UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
UNIVERSITY OF SZEGED
Department of Software Engineering

Okosóra, Okostelefon és OkosTV - Apple Swift alapú alkalmazás fejlesztés

Dr. Bilicki Vilmos

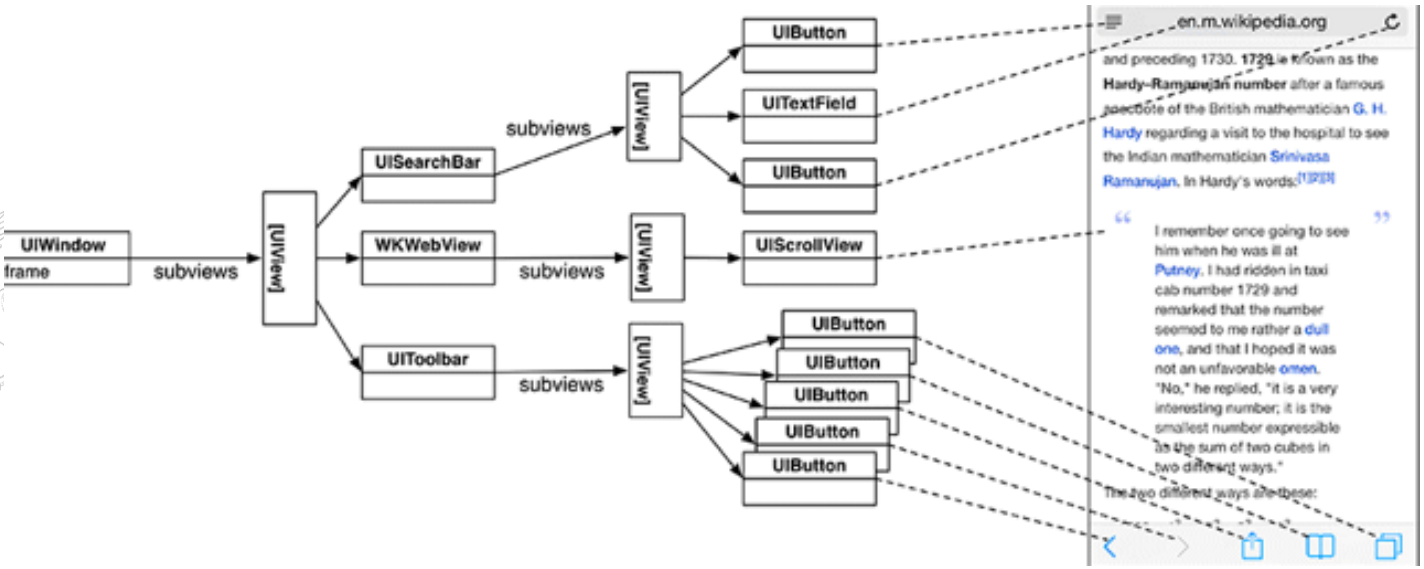
Szoftverfejlesztés Tanszék

MVC 2

- View elemek használata
- Koordináta-rendszer,
  - frame, view, bound
- Autolayout
- Elemek csoportosítása (Stack View)
- UIViewController
  - Outlet
  - Action
  - viewDidLoad()
- Storyboard belépési pont (Initial View Controller)

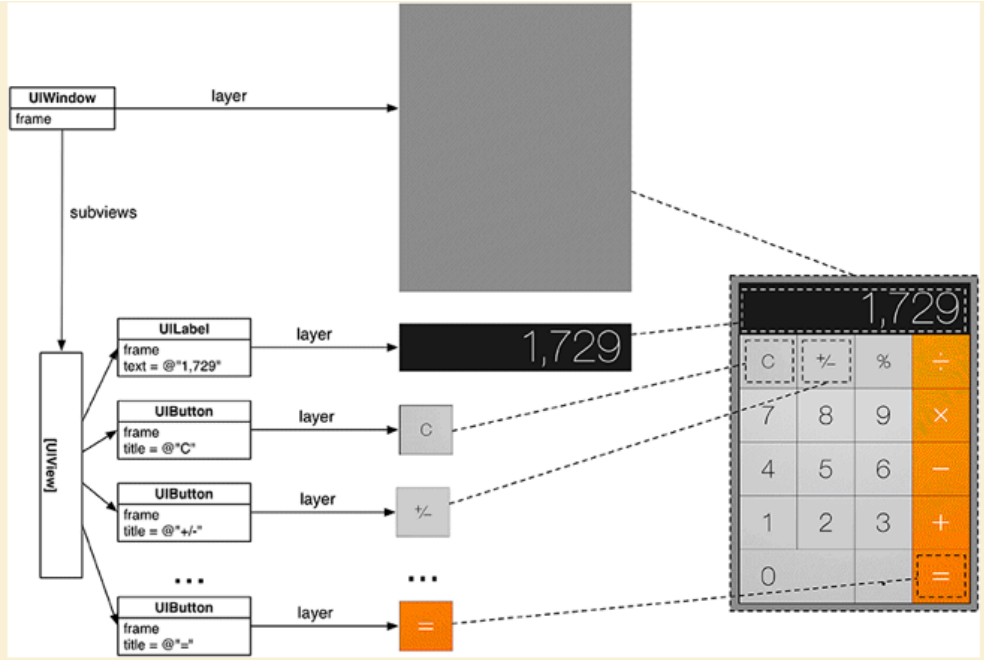
View

- UIView példány vagy származtatott osztálya
- Tudja hogyan rajzolja ki magát
- Eseményeket tud kezelni (érintés)
- View hierarchiába van melynek gyökere az alkalmazás ablak (UIWindow)
  - X db subview
  - 1 db superview



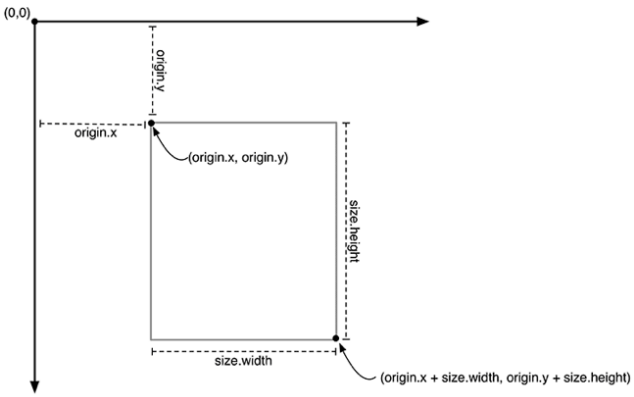
View hierarchia

- Minden view saját magát rajzolja ki
- Ezek a rétegek együtt kerülnek a képernyőre



