

Jak ugrýźć iOS?

wprowadzenie do platformy i języka Swift

Jakub Tudruj

Dlaczego iOS?





Jak zacząć?

Sprzęt

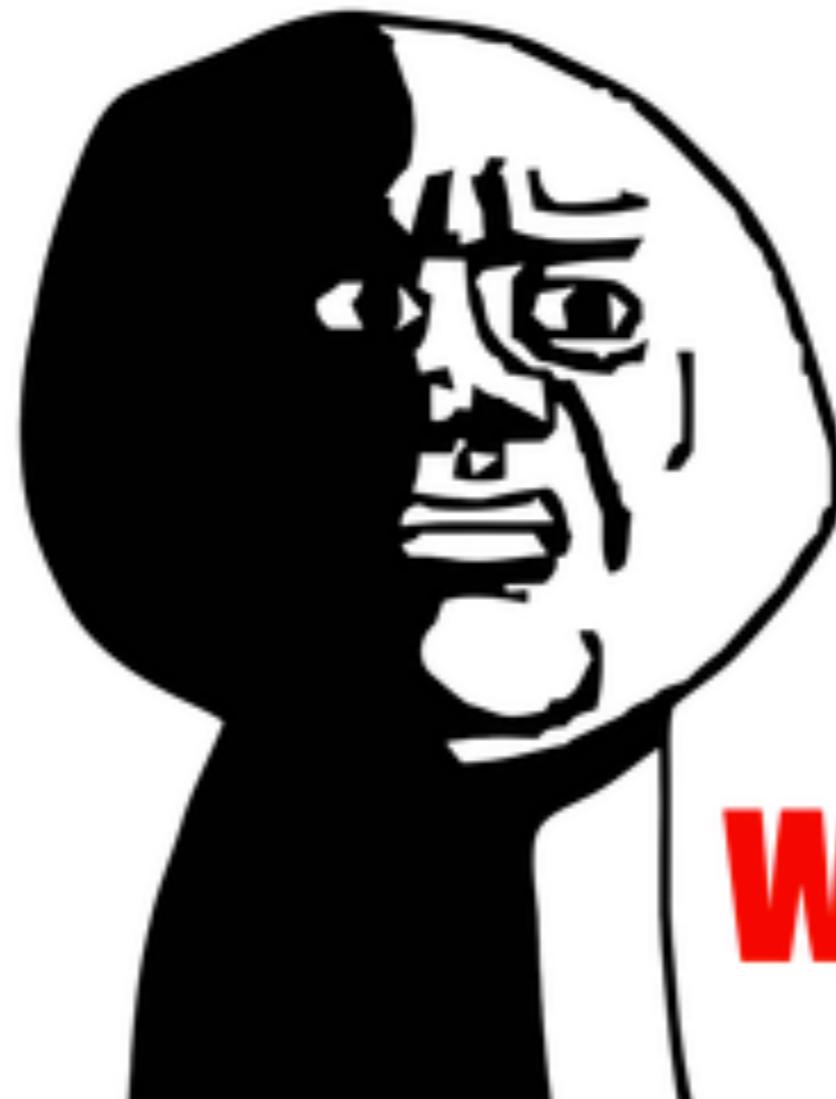


=

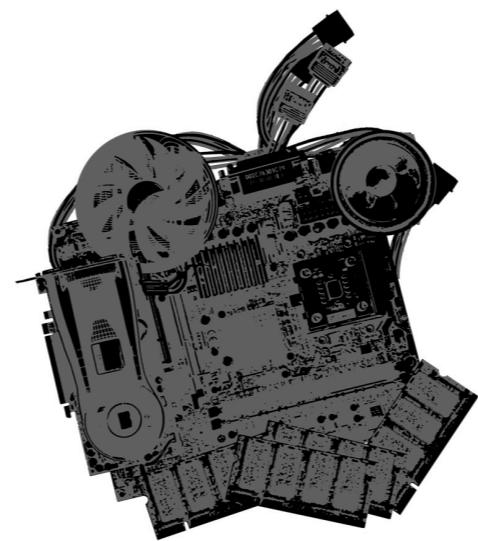


Sprzęt

OH GOD



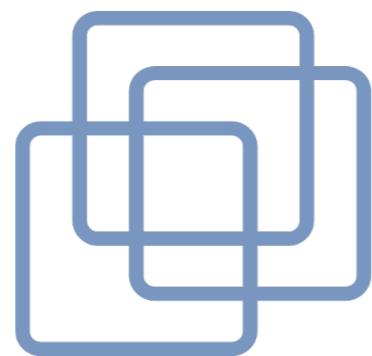
Alternatywa



