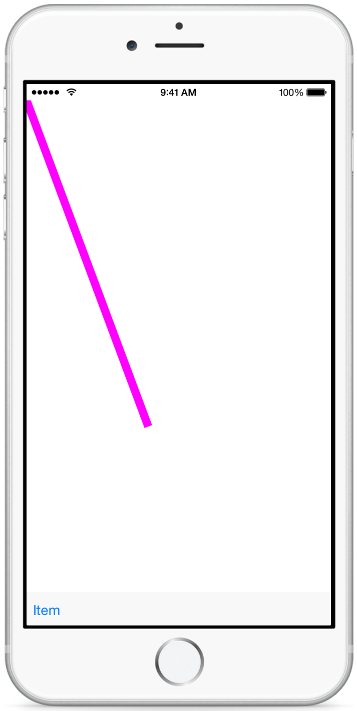
FingerPainter
Lesson 2

Description

Override the `UIResponder` methods `touchesBegan:withEvent:` and `touchesMoved:withEvent:` to illustrate how the device responds to moving touches.



Learning Outcomes

- Describe the inheritance hierarchy of view controllers.
- Implement `UIResponder` methods in a controller to handle touch events.
- Practice creating custom breakpoint actions to print console messages.
- Discover how touching the device screen generates event-driven method calls.

Vocabulary

inheritance	`UIViewController`	`UIResponder`
override	breakpoint	

Materials

- **FingerPainter Lesson 2** Xcode project

Opening

Using `drawRect:` is ok for simple custom views that don't change very much, but how can we create a custom view that continuously updates as we touch the screen?

Agenda

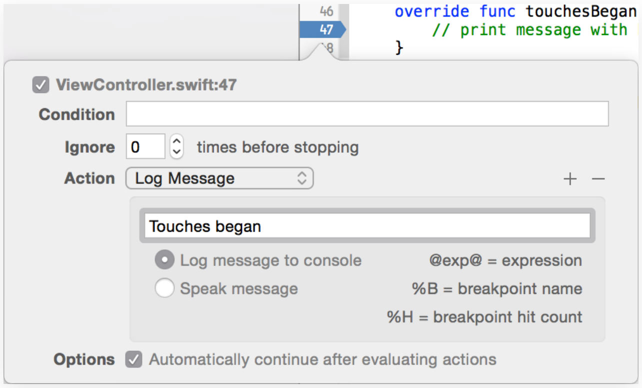
- Discuss the desire to draw a line as the user touches the screen.
- Explain how one approach is to draw a line from point to point as the user drags a finger across the screen.
- Discuss how app view controllers inherit from `UIViewController`, which inherits from `UIResponder`.
- Using the Xcode Documentation and API Reference (⇧⌘0), explore the `UIResponder` class reference, drawing attention to the `touchesBegan:withEvent:` and `touchesMoved:withEvent:` methods.
- Discuss how overriding `touchesBegan:withEvent:` and `touchesMoved:withEvent:` in the view controller can facilitate drawing a continuous line as the user drags a finger on the screen.
- In the `ViewController` class, add an implementation of `touchesBegan:withEvent:`.

```
override func touchesBegan(touches: Set<UITouch>,
   withEvent event: UIEvent?) {
   // print message with breakpoint here
}
```

- Explain that `touchesBegan:withEvent:` is called as soon as a user touches the screen.
- Add an implementation of `touchesMoved:withEvent:`.

```
override func touchesMoved(touches: Set<UITouch>,
   withEvent event: UIEvent?) {
   // print message with breakpoint here
}
```

- Explain that `touchesMoved:withEvent:` is called repeatedly, as the user drags a finger across the screen.
- Add custom breakpoints to the bodies of both the `touchesBegan:withEvent:` and `touchesMoved:withEvent:` methods that use a **Log Message** action to print a console message and automatically continue.



```
override func touchesBegan
    // print message with
}
```

☑ ViewController.swift:47

Condition

Ignore 0 times before stopping

Action Log Message

Touches began

● Log message to console

○ Speak message

@exp@ = expression

%B = breakpoint name

%H = breakpoint hit count

Options ☑ Automatically continue after evaluating actions

- Run the app (⌘R), click on the screen to simulate a touch, and observe the console (⇧⌘C) output reporting the start of a touch event.
- Click and drag on the Simulator screen to simulate a touch that moves, and observe the console (⇧⌘C) output reporting the movement of a touch.

Closing

What is the `touches` argument that both the `touchesBegan:withEvent:` and `touchesMoved:withEvent:` methods receive?

Modifications and Extensions

- Investigate the additional methods in the `UIResponder` class, implement the `touchesEnded:withEvent:` method, and add a custom breakpoint to print a message to the console when the method is called.
- Update the `touchesBegan:withEvent:` and `touchesMoved:withEvent:` methods to print the number of touches on the screen, simulate two touches with the simulator, and describe the number of touches your app prints to the console.

Resources

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

UIViewController Class Reference https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIViewController_Class/index.html

UIResponder Class Reference http://developer.apple.com/library/ios/documentation/uikit/reference/UIResponder_Class/Reference/Reference.html

Event Handling Guide for iOS https://developer.apple.com/library/ios/documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/Introduction/Introduction.html

Setting Breakpoint Actions and Options http://developer.apple.com/library/ios/recipes/xcode_help-breakpoint_navigator/articles/setting_breakpoint_actions_and_options.html