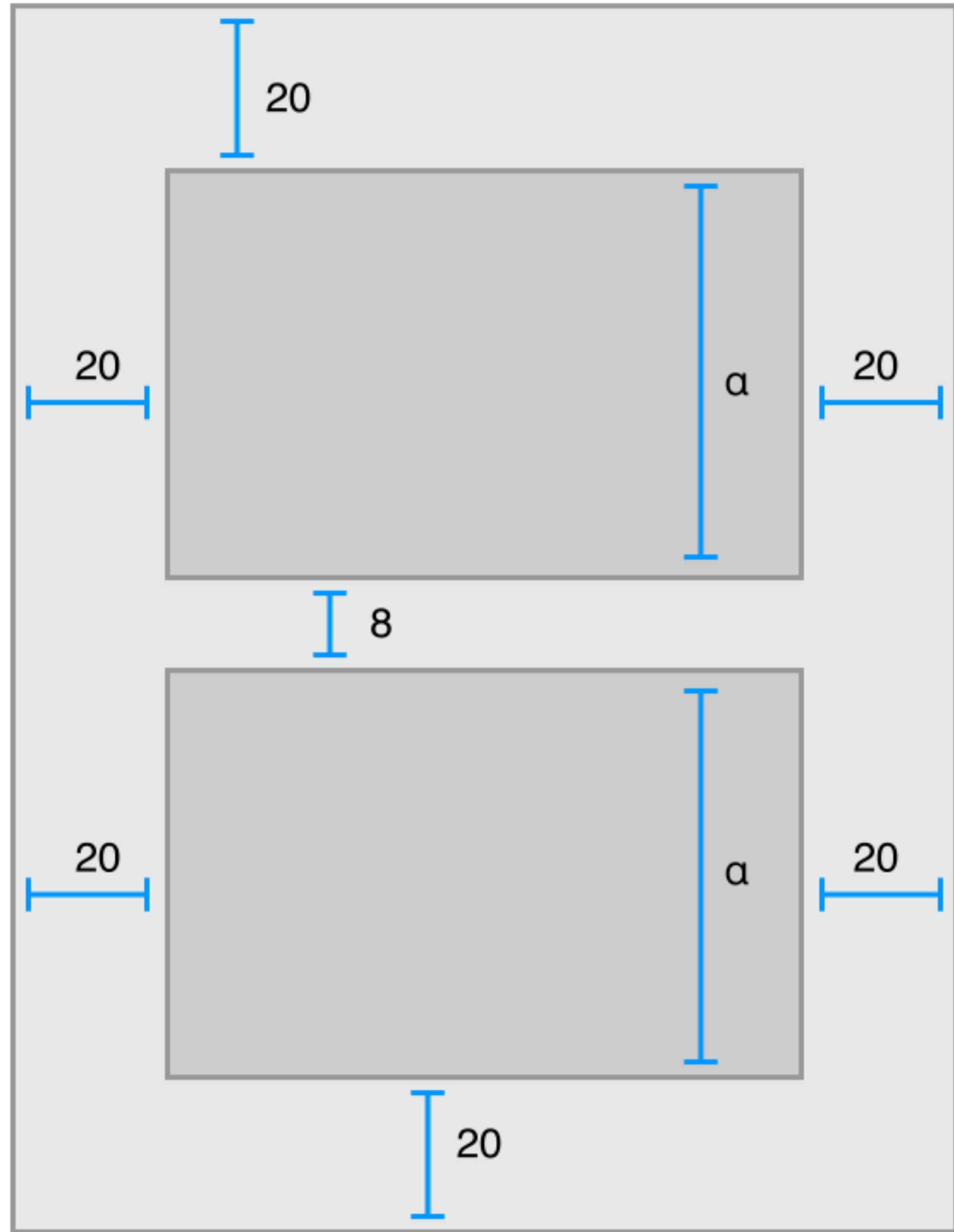
Autoresizing mask

```
public struct UIViewAutoresizing : OptionSet {  
  
    public init(rawValue: UInt)  
  
    public static var flexibleLeftMargin: UIViewAutoresizing { get }  
  
    public static var flexibleWidth: UIViewAutoresizing { get }  
  
    public static var flexibleRightMargin: UIViewAutoresizing { get }  
  
    public static var flexibleTopMargin: UIViewAutoresizing { get }  
  
    public static var flexibleHeight: UIViewAutoresizing { get }  
  
    public static var flexibleBottomMargin: UIViewAutoresizing { get }  
}
```

