Workshop:
My first app: Apple Pie (Continue)

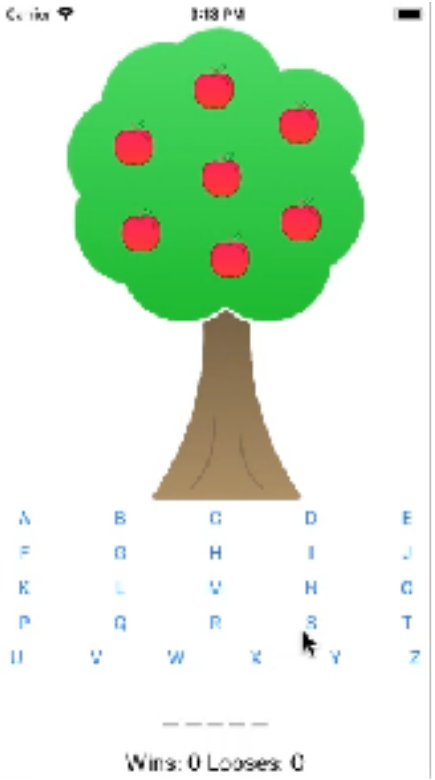
Ana Nogal
ana.nogal@gmail.com
@anainogal
http://ananogal.com

Agenda

What are we learning today?

- ***IBOutlets***
- ***IBActions***
- ***Finish our app***
  - Refactoring

Apple Pie



IBOutlets/ IBActions

They create a connection between UI elements in the storyboard and source code

Demo

IBOutlets / IBActions

Challenge 1

Create IBOutlets to:

- The imageView
- The two labels
- And all the buttons

  (Tip: change the connection to Outlet collection)

Add all buttons to the IBAction

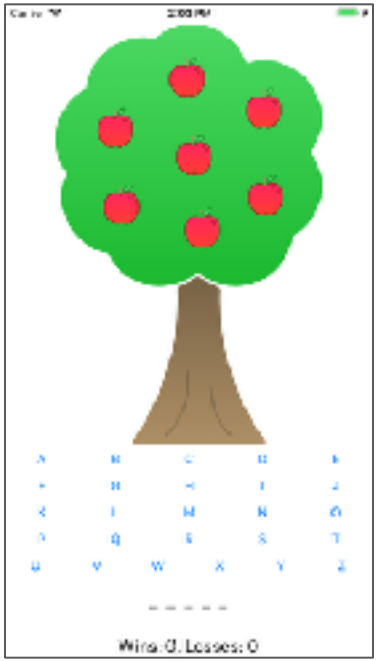
What have we learned so far?

- ***IBOutlets***
  - They connect controls such as Buttons and Labels to the code file
- ***IBActions:***
  - They execute code when an event occurs in the control that has the action

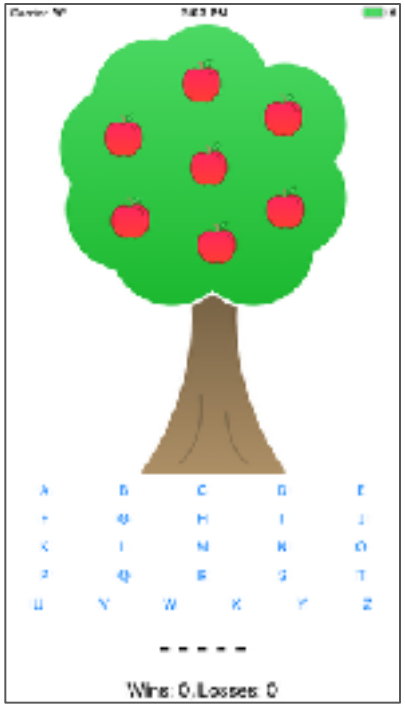
Challenge 2: Start a new round

Try to figure out what you need to do to start a new round.

