



Okosóra, Okostelefon és OkosTV - Apple Swift alapú alkalmazás fejlesztés

Dr. Bilicki Vilmos
Szoftverfejlesztés Tanszék

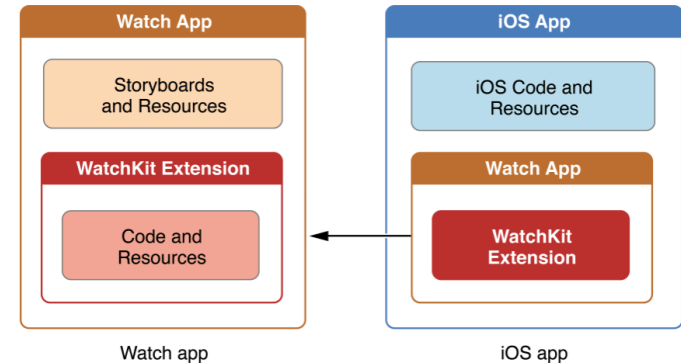
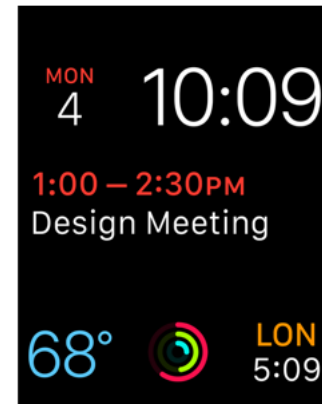
Alkalmazás készítése okosóra

- ▶ WatchApp (storyboard, források)
- ▶ WatchKit (felhasználói interakciók)
- ▶ WatchApp architektúra
 - Adatmegosztás a WatchApp és a WatchKit között
 - Notifications, complications, dock

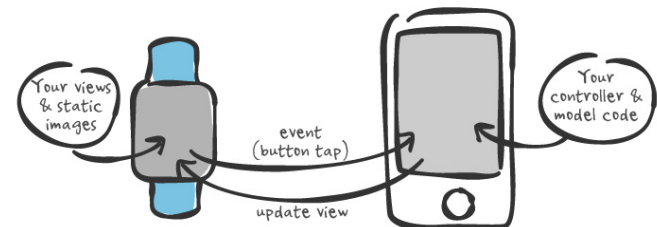


Apple Watch

- ▶ Watch App
 - Storyboards
 - Erőforrások
- ▶ Watch Kit
 - Watch App-on belül
 - Interfészekhez tartozó kód
 - Interakciók lekezelése
- ▶ iOS app része
 - Az órára másolódik



Régen: WatchOS 1



Watch App

- ▶ Apple Watch Home Screen-ről elindított alkalmazás
 - Több képernyő
 - Tartalom
 - Interakció
- ▶ Complications
 - the act of forming a unified idea or impression from a number of sense data, memories, etc.
 - *complication* comes from watch making, where the addition of features added complexity to the watch construction
 - Kis vizuális elemek
 - A felhasználó által testreszabhatóak
 - Ajánlott minden alkalmazáshoz legalább egy ilyen létrehozni



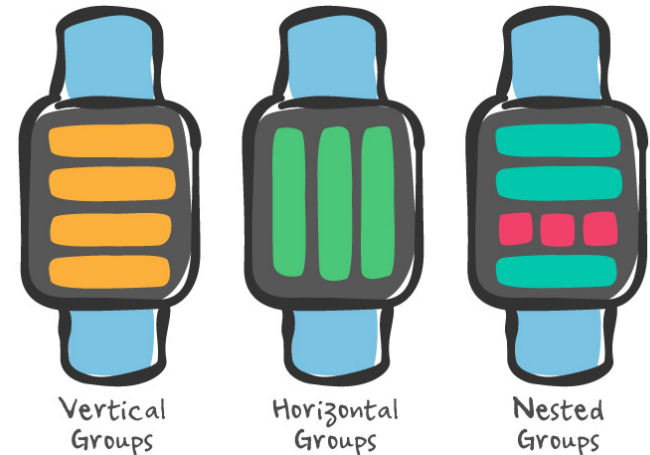
Watch App

► Értesítések

- Első megjelenítés – short look
- Másodlagos – long look (ha a csukló felemelve marad)
 - Egyedi grafika
 - Dinamikus tartalom
 - ...

