

IOS

MKZ 2017

L. Pešička

iOS

- ◉ mobilní operační systém od firmy Apple
- ◉ vydán 2007 pro iPhone, iPod Touch
- ◉ 2010 rebranding iPhone OS => iOS
- ◉ pouze na vlastním HW
 - většina zařízení používá poslední verzi systému
 - možnost vyladění spolupráce HW/SW
- ◉ Pravidelné updaty OS pro drtivou většinu používaných zařízení (x Android)
- ◉ obchod AppStore (70 procent vývojář, 30 Apple)

MULTITASKING

- ◉ **Není to stejné jako u PC**
 - Výkonnost zařízení
 - Výdrž baterie
- ◉ **Stisk HOME tlačítka**
 - Uspání aplikace
 - Ukončení aplikace (v případě potřeby uvolní paměť)

<http://jablickar.cz/jak-skutecne-funguje-multitasking-v-ios/>

STAV APLIKACE

- ◉ Běží
 - Spuštěna a běží na popředí
- ◉ Na pozadí
 - Stále spuštěna a běží na pozadí
- ◉ Pozastavena
 - Do 5s od HOME tlačítka, když není používána
 - Pokud je používána -> o 10 min odloženo pozast.
 - Není spuštěna, ale stále zabírá paměť
- ◉ Neaktivní
 - Spuštěna, ale nepřijímá příkazy (zamknutí)
- ◉ Neběží
 - Aplikace byla ukončena

APLIKACE NA POZADÍ

- ◉ Navigace
- ◉ Přehrávání zvuku (pozastavení při telefonování)
- ◉ Přijímání VoIP hovorů (Skype)
- ◉ Automatické stahování (novinky, podcasty)

STAVY APLIKACE - UDÁLOSTI

- ◉ ApplicationWillResignActive
 - Aplikace přestane být na popředí
 - Např. zastavit činnost při hovoru
- ◉ ApplicationDidEnterBackground
 - Aplikace přejde do pozadí
- ◉ ApplicationWillEnterForeground
 - Uspaná aplikace jde do popředí
 - Obnovení spojení se servery, nastavení časovačů, načtení obrázků...
- ◉ ApplicationDidBecomeActive
 - Aplikace je aktivní

STAVY APLIKACE - UDÁLOSTI

◉ ApplicationWillTerminate

- Před ukončením aplikace
- Z multitaskingu, vypínání přístroje
- Uložení dat, pokud již nebylo dříve

◉ ApplicationDidReceiveMemoryWarning

- Výzva k uvolnění prostředků

RYCHLÉ PŘEPÍNÁNÍ MEZI APLIKACEMI



Dvojklik tlačítko HOME

Přepnout na aplikaci, ukončit ji

Zdroj obrázku:

<https://support.apple.com/cs-cz/HT4211>

AKTUALIZACE APLIKACÍ NA POZADÍ (POKUD ZAPNUTO)

- ◉ Šetření baterie

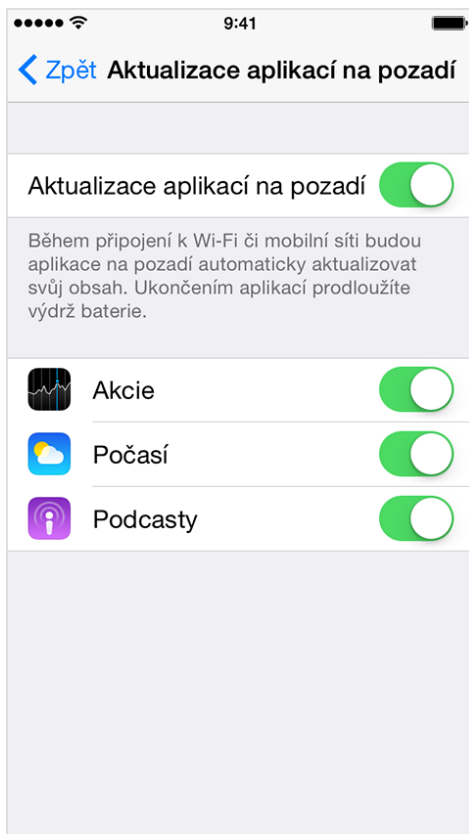
- V určitých časech - připojení k WiFi, připojení k napájení, aktivní používání

- ◉ „IOS se učí vzorce chování založené na vašem používání zařízení a snaží se předpovídat, kdy by se aplikace na pozadí měla obnovit.“

- ◉ „Taky se učí, kdy je zařízení obvykle neaktivní, jako třeba v noci, aby se aplikace aktualizovaly, když zařízení nepoužíváte“

- ◉ „Aplikace také mohou plánovat aktualizace na pozadí podle vaší polohy.“

AKTUALIZACE APLIKACÍ NA POZADÍ



Zdroj citátů na předchozím slidu a obrázku:

<https://support.apple.com/cs-cz/HT4211>

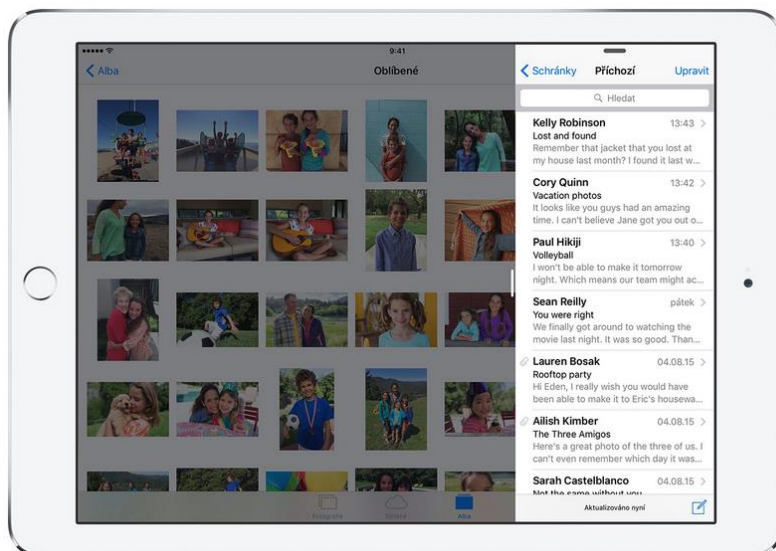
MULTITASKING

Od IOS 4	
Backgroud audio	Přehrává multimedia
Voice over IP	Suspendovaná, když není hovor
Background location	Notifikace o změnách polohy
Push notification	Notifikace přes síť
Local Notification	V předdefinovaný čas
Task completion	Požádá systém o čas k dokončení tasku
Fast app switching	Nevykonává kód, lze ji odstranit

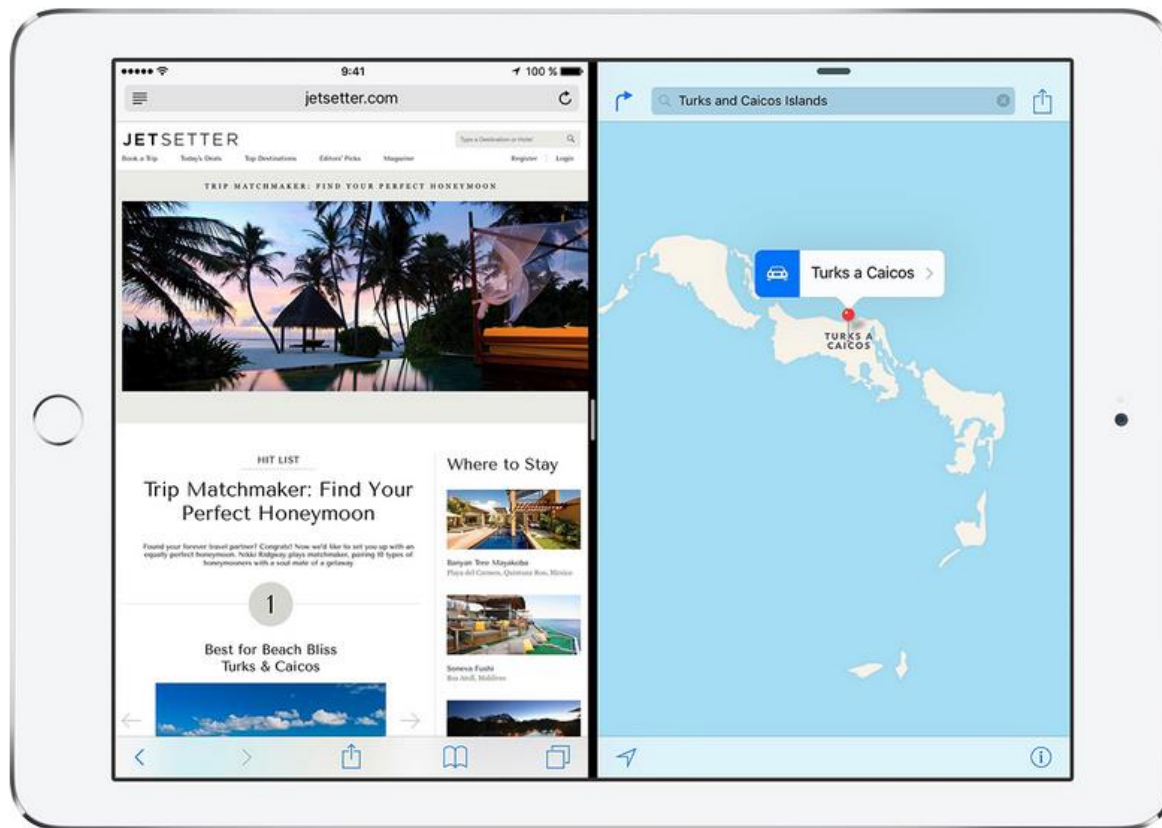
Od IOS 5 navíc	
Newsstand	Stahování obsahu na pozadí
External Accessory	Sdílí data v pravidelných intervalech
Bluetooth Accessory	Sdílí data

IPAD - SLIDE OVER

- Práce s druhou aplikací bez opuštění té používané
- Na pravém okraji obrazovky prstem shora dolů