Apple Pie



Solution



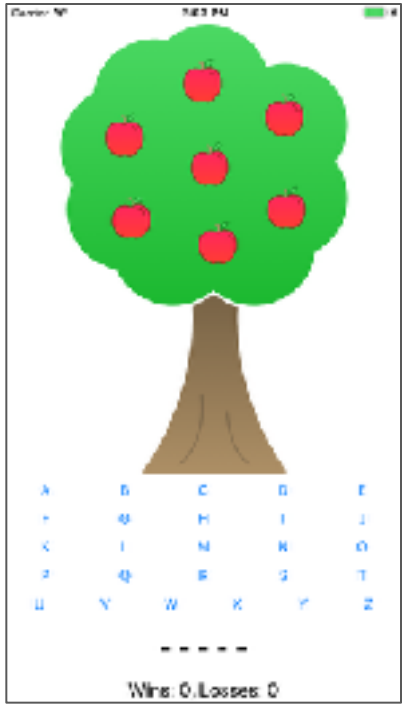
The tree should have 7 apples

All buttons are enabled

The wordLabel has a word hidden with a dash format

The scoreLabel has Wins and Losses set to 0

DEMO



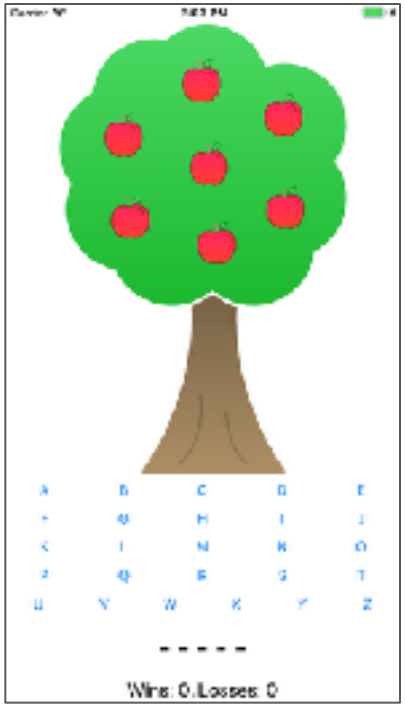
The tree should have 7 apples

All buttons are enabled

The wordLabel has a word hidden with a dash format

The scoreLabel has Wins and Losses set to 0

Challenge 3



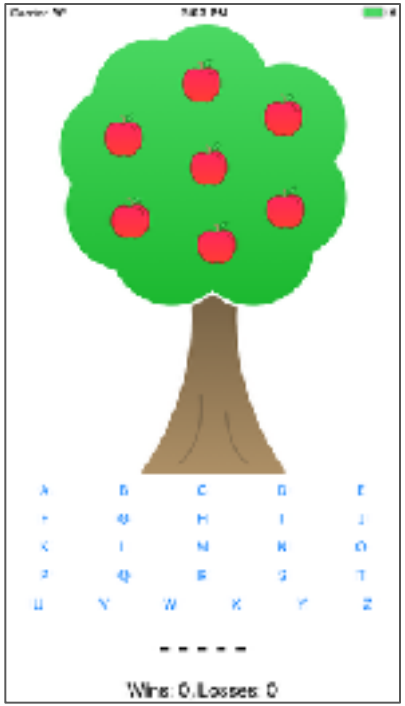
The tree should have 7 apples

All buttons are enabled

The wordLabel has a word hidden with a dash format

The scoreLabel has Wins and Losses set to 0

Demo



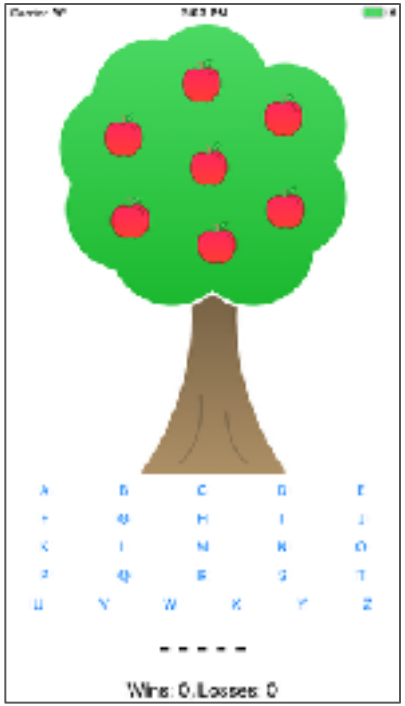
The tree should have 7 apples

All buttons are enabled

The wordLabel has a word hidden with a dash format

The scoreLabel has Wins and Losses set to 0

Challenge 4