► Felhasználói interfészek

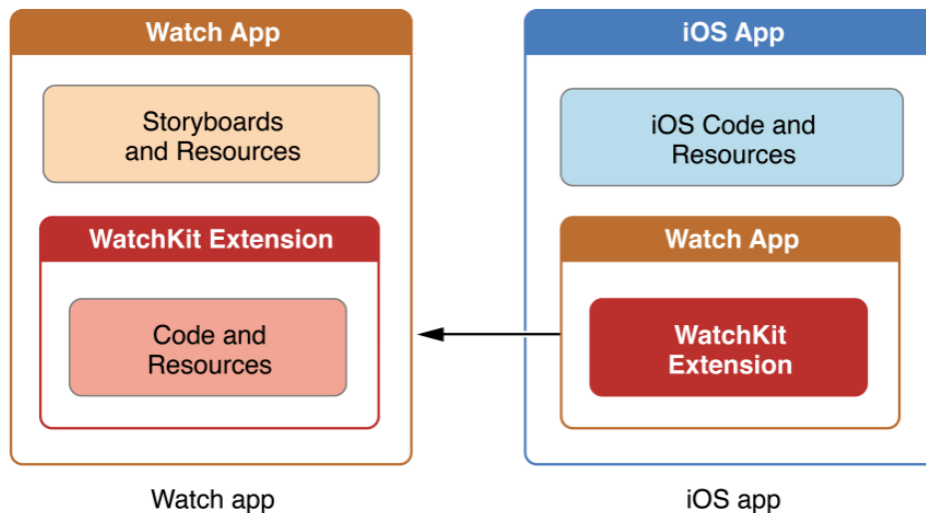
- Gyors információ megjelenítés
- Gyors navigáció, interakció
- Kiegészítot a telefonon lévő alkalmazást





App Struktúra

- ▶ Két módon hozhatunk létre Watch app-ot
 - Meglévő iOS projekthez adunk Watch – appot
 - iOS app, Watch app, WatchKit kiterjesztés
 - Új Watch appot hozunk létre



Watch app

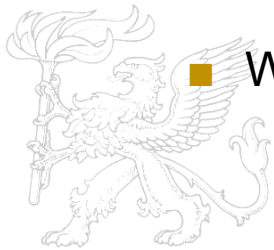
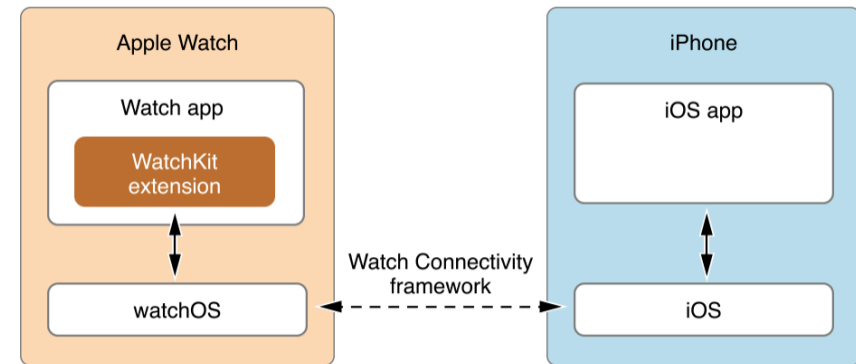
iOS app

Architektúra

- ▶ A Watch app-ot a Home screen-en kiválasztva
 - a watchOS elindítja az előtérben
 - Addig marad ott amíg a felhasználó el nem tünteti
 - Akkor is lefuthat amikor nincs előtérben (periodikusan)
 - Háttérben futnak amennyiben
 - A complication látható
 - Az app a dock-ban van
 - Amikor értesítéseket kezel