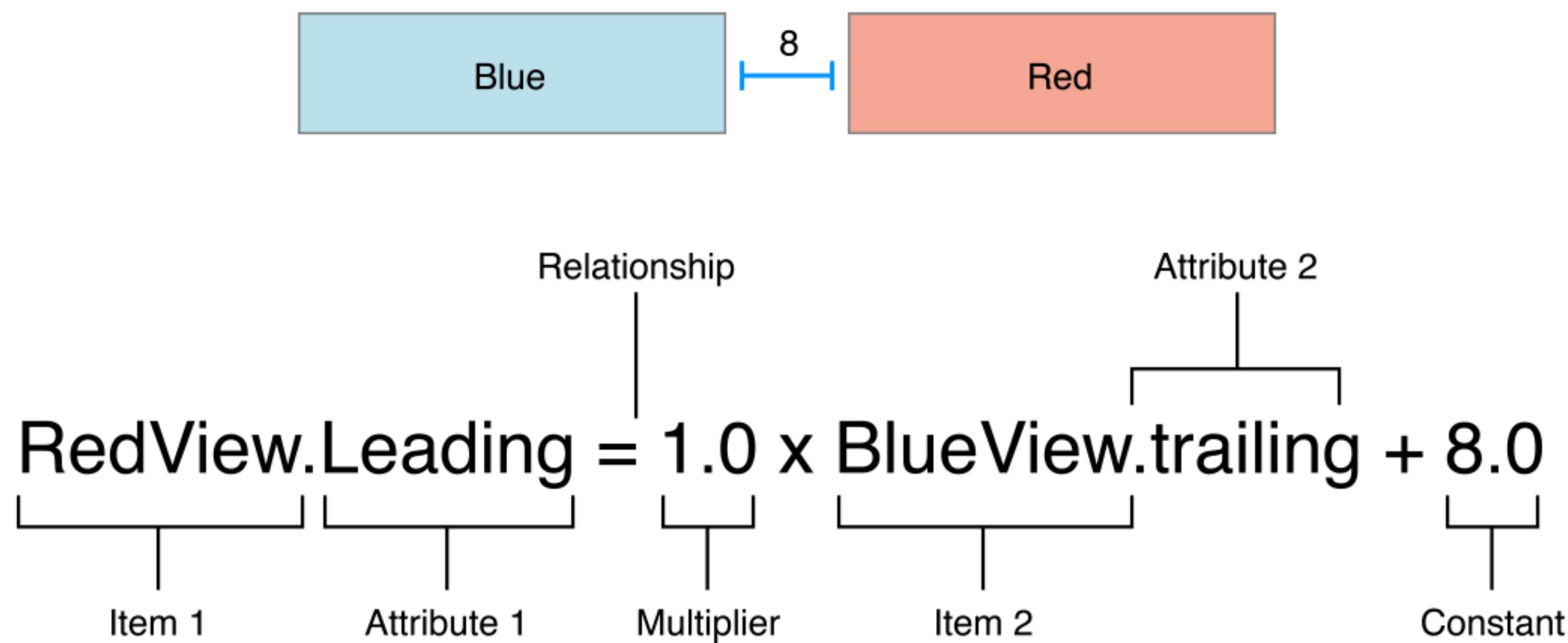
AutoLayout



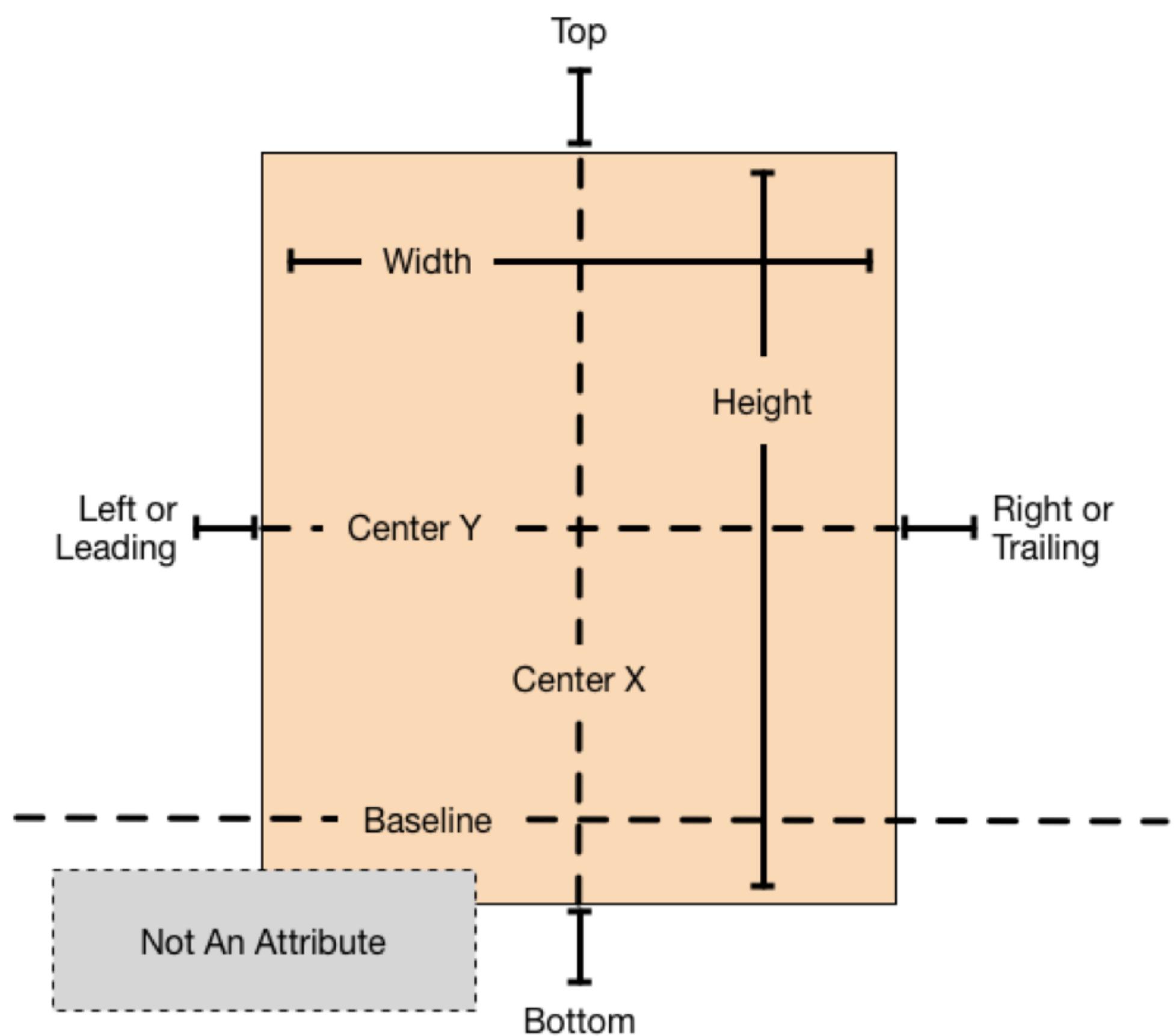
AutoLayout



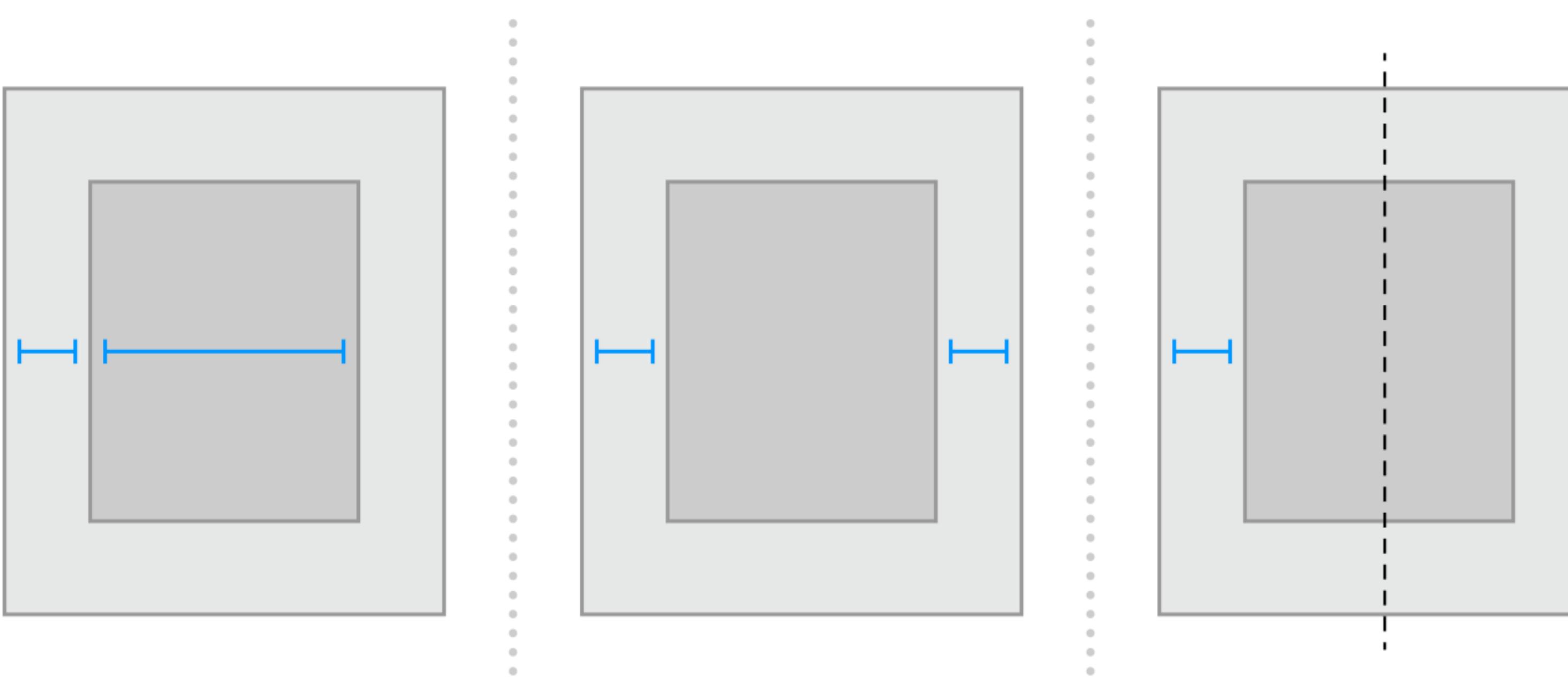
AutoLayout

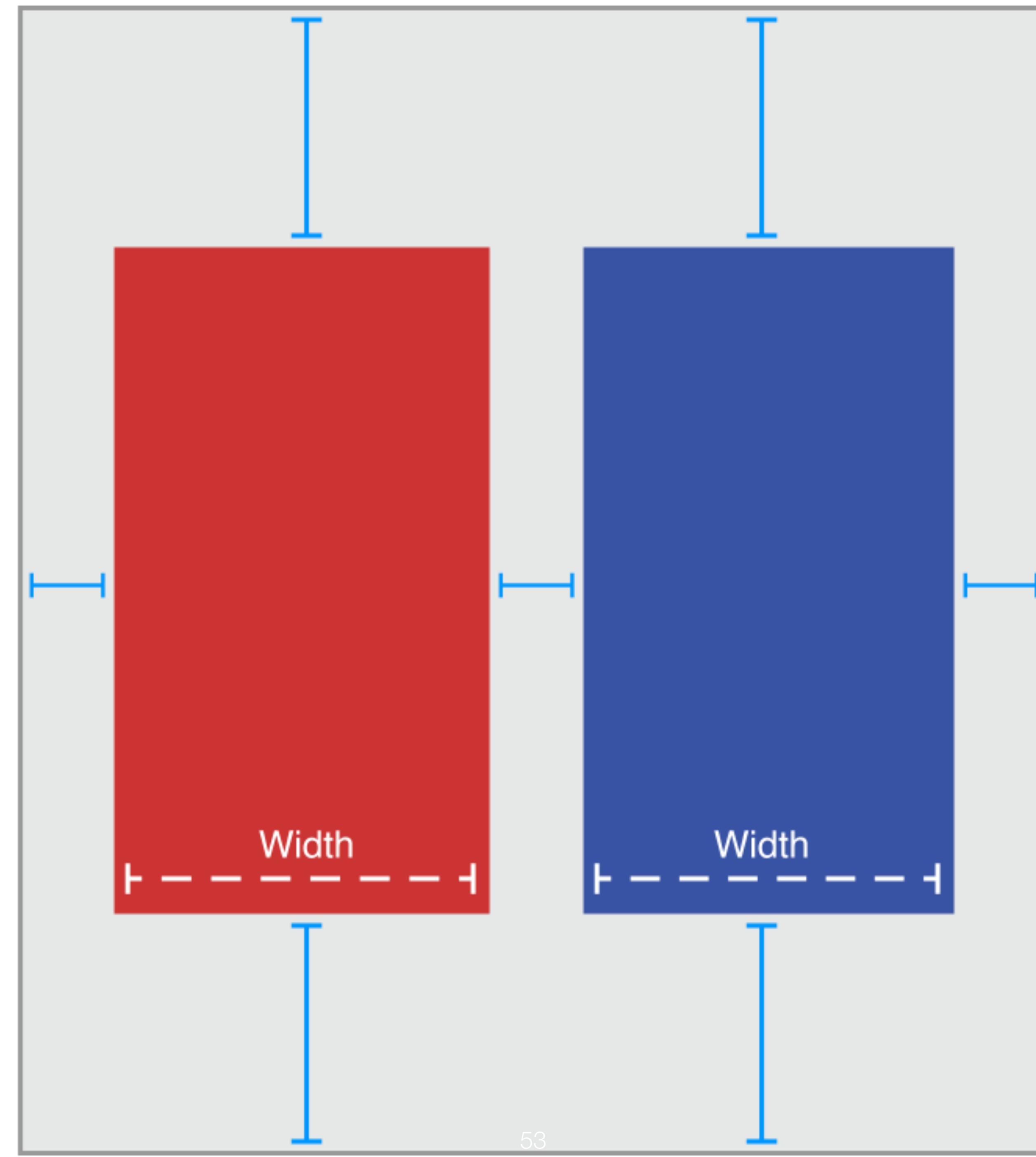


AutoLayout



Nonambiguous, satisfiable layout

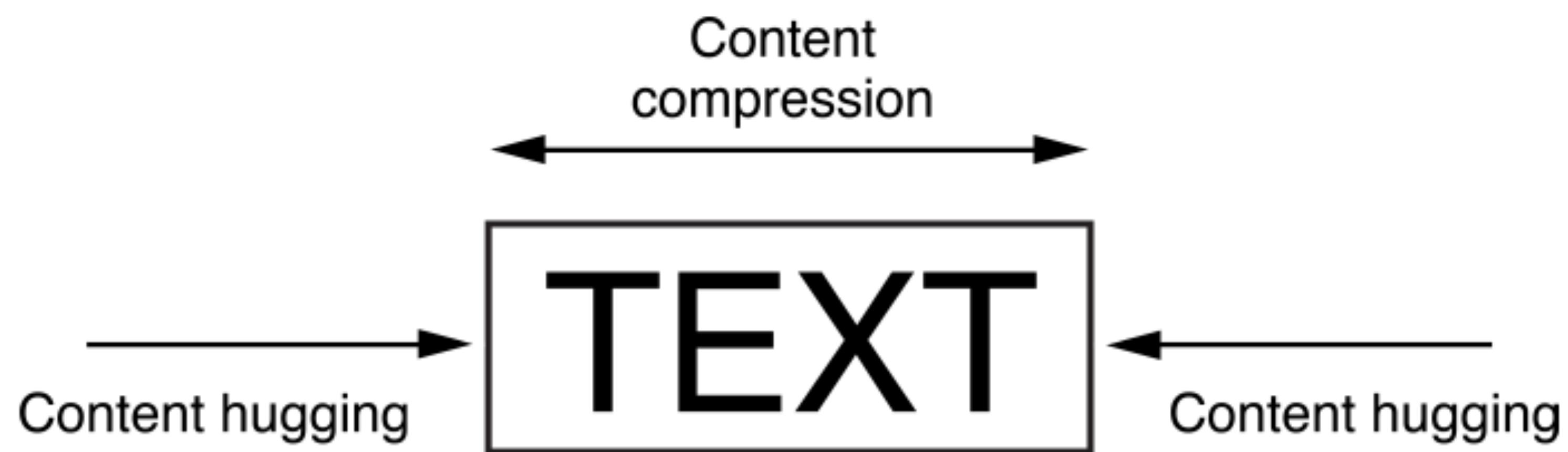




Intrinsic Content Size

View	Intrinsic content size
UIView and NSView	No intrinsic content size.
Sliders	Defines only the width (iOS). Defines the width, the height, or both—depending on the slider's type (OS X).
Labels, buttons, switches, and text fields	Defines both the height and the width.
Text views and image views	Intrinsic content size can vary.

