Beginner

101: iOS Apprentice Review Part 1

Part 3: Lab

iOS Apprentice Review Part 1
Part 3: Lab Instructions

The Story: Generate & Reset

In this Lab, you'll be adding some basic code for the Story Time app so that when you tap on the "Generate Story" button, the story will display in the textview.

Step 1: Add Code To Generate Story

Locate `generateStory()` and add the following code:

```
switch sender.tag {
  case 1:
    sender.tag = 2
    sender.setTitle("Start Over", forState: UIControlState.Normal)
    populateStory()
  case 2:
    sender.tag = 1
    sender.setTitle("Generate Story", forState:
UIControlState.Normal)
    resetStory()
  default:
    break
}
```

In the previous block of code you are checking the tag value of the button (identified here as sender), and using a case statement to instruct the sytem to either populate or reset the sotry based on that value. You're also re-setting the tag and the title of the button.

Note: A sender is the object that sent the message to the selector.

Step 2: Change Button Tag

You'll notice in `generateStory()` you're using a tag value of the button to determine what action to take when the button is pressed. Whenever you add a new control to the view, the tag value is set to 0. Find the button and change its tag value to 1. This setting is located in the View section of the button's Attributes Inspector.

Step 3: Uncomment Code

Uncomment the code in `populateStory()`. Once you have done that, the colde will be the following:

```
if (currentSwitchValue) {
  textview.text = "\(textField1.text) entered the room and saw
\(currentNumber) \(monsters)! \(textField1.text)
\(textField2.text) down the hall. Sadly, \(textField1.text) was
killed by all of the \(monsters)! \n\nPoor \(textField1.text).
\n\nBetter luck next time!"
} else {
  textview.text = "\(textField1.text) entered the room and saw
\(currentNumber) \(monsters)! \(textField1.text)
\(textField2.text) down the hall. Without missing a beat,
\(textField1.text) killed all of the \(monsters)! \n\nFantastic!
\n\nOur hero will live to fight another day."
}
```

The code you just uncommented determines, based on the `currentSwitchValue` setting, which story to display (monsters win or monsters lose). Then, it builds a string that will be displayed in the textview using string interpolation. String interpolation allows you to create a string by replacing the placeholders (the items wrapped in parentheses, prefixed by a backslash) with the value of that item.

Step 4: Add Code To Reset Story

Add the following code to `resetStory()`.

```
textField1.text = ""
textField2.text = ""
sliderControl.value = 50
switchControl.on = true
textview.text = "your generated story will appear here"

currentNumber = 50
currentSwitchValue = true

button.setTitle("Generate Story", forState: UIControlState.Normal)
button.tag = 1
```

This code simply resets all of your default values and clears any values in the text fields. It also resets the button text and tag for the Generate Story button.

Step 5: Add Code To Hide Keyboard

Add the following code to `hideKeyboard()`:

```
textField1.resignFirstResponder()
textField2.resignFirstResponder()
```

Using those two lines, you can notify the text fields to resign their first responder status; this will hide the keyboard if it's on-screen.

Step 6: Obtain Value For Slider

Add the following code to `sliderMoved()`:

```
currentNumber = lroundf(sender.value)
```

Here you are simply storing the value from the slider control into the `currentNumber` variable so you may use it later when generating the story.

Step 7: Obtain Value For Switch

Add the following code to `switchValueChanged()`:

```
if (sender.on) {
  currentSwitchValue = true
} else {
  currentSwitchValue = false
}
```

These four lines of code simply check the current value of the switch and stores it in the variable `currentSwitchValue`.

Step 8: Build & Run

At this point you have a *mosty* working Story Time app. In the Lab, you will work with the segmented control and image views which will allow the user to choose between a zombie story and a vampire story.

Finishing Touches

At this point, you have the basic functionality complete for the Story Time app.

However, there are two major features the app still needs:

1. **Ability to change the story**. When the segmented control value is selected (zombies or vampires), the story line should change as well as the background image.
2. **Showing the number selected on the slider**. It's no fun not knowing the value of the slider (number of monsters). This value should be shown.

Let there be vampires!