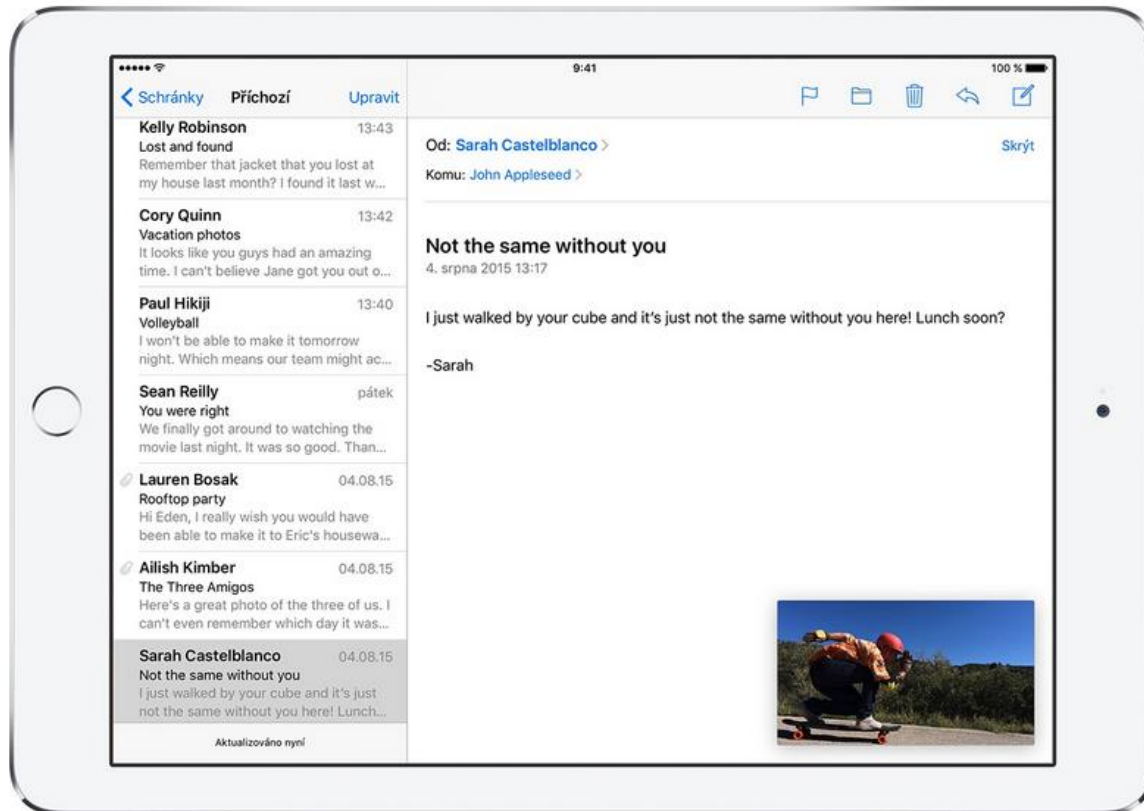


IPAD - SPLIT VIEW



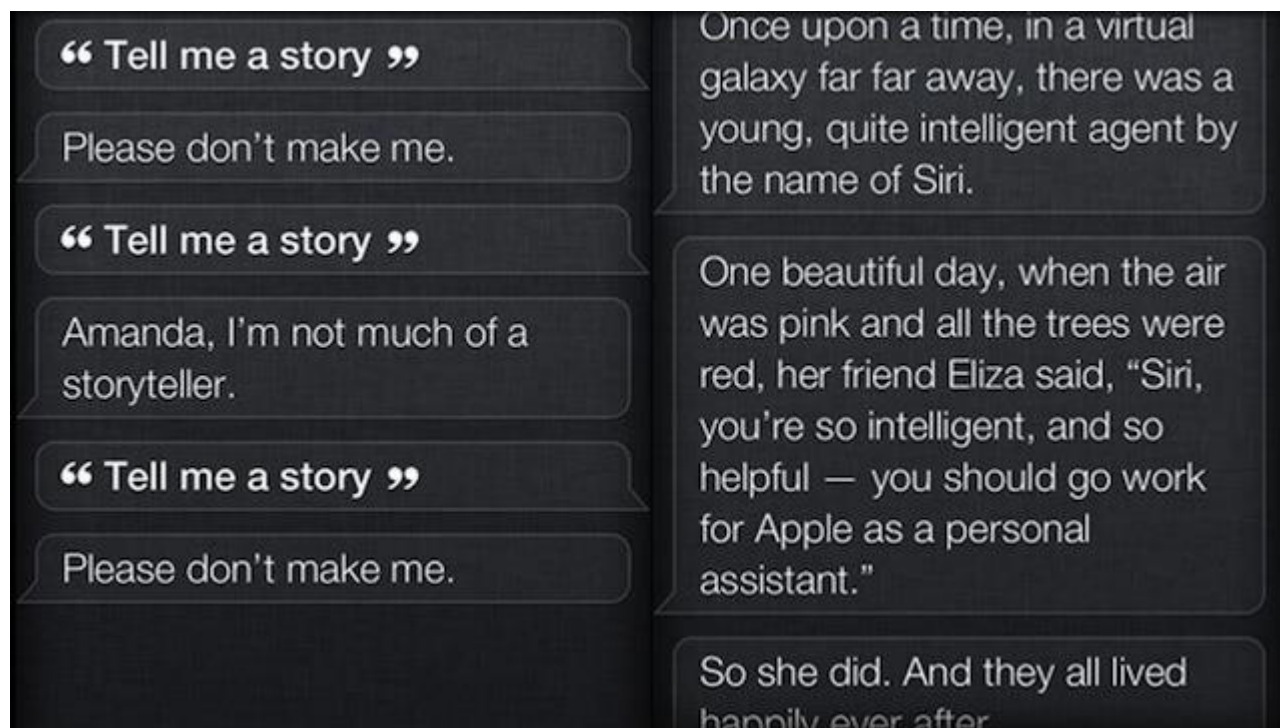
IPAD - OBRAZ V OBRAZE

Přehrávání videa pokračuje v malém okně, můžete je přesunout



SIRI

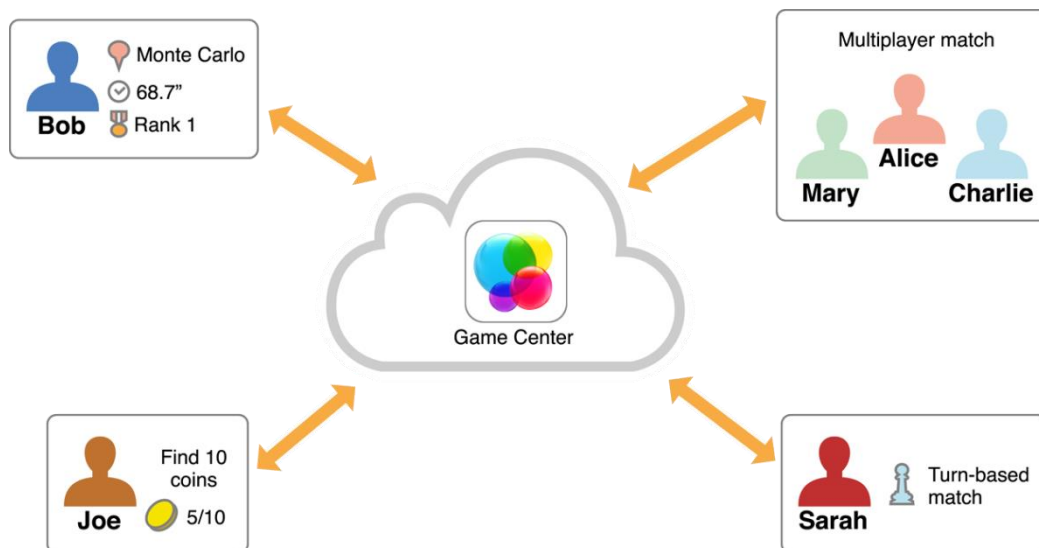
- ◉ hlasový asistent(ka)
- ◉ vyžaduju síťové připojení
- ◉ iPhone 4S a výše, iPade 3th generace a výše



Zdroj: <http://pixshark.com/siri-jokes.htm>

GAME CENTER

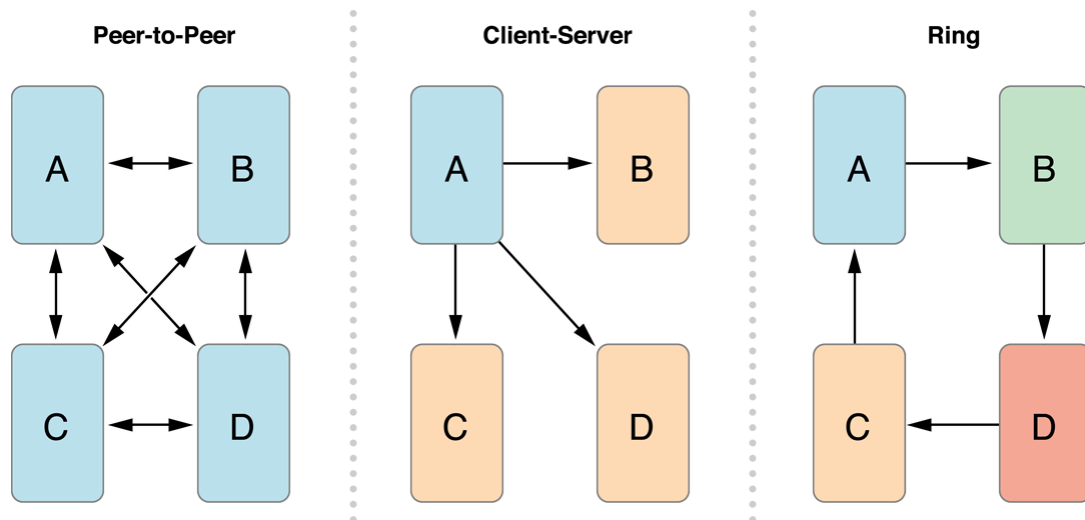
- ◉ online multiplayer síť
- ◉ nejlepší skóre hráče /globální (Bob)
- ◉ nalezení spoluhráče, online hraní (M,A,C)
- ◉ Turn based hraní - notifikace (Sarah)



TOPOLOGIE HRANÍ

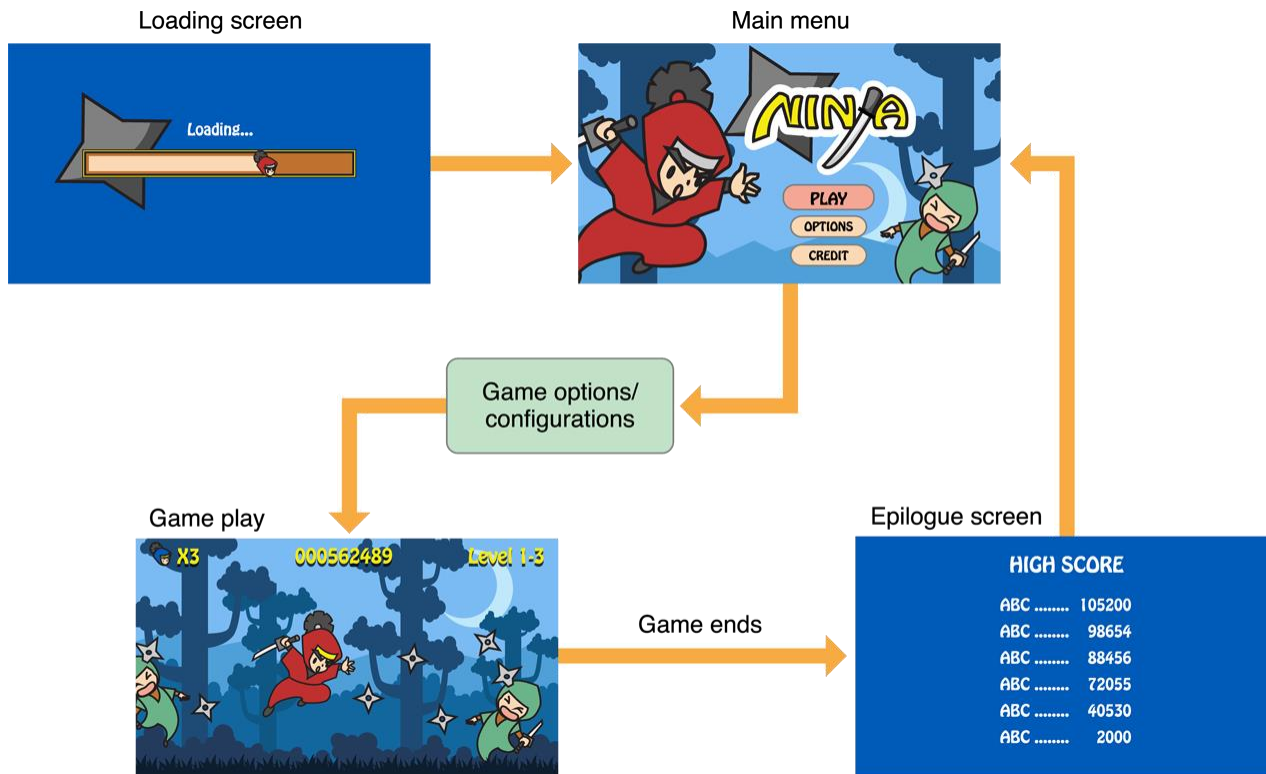
Podrobný popis pro programátora:

https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/GameKit_Guide/Matchmaking/Matchmaking.html#//apple_ref/doc/uid/TP40008304-CH9-SW1