The tree should have 7 apples

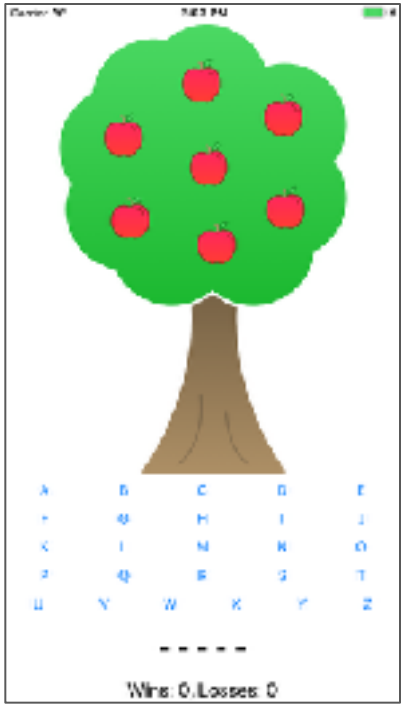
All buttons are enabled

The wordLabel has a word hidden with a dash format

(Arrays have a method called joined, that joins all elements of the Array in a String separated by the value passed in)

The scoreLabel has Wins and Losses set to 0

Demo



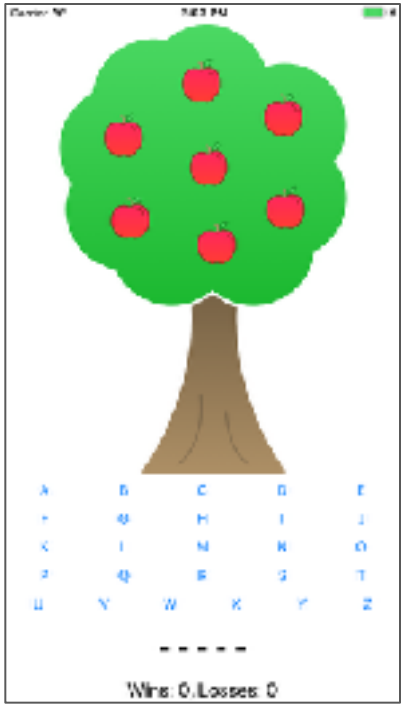
The tree should have 7 apples

All buttons are enabled

The wordLabel has a word hidden with a dash format

The scoreLabel has Wins and Losses set to 0

Challenge 6



The tree should have 7 apples

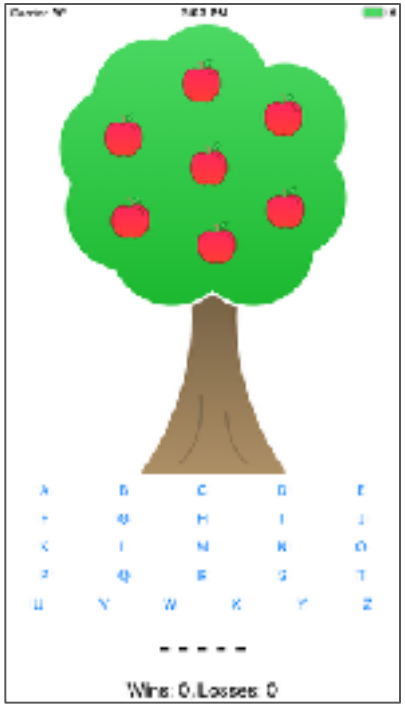
All buttons are enabled

The wordLabel has a word hidden with a dash format

The scoreLabel has Wins and Losses set to 0

(give a meaning to the zeros)

Refactoring



Is the process of restructuring existing code without changing its external behaviour.