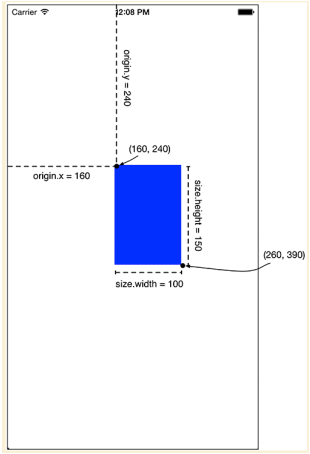
View: Frame, Bound

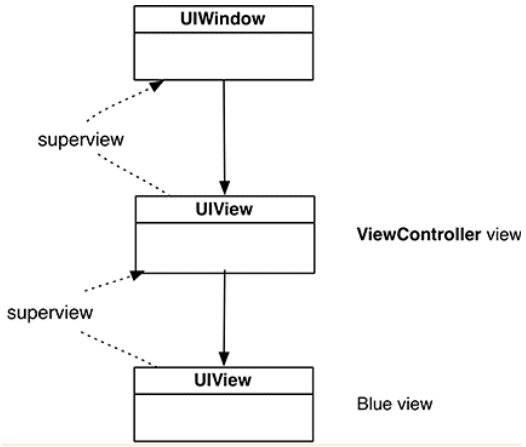
- View init(frame:), CGrect a superview koordináta rendszerében
  - View méret
  - Pozíciója a superview-hez képest
- Bound az aktuális view koordinátrendszerében



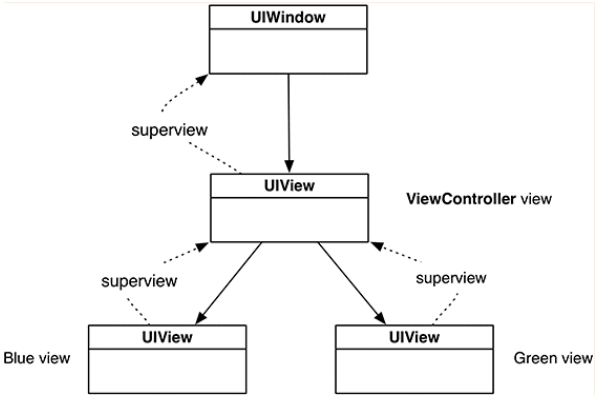
Példa

```
class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        let firstFrame = CGRect(x: 160, y: 240,
width: 100, height: 150)
        let firstView = UIView(frame: firstFrame)
        firstView.backgroundColor =
UIColor.blueColor()
        view.addSubview(firstView)
    }

}
```



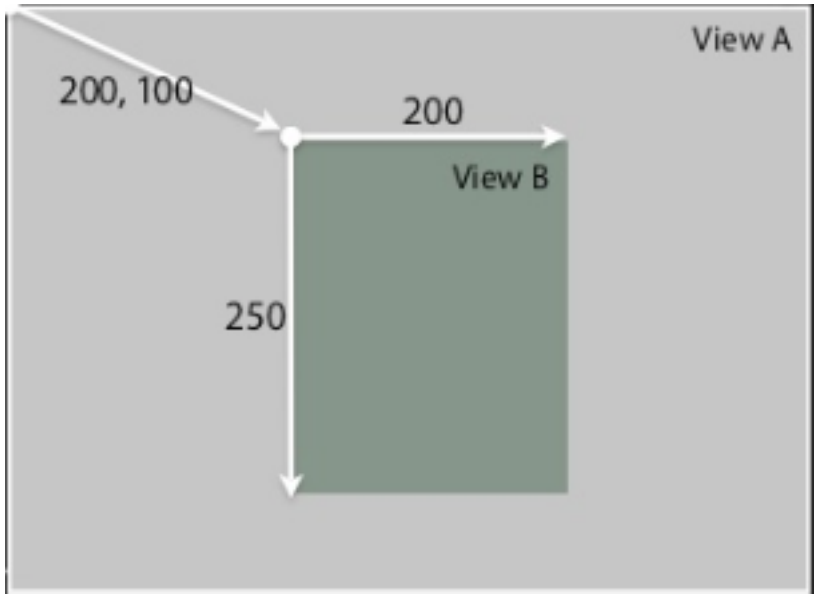



```
override func viewDidLoad() {
    super.viewDidLoad()

    let firstFrame = CGRect(x: 160, y: 240, width:
100, height: 150)
    let firstView = UIView(frame: firstFrame)
    firstView.backgroundColor =
UIColor.blueColor()
    view.addSubview(firstView)

    let secondFrame = CGRect(x: 20, y: 30, width:
50, height: 50)
    let secondView = UIView(frame: secondFrame)
    secondView.backgroundColor =
UIColor.greenColor()
    view.addSubview(secondView)
}
```

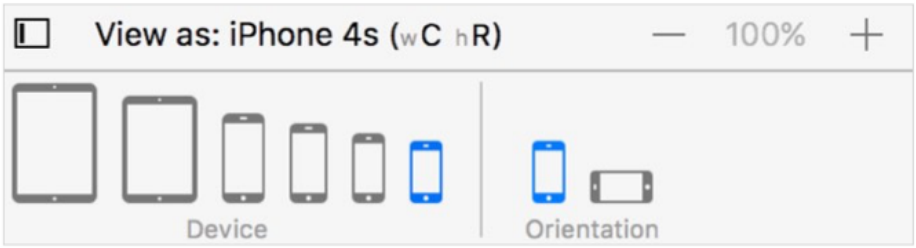


Frame vs. Bound

- Ha View-et használunk: Frame
- Ha View-et készítünk: Bound
- Ha eseménykezelőt írunk: Bound