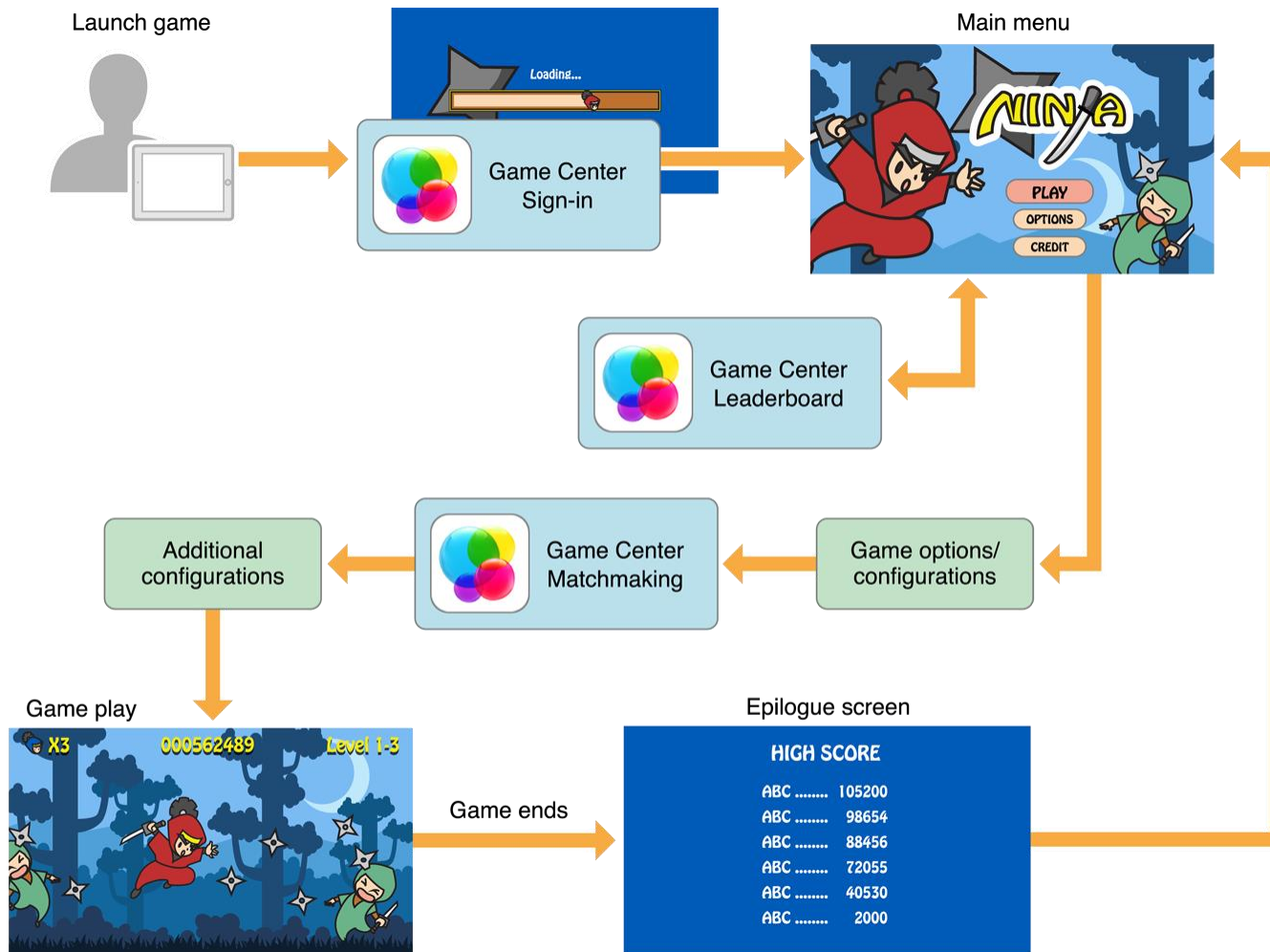


Zdroj obrázku: viz odkaz nad ním

OBRAZOVKY HRY

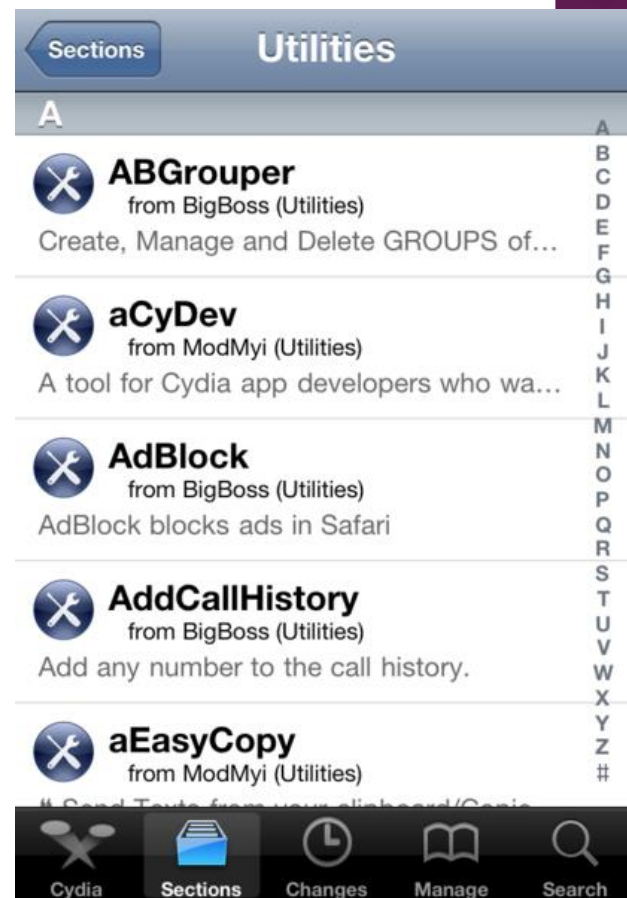


MULTIPLAYER HRA



JAILBREAKING

- přístup k filesystému
- modifikace SpringBoardu (domovská obrazovka)
- Cydia
 - instalace aplikací mimo AppStore
 - decentralizovaný systém úložišť (repositáře)
 - vlastní obchod Cydia store



zdroj obrázku: <http://cs.wikipedia.org/wiki/Cydia>

SDK

- ◉ Xcode pro Intel based Mac zdarma
- ◉ lze testovat aplikace na simulátoru
- ◉ roční poplatek 99\$ pro vývojáře
- ◉ nebo 299\$ enterprise (store pro firmu)
- ◉ univerzitní program
- ◉ testování na reálném zařízení
- ◉ publikace v AppStore
 - zdarma
 - od minimální ceny stanovené Apple

CERTIFIKÁTY

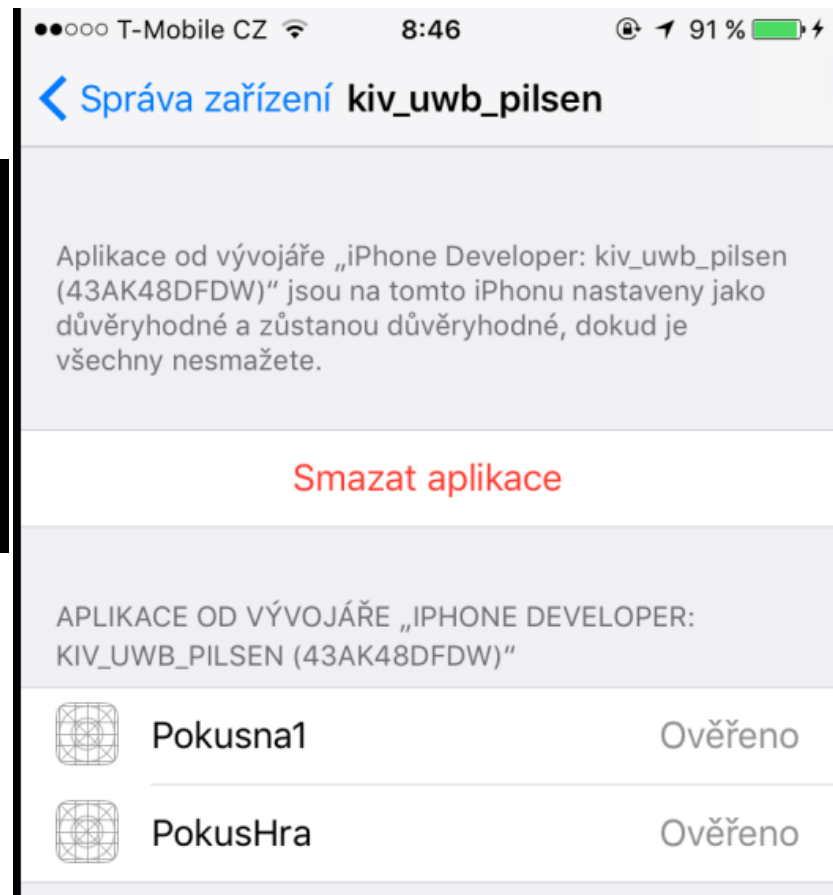
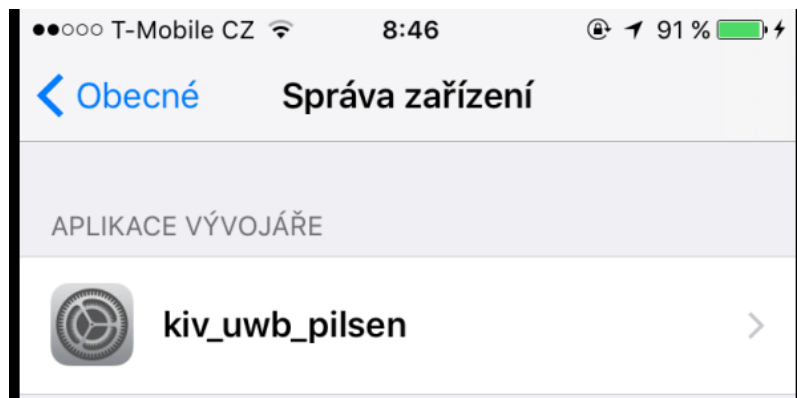
- uživatelský certifikát (development)



CERTIFIKÁT

- ◉ identifikátor **aplikace (AppID)**
- ◉ ID zařízení (UDID), získané např. z iTunes
- ◉ provisioning profile
 - jméno
 - typ (development)
 - App ID
 - Certifikáty
 - Devices
- ◉ iPhone: Nastavení - Obecné - Správa zařízení
 - Vývojář - Aplikace

VÁMI VYVÍJENÉ APLIKACE NA ZAŘÍZENÍ (Z XCODE -> MOBIL)



JAK PROGRAMOVAT?

