EE443/EE593
MOBILE APPLICATION DEVELOPMENT
CHAPTER 26

YEHIA ARAFA

KLIPSCH SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

OUTLINE

1. Where are we (quick reminder) & chapter goal.
2. Chapter 26:
   - Improve the user experience.
   - Add a HUD.
   - Handle the navigation.
3. Additional stuff:
   - XML Parsing
   - CocoaPods

OUTLINE

1. Where are we (quick reminder) & chapter goal.
2. Chapter 26:
   - Improve the user experience.
   - Add a HUD.
   - Handle the navigation.
3. Additional stuff:
   - XML Parsing
   - CocoaPods

CHAPTER 25 GOALS

- The main goal of this chapter is to improve the user experience; by adding some polish to the **tag location screen.**

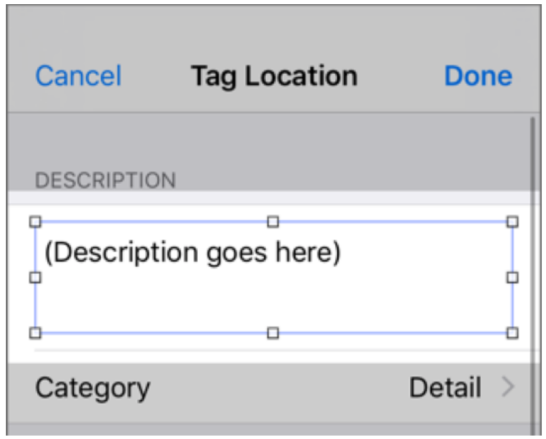
- User adoption is critical. Projects fail without happy users. Customers who have a positive user experience are going to be more likely to stick with your apps.

OUTLINE

1. Where are we (quick reminder) & chapter goal.
2. Chapter 26:
   - Improve the user experience.
   - Add a HUD.
   - Handle the navigation.
3. Additional stuff:
   - XML Parsing
   - CocoaPods

IMPROVE THE USER EXPERIENCE

- The cell having the description text view have a 10-point margin between the text view and the cell border. So it is possible to tap on the cell and be just outside the text view area.
- This is bad UX and annoying when you want to start typing and the keyboard just doesn't appear.
- How can we fix this ?
  - **Keyboard activation for cells.** When the user taps anywhere inside that first cell, the text view should activate, even if the tap wasn't on the text view itself.



IMPROVE THE USER EXPERIENCE

- Gesture Recognizer.
  - **UIGtureRecognizer** tutorial: https://www.raywenderlich.com/162745/uigesturerecognizer-tutorial-getting-started

- #selector () keyword
  - This pattern is known as **target-action**, The "target" is the object that the message should be sent to, which is often self, and "action" is the message to send [named by the #selector()]

IMPROVE THE USER EXPERIENCE

- Gesture Recognizer.
  - **UIGtureRecognizer** tutorial: https://www.raywenderlich.com/162745/uigesturerecognizer-tutorial-getting-started

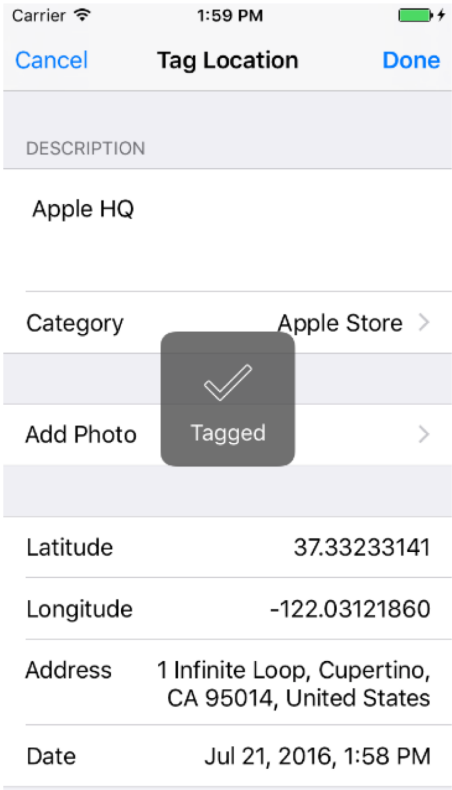
- #selector () keyword
  - This pattern is known as **target-action**, The "target" is the object that the message should be sent to, which is often self, and "action" is the message to send [named by the #selector()]

OUTLINE

1. Where are we (quick reminder) & chapter goal.
2. Chapter 26:
   - Improve the user experience.
   - Add a HUD.
   - Handle the navigation.
3. Additional stuff:
   - XML Parsing
   - CocoaPods