Content Hugging



iOS Auto Layout

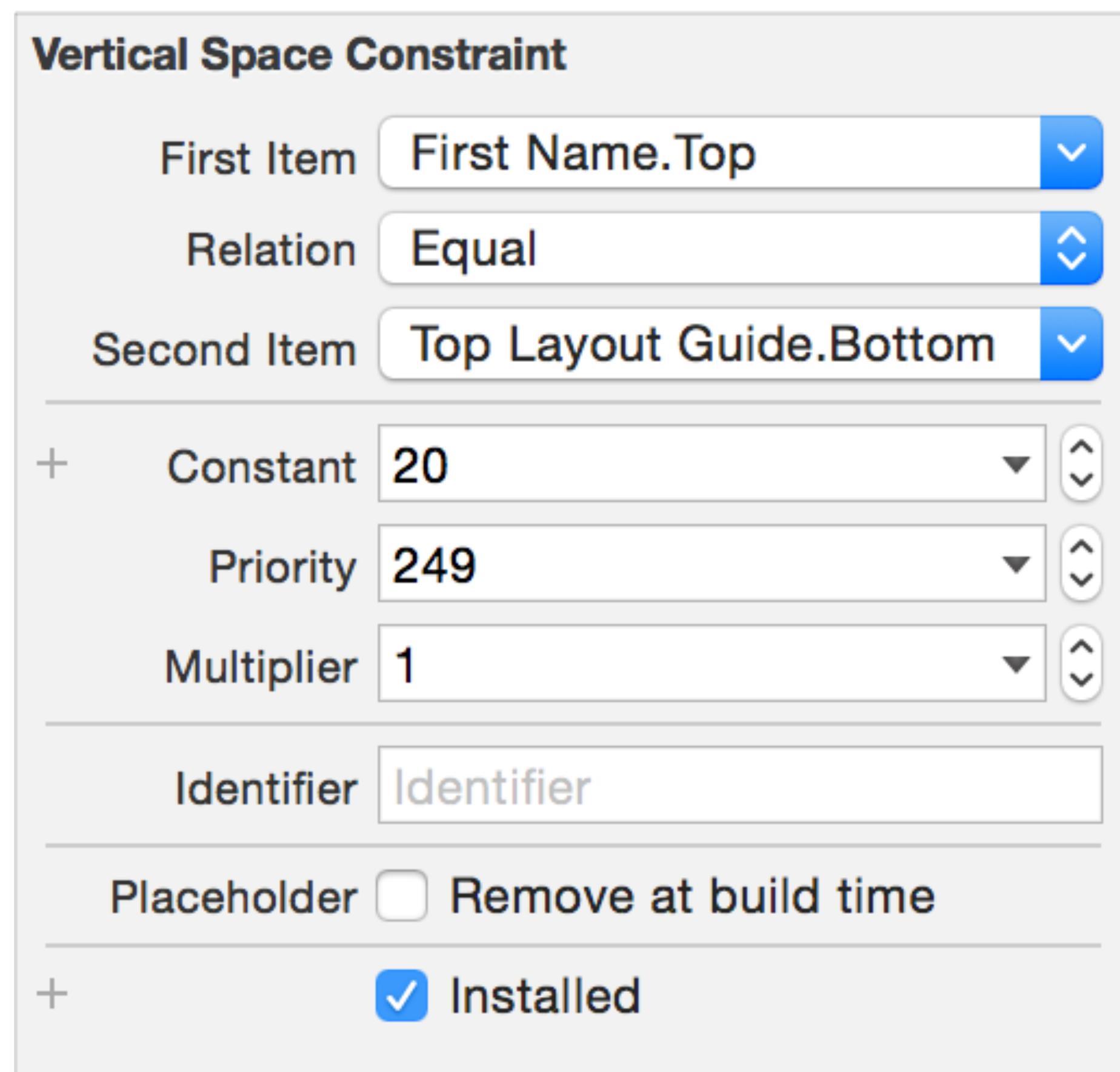
CONTENT HUGGING



CONTENT COMPRESSION RESISTANCE



Constraints and IB



User Interface

Autolayout

Коллекции данных