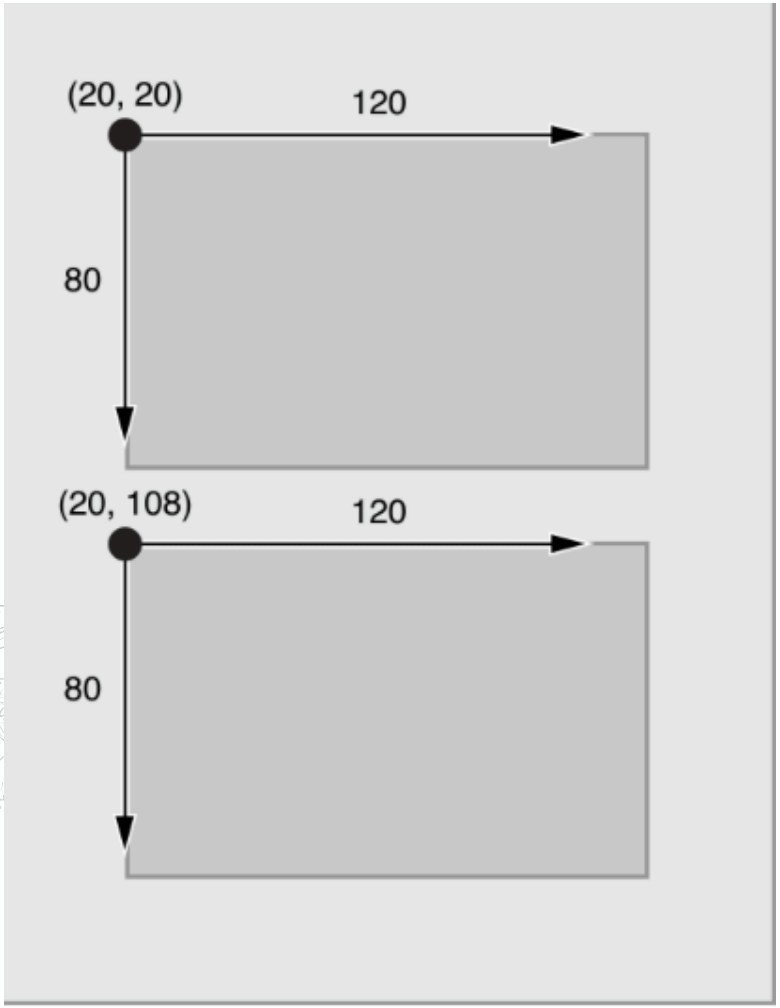


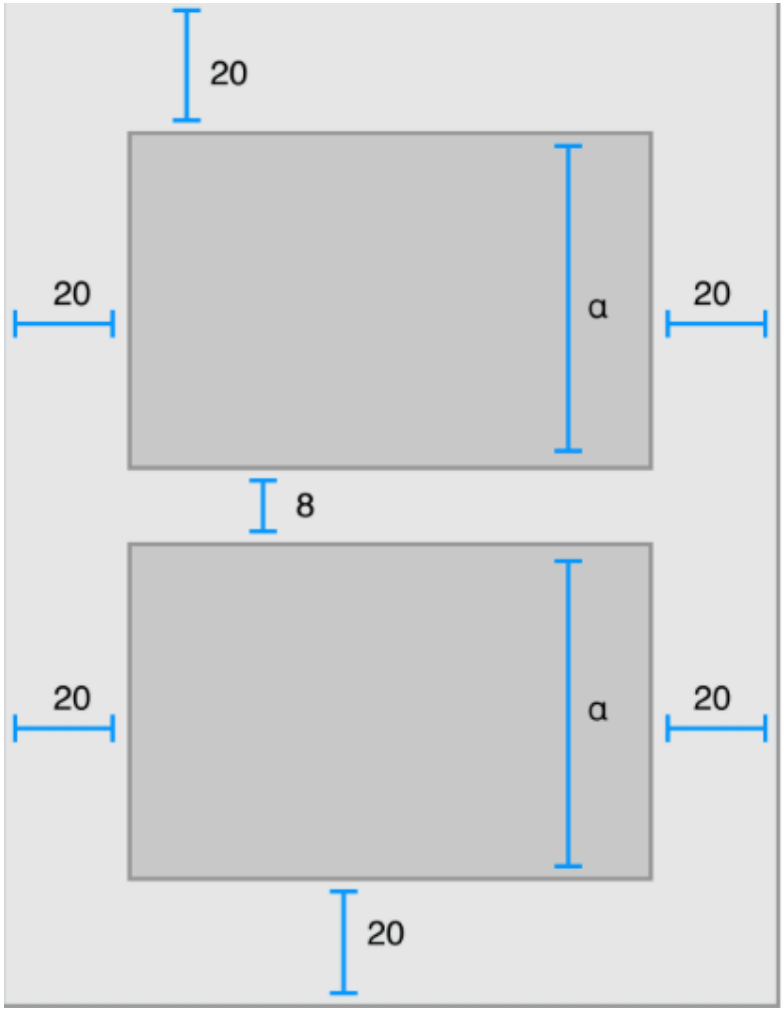
Autolayout



- A view hierarchiában lévő view elemek méretét és pozícióját automatikusan kiszámítja a megadott kényszerek alapján
- Dinamikusan reagál a külső és belső változásokra
  - Rotálás, megosztott ablak, képernyő méretek, felbontások, …
  - Nyelvek, ...
- Típusai
  - Kényszer mentes (verem nézetek)
  - Kényszerrel