- ◉ Macové zařízení
 - MacBook, MacBook Air, iMac, Mac mini, ...
- ◉ Nainstalovaný Xcode
 - Vývojové prostředí zdarma
 - Lze pouštět aplikace v emulátoru
- ◉ Vývojářský účet
 - Instalace aplikací do zařízení vývojáře
 - Publikování do obchodu
- ◉ Programovací jazyk
 - Objective-C, Swift
 - Multiplatformní vývoj - Xamarin (C#)

JAZYK OBJECTIVE-C, COCOA TOUCH

- ◉ objektově orientovaný
- ◉ původně hlavní jazyk OS NeXTSTEP
- ◉ rozšíření standardního ANSI C
- ◉ aplikační vývojové prostředí Cocoa Touch
 - Objective-C runtime
 - Foundation framework
 - ◉ root class NSObject - definuje základní chování objektu
 - UIKit framework
 - ◉ obsluha událostí, kreslení, zpracování textu..
 - ◉ komponenty UI

```
#import <Foundation/Foundation.h>

@interface MyClass : NSObject
{
    // oblast deklarace instančních proměnných
    @public NSInteger myInt;
}

// oblast deklarace metod
- (void)myMethodWithParameter:(id)parameter;

@end

@implementation MyClass

// oblast implementace metod
- (void)myMethodWithParameter:(id)parameter
{
    // implementace metody
}

@end
```

často oddělené soubory
header file: .h a implementace: .m

UKÁZKA

```
#import <UIKit/UIKit.h>

int main(int argc, char *argv[])
{
    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc]
    init];
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    [pool release];
    return retVal;
}
```

METODA TŘÍDY, INSTANCE

```
@interface MyClass : NSObject
```

```
+ (id) classMethod;
```

```
- (id) instanceMethod;
```

```
@end
```

```
@implementation MyClass
```

```
...
```

```
@end
```

ZASÍLÁNÍ ZPRÁV OBJEKTU

[příjemce zpráva]

[[příjemce zpráva1] zpráva2]

[myObject setMyInt:1];

myObject.myInt = 1;

[myArray count]; // počet objektů v poli

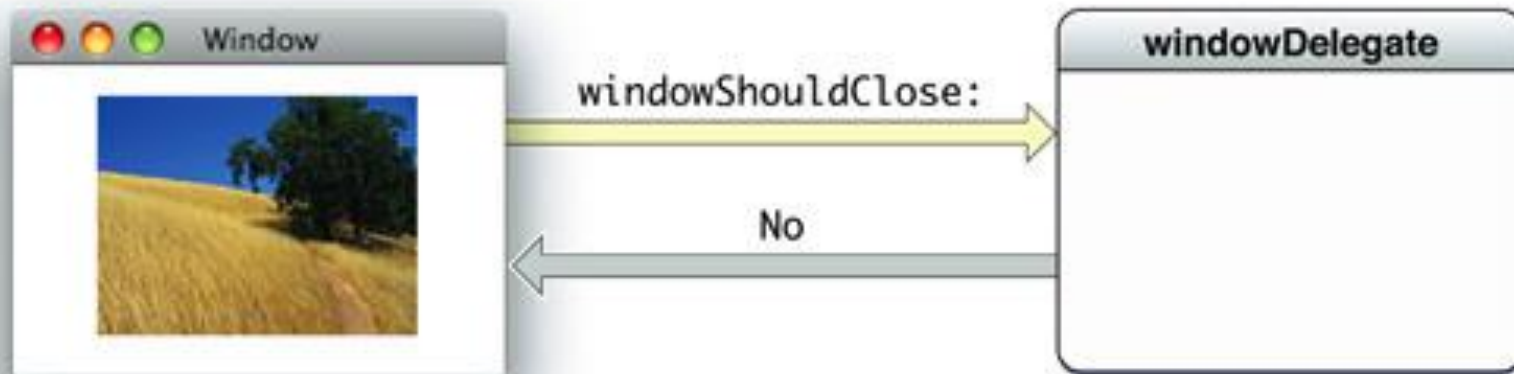
[[NSMutableArray alloc] initWithCapacity:1];

PROTOKOL

@interface Trida : Rodic <Protokoly>

Třída může přijímat více protokolů (viz Java a interface)

DELEGACE



NSWindows třída z **AppKit frameworku**

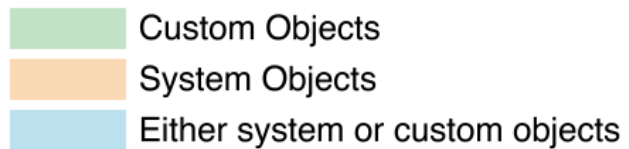
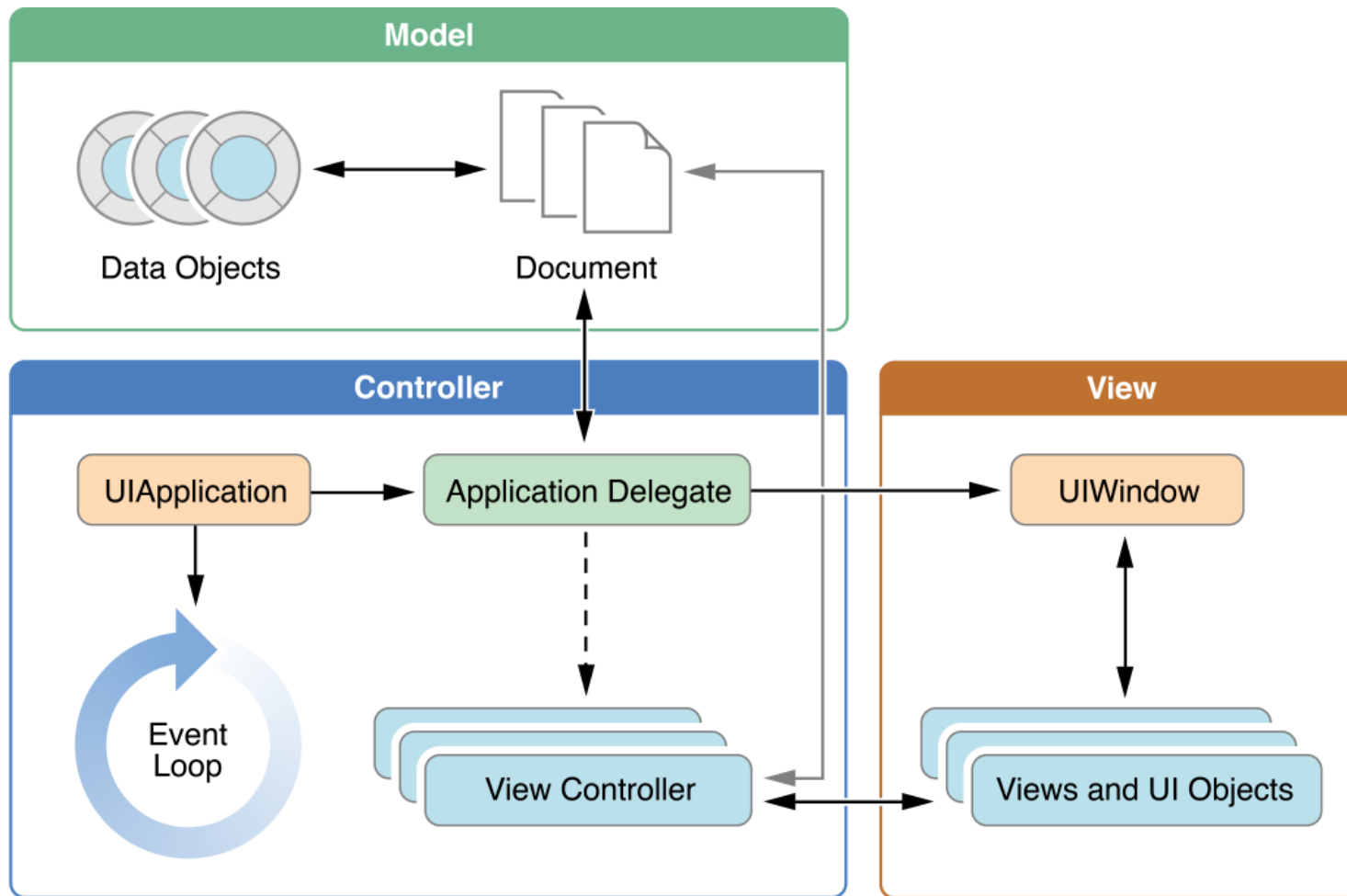
NSWindow deklaruje **protokol**, mj. metoda **windowShouldClose:**

uživatel klikne na uzavírací tlačítko

objekt pošle **windowShouldClose:** **delegátovi** a zeptá se jej na potvrzení akce

delegát vrátí *true* / *false*, kontroluje chování objektu okna

obrázek - viz dokumentace Apple

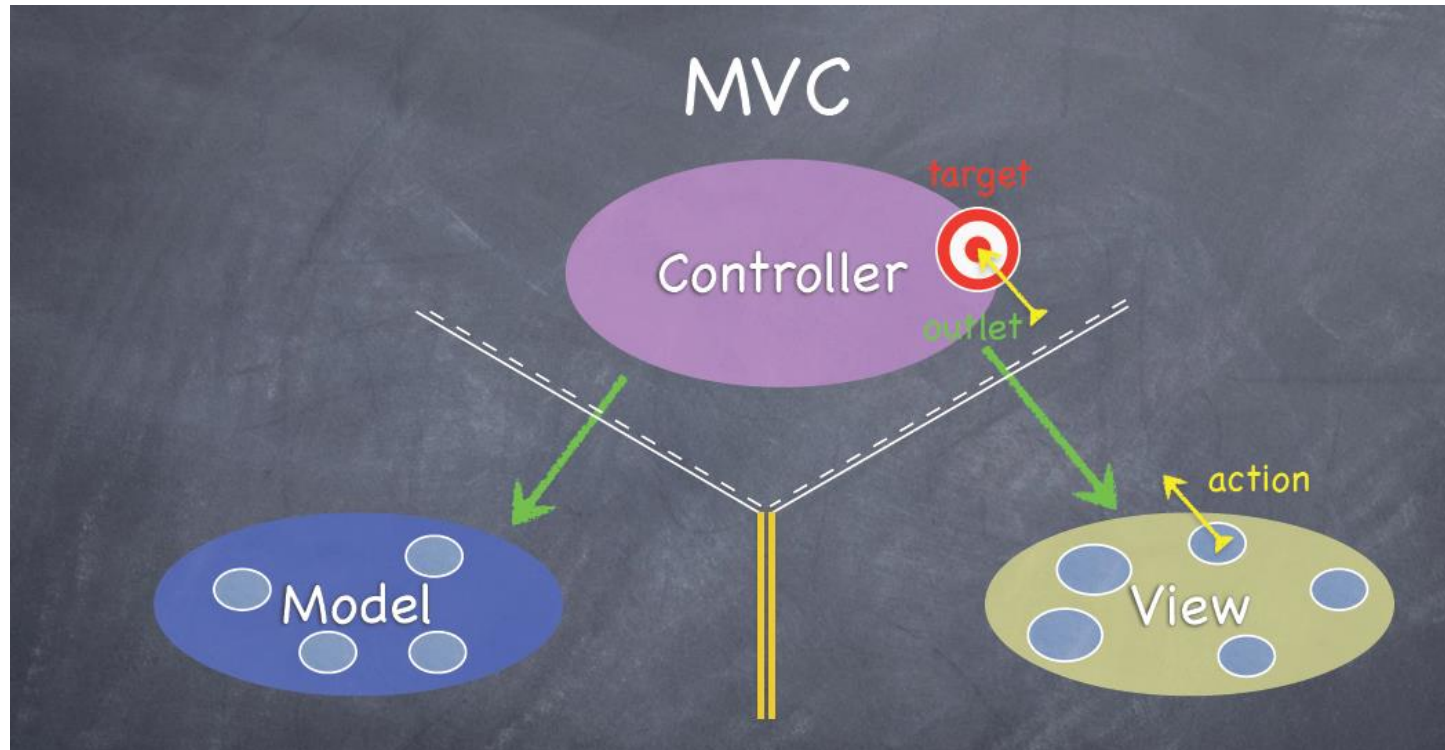


MODEL - VIEW - CONTROLLER

- ◉ Programové objekty do těchto tří skupin
- ◉ Model
 - co je aplikace
 - ne jak je zobrazena
- ◉ Controller
 - jak je model prezentován uživateli (UI logika)
- ◉ View
 - zobrazení