Состояния экранов

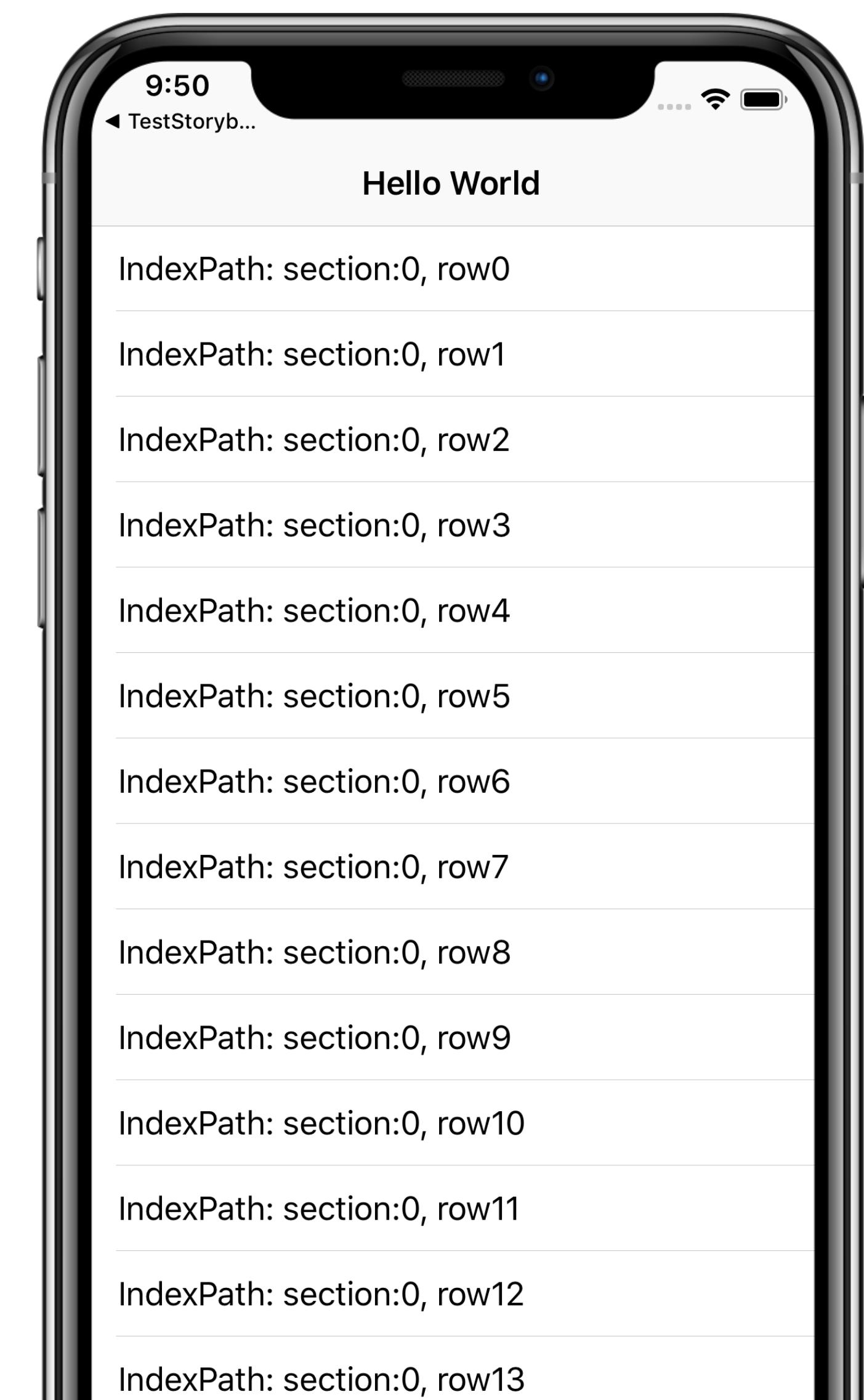
User Interface

Autolayout

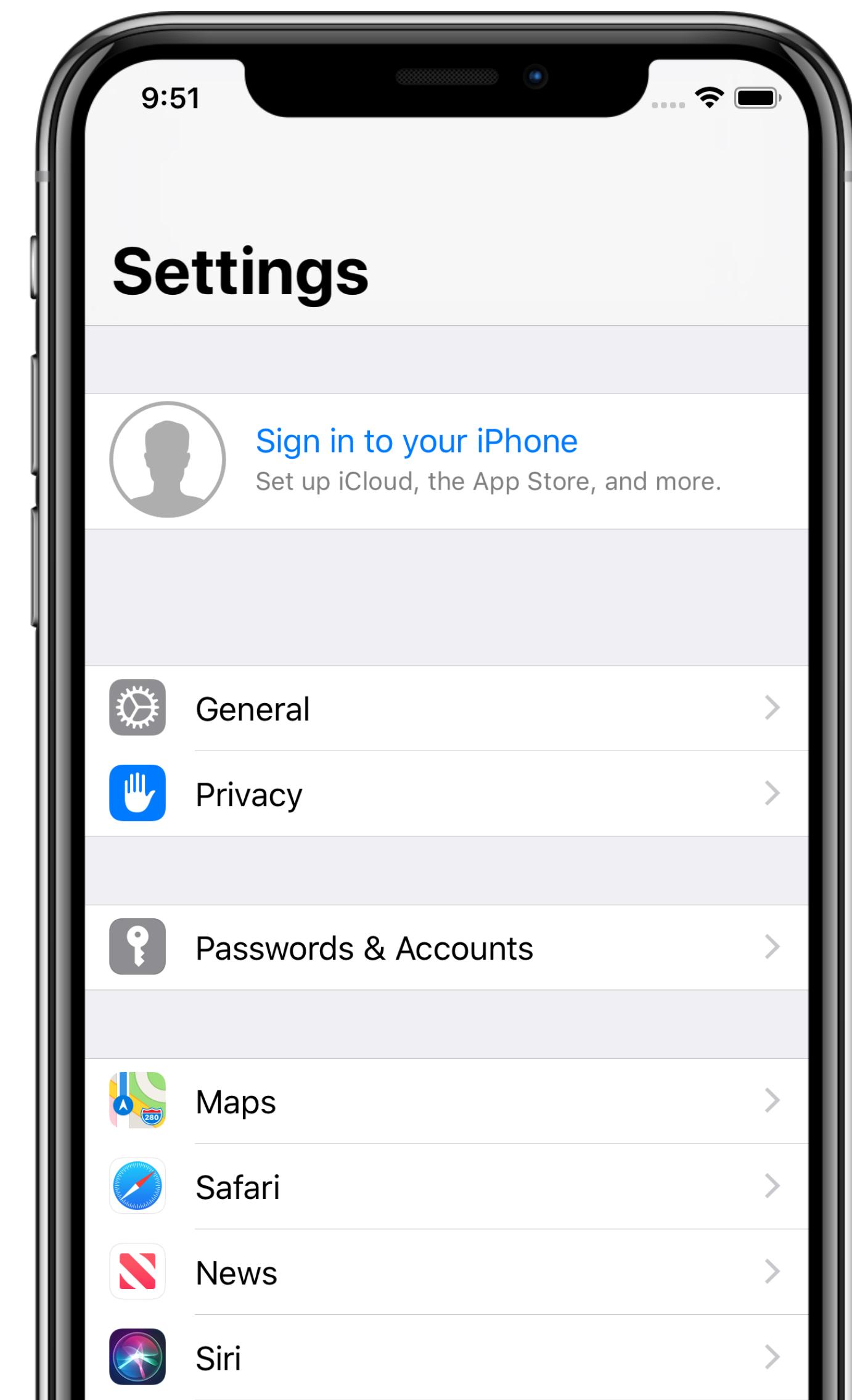
Коллекции данных

Состояния экранов

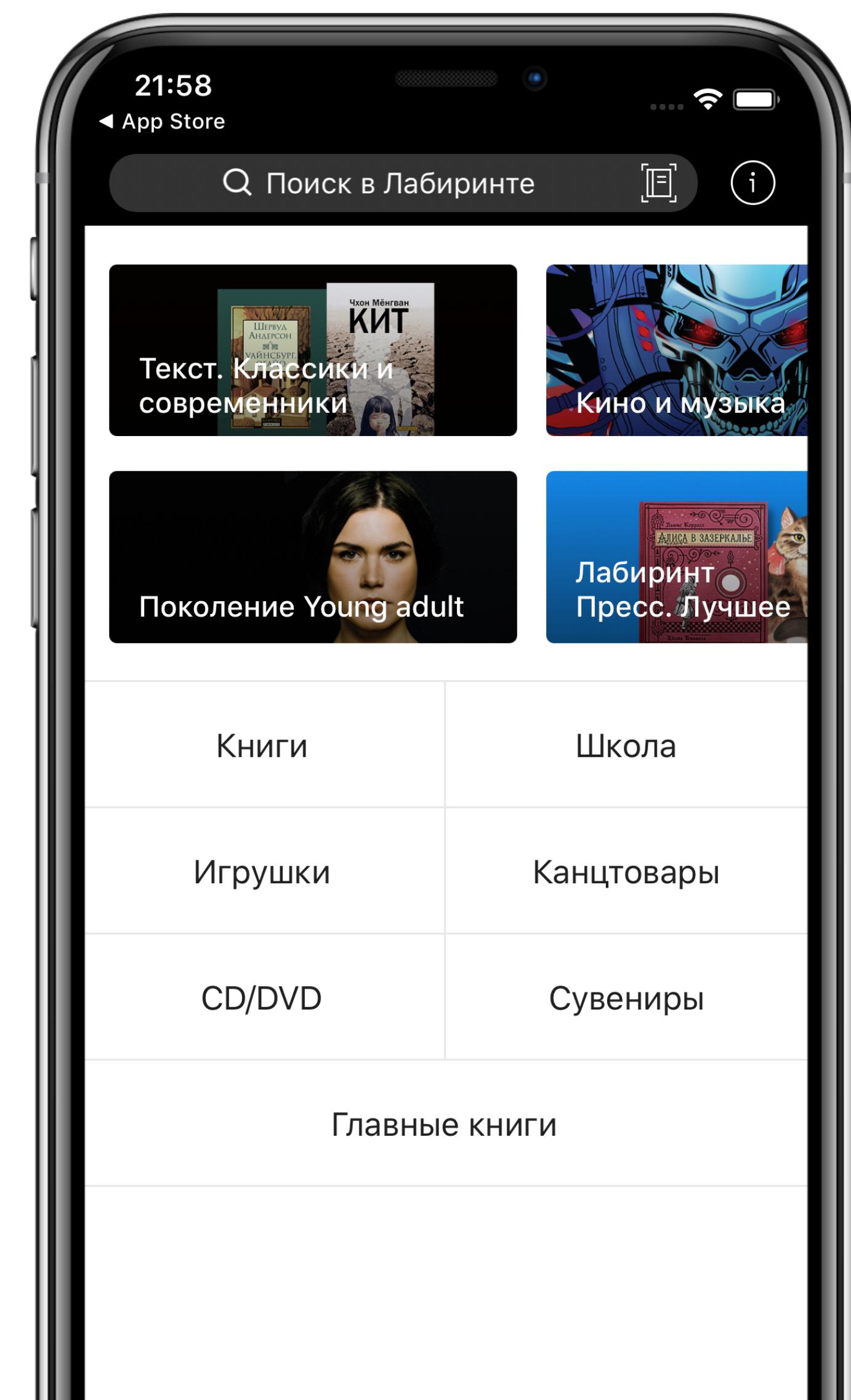