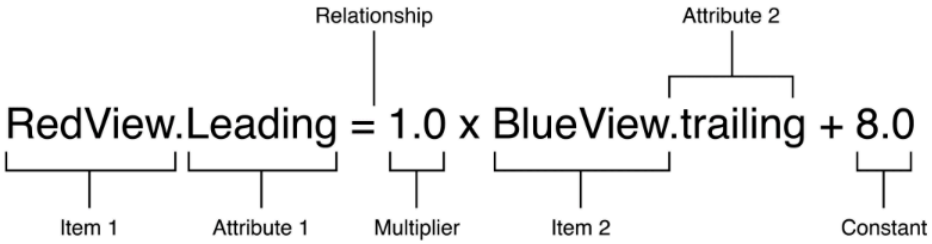
Autolayout vs Frame





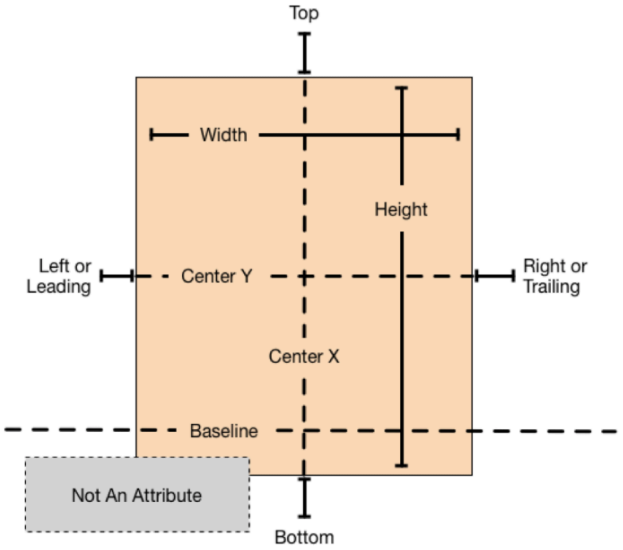
Autolayout - Kényszerek

- Lineáris egyenletek halmaza
- Minden kényszer egy egyenlet
- A cél olyan egyenletrendszert írni amelynek csak egy megoldása van



Autolayout attribútumok

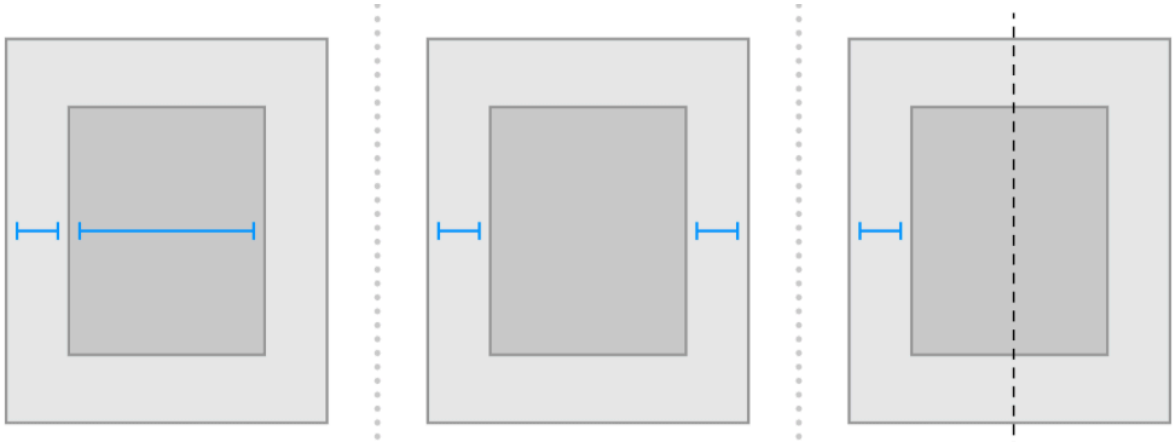
- Méret attribútumok
- Lokáció attribútumok



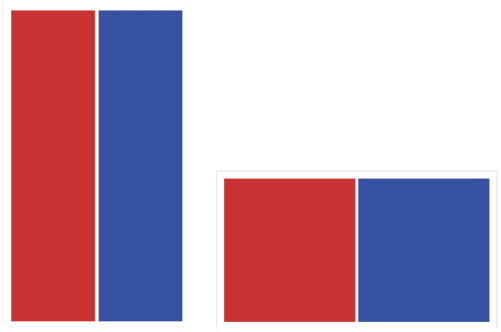

```
// Setting a constant height
View.height = 0.0 * NotAnAttribute + 40.0

// Setting a fixed distance between two buttons
Button_2.leading = 1.0 * Button_1.trailing + 8.0

// Aligning the leading edge of two buttons
Button_1.leading = 1.0 * Button_2.leading + 0.0

// Give two buttons the same width
Button_1.width = 1.0 * Button_2.width + 0.0

// Center a view in its superview
View.centerX = 1.0 * Superview.centerX + 0.0
View.centerY = 1.0 * Superview.centerY + 0.0

// Give a view a constant aspect ratio
View.height = 2.0 * View.width + 0.0
```

Egyértelmű, megoldható

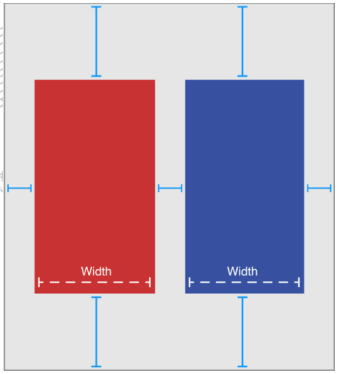
- Superview -adott
- Két kényszer minden view-ra minden dimmenzióra
- Számos megoldás létezik



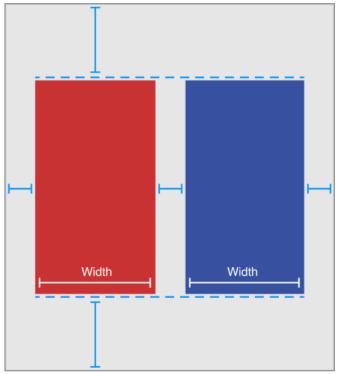
Példa



```
1   // Vertical Constraints
2   Red.top = 1.0 * Superview.top + 20.0
3   Superview.bottom = 1.0 * Red.bottom + 20.0
4   Blue.top = 1.0 * Superview.top + 20.0
5   Superview.bottom = 1.0 * Blue.bottom + 20.0
6
7   // Horizontal Constraints
8   Red.leading = 1.0 * Superview.leading + 20.0
9   Blue.leading = 1.0 * Red.trailing + 8.0
10  Superview.trailing = 1.0 * Blue.trailing + 20.0
11  Red.width = 1.0 * Blue.width + 0.0
```




```
1   // Vertical Constraints
2   Red.top = 1.0 * Superview.top + 20.0
3   Superview.bottom = 1.0 * Red.bottom + 20.0
4   Red.top = 1.0 * Blue.top + 0.0
5   Red.bottom = 1.0 * Blue.bottom + 0.0
6
7   //Horizontal Constraints
8   Red.leading = 1.0 * Superview.leading + 20.0
9   Blue.leading = 1.0 * Red.trailing + 8.0
10  Superview.trailing = 1.0 * Blue.trailing + 20.0
11  Red.width = 1.0 * Blue.width + 0.0
```