Advantages:

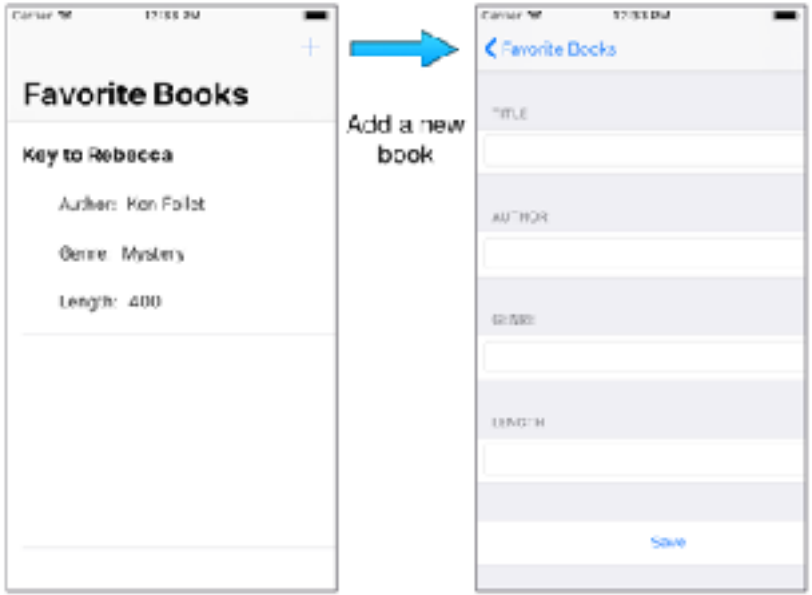
- **Code readability**
- **Reduced complexity**

What have we learned so far?

- ***IBOutlets:***
  - They connect controls such as Buttons and Labels to the code file
- ***IBActions:***
  - They execute code when an event occurs in the control that has the action
- ***Refactoring:***
  - After writing a piece of code, always take time to refactoring -> make your code readable by others

View Controllers

Manage a Screen or a portion of a Screen



Single Responsibility Principle (Separation of concerns)

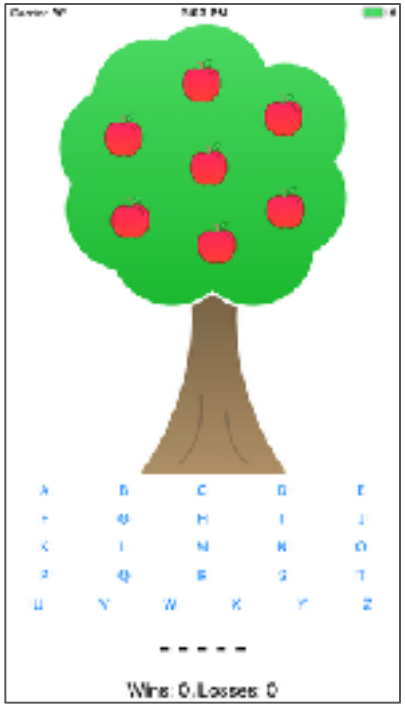
A class should have only one reason to change.

Refactoring Demo

Challenge 7: Guessing a Letter

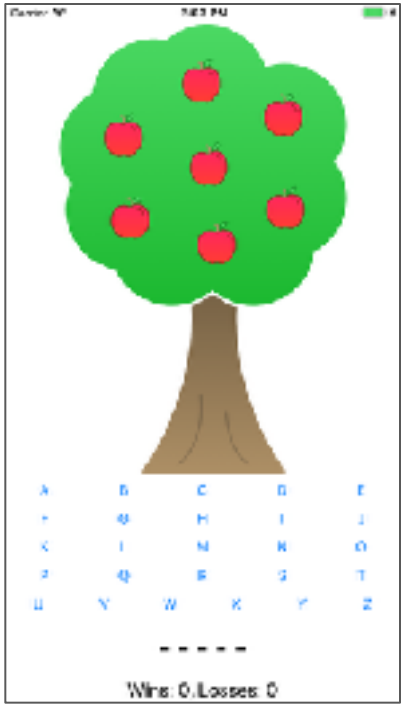
Try to figure out what happens when a player tries to guess a letter.