UITableView



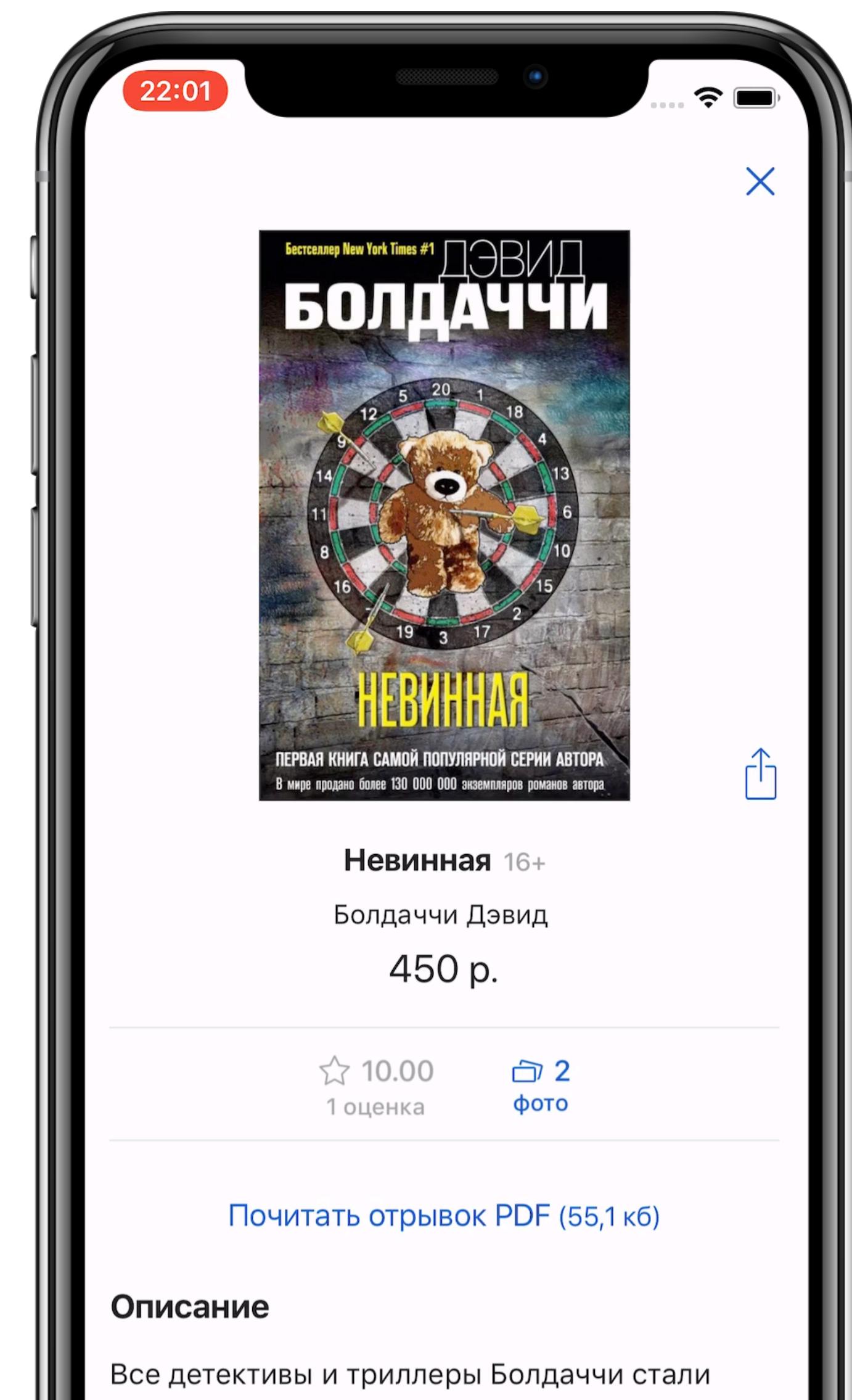
UITableView: пример



UITableView: пример



UITableView: пример

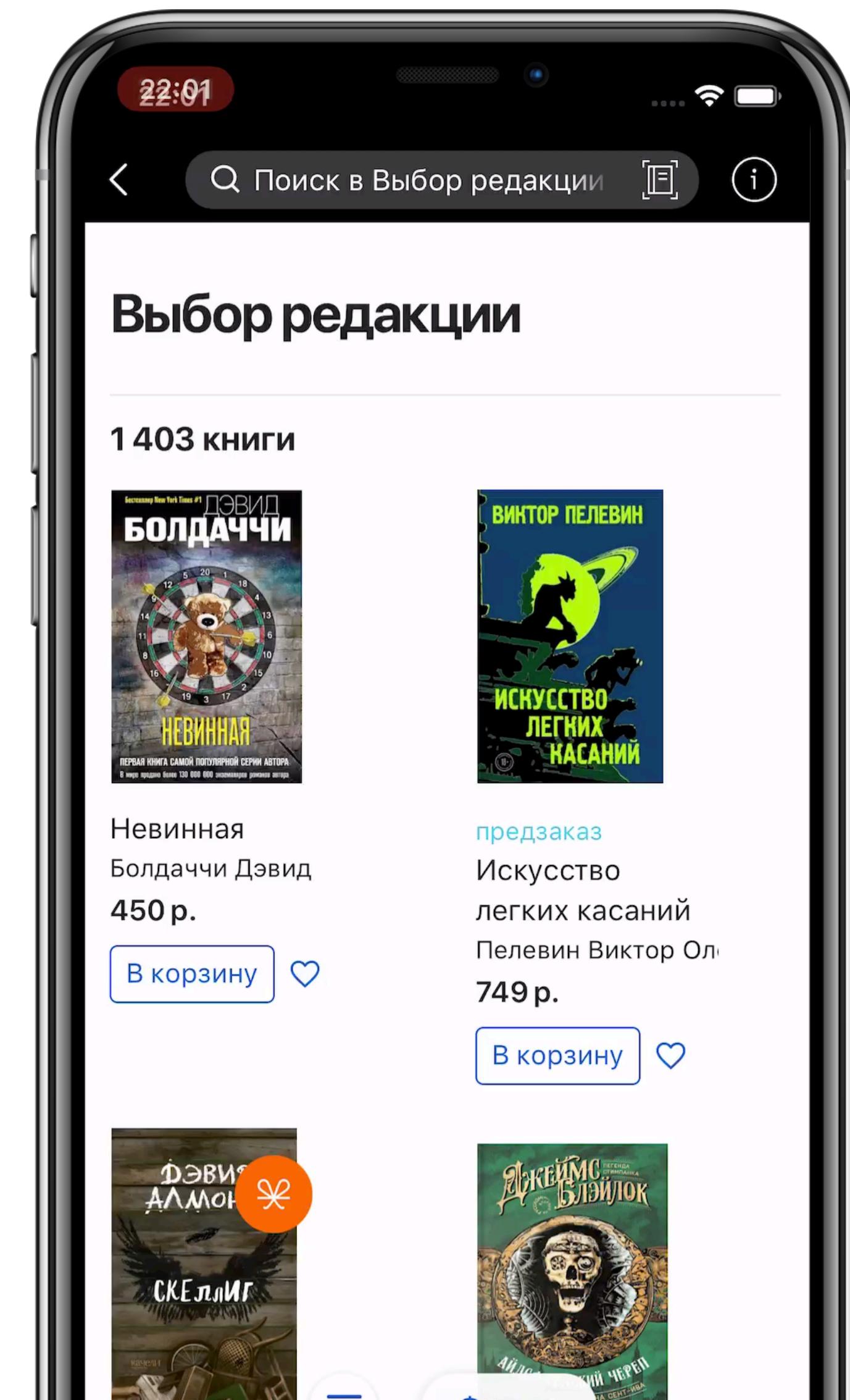


[Почитать отрывок PDF \(55,1 кб\)](#)