by thü / www.ecliptic.ch

Hackintosh

Think *really* different



vmware®

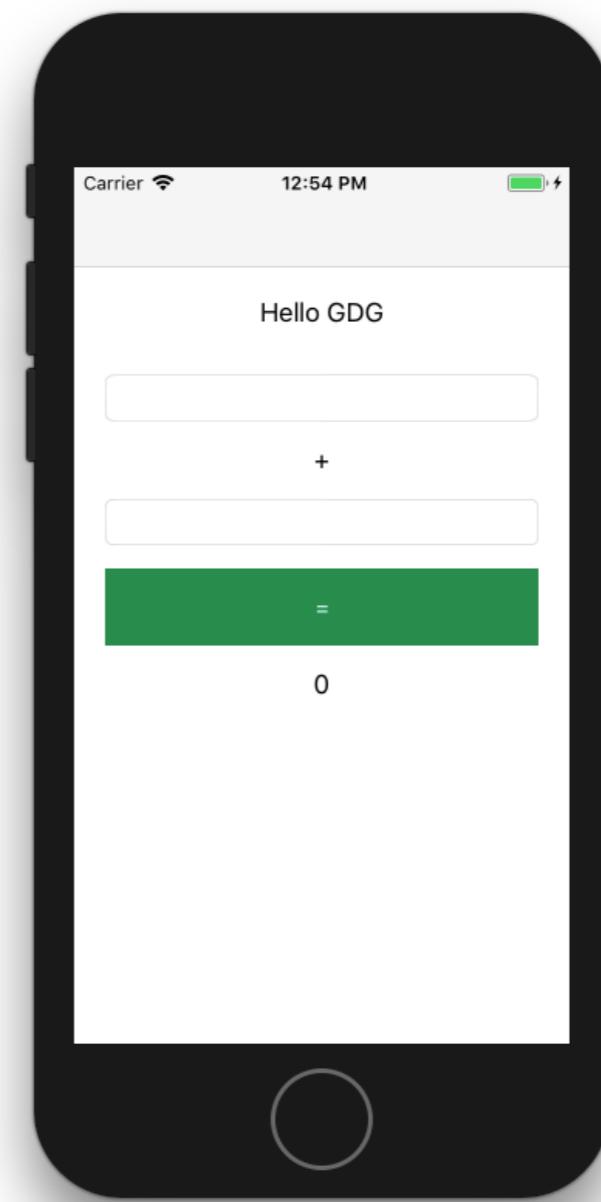
Windows? Linux?

- swiftforwindows.github.io
- <https://swift.org/download/#using-downloads>
- <https://wyszo.wordpress.com/2016/02/14/programming-in-swift-on-linux/>