▶ Két csomag

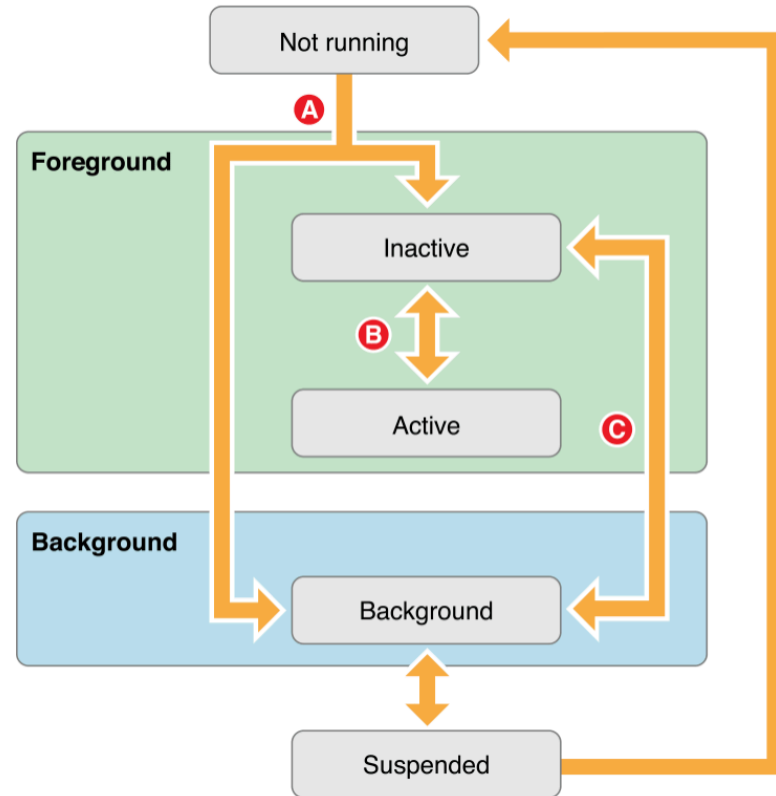
- Watch app
 - Storyboard
- WatchKit bővítmény csomag
 - Kód
 - Erőforrások
 - Complications (CLKComplicationDataSource protocol)
- IOS app – WatchKit kommunikáció – Watch Connectivity keretrendszer





Alkalmazás élelciklus

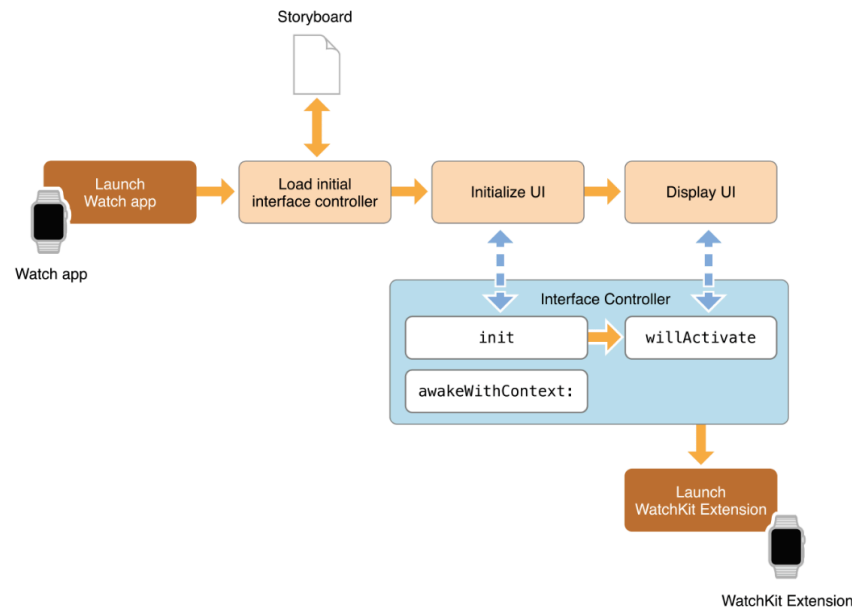
- WKExtensionD
elegate protocol





Scene menedzsment

- ▶ Teljes képernyősek
 - Egy app több scene
- ▶ WatchKit extension konktroller: WKInterfaceController



Dock és pillanatképek

- ▶ A tartalom gyors áttekintése és az alkalmazás indítása
- ▶ WKSnapshotRefreshBackgroundTask
- ▶ `scheduleSnapshotRefreshWithPreferredDate:userInfo:scheduledCompletion:`