Описание

Все детективы и триллеры Болдаччи стали

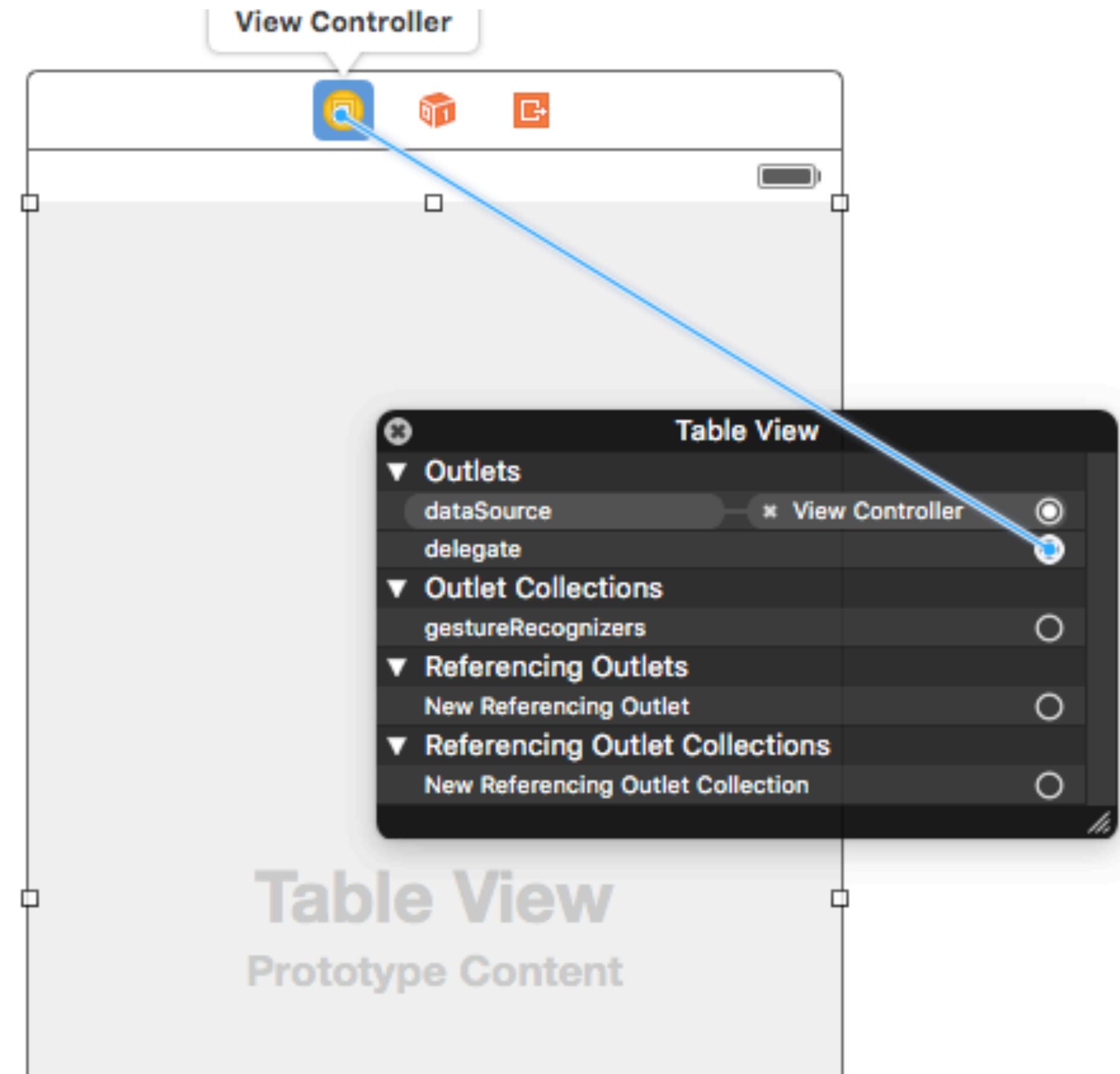
UICollectionView: пример



UICollectionView: пример



DataSource и Delegate



User Interface

Autolayout

Коллекции данных

Состояния экранов

User Interface

Autolayout

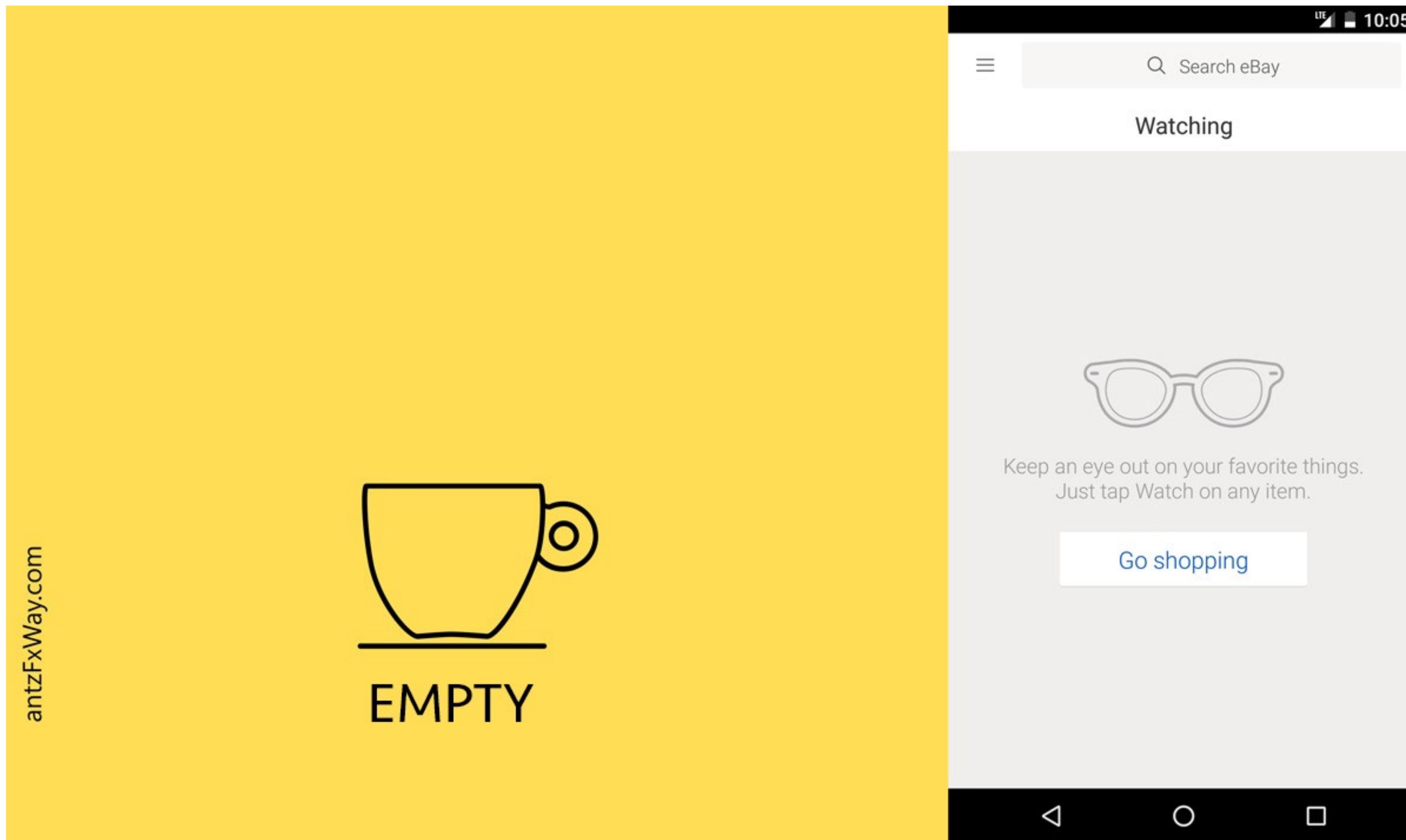
Коллекции данных

Состояния экранов

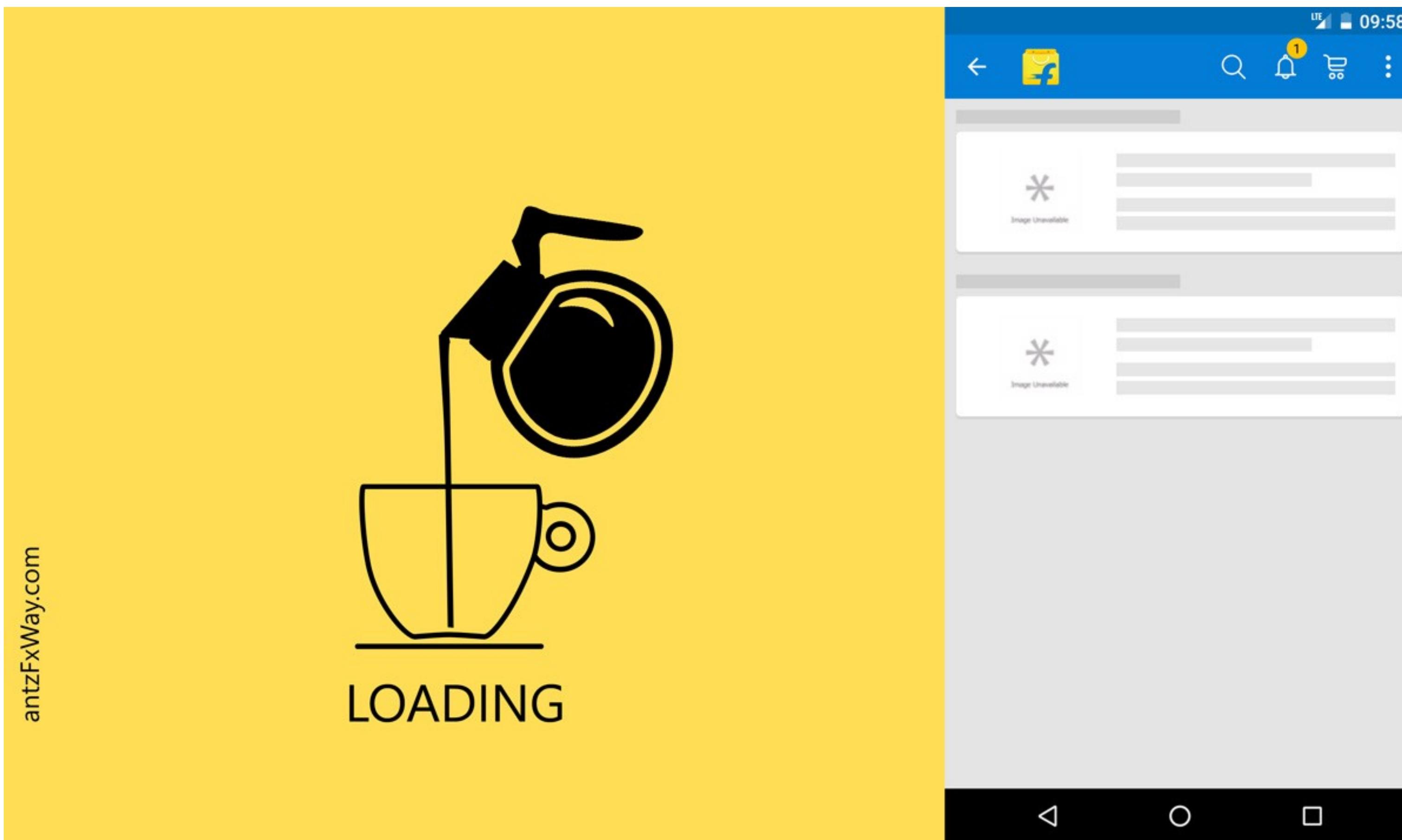
Чашечка кофе и состояния экранов

- Пустая чашка
- В чашку наливают кофе
- Наполовину наполненная кружка
- Кружка сломалась
- Полная кружка