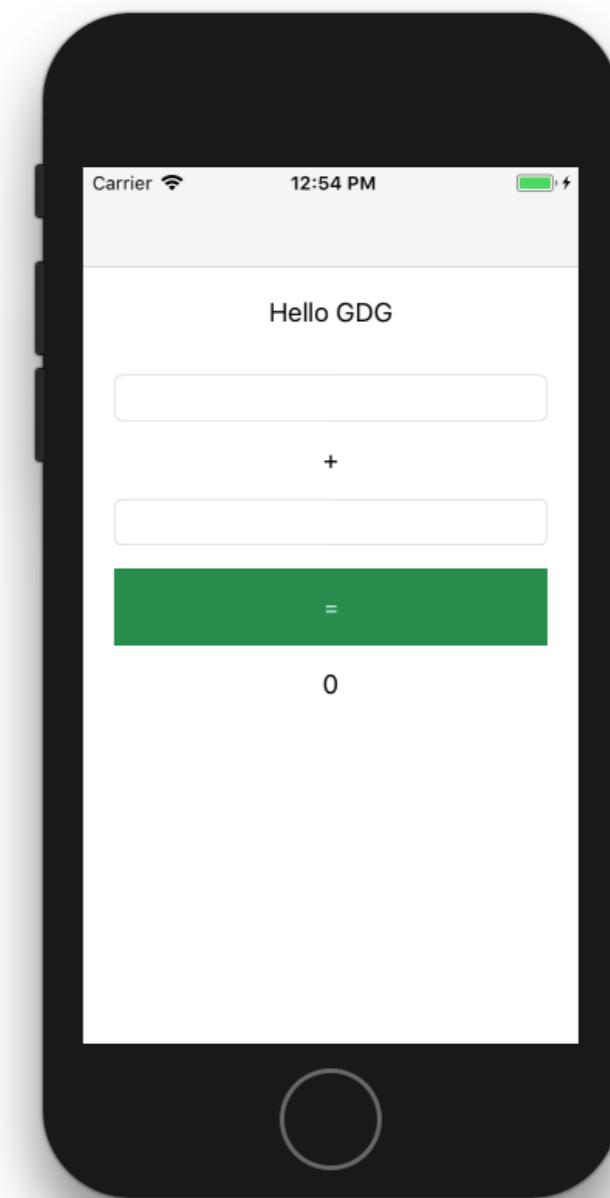
IDE - Xcode



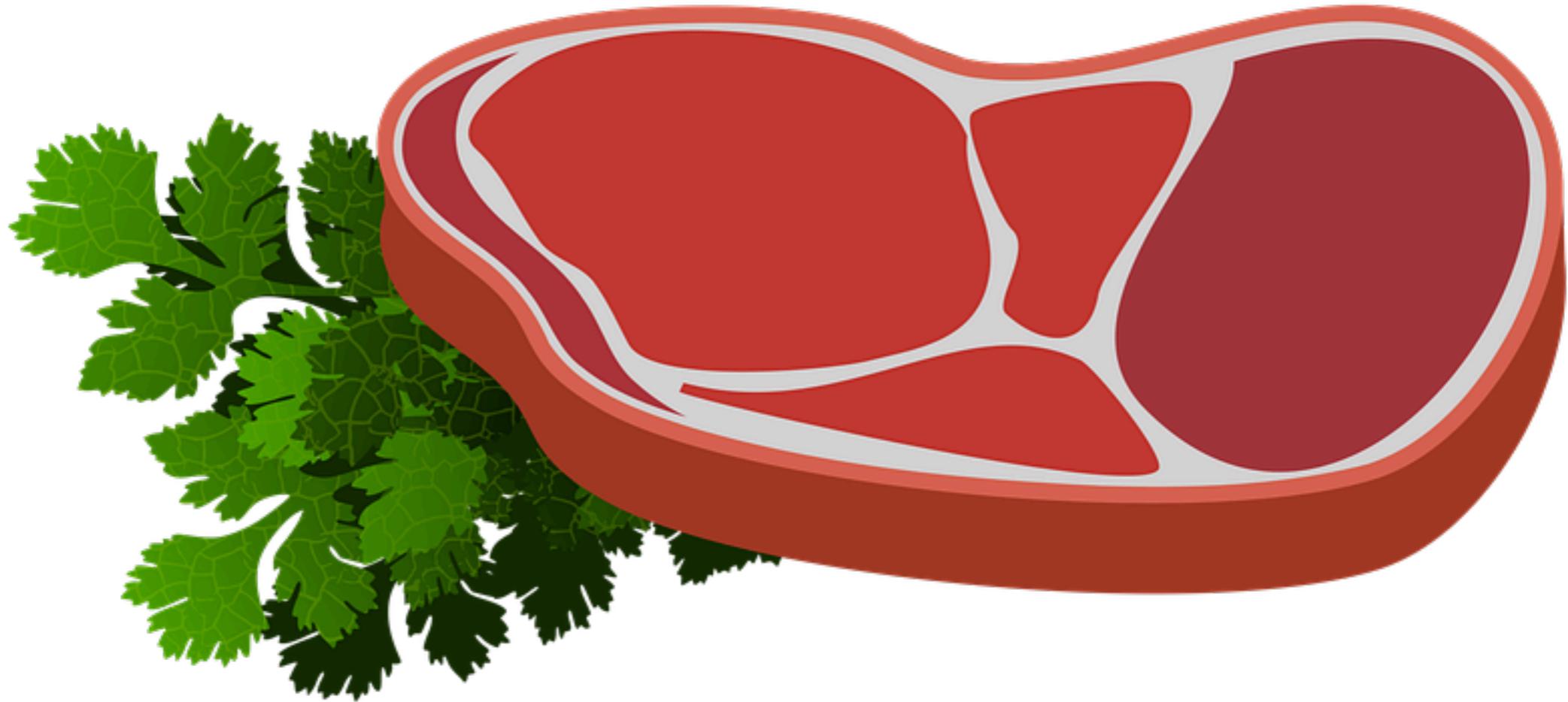
Jak uruchomić aplikację?