MVC

Action - stisk tlačítka, obsluha
Outlet - potřebuji změnit
hodnotu v textovém poli



Zdroj - Stanford university

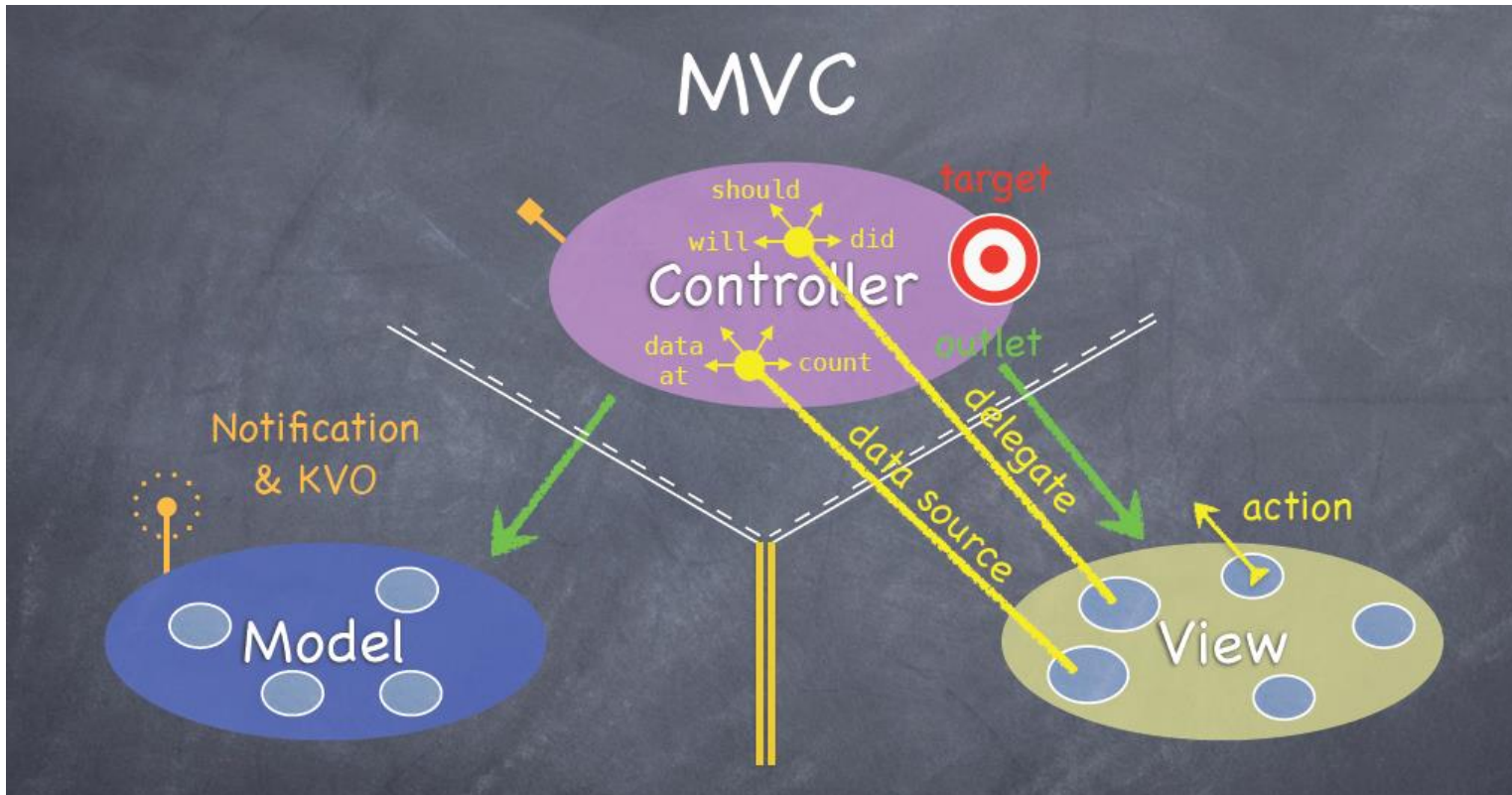
Controller může komunikovat s **Modelem** i s **View**

View pošle **akci**, směřovanou targetu v **Controlleru**

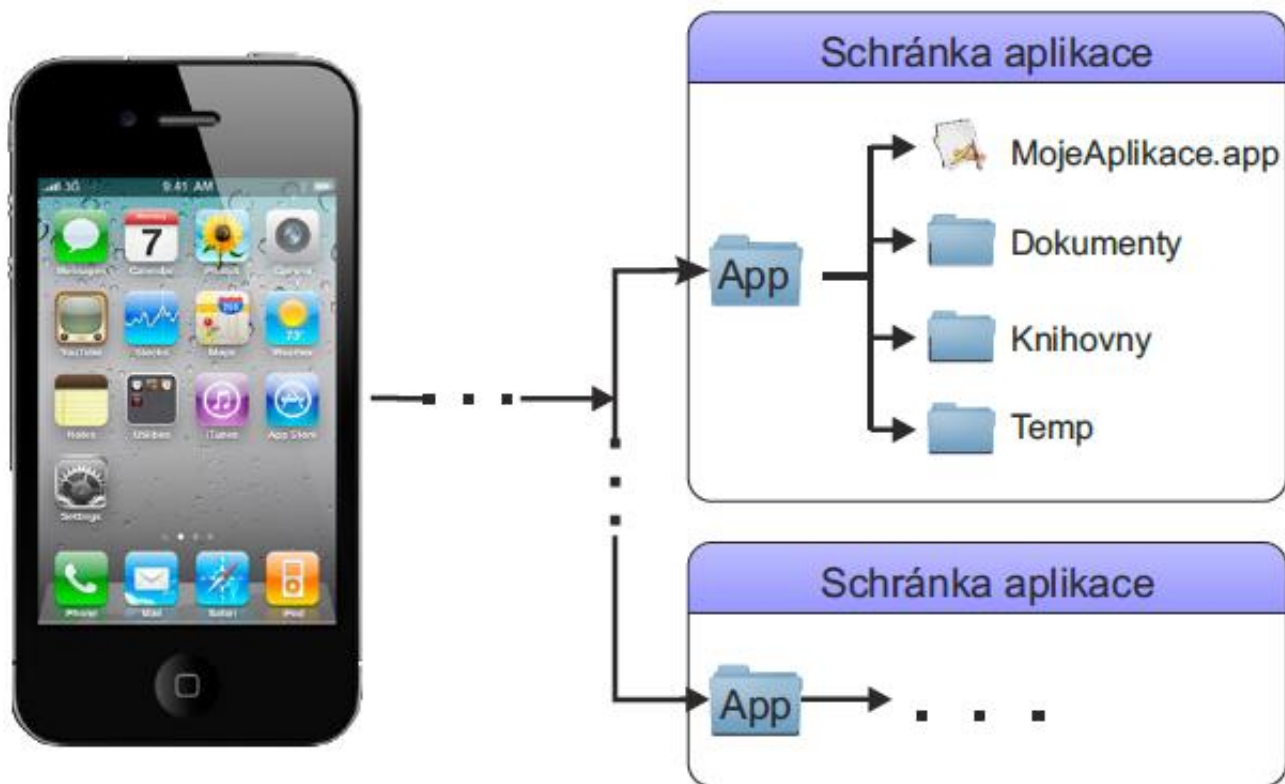
View se potřebuje synchronizovat s **Controllerem** => Controller bude **delegátem** pro **View**

Delegát je představován protokolem

MVC



IZOLOVANOST APLIKACÍ



PRÁCE SE SOUBORY - STREAMY

- ◉ Sekvenční přístup, asynchronní
- ◉ **NSOutputStream** - zápis na disk
- ◉ **NSInputStream** - čtení z disku

- ◉ Vstupní stream probudí smyčku běhu a informuje delegáta, že jsou dostupná data ke čtení
- ◉ Běhová smyčka může být vzbuzena několikrát