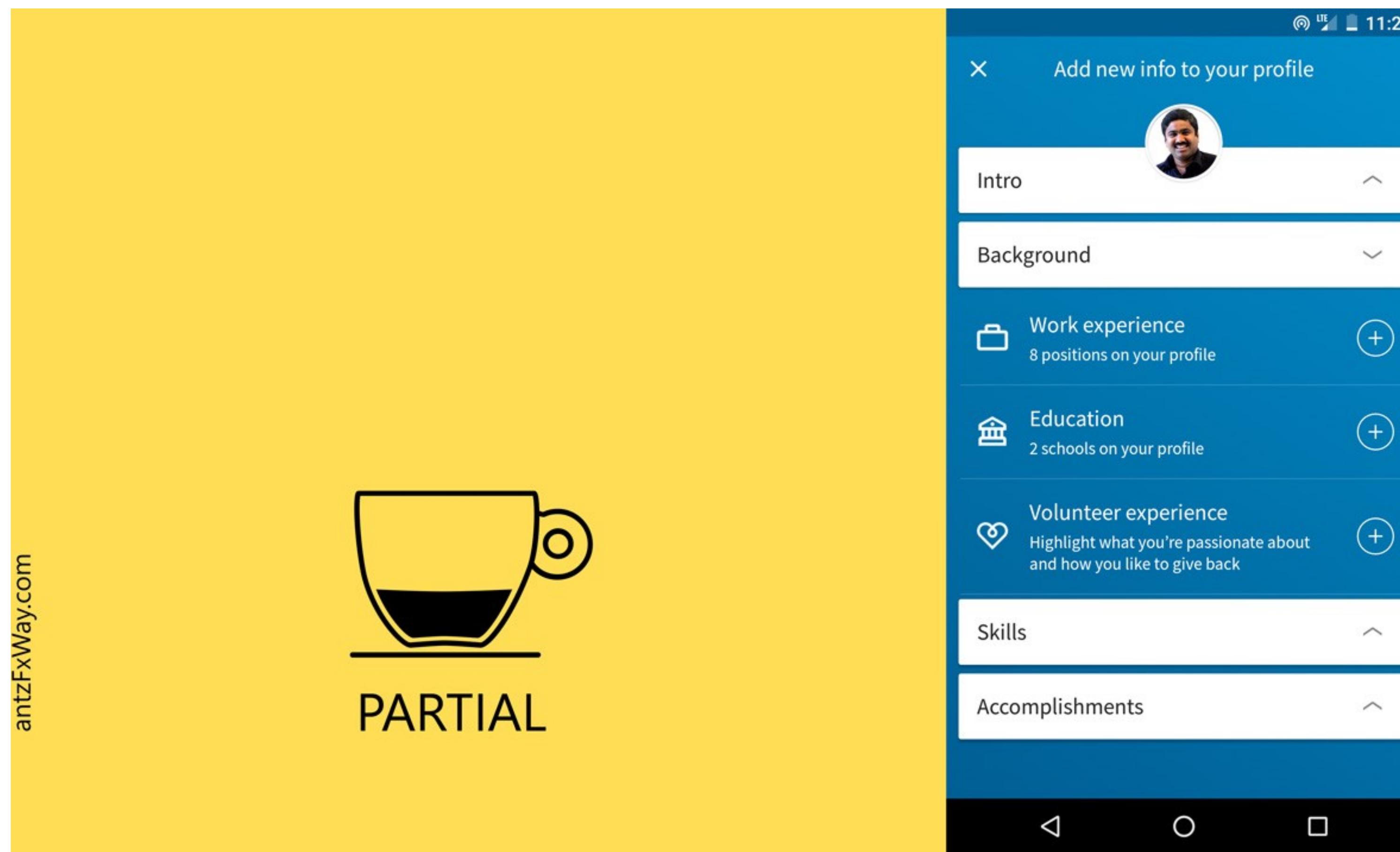
Blank/Empty State



Loading State



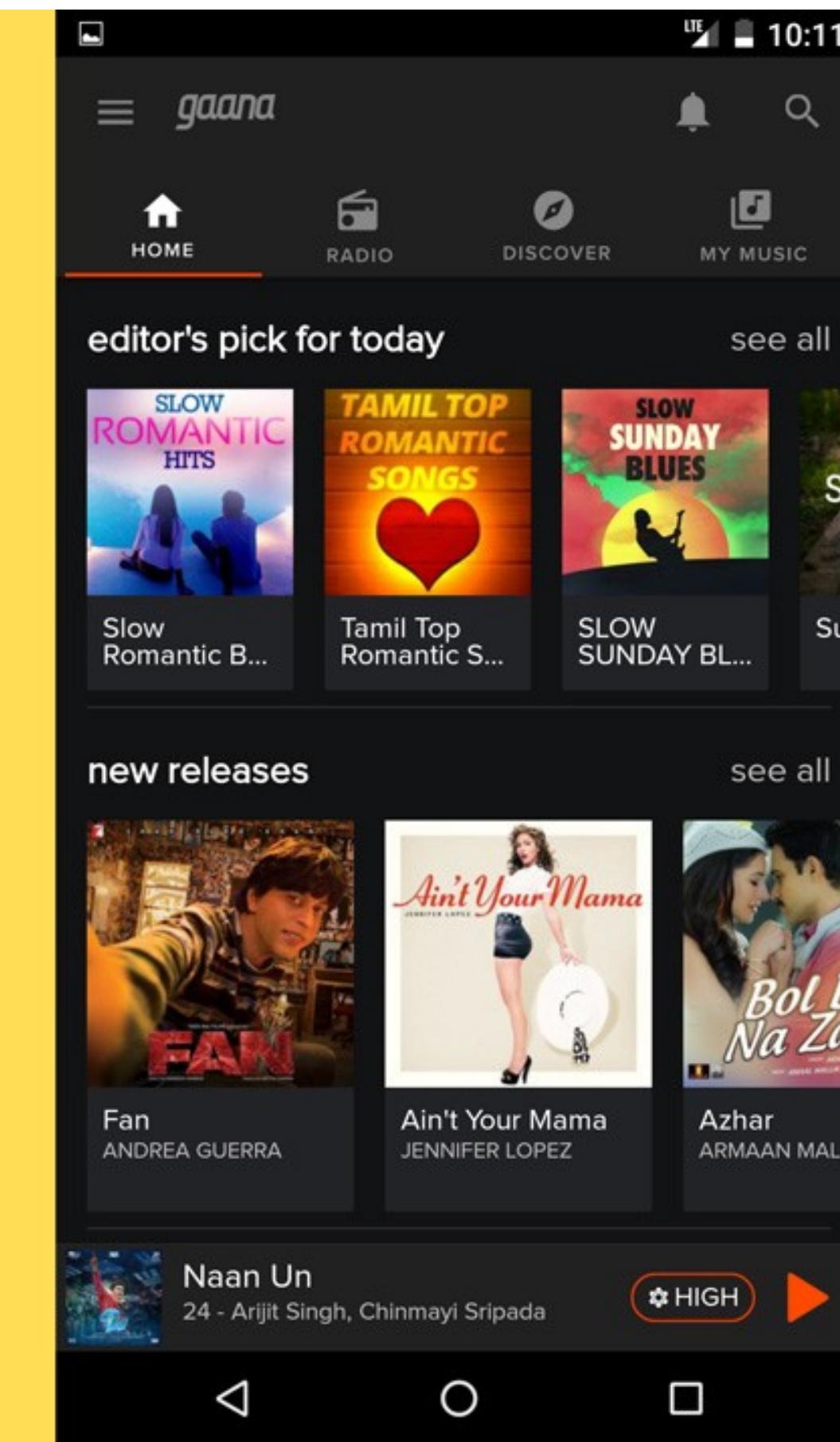
Partial State



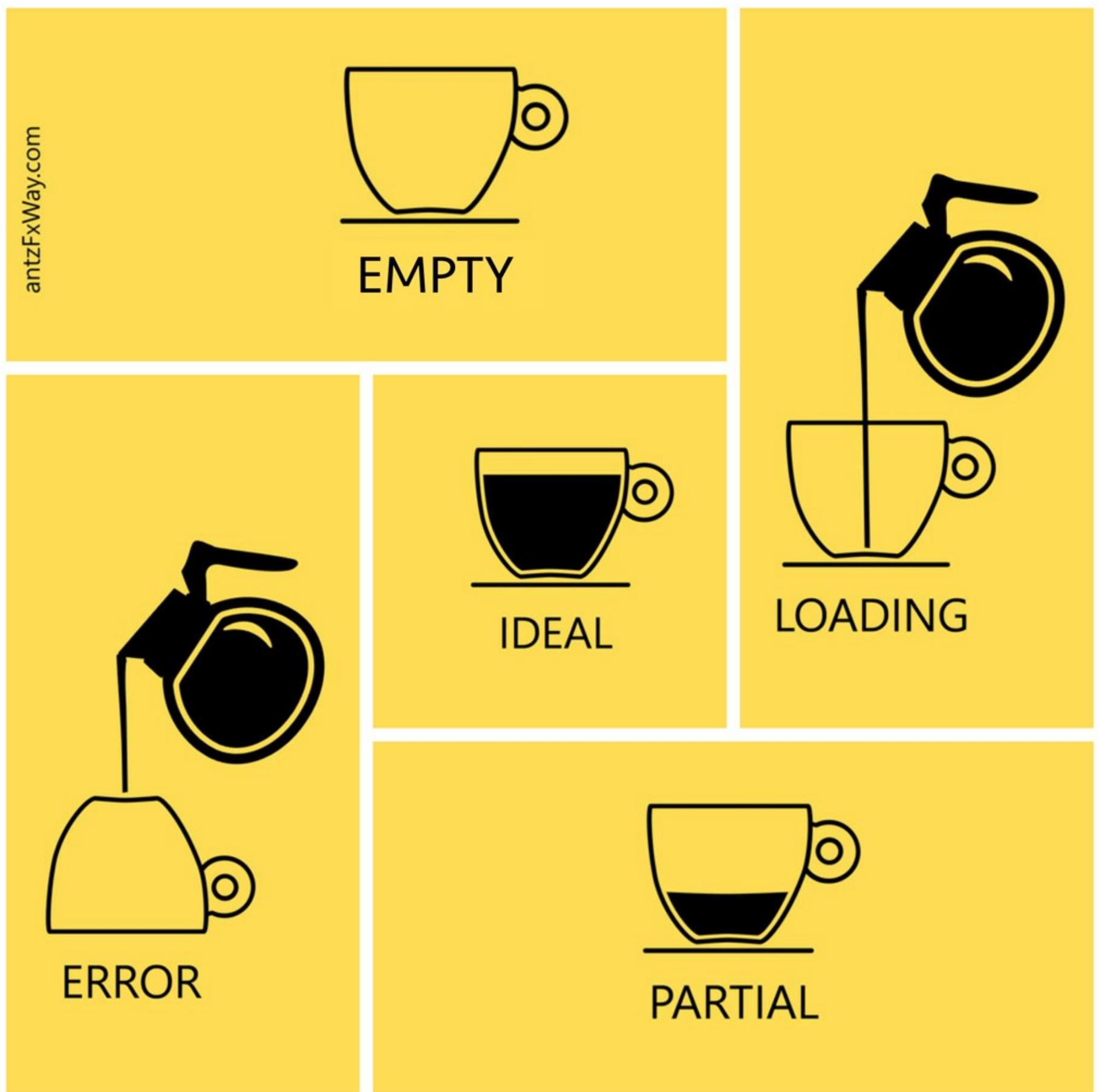
Ideal State



antzFxWay.com



Takeaway



User Interface

Autolayout

Коллекции данных

Состояния экранов

Материалы

<http://bit.ly/iossurf19lecture5>