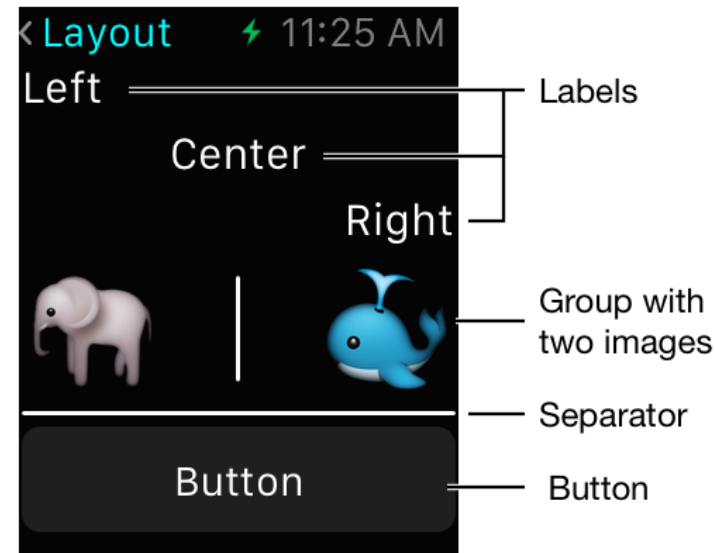
Adatmegosztás

- ▶ Watch app – WatchKit
 - Külön fájlok (font,...)
 - WatchKit extension-ba egy fájl
 - Megosztott csoport tároló
- ▶ Watch Connectivity
 - Kétirányú kommunikáció (fájl/adat)
 - WatchKit extension – IOS App
 - Pl.: `updateApplicationContext:error:`



UI Alapok

- ▶ Storyboard scene-k
 - Minden scene egy képernyő darab
- ▶ WatchApp interfész
 - Scene-k vertikális verem nézetben
 - Méret/Pozíció/Rendezés
 - Csoport objektumok
 - Vertikális/Horizontalis rendezés
 - Futásidőben változtatható
 - Adat értékek
 - Vizuális megjelenés
 - Méret,..., transzparencia
 - Megjelenés, elfedés



Navigáció

► Képernyők között

■ Lap alapú

- Nincs szoros kapcsolat az oldalak között
- Jobbra/Balra lépés

■ Hierarchikus

- Mestre/Részlet interfészek
- Push/Pop (gomb/csoport/táblázat sor)
- `pushViewControllerWithName:context:`

■ Modális ablakok

- `presentControllerWithName:context:`
- `presentControllerWithNames:contexts:`



Interfész objektumok

- ▶ WatchKit extension -> WatchApp UI
 - WKInterfaceObject
 - Proxy objektum
 - Egyirányú
 - Outlet

```
class MySwiftInterfaceController {
    @IBOutlet weak var label: WKInterfaceLabel!
}

override init() {
    // Initialize properties here.
    super.init()

    // It is now safe to access interface objects.
    label.setText("Hello New World")
}
```



Felhasználói interakciók

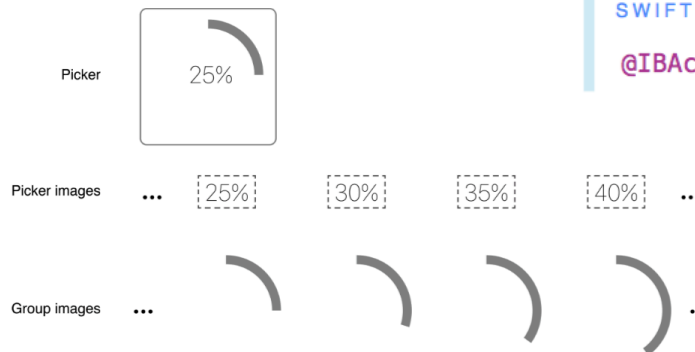
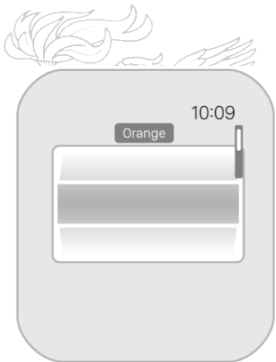
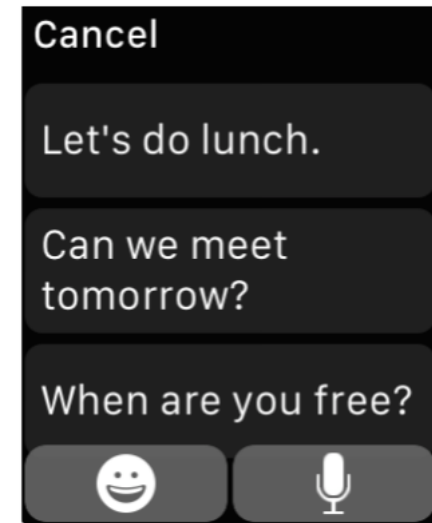
- ▶ Gomb
- ▶ Kiválasztó
- ▶ Kapcsoló
- ▶ Csúszka
- ▶ Menü