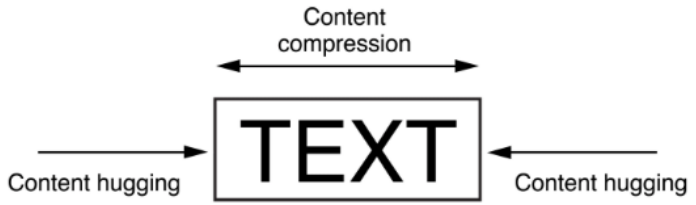


Kényszer prioritások

- Alapértelmezésben minden kényszer kötelező
- Lehet opcionális kényszereket is specifikálni
  - 1-1000 prioritás
  - 1000 – kötelező
  - A nagyobbtól a kisebb felé próbálja kielégíteni, ha egyet nem tud akkor megy a következőre

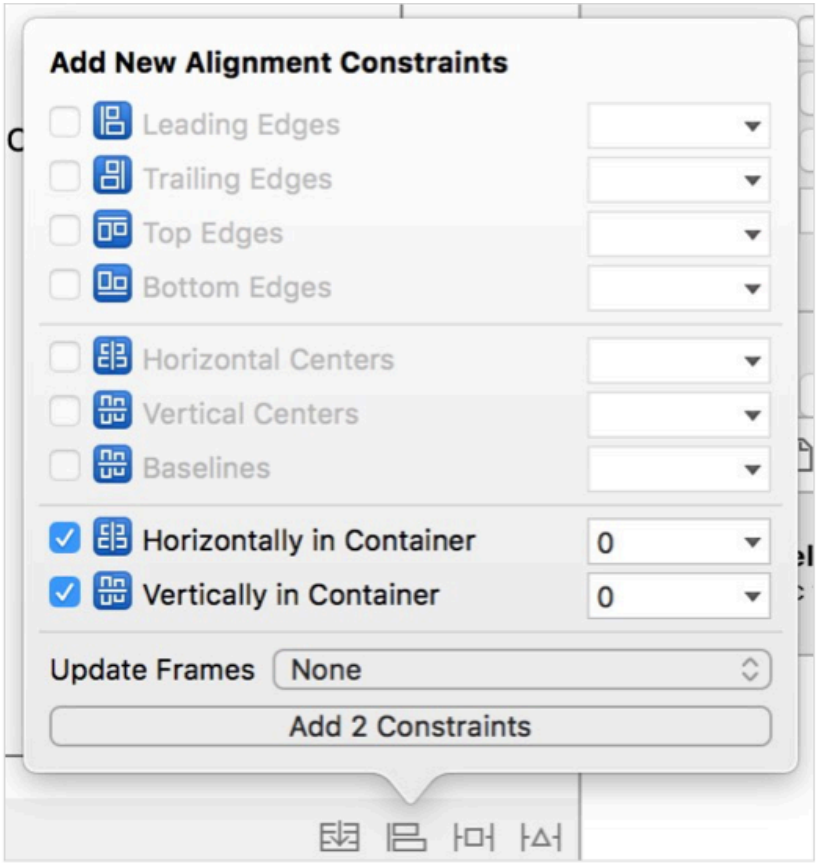
Belső tartalom méret

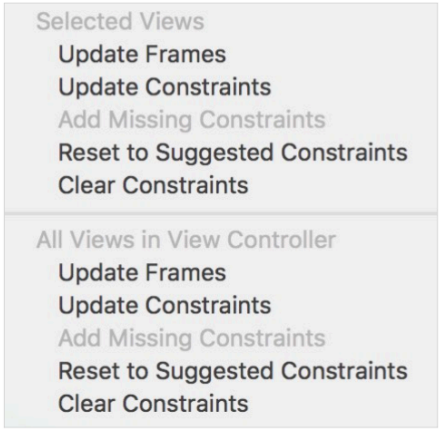
- Egyes view-oknak van természetes mérete
- Intrinsic content size
  - Széthúzás
  - Összenyomás
- Mindegyiknek prioritásai vannak

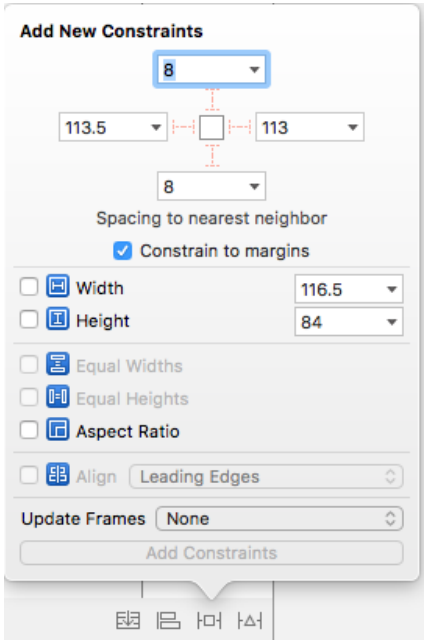



```
1   // Compression Resistance
2   View.height >= 0.0 * NotAnAttribute + IntrinsicHeight
3   View.width >= 0.0 * NotAnAttribute + IntrinsicWidth
4
5   // Content Hugging
6   View.height <= 0.0 * NotAnAttribute + IntrinsicHeight
7   View.width <= 0.0 * NotAnAttribute + IntrinsicWidth
```

XCode

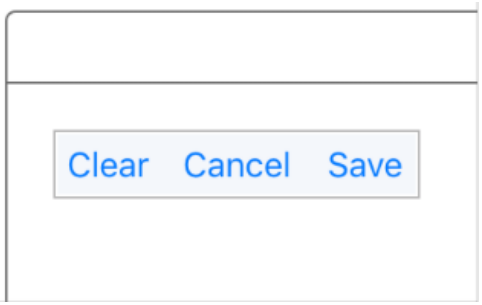

Add New Alignment Constraints
Leading Edges
Trailing Edges
Top Edges
Bottom Edges
Horizontal Centers
Vertical Centers
Baselines
Horizontally in Container
0
Vertically in Container
0
Update Frames
None
Add 2 Constraints