PRÁCE SE SOUBORY - NSFILEHANDLE

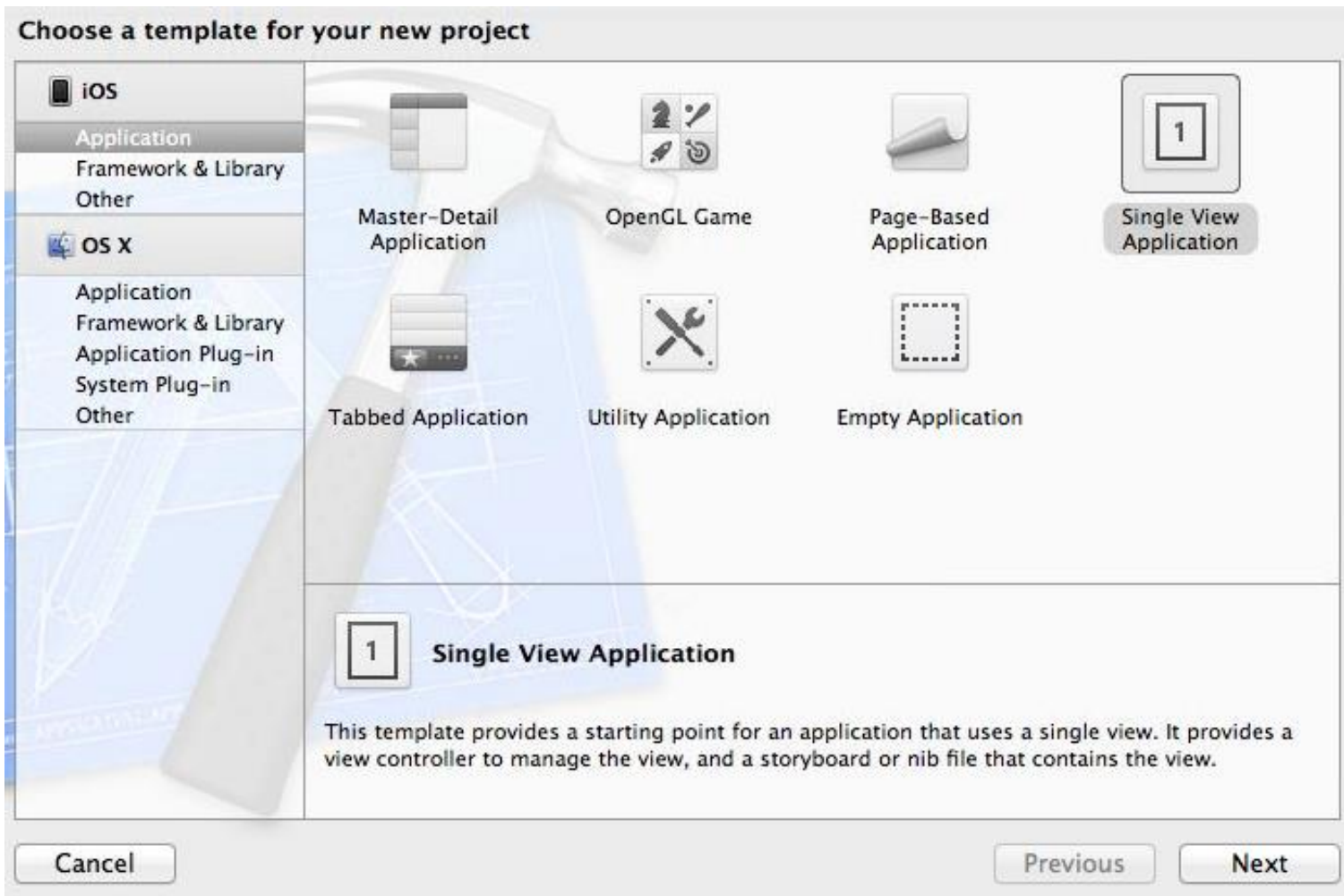
- ◉ Otevření souboru
- ◉ Volání čtení - zápisů
- ◉ Uzavření souboru

- ◉ Synchronní přístup

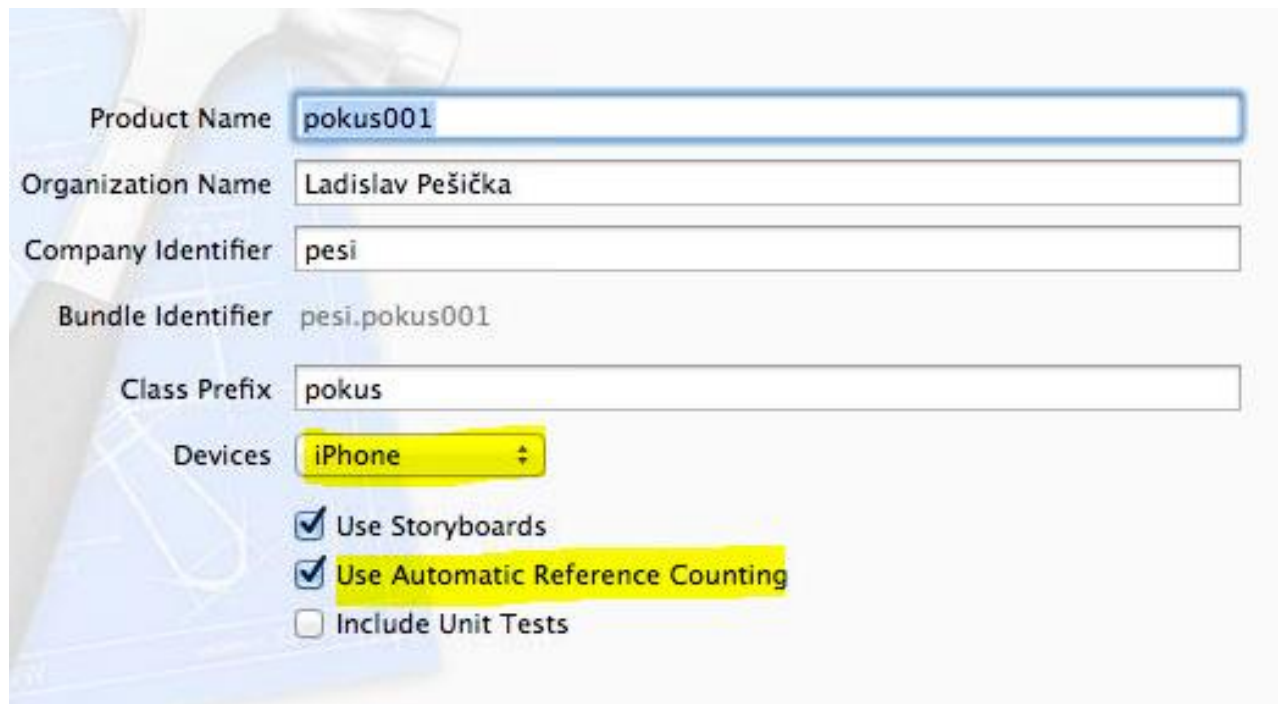
XCODE

- ◉ Xcode - File - New project - IOS
 - Master - detail application
 - OpenGL game
 - Page based application
 - Single View application
 - Tabbed application
 - Utility application
 - Empty application
- ◉ Xcode kromě iOS aplikace umožňuje i OS X vývoj (Cocoa application, command line tool)

XCODE



XCODE



Product Name pokus001

Organization Name Ladislav Pešíčka

Company Identifier pesi

Bundle Identifier pesi.pokus001


Class Prefix pokus

Devices iPhone


- ☒ Use Storyboards
- ☒ Use Automatic Reference Counting
- ☐ Include Unit Tests





Pro zařízení: iPhone, iPad, univerzální aplikace

PROJECT

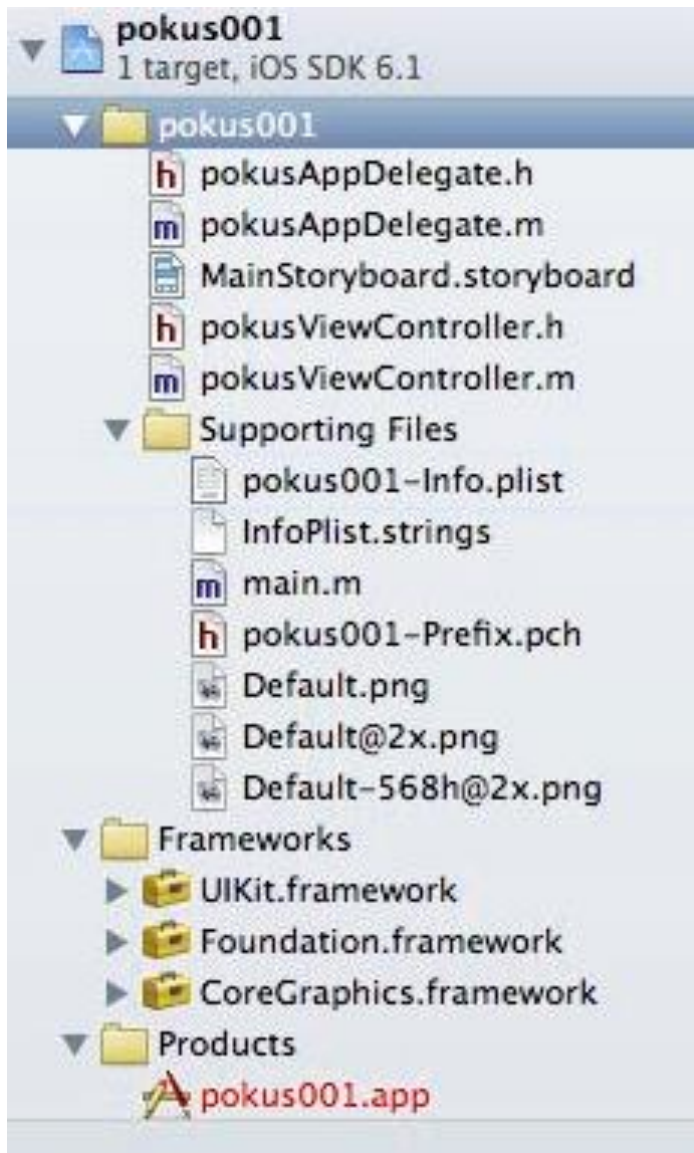
 pokus001

TARGETS

 pokus001

Summary	Info	Build Settings	Build Phases	Build Rules
iOS Application Target				
Bundle Identifier <input type="text" value="pesi.pokus001"/>				
Version <input type="text" value="1.0"/>		Build <input type="text" value="1.0"/>		
Devices <input type="text" value="iPhone"/>				
Deployment Target <input type="text" value="6.1"/>				
▼ iPhone / iPod Deployment Info				
Main Storyboard <input type="text" value="MainStoryboard"/>				
Main Interface <input type="text"/>				
Supported Interface Orientations				
<div> Portrait</div> <div> Upside Down</div> <div> Landscape Left</div> <div> Landscape Right</div>				
Status Bar				
Style <input type="text" value="Default"/>				
Visibility <input type="checkbox"/> Hide during application launch				
Tinting <input type="text" value="Disabled"/>				
Tint Color <input type="color" value="blue"/>				

SOUBORY PROJEKTU



AppDelegate (.h, .m)
MainStoryboard (**View**)
View**Controller** (.h, .m)

Pokus001.app

MVC
Model - vytvorime zvlast
View - StoryBoard
Controller - viz soubory

POKUSAPPDELEGATE.M

```
#import "pokusAppDelegate.h"

@implementation pokusAppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:
    (NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    // Sent when the application is about to move from active to inactive state. This
    // can occur for certain types of temporary interruptions (such as an incoming
    // phone call or SMS message) or when the user quits the application and it
    // begins the transition to the background state.
    // Use this method to pause ongoing tasks, disable timers, and throttle down
    // OpenGL ES frame rates. Games should use this method to pause the game.
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    // Use this method to release shared resources, save user data, invalidate
    // timers, and store enough application state information to restore your
    // application to its current state in case it is terminated later.
    // If your application supports background execution, this method is called
    // instead of applicationWillTerminate: when the user quits.
}
```

POKRAČOVÁNÍ

```
- (void)applicationWillEnterForeground:(UIApplication *)application
{
    // Called as part of the transition from the background to the inactive state;
    // here you can undo many of the changes made on entering the background.
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    // Restart any tasks that were paused (or not yet started) while the application
    // was inactive. If the application was previously in the background, optionally
    // refresh the user interface.
}

- (void)applicationWillTerminate:(UIApplication *)application
{
    // Called when the application is about to terminate. Save data if appropriate.
    // See also applicationDidEnterBackground:.
}

@end
```