Solution - IBAction



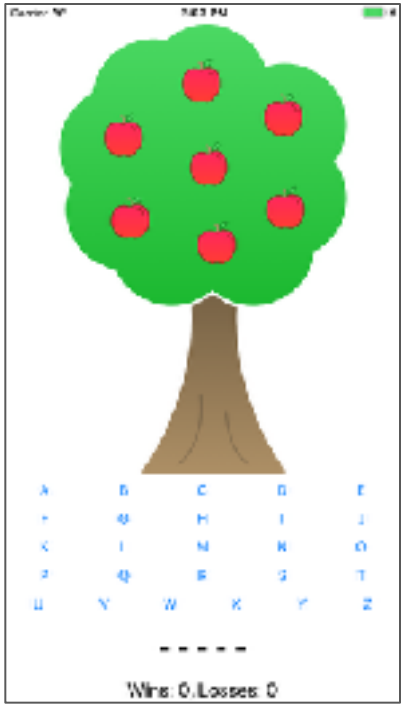
- The player hits a button, and the IBAction calls the game passing the guessed letter.
- The IBAction sets the button to disabled.
- The IBAction calls updateUI.

Solution - Game



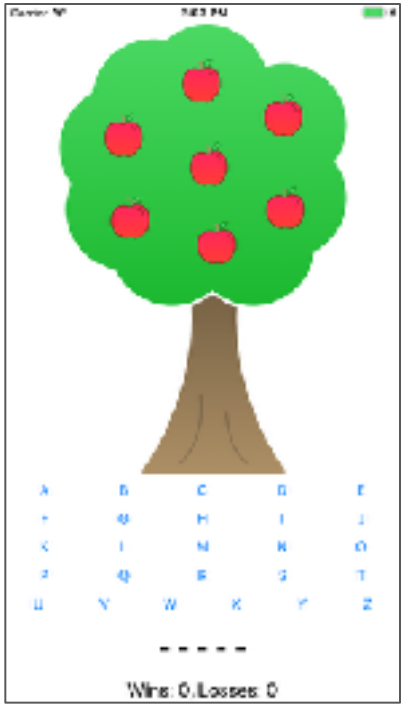
- The game adds the letter to the guessedLetters collection.
- If the word doesn't contain the letter, the game updates the remainingAttempts by subtracting one.
- Update formattedWord to show the guessedLetters.

Challenge 8.1



- The player hits a button, and the IBAction calls the game passing the guessed letter.
- The IBAction sets the button to disabled.
- The IBAction calls updateUI.

Challenge 8.2

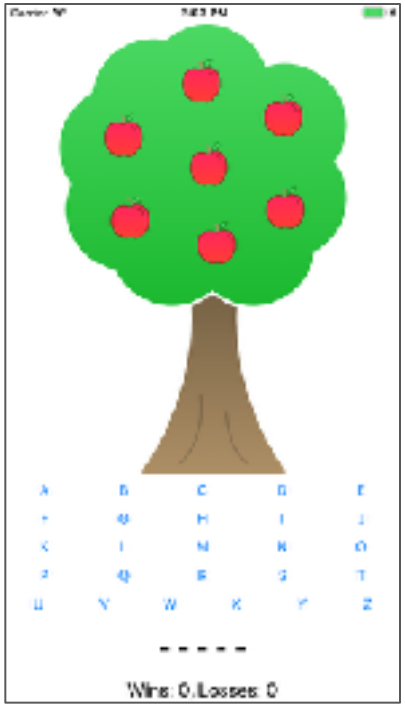


- The game adds the letter to the guessedLetters collection.
- If the word doesn't contain the letter, the game updates the remainingAttempts by subtracting one.
- Update formattedWord to show the guessedLetters.