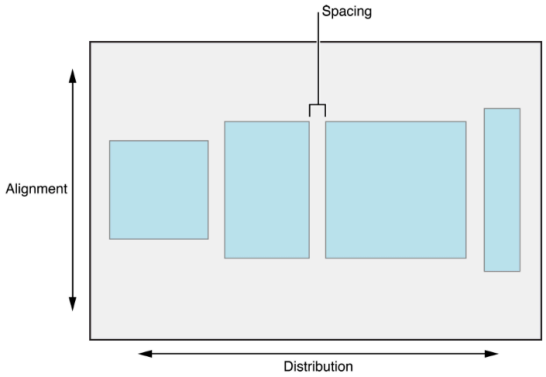

Selected Views
Update Frames
Update Constraints
Add Missing Constraints
Reset to Suggested Constraints
Clear Constraints
All Views in View Controller
Update Frames
Update Constraints
Add Missing Constraints
Reset to Suggested Constraints
Clear Constraints


Add New Constraints
8
113.5
113
8
Spacing to nearest neighbor
Constrain to margins
Width
116.5
Height
84
Equal Widths
Equal Heights
Aspect Ratio
Align
Leading Edges
Update Frames
None
Add Constraints

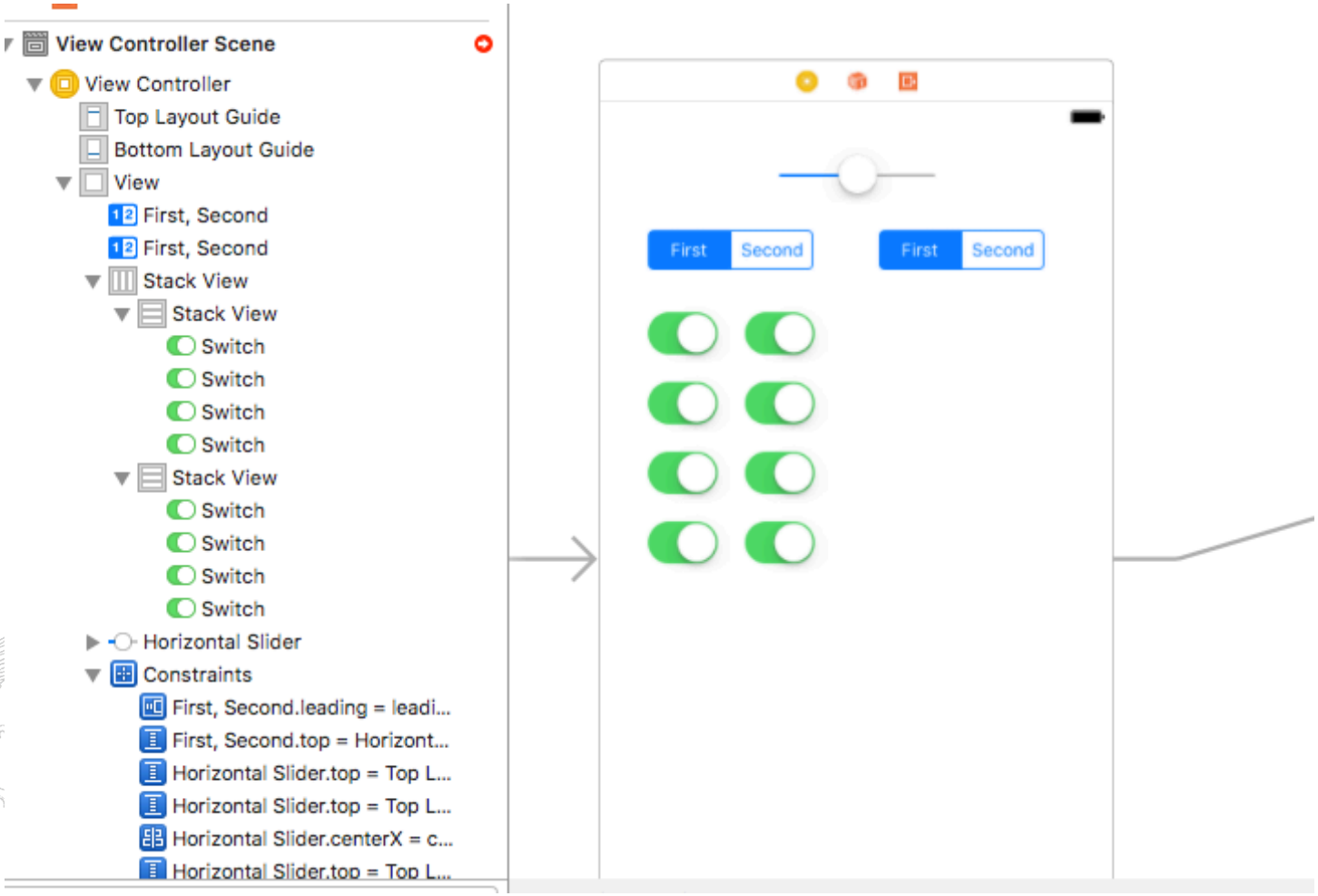
Kényszerek nélküli auto layout

- Stack View – egyszerű GUI gyors definiálása
- Egy Stack View sorokba vagy oszlopokba rendezi az elemeket
- Az elemek tulajdonságai alapján lesznek elrendezve
  - axis – horizontális vs vertikális
  - orientation – ua (NSStackview)
  - distribution – az tengelyen az elhelyezés
  - alignment – stack view-ben a nem sorrendi elrendezés
  - spacing - térköz





Példa


View Controller Scene
View Controller
Top Layout Guide
Bottom Layout Guide
View
First, Second
First, Second
Stack View
Stack View
Switch
Switch
Switch
Switch
Stack View
Switch
Switch
Switch
Switch
Horizontal Slider
Constraints
First, Second.leading = leadi...
First, Second.top = Horizont...
Horizontal Slider.top = Top L...
Horizontal Slider.top = Top L...
Horizontal Slider.centerX = c...
Horizontal Slider.top = Top L...
First
Second
First
Second

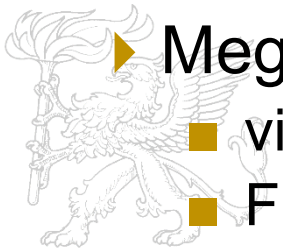
UIViewController

- Az alkalmazás belső struktúrájának alapja
- Minden alkalmazásnak van legalább egy ViewController-je
- UIViewController a view-ek kezelésére szolgáló metódusokat és tulajdonságokat definiál
  - Tartalom
  - Tároló
- Specializált osztályok
  - UITableViewController
  - UICollectionViewController
  - …