POKUS VIEW CONTROLLER

```
#import "pokusViewController.h"

@interface pokusViewController ()

@end

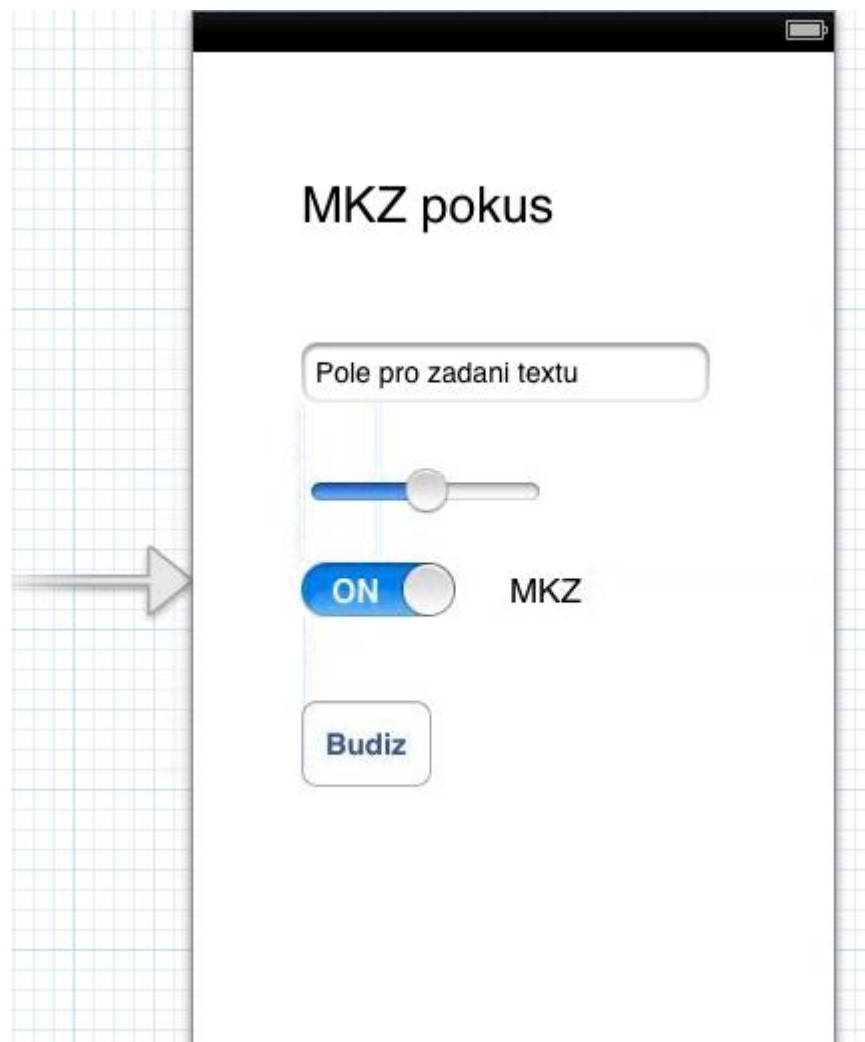
@implementation pokusViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

@end
```

STORYBOARD



Spuštění aplikace v emulátoru



TAB BAR CONTROLLER

- ◉ Organizace aplikace do několika oddělených sekcí
- ◉ Tab bar - spodní lišta aplikace
- ◉ Custom view - nad lištou (root view controller)
- ◉ Jako root view controller může být navigační controller

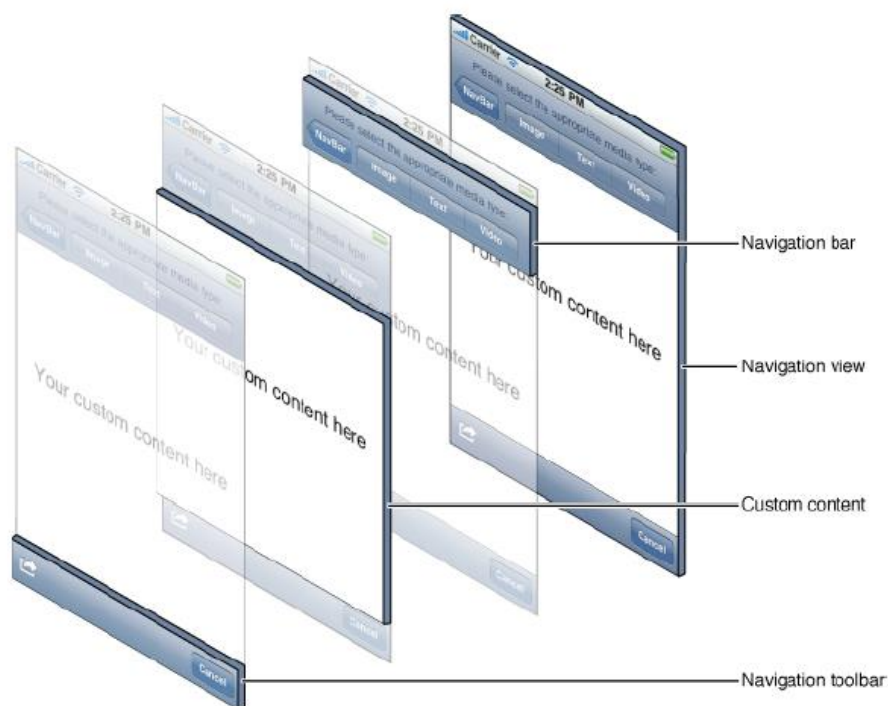
TAB BAR CONTROLLER



Obrázek 6.6: Tab bar controller view v aplikaci Hodiny.

NAVIGATION CONTROLLER

- Hierarchická data
- Navigation toolbar, navigation bar, custom view (dohromady navigation view)



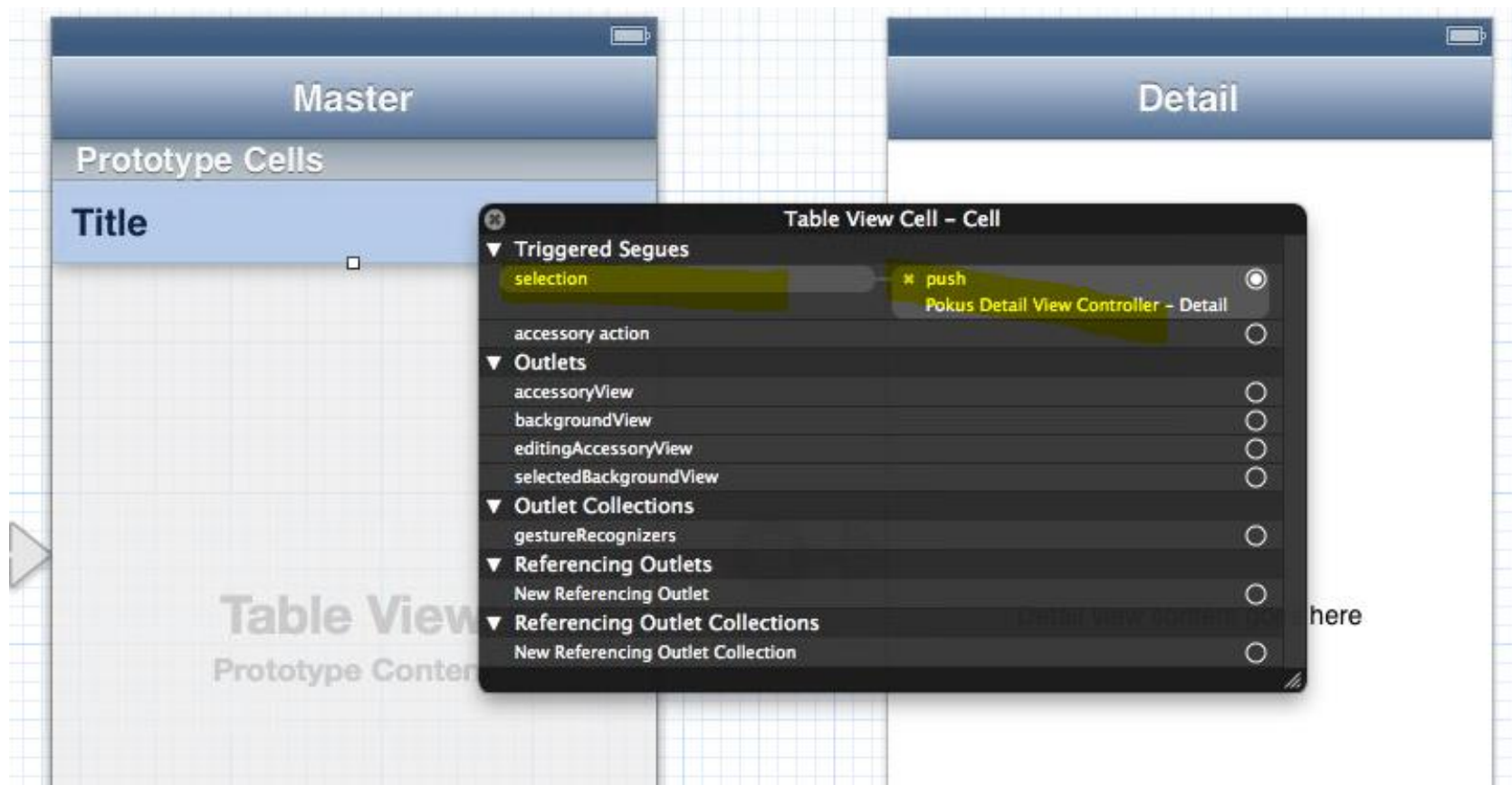
Obrázek 6.8: Struktura navigačního rozhraní.

MASTER - DETAIL APLIKACE

- ◉ Vytvoříme Master - Detail application
- ◉ UITableView (často používaný, např. v Settings)
- ◉ Vygeneruje kostru aplikace
 - Přidat událost
 - Podívat se na detailní zobrazení
- ◉ pokusAppDelegate
 - Žádná změna

