Workshop:
My first app: Apple Pie

Ana Nogal

ana.nogal@gmail.com

@anainogal

http://ananogal.com

Agenda

What are we learning today?

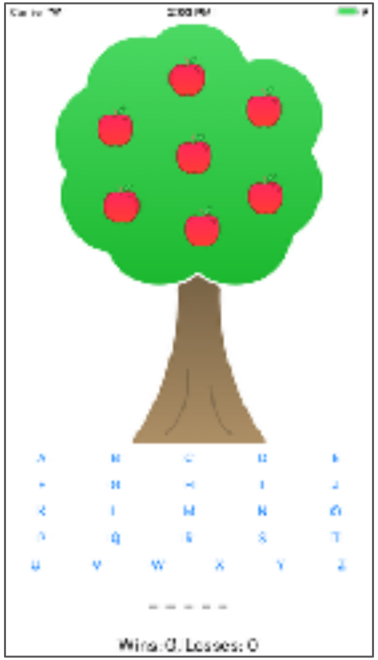
- Xcode: where things are and how to use it
- Auto Layout and Stack Views
- Swift Language:
  - Constants/Variables
  - Data Types
  - Conditional statements
  - Functions
  - Loops

Before we start

Make sure you have Xcode 9.x installed, and that you have downloaded the images from:

https://github.com/ananogal/Workshop-Apple-Pie-I

Apple Pie



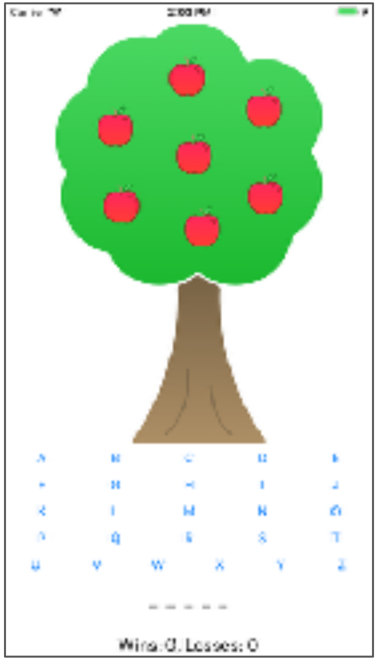
_ _ _ _ _

Wins: 0, Losses: 0

Challenge

Create a TODO list of all the things you need to do to make this app work.

Apple Pie



Wins: 0, Losses: 0

Solution

- Have different words to guess
- Show the letters that the player did guess
- Show dashes for each letter not guessed
- Disable selected letter
- Make an apple fall for wrong letters
- Add 1 to wins when player guesses the word
- Add 1 to losses when there are no more apples
- Enable all letters when starting a new round

Demo Playground

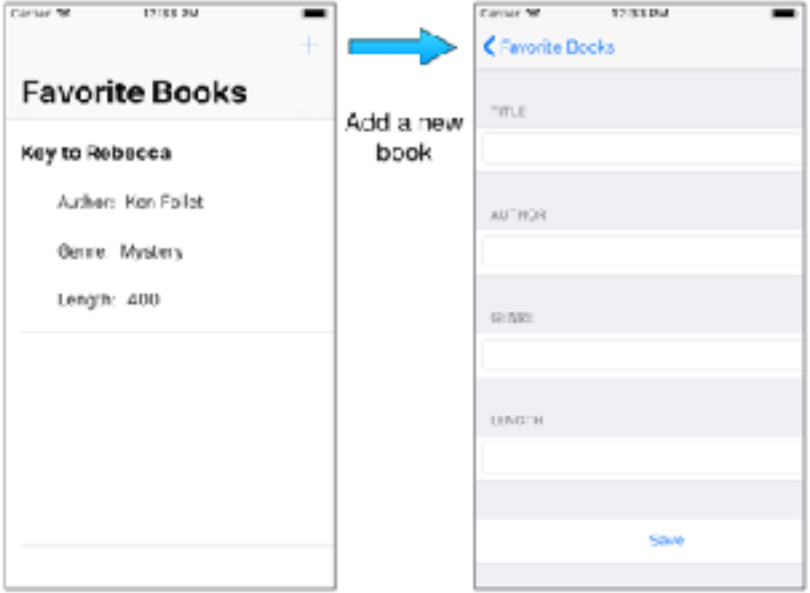
What have we learned so far?