Jak uruchomić aplikację?

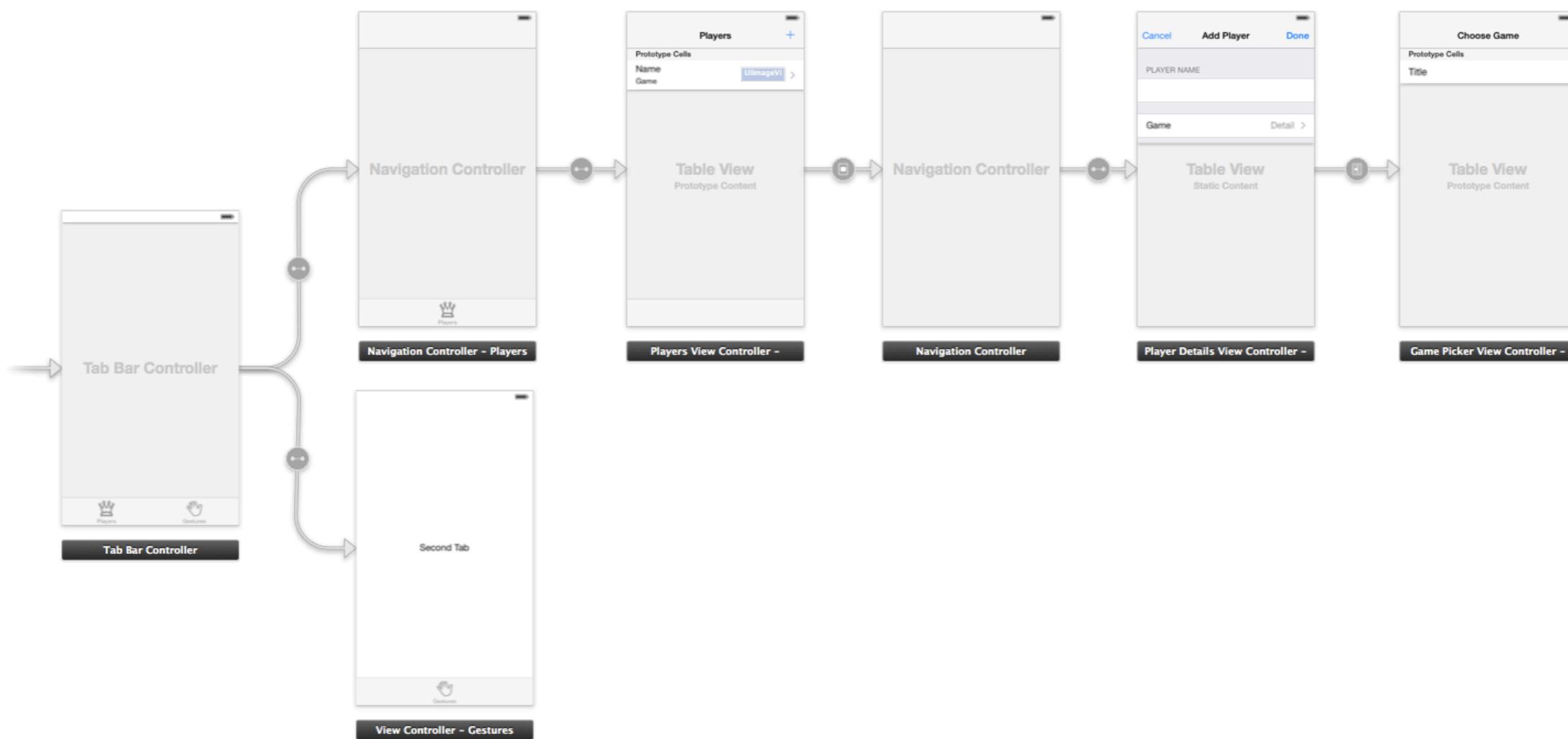


Mięsko



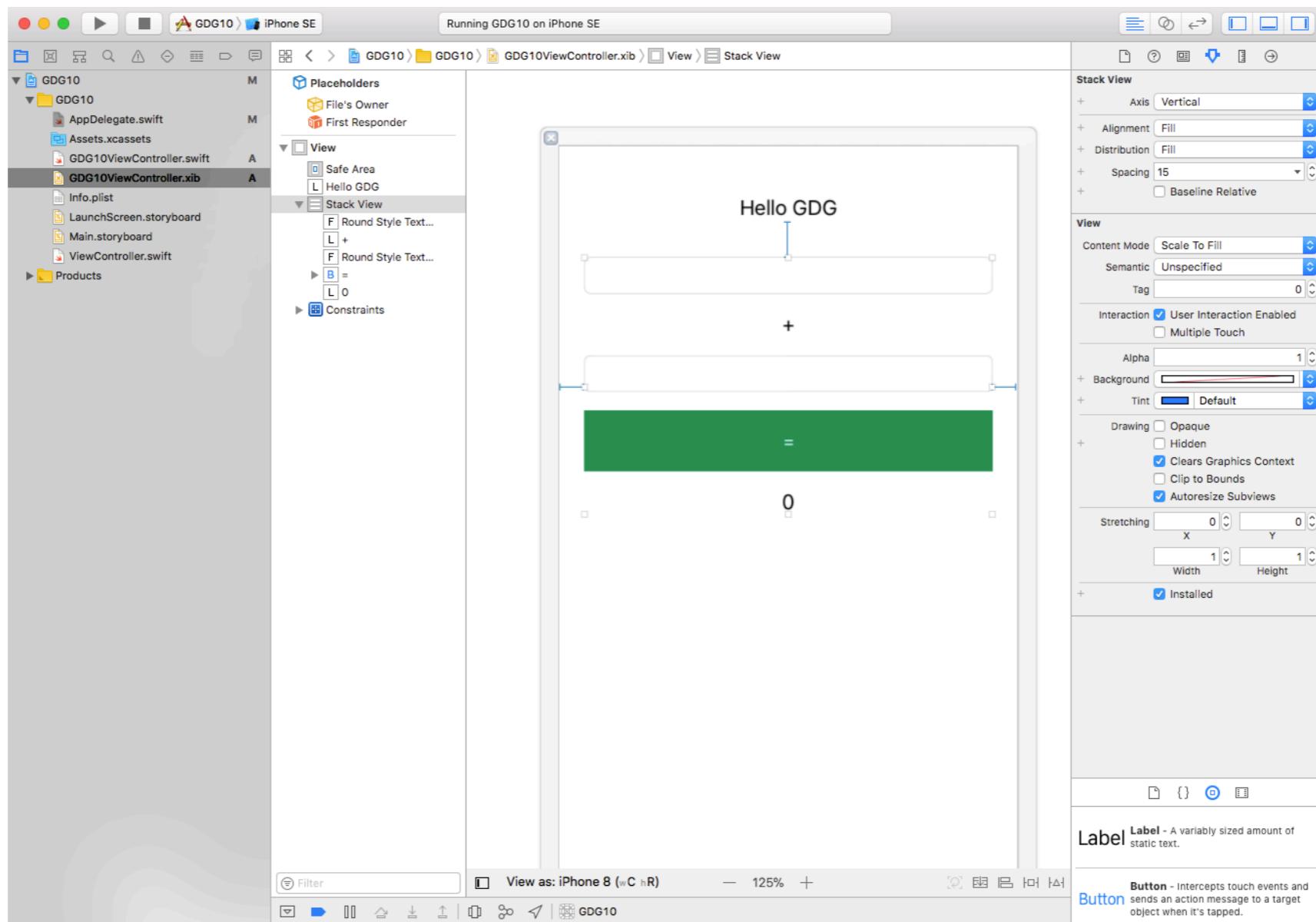
Tworzenie widoków

Storyboards



Tworzenie widoków

XIBs



Tworzenie widoków

Kod

```
1. let textField = UITextField()  
2. view.addSubview(textField)  
3. textField.translatesAutoresizingMaskIntoConstraints = false  
4. textField.leadingAnchor.constraint(equalTo: view.leadingAnchor, constant: 20).isActive = true  
5. textField.trailingAnchor.constraint(equalTo: view.trailingAnchor, constant: 20).isActive = true  
6. textField.topAnchor.constraint(equalTo: view.topAnchor, constant: 45).isActive = true
```

Swift

- 2014 - rok powstania
- Open Source
- Następca Objective C
- Podobieństwo do Kotlin



Składnia

Objective C:

```
1  UINavigationController *navigationController =
2      [[UINavigationController alloc] initWithRootViewController:
3          [[CarsListViewController alloc] initWithList:
4              [CarsListDataSource carsWithType: CarsListDataSourceTypeForSale]
5          ]]
6      ];
```