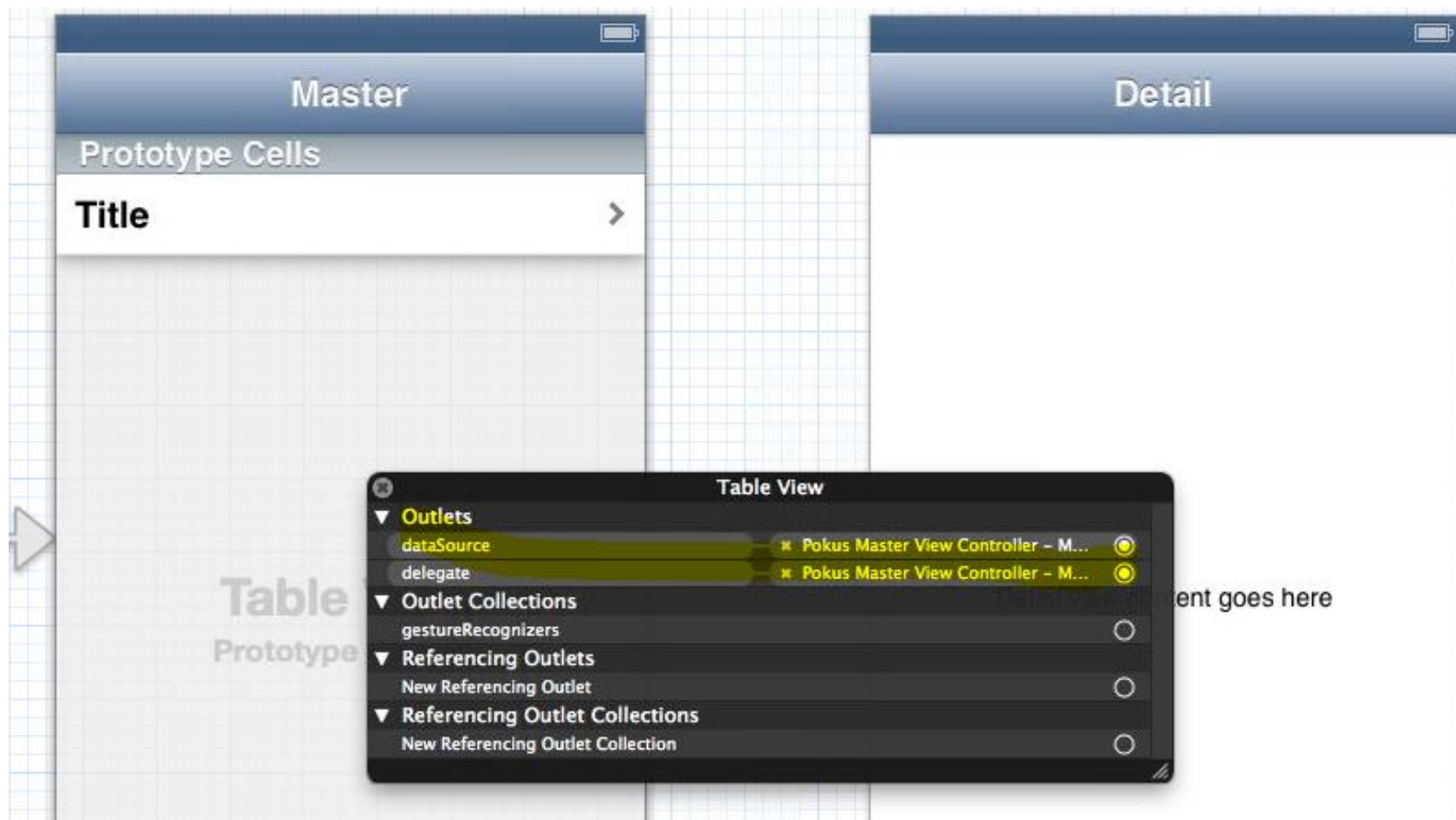
STORYBOARD

PRAVÁ MYŠ NA POLOŽCE TITLE



STORYBOARD

PRAVÁ MYŠ POD POLOŽKOU TITLE



OUTLET

Držet Ctrl, soubor .h

The screenshot shows the Xcode interface with a storyboard on the left and a code editor on the right. In the storyboard, a yellow line connects an IBOutlet labeled 'MKZ pokus' to a UI element. The code editor shows the implementation of this outlet in a header file.

```
// pokusViewController.h
// pokus003
// Created by Ladislav Pešička on 08.05.13.
// Copyright (c) 2013 Ladislav Pešička. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface pokusViewController :
    UIViewController
@property (weak, nonatomic) IBOutlet UILabel *
    display;

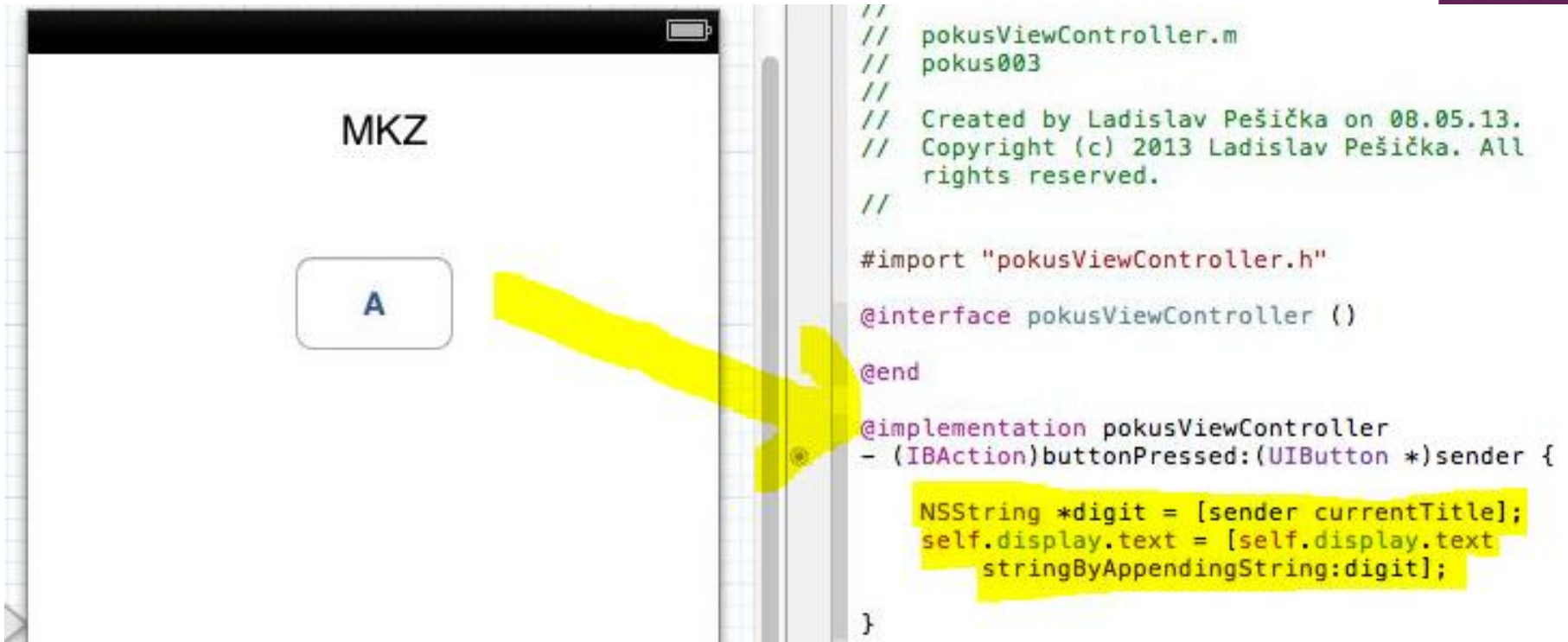
@end
```

Below the code editor, a panel titled 'Label - MKZ pokus' displays the 'Referencing Outlets' section. It shows a table with one entry: 'display' from 'Pokus View Controller', which is selected with a radio button.

Outlet Name	Referencing Outlet	Selected
display	Pokus View Controller	<input checked="" type="radio"/>

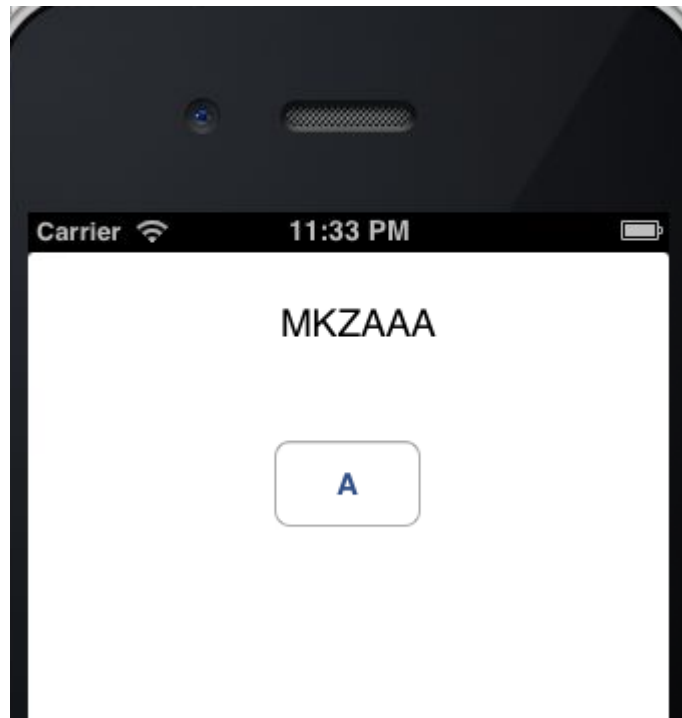
ACTION

Držet Ctrl, soubor .m



Po stisknutí tlačítka A dojde k přidání currentTitle, tedy A, k nápisu na který ukazuje display (výchozí MKZ)

3X STISKNUTO TLAČÍTKO A



TUTORIAL

- ◉ <http://www.raywenderlich.com/1797/how-to-create-a-simple-iphone-app-tutorial-part-1>

UKÁZKA Z EMULÁTORU



SWIFT

- ◉ Pro vývoj na iOS i Mac OS X
- ◉ Alternativa k Objective-C
- ◉ Lze s Objective-C mixovat ve stejném programu
- ◉ Nevyužívá pointery (nenutí k tomu programátora), ale je možné je využít
- ◉ Chris Lattner (2010 Swift)
- ◉ Open source (swift.org)

SWIFT

- ◉ Zdroj ukázek: wikipedia

```
NSString *str = @"hello,";  
str = [str stringByAppendingString:@" world"];
```

```
var str = "hello,"  
str += " world"
```

SWIFT

⦿ Konstanty

- `let constant = 1`

⦿ Proměnné

- `var prom1 = 1, prom2 = 2`
- `var double1, double2 : Double`
- `var prom: Int`
- `prom = 1`

UNICODE CHARACTERS IN CONSTANT & VARIABLE NAMES

```
let 🍀🍀🍀🍀 = 4  
var 🌸🌸🌸🌸🌸 = 5
```

SWIFT - ŘÍDÍCÍ KONSTRUKCE, PODMÍNKY

```
// old school c-style for loop
for var index = 0; index < 2; index++ { /* use index */ }

// new iterator-style for-in loop
for value in 0..<2 { /* use value */ }
```

```
let temperature = 40

var feverish: Bool
if temperature > 37 {
    feverish = true
} else {
    feverish = false
}
```

```
switch x { // break by default
case value1:
    /* ... */
case value2, value3:
    fallthrough // to not break
default:
    /* ... */
} // switch must be exhaustive
```


SWIFT - FUNKCE

```
1  func greet(person: String, day: String) -> String {  
2      return "Hello \(person), today is \(day)."  
3  }  
4  greet(person: "Bob", day: "Tuesday")
```

SWIFT - TŘÍDY

(DEVELOPER.APPLE.COM)

```
1  class Shape {  
2      var numberOfSides = 0  
3      func simpleDescription() -> String {  
4          return "A shape with \(numberOfSides) sides."  
5      }  
6  }
```

```
1  var shape = Shape()  
2  shape.numberOfSides = 7  
3  var shapeDescription = shape.simpleDescription()
```

SWIFT PLAYGROUND

- Okamžité otestování kódu bez kompilací velkých celků (hned vidíme efekt)



Zdroj
obrázku:
Internet

SWIFT PLAYGROUND

Swift Playgrounds

[View More by This Developer](#)

By Apple

Open iTunes to buy and download apps.



[View in iTunes](#)

Description

Swift Playgrounds is a revolutionary app for iPad that makes it fun to learn and experiment with code. You solve interactive puzzles in the guided “Learn to Code” lessons to master the basics of coding, while additional challenges let you explore code and create programs that are engaging and unique.

[Apple Web Site](#) › [Swift Playgrounds Support](#) › [Application License Agreement](#) ›

[...More](#)

What's New in Version 1.2

- Add new pages to your template-based playgrounds with a tap of the Edit button in the table of contents
- You can now use the MapKit framework and Swift 3.1 code in your playgrounds
- Learn to Code pages start much faster

KURZY PROGRAMOVÁNÍ PRO IOS

- ⦿ <http://www.stanford.edu/class/cs193p/cgi-bin/drupal/downloads-2011-fall>
- ⦿ <http://web.stanford.edu/class/cs193p/cgi-bin/drupal/downloads-2013-fall>