- ***Swift Language:***
  - Variables and Constants
  - Data Types
  - Operators
  - Conditional Statements
  - Functions
  - Arrays

Demo Project

View Controllers

Manage a Screen or a portion of a Screen



Favorite Books

Key to Rebecca

Author: Ken Follett

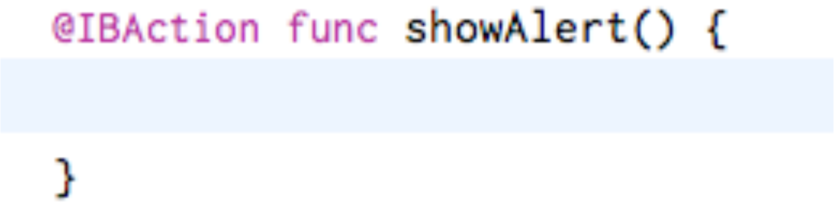
Genre: Mystery

Length: 400

View Controllers

Main.storyboard
(Design)

ViewController.swift
(Code)



```
@IBAction func showAlert() {

}
```

Add a button to the VC

- Go to Main.storyboard
- Drag and drop a button to the VC
- Center it in the VC
- Run the app

Different Sizes

- Different Devices -> Different screen sizes.
- Change the device
- Change the orientation
- What Happened?

Auto Layout

- Different sizes of screens
- Defines constraints or rules for all screens
- Stack Views
- Size Classes

Demo Auto Layout

Challenge 1

- Create three more Buttons “C”, “D” and “E”.
- All buttons should be 20 pixels from the top and 30 from the left.
- Add a different background-colour for each one.

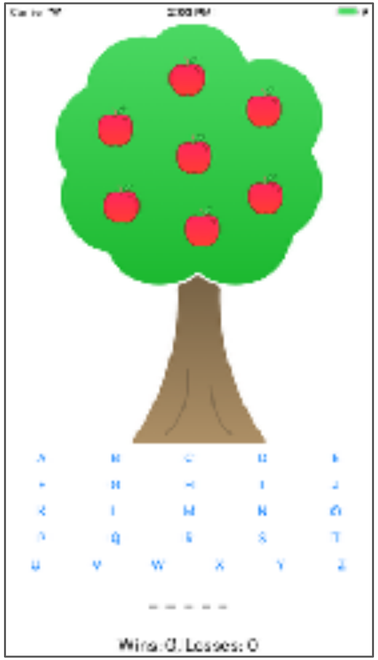
Stack Views

- Allows you to manage several constraints
- Vertical (Column)
- Horizontal (Row)

Demo Stack Views

- Go to Main.Storyboard
- Select all the buttons and embed them in a StackView
- Try out some of the properties of the StackView
- Add a top vertical StackView
- Add constraints to the top StackView

Challenge 2

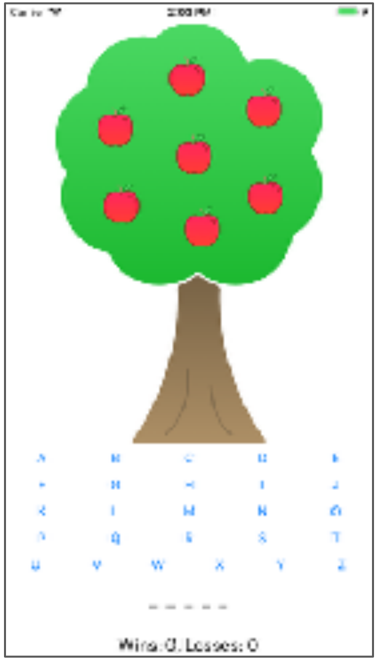


Wins: 0, Losses: 0

Walkthrough add images

- In the Navigator panel go to the Project Navigator
- Select the Assets folder
- Go to the folder where you download the images
- Drag and Drop them after the AppIcon
- Go back to the Main.storyboard
- Add an ImageView to the top StackView
- Set its image property to Tree 7
- Set “Content mode” to “Aspect Fit”.

Challenge 3



- Add two labels to the top Stack view
- Add a height constraint to the first label with the value 60
- Set the font size to 30 and centre the text
- Select the second label and add a height constraint of 30
- Set the font size to 20 and centre the text

IBOutlets

Establishes a relationship between Design file (Main.storyboard) and Code (ViewController.swift)

Demo Project

What have we learned so far?

- ***Swift Language:***
  - Variables and Constants
  - Data Types
  - Operators
  - Conditional Statements
  - Functions
  - Arrays

What have we learned so far?

- ***ViewControllers:***
  - Controls a screen.
  - Are made of 2 parts: the design and code.
  - Use IBOutlets to establish a relationship between design and code
- ***Auto Layout:***
  - Defines constraints or rules that can be applied to all screens.
- ***Stack Views:***
  - Manage the constraints of its children.

App Labs

- Unit 1
    - Units 1.1 to Unit 1.4
- Unit 2
    - Unit 2.1 to 2.7

Resources

App Development with Swift

Apple Swift Book

Human Interface Guidelines iOS