Swift:

```
1  let navigationController =
2      UINavigationController(rootViewController:
3          CarsListViewController(list: CarsListDataSource.cars(type: .forSale)))
4      )
```

Null safety

```
1 var errorCodeString: String?  
2 var errorDescription: String! = nil  
3  
4 errorCodeString = "404"  
5 errorDescription = "Resource not found"  
6  
7 print(errorCodeString) //Optional("404")  
8 print(errorDescription) //Resoruce not found  
9
```

Null safety

```
10 /*Force unwrapping*/
11 print(errorCodeString!) //404
12 print(errorCodeString ?? "No error") //404
13
14 /*Optional binding*/
15 if errorCodeString != nil {
16     let theError = errorCodeString!
17     print(theError) //404
18 }
```

Null safety

```
19  /*Optional binding*/
20  if let theError = errorCodeString {
21      print(theError)
22  } else {
23      print("errorCodeString is nil")
24 }
```

```
27  guard let theError = errorCodeString else {
28      print("errorCodeString is nil")
29      return
30 }
31 print(theError)
```

Null safety

```
33  /*Optional binding and additional checks*/
34  if let theError = errorCodeString,
35  -    let errorCodeInteger = Int(theError) {
36      print("\(theError): \(errorCodeInteger)")
37 }
```

```
39  /*Optional chaining*/
40  var errors: [String]?
41  errors?.append("404")
42  errors?.append("500")
43  print(errors) //nil
44
45  errors = ["404"]
46  errors?.append("500")
47  print(errors) //Optional(["404", "500"])
```

Funkcje

```
1 func add(_ a: Int, _ b: Int) -> Int {  
2     return a + b  
3 }  
4  
5 let sum = add(2, 3) //5  
6  
7 func printGreetings(name: String) {  
8     print("Hello \(name)!")  
9 }
```

Funkcje

```
7 func printGreetings(name: String) {  
8     print("Hello \(name)!")  
9 }  
10  
11 printGreetings(name: "Krzysztof")  
12
```

Funkcje

```
13 func printGreetings(to firstName: String, and secondName: String? = nil) {  
14     var greetings = "Hello \(firstName)"  
15     if let secondName = secondName {  
16         greetings += " and \(secondName)"  
17     }  
18     print(greetings)  
19 }  
20  
21 printGreetings(to: "Jacek") //Hello Jacek  
22 printGreetings(to: "Jacek", and: "Halina") //Hello Jacek and Halina
```

Value type vs Reference type

```
1 - class ErrorHandler {  
2     let error: HttpError  
3 -     init(error: HttpError) {  
4         self.error = error  
5     }  
6     //some implementation  
7 }
```

```
9 - struct HttpError {  
10    let error: Int  
11    let description: String  
12 }
```

Enum

```
26 enum SourceType {  
27     case web  
28     case cache  
29     case database  
30 }
```

```
8 public enum Result<Value> {  
9     case success(Value)  
10    case failure(Error)  
11  
12 func doSomething() {  
13     //do something  
14 }
```

Switch

```
1 switch response.result {  
2     case .success(let value):  
3         /*some code*/  
4     case .failure(let error):  
5         /*some code*/  
6 }
```

```
1 switch statusCode {  
2     case 400, 401, 403, 404:  
3         return "Client error, 4xx."  
4     case 500...505:  
5         return "Server error, 5xx."  
6     case let unknownCode where (unknownCode >= 200 && unknownCode < 300):  
7         return "\u{unknownCode} is not a known error code."  
8     default:  
9         return "Unexpected error."  
10 }
```

Extensions

```
1 extension String {  
2     mutating func kittyfy() {  
3         self = "😺" + self + "😺"  
4     }  
5 }  
6  
7 var someString = "Hello GDG10"  
8 someString.kittyfy()  
9 print(someString) //😺Hello GDG10😺
```

POP

```
1 protocol Loggable {  
2     func log(_ message: String)  
3 }  
4  
5 extension Loggable {  
6     func log(_ message: String) {  
7         print("LOG MESSAGE \(Date()): \(message)")  
8     }  
9 }
```

```
17 class ImageDownloader: Loggable {  
18     func download(completionHandler: ()->()) {  
19         //download image...  
20         completionHandler()  
21     }  
22 }
```