Object	Objective-C	Swift
Button	- (IBAction)buttonAction	@IBAction func buttonAction()
Picker	- (IBAction)pickerAction: (NSInteger)index	@IBAction func pickerAction(index: Int)
Switch	- (IBAction)switchAction: (BOOL)on	@IBAction func switchAction(value: Bool)
Slider	- (IBAction)sliderAction: (float)value	@IBAction func sliderAction(value: Float)
Menu Item	- (IBAction)menuItemAction	@IBAction func menuItemAction()

Szöveg bevitel/Kiválasztó

- ▶ Bevitel
 - Diktálás
 - Előre megadott lista
- ▶ Picker (Digitális Korona)
 - Picker items
 - Coordinated interface objects

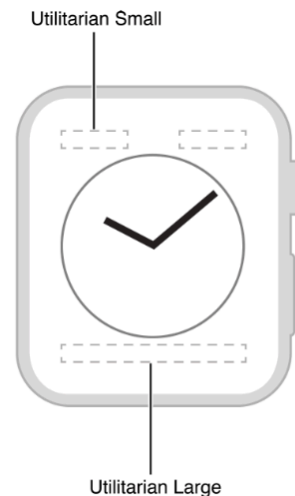
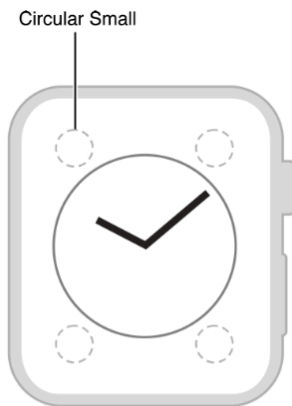
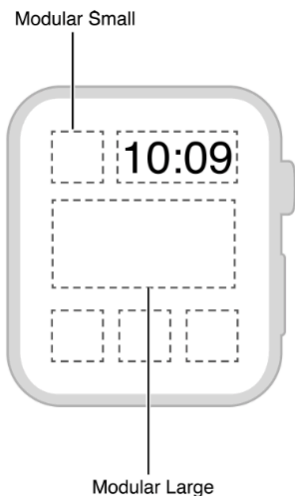
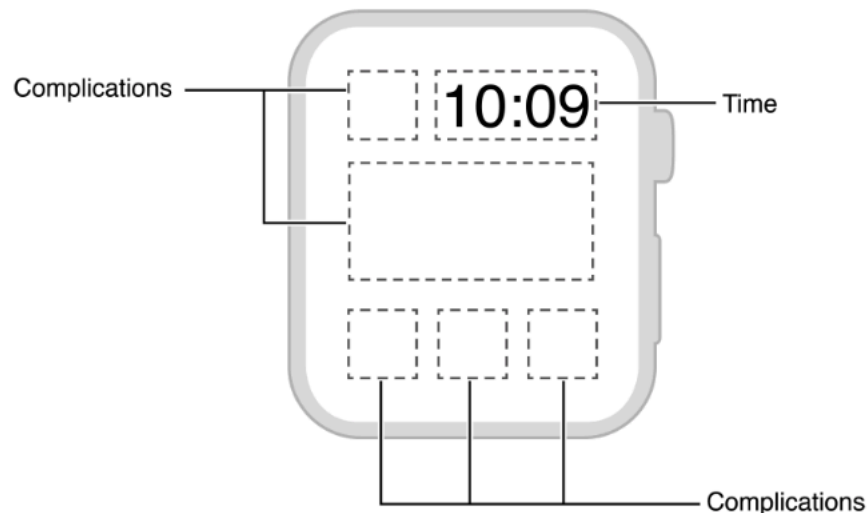