UIViewController

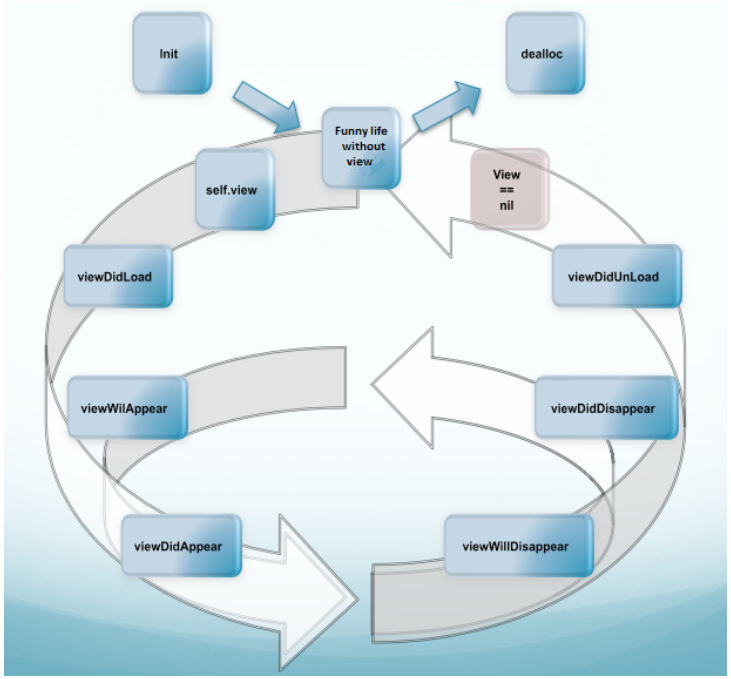
```
class MyViewController: UIViewController {
    @IBOutlet weak var myButton : UIButton!
    @IBOutlet weak var myTextField : UITextField!

    @IBAction func myButtonAction(sender: id)
}
```

- View-et lehet hozzá kötni
- UIKit View betöltés
  - Storyboard alapján létrehozza
  - Összeköti az outlet és action elemeket
  - Inicializálja a ViewController view adattagját a root view-val.
  - Meghívja a controller awakeFromNib metódusát
  - Meghívja a viewDidLoad metódust (itt kezeljük az al view-okat)
- Megjelenítés előtt
  - viewWillAppear
  - Frissíti a view kinézetet
  - Megjeleníti a View-ot a képernyőn
  - Meghívja a viewDidAppear metódust



UIViewController fontosabb metódusok

- viewDidLoad – osztály létrehozásakor és betöltésekor a xib fájlból. Kezdő inicializálás, egyszeri dolgok. (ha a memóriából törlődik akkor többször)
- viewWillAppear – Mielőtt megjelenne. Mezők, elemek elrejtése, olyan műveletek aminek minden megjelenés előtt meg kell valósulnia
- viewDidAppear – animációk elkezédse, külső adatok betöltése
- viewDissapear
- viewDidUnload/viewDidDispose – takarítás (Objective C)



viewDidLoad()

- Akkor hívódik meg amikor a view hierarchia a memóriába kerül
- Nib/Xib és program alapú létrehozásnál is meghívódik
- További inicializációt érdemes ide tenni


```
    override func loadView() {
        // Create a map view
        mapView = MKMapView()

        // Set it as *the* view of this view
controller
        view = mapView
    }

    override func viewDidLoad() {
        super.viewDidLoad()

        print("MapViewController loaded its
view.")
    }
```

Outlet/Action

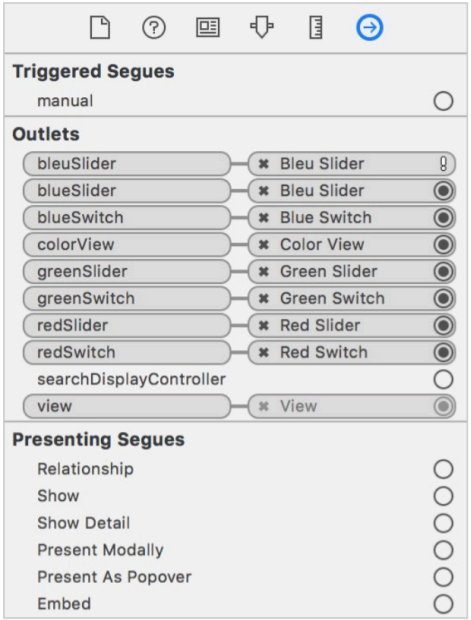
- @IBOutlet, @IBAction – interface builder makró .nib fájl vs .h és .m fájlok

```
import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var myLabel: UILabel! {
        didSet {
            myLabel.textColor = UIColor.purpleColor()
        }
    }

    @IBOutlet weak var myOtherLabel: UILabel! {
        didSet {
            myOtherLabel.textColor = UIColor.yellowColor()
        }
    }

    @IBOutlet weak var myButton: UIButton! {
        didSet {
            myButton.tintColor = UIColor.magentaColor()
        }
    }

    override func viewDidLoad() {
        super.viewDidLoad()
    }
}
```