You'll start by adding code to handle the segmented control value change. In **ViewController.swift**, locate `segmentedControllerChanged()` and add the following code:

```
storyType = sender.selectedSegmentIndex

var image : UIImage!

switch storyType {
  case 0:
    image = UIImage(named:"zombies")
    monsters = "zombies"
  case 1:
    image = UIImage(named:"vampires")
    monsters = "vampires"
  default:
     break
}

resetStory()
backgroundImage.image = image
```

In the code above, you are setting the variable `storyType` which will be used in determining which story to display. You are also setting the image view to match the story selection, and resetting the story in the event the user changes the selection after generating the story. Finally, you are setting the newly created image as the image for the `backgroundImage.image`.

How many monsters did you say there were?

Next, let's add a way to show the user how many monsters they selected on the slider.

Adding Properties

Add the following property to the top of **ViewController.swift**:

```
@IBOutlet weak var numberLabel: UILabel!
```

Then, connect this IBOutlet to the corresponding "Number" label on the view.

Showing the Value

Next, you will need to add the same line of code to three separate methods: `viewDidLoad()`, `sliderMoved()`, and `resetStory()`. Add the the following line of code to the end of each method specified above:

```
numberLabel.text = "Number (\(currentNumber))"
```

That line of code sets the text for the newly created IBOutlet `numberLabel` to the value stored in the `currentNumber` property. It does so on load, when the slider value changes, and when the story has been reset.

If you have time… please move on to the next page to see how you can extend your app to include an additional story type.

Bonus Lab – Time Permitting

Only proceed with the following if you have time left in your Lab.

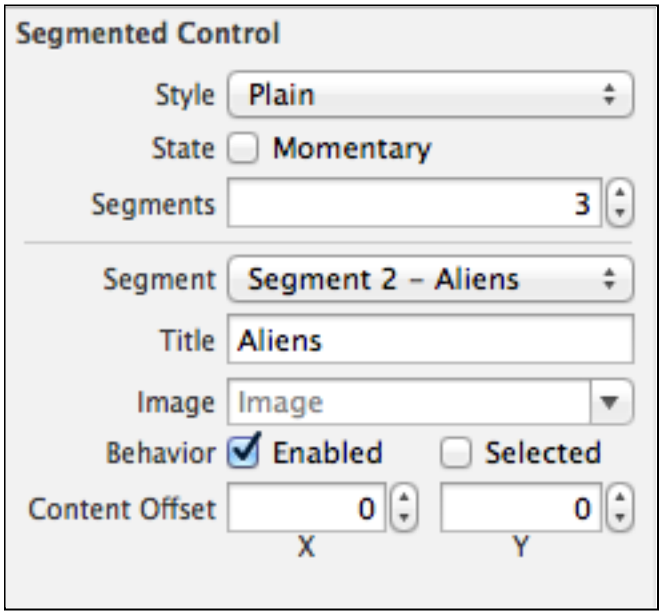
But what about the aliens?

A monster app is nothing without Aliens. Right? In this next section, you will learn how to add a third option to your segmented control.

Adding Aliens

The movie poster for aliens has already been added to the project. In order to allow the user to select this as an option, you will need to add an additional segment to your segmented control and give it the title, "Aliens".

In **Main.storyboard** set the segmented control like so:



Segmented Control
Style: Plain
State: Momentary [ ]
Segments: 3
Segment: Segment 2 – Aliens
Title: Aliens
Image: Image
Behavior: Enabled [✓] Selected [ ]
Content Offset: X 0, Y 0

In **ViewController.swift**, you'll also need to update `segmentedControllerChanged()` with the following code:

```
storyType = sender.selectedSegmentIndex

var image : UIImage!

switch storyType {
  case 0:
    image = UIImage(named:"zombies")
    monsters = "zombies"
  case 1:
    image = UIImage(named:"vampires")
    monsters = "vampires"
  case 2:
    image = UIImage(named:"aliens")
    monsters = "aliens"
  default:
    break
}

resetStory()
backgroundImage.image = image
```

That’s it! You now have a fully functioning Story Time app. In the challenge section, you’ll be adding a some validation and user feedback.