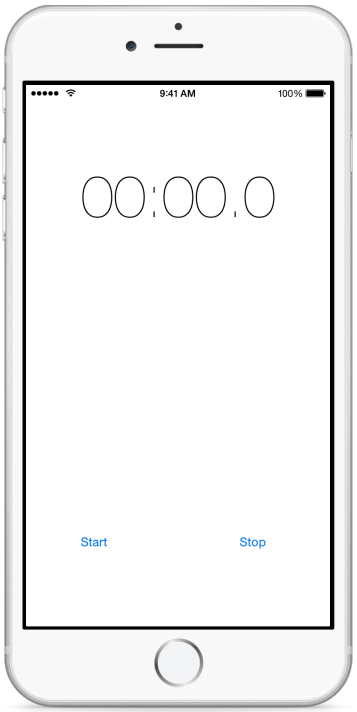
Stopwatch
Lesson 2

Description

Review the Model-View-Controller design pattern, analyze model requirements, and add a Stopwatch model to the project.



Learning Outcomes

- Define the Model-View-Controller pattern and identify the model in an Xcode project.
- Analyze model requirements and relate state and behavior to properties and methods.
- Practice implementing a class definition consisting of properties and methods.
- Interpret the meaning of a Swift optional, and observe optional binding with `if let`.

Vocabulary

Model-View-Controller	model	state
behavior	property	`var`
optional	initialization	`NSDate`
computed property	accessor	mutator
`NSTimeInterval`	optional binding	`if let`
method	`nil`	

Materials

- **Stopwatch Lesson 2** Xcode project
- **Model-View-Controller** presentation

Opening

We've created the view components, now how do we create the model? What should the model be?

Agenda

- Present the MVC design pattern, focusing on how models, views and controllers manifest in an Xcode project.
- Add a new class (⌘N) to the project called `Stopwatch`.

```
class Stopwatch {

}
```

- Discuss what state and behavior a `Stopwatch` model should have, including a start time, elapsed time, starting and stopping.
- Add a `startTime` property to the `Stopwatch` class.

```
private var startTime: NSDate?
```

- Explain why it is necessary to declare `startTime` as a `var` and an optional, because its initial value will be set when the `Stopwatch` starts to track elapsed time, rather than during initialization.
- Discuss how a `Stopwatch` might represent elapsed time as the difference between its `startTime` and the current time.
- Using the Xcode Documentation and API Reference (⇧⌘0), explore the `NSDate` class reference and the `timeIntervalSinceNow` method.
- Declare a computed property for `elapsedTime`.

```
var elapsedTime: NSTimeInterval {
   if let startTime = self.startTime {
      return -startTime.timeIntervalSinceNow
   } else {
      return 0
   }
}
```

- Explain the computed property syntax, and how Swift computed property declarations generate accessor and, optionally, mutator methods (getters and setters).
- Discuss the the use of if let to unwrap the optional `startTime` property, and using it to return an `NSTimeInterval`.
- Using the Xcode Documentation and API Reference (⇧⌘0), explore the `NSTimeInterval` data type, and discuss its expressiveness in comparison to `Double`.
- Add a `start` and `stop` method to the `Stopwatch` model.

```
func start() {
   startTime = NSDate()
}

func stop() {
   startTime = nil
}
```

- Discuss how, when a `Stopwatch` starts, it keeps track of its start time by assigning a new `NSDate` to its `startTime` property.
- Explain the assignment of `nil` to an optional property to indicate that it has "no value."
- Run the app (⌘R), and observe that the model, view and controller exist but lack integration.

Closing

Why is it important to make the start time a private property, and the elapsed time a read-only, computed property?

Modifications and Extensions

- Replace the `elapsedTime` computed property with a method that exhibits similar behavior. Decide which approach you feel is better, and explain why.
- Add a `pause` and `resume` behavior to the `Stopwatch` model.

Resources

Cocoa Core Competencies: Model Object http://developer.apple.com/library/ios/documentation/general/conceptual/devpedia-cocoacore/ModelObject.html

The Swift Programming Language: Classes and Structures https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/ClassesAndStructures.html

The Swift Programming Language: Properties https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Properties.html

NSDate Class Reference https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSDate_Class/Reference/Reference.html

The Swift Programming Language: Optional Binding https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/TheBasics.html#//apple_ref/doc/uid/TP40014097-CH5-ID333

Foundation Data Types Reference https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Miscellaneous/Foundation_DataTypes/

The Swift Programming Language: Methods https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Methods.html

Start Developing iOS Apps Today: Finding Information https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/FindingInformation.html