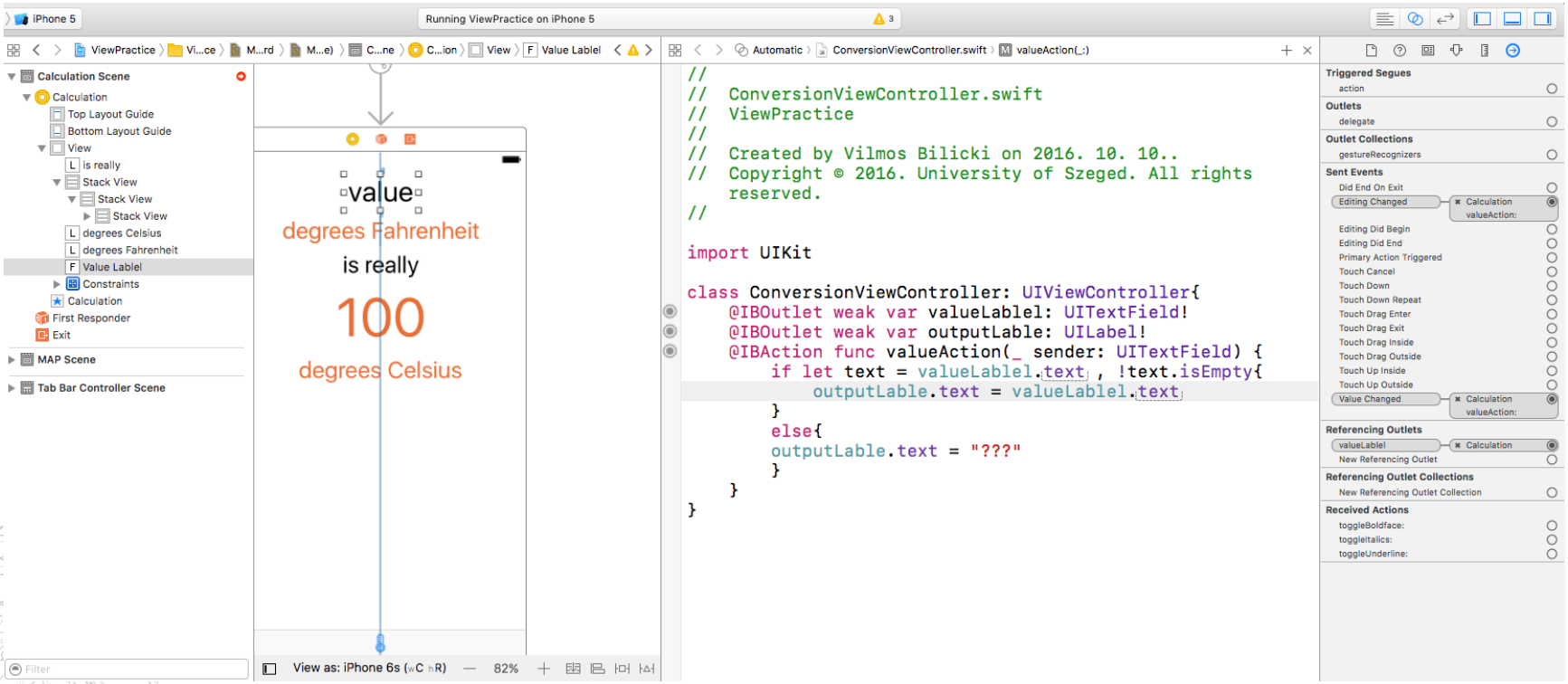
```
@IBAction func mainButton(sender: UIButton) {
    println(sender.tag)
}
```

Outlet/Action módosítás

- Nem elegendő a swift forráskódban átírni
- A NIB fájlban is le van tárolva a név, ezt nem cseréli ki automatikusan

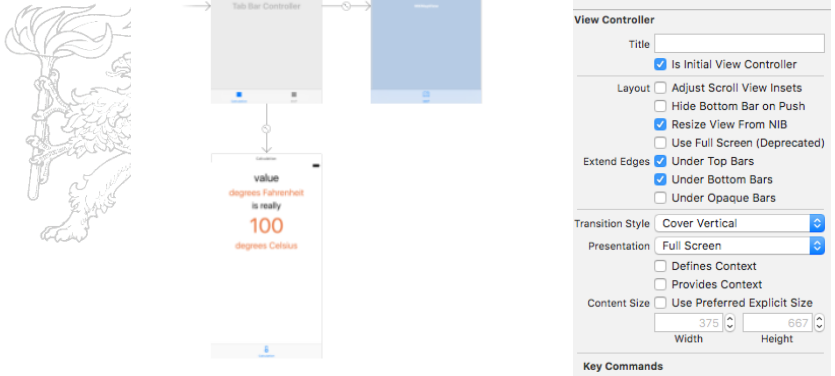


XCode


iPhone 5
Running ViewPractice on iPhone 5
ViewPractice
Calculation Scene
Calculation
Top Layout Guide
Bottom Layout Guide
View
is really
Stack View
Stack View
Stack View
degrees Celsius
degrees Fahrenheit
Value Lablel
Constraints
Calculation
First Responder
Exit
MAP Scene
Tab Bar Controller Scene
value
degrees Fahrenheit
is really
100
degrees Celsius
View as: iPhone 6s
82%
ConversionViewController.swift
valueAction(_:)
//
// ConversionViewController.swift
// ViewPractice
//
// Created by Vilmos Bilicki on 2016. 10. 10..
// Copyright © 2016. University of Szeged. All rights reserved.
//
import UIKit
class ConversionViewController: UIViewController{
@IBOutlet weak var valueLablel: UITextField!
@IBOutlet weak var outputLable: UILabel!
@IBAction func valueAction(_ sender: UITextField) {
if let text = valueLablel.text , !text.isEmpty{
outputLable.text = valueLablel.text
}
else{
outputLable.text = "???"
}
}
}
Triggered Segues
action
Outlets
delegate
Outlet Collections
gestureRecognizers
Sent Events
Did End On Exit
Editing Changed
Editing Did Begin
Editing Did End
Primary Action Triggered
Touch Cancel
Touch Down
Touch Down Repeat
Touch Drag Enter
Touch Drag Exit
Touch Drag Inside
Touch Drag Outside
Touch Up Inside
Touch Up Outside
Value Changed
Referencing Outlets
valueLablel
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Received Actions
toggleBoldface:
toggleItalics:
toggleUnderline:

Storyboard belépési pont (Initial View Controller)

- UIStoryboard
  - Tartalmazza a teljes vagy részleges ViewController gráfot
  - InitalViewController – az indulás után betöltődő ViewController



MVC 2

- View elemek használata
- Koordináta-rendszer,
  - frame, view, bound
- Autolayout
- Elemek csoportosítása (Stack View)
- UIViewController
  - Outlet
  - Action
  - viewDidLoad()
- Storyboard belépési pont (Initial View Controller)

A következő előadás tartalma

- MVC modellek összekötése
- TabBar, SplitView, NavBar
- Segue: Modal, Unwind, Popover, Embed
- Navigation Controller (embed)
- prepareForSegue()