POP

```
1 protocol Loggable {  
2     func log(_ message: String)  
3 }  
4  
5 extension Loggable {  
6     func log(_ message: String) {  
7         print("LOG MESSAGE \(Date()): \(message)")  
8     }  
9 }
```

```
24 let imageDownloader = ImageDownloader()  
25 imageDownloader.download {  
26     imageDownloader.log("😎image downloaded! 🤘")  
27     //LOG MESSAGE 2017-11-29 15:16:31 +0000: image downloaded  
28 }
```

POP

```
11 extension Loggable where Self: UIViewController {  
12     func log(_ message: String) {  
13         print("ViewController message: \(message)")  
14     }  
15 }
```

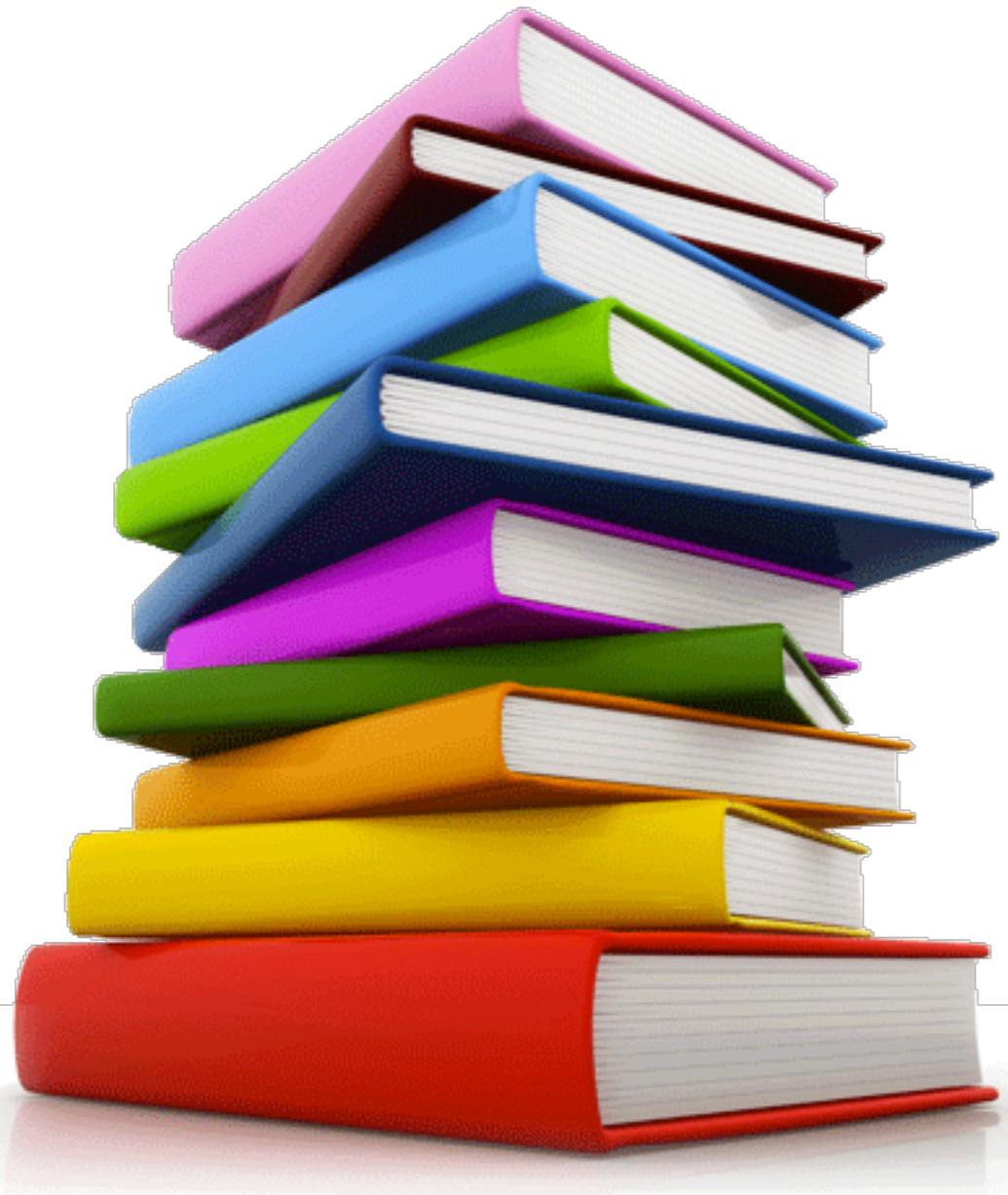
```
31 class MyViewController: UIViewController, Loggable {  
32     override func viewDidLoad() {  
33         log("view did load")  
34         //ViewController message: view did load"  
35     }  
36 }  
37  
38 let vc = MyViewController()  
39 vc.log("Hello, GDG10!") //ViewController message: Hello, GDG10!"
```