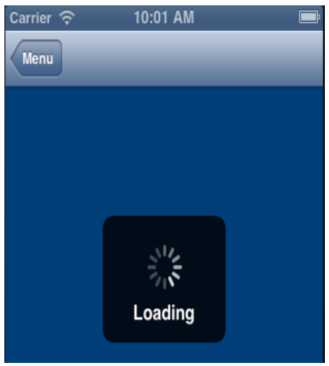
ADD A HUD

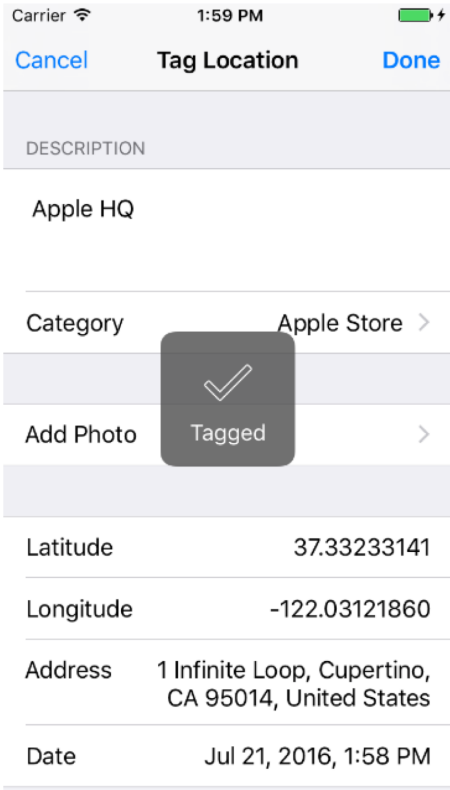
- Add a little spice:
  - When the user tap the Done button to close the screen, the app will show a quick animation to let you know it successfully saved the location.
- What is a HUD ?
  - **H**eads-**U**p **D**isplay, which is simply a UIView subclass that you can add on top of your main view class.
  - We are going to make our own view for this one as the HUD is not available as a standard view.



ADD A HUD

- Add a little spice:
  - When the user tap the Done button to close the screen, the app will show a quick animation to let you know it successfully saved the location.
- What is a HUD ?
  - **H**eads-**U**p **D**isplay, which is simply a UIView subclass that you can add on top of your main view class.
  - We are going to make our own view for this one as the HUD is not available as a standard view.





ADD A HUD

- Static vs Class vs Global Functions in Swift
  1. **Static functions** are invoked by the class itself, not by an instance. This makes it simple to invoke utility functions without having to manage an object to do that work for you.
     - We can access static function as **AppUtils.appUtility().**
     - Static functions can not be overridden.

```
class AppUtils {
    static func appUtility() {
    }
}
```

ADD A HUD

- Static vs Class vs Global Functions in Swift

  2. **Class functions** (not instance methods) are also static functions but they are dynamically dispatched and can be overridden by subclasses unlike static functions.
     - We can access them similar to static functions as **AppUtils.appUtility()** and **AppOtherUtils.appUtility()**.
     - Class method is a method that works on the class as a whole and not any particular instance

```
class AppUtils{
    class func appUtility(){
        println("In AppUtils")
    }
}
class AppOtherUtils: AppUtils{
    override class func appUtility(){
        println("In AppOtherUtils")
    }
}
```

ADD A HUD

- Static vs Class vs Global Functions in Swift
  3. **Global functions** are stand alone functions in Swift which is not within a class and access it anywhere in the project. The global functions can be kept in a separate file that we can import into any project as per requirement.
     - We can just access them as appUtility() anywhere in the projectStatic is same as class final.
     - In case of static functions, if we access one of the static member, entire class gets loaded in memory. But in case of global function, only that particular function will be loaded in memory.

```
func appUtility() {
}
```

OUTLINE

1. Where are we (quick reminder) & chapter goal.
2. Chapter 26:
   - Improve the user experience.
   - Add a HUD.
   - Handle the navigation.
3. Additional stuff:
   - Request data from Google API & parse it
   - CocoaPods

HANDLE THE NAVIGATION

- Dismiss the **Tag Location** screen after giving the user a chance to see the HUD.
- When the user tap on the **done** button, we want to tell the app to close the **Tag Location** screen after 0.6 seconds not right away.
- To do that we are going to use the **G**rand **C**entral **D**ispatch framework, or GCD.
  - (GCD) dispatch queues are a powerful tool for performing tasks. Dispatch queues let you execute arbitrary blocks of code either asynchronously or synchronously with respect to the caller.
  - You can use dispatch queues to perform nearly all of the tasks that you used to perform on separate threads.
  - GCD provides some dispatch queues for you automatically, but others you can create for specific purposes.
  - Tutorial: https://www.raywenderlich.com/148513/grand-central-dispatch-tutorial-swift-3-part-1