SWIFT

```
@IBAction func pickerAction(index: Int)
```

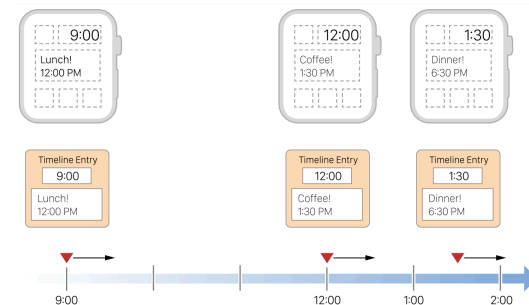

Complications alapok

- Az óra számlapján megjelenő apró elemek



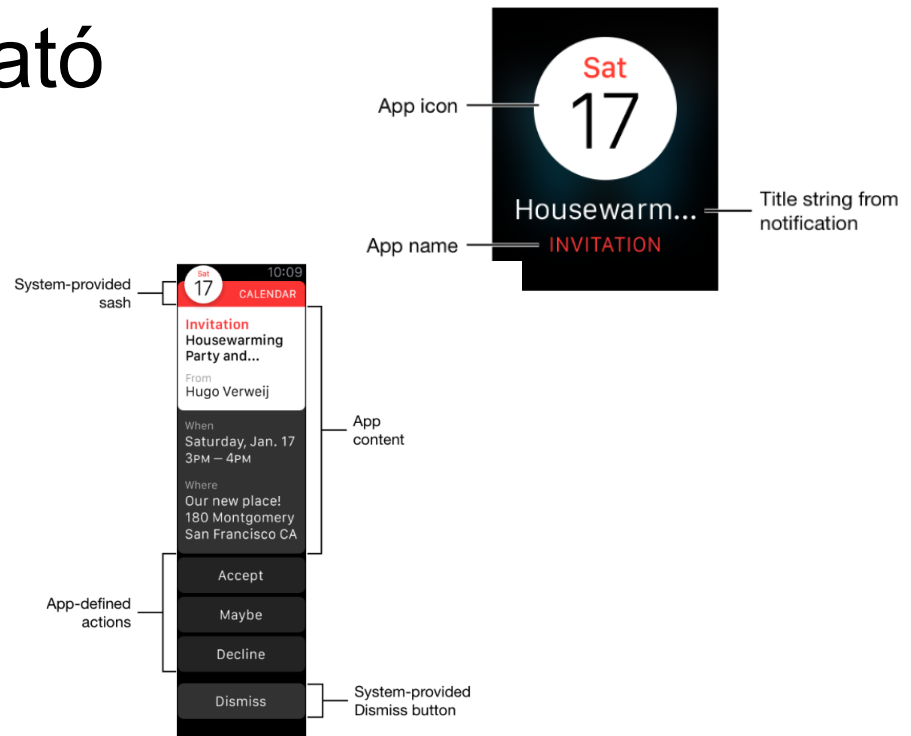
Működése

- ▶ Amennyiben az alkalmazás rendelkezik aktív Complication-nal
 - Rövid infó a felhasználónak
 - Memóriában marad az alkalmazás
 - Több háttér idő
- ▶ ClockKit előre elkészíti
 - Előre elkéri az adatot is
 - CLKComplicationDataSource
 - Timeline - NSDate



Értesítések

- ▶ Amikor az alkalmazás nem fut, így tud a felhasználóval kommunikálni
- ▶ Idő/Lokáció segítségével időzíthető
- ▶ Push is használható
- ▶ Short-look
- ▶ Long-look



Notification Type	Source	Destination
Local	iOS app	Either Apple Watch or iPhone, depending on the locked/unlocked state of both devices.
Local	WatchKit extension	Apple Watch only
Remote	Your server	Either Apple Watch or iPhone, depending on the locked/unlocked state of both devices.
Silent	Your server	iPhone only

Alkalmazás készítése okosóra

- ▶ WatchApp (storyboard, források)
- ▶ WatchKit (felhasználói interakciók)
- ▶ WatchApp architektúra
 - Adatmegosztás a WatchApp és a WatchKit között
 - Notifications, complications, dock