Integracja kodu z widokiem

```
1. @IBOutlet weak var firstTextField: UITextField!
2. @IBOutlet weak var secondTextField: UITextField!
3.
4.
5. @IBAction func myButtonTapped(_ sender: UIButton) {
6.     print("Hello GDG10!")
7. }
```

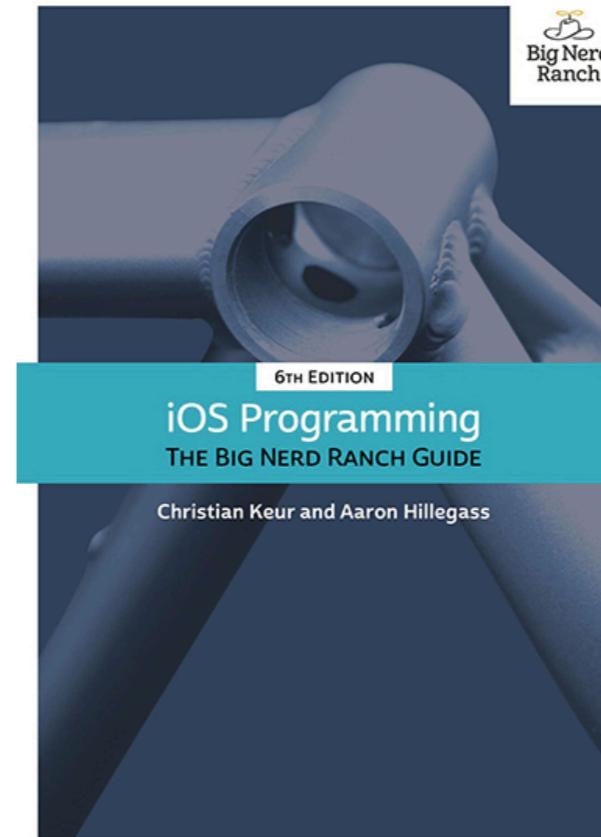
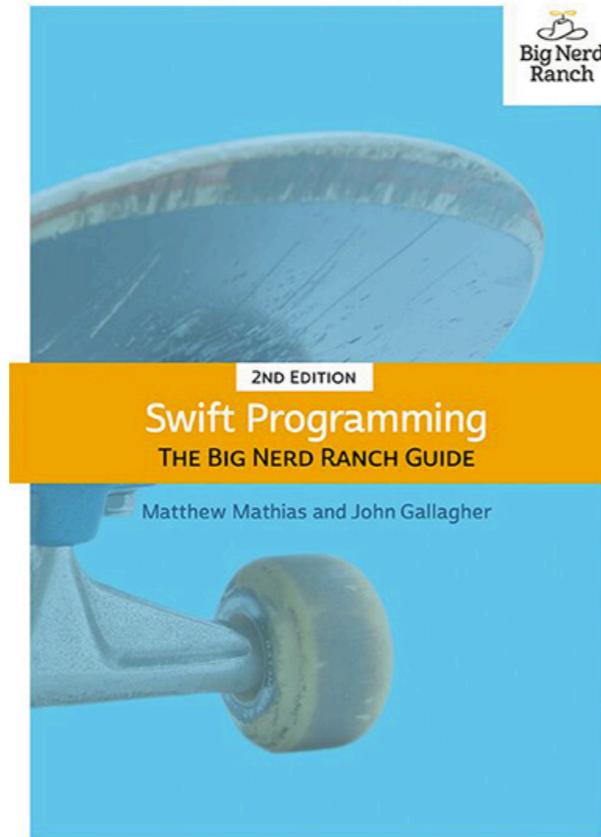
Demo

Źródła wiedzy



Książki

- raywenderlich.com
- bignerdranch.com/books/



Internet

- raywenderlich.com
- appcoda.com
- youtube.com
 - lets build that app

Internet

- iOS Dev Weekly
- iOS Goodies
- Little bites of cocoa
- Swift Dev Weekly
- Weekly Cocoa

Dziękuję! Q&A

github.com/JakubTudruj/GDG10