HANDLE THE NAVIGATION

- **Synchronous execution**

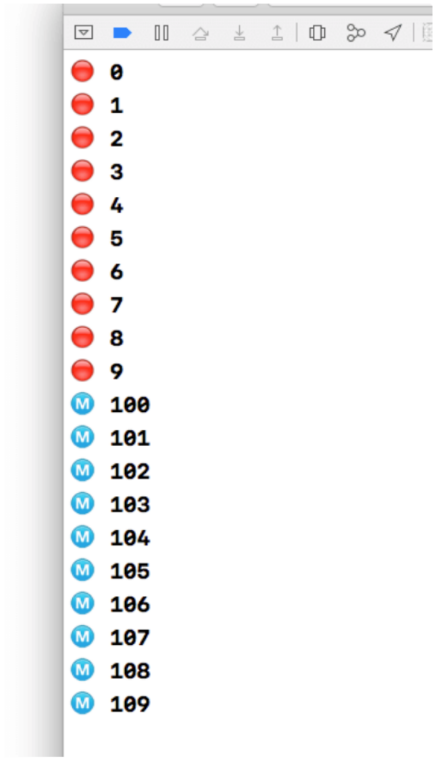
```
override func viewDidAppear(_ animated: Bool) {
    super.viewDidAppear(animated)

    simpleQueues()
}


func simpleQueues() {
    let queue = DispatchQueue(label: "com.appcoda.myqueue")

    queue.sync {
        for i in 0..<10 {
            print("🔴", i)
        }
    }

    for i in 100..<110 {
        print("Ⓜ️", i)
    }
}
```



https://www.appcoda.com/grand-central-dispatch

HANDLE THE NAVIGATION

- **Asynchronous execution**

```
override func viewDidAppear(_ animated: Bool) {
    super.viewDidAppear(animated)

    simpleQueues()
}


func simpleQueues() {
    let queue = DispatchQueue(label: "com.appcoda.myqueue")

    queue.async {
        for i in 0..<10 {
            print("🔴", i)
        }
    }

    for i in 100..<110 {
        print("Ⓜ️", i)
    }
}
```

https://www.appcoda.com/grand-central-dispatch

OUTLINE

1. Where are we (quick reminder) & chapter goal.
2. Chapter 26:
   - Improve the user experience.
   - Add a HUD.
   - Handle the navigation.
3. Additional stuff:
   - XML Parsing
   - CocoaPods

XML PARSING

- XML is a software -and hardware- independent tool for storing and transporting data.
- It stands for e**X**tensible **M**arkup **L**anguage.
- XML is a markup language much like HTML, and was designed to be self-descriptive.
- Tons of document formats use XML syntax, like RSS, Atom, SOAP and XHTML, so it's good to know, how to work with them especially if you are planning to use a certain API in your future app.

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

XML PARSING

- I will perform a request to get RSS news feed from NMSU news center and parse the data returned.
- https://newscenter.nmsu.edu/Articles/news

```
▼ <item>
    <title> NMSU honored as family-friendly workplace </title>
    <link> http://newscenter.nmsu.edu/Articles/view/13040/nmsu-honored-as-family-friendly-workplace </link>
    <guid isPermaLink="true"> http://newscenter.nmsu.edu/Articles/view/13040/nmsu-honored-as-family-friendly-workplace </guid>
    <pubDate> Thu, 29 Mar 2018 16:00:00 -0600 </pubDate>
    <dc:creator> Tiffany Acosta </dc:creator>
  ▼ <description>
      <![CDATA[ <p>NMSU receives New Mexico Family Friendly Business Award. </p> ]]>
    </description>
  ▶ <content:encoded>
  </item>
```

OUTLINE

1. Where are we (quick reminder) & chapter goal.
2. Chapter 26:
   - Improve the user experience.
   - Add a HUD.
   - Handle the navigation.
3. Additional stuff:
   - XML Parsing
   - CocoaPods

COCOAPODS

- CocoaPods is an application level dependency manager for swift.
- It installs dependencies (e.g. libraries and 3rd party apps) for an application by specification of dependencies rather than manual copying source files.
- It has over 30 thousand libraries and is used in over 1.9 million apps. CocoaPods can help you scale your projects elegantly.
- Tutorial: https://www.raywenderlich.com/156971/cocoapods-tutorial-swift-getting-started
- We will use CocoaPods to add **FeedKit** [https://github.com/nmdias/FeedKit] in our app and perform network request + parsing the data with only one single line of code!

ALL DONE FOR TODAY

Thank you!