Challenge 9:
Winning or Losing a round

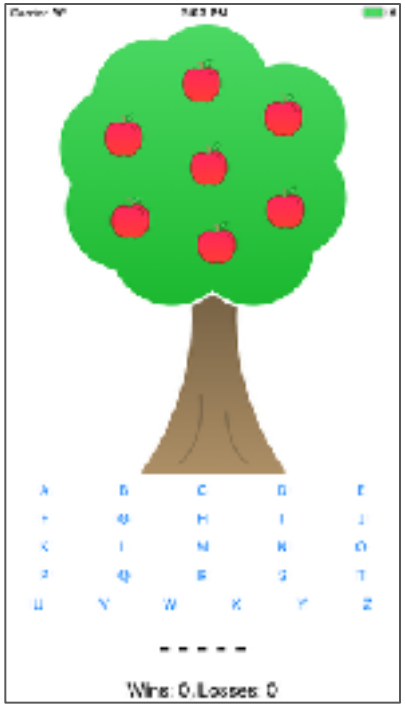
What happens when a player wins a round?

Solution



- The game calls *updateState*.
- *updateState* verifies if the player won the round -> updates wins by one
- *updateState* verifies if the player lost the round -> updates looses by one
- Update the IBAction code to decide if it starts a new round or just updates the UI.

Challenge 9.1



- *updateState* verifies if the player won the round -> updates wins by one
- *updateState* verifies if the player lost the round -> updates looses by one
- Start a new round

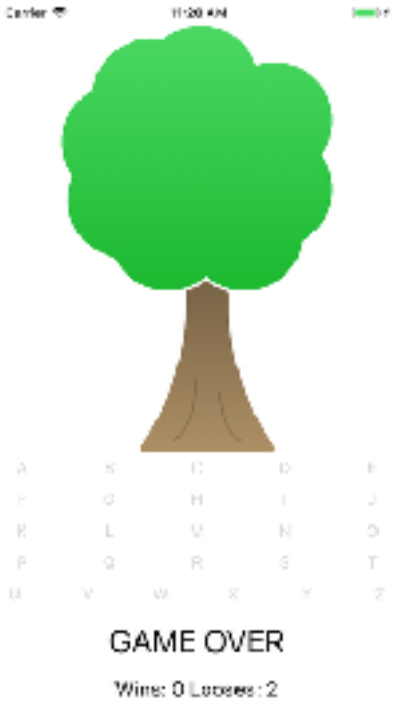
What happens if there are no more words?

Challenge 11:
Game over

What happens when there are no more words to guess?

(Think what you can do to tell the Player that the game is over.)

Solution



- In the ViewController verify if the game is over before updating the UI.
- If it’s over set the UI to look like the Image.

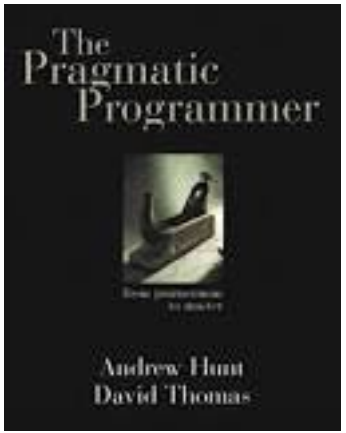
What have we learned so far?

- ***IBOutlets:***
  - They connect controls such as Buttons and Labels to the code file
- ***IBActions:***
  - They execute code when an event occurs in the control that has the action
- ***Refactoring:***
  - After writing a piece of code, always take time to refactoring -> make your code readable by others

What have we learned so far?

- ***Single Responsibility Principle:***
  - A class should only have one reason to change

Resources



Books:

The Pragmatic Programmer

Summary

Resources

RayWenderlich.com

HackingWithSwift.com

Apple Swift Book

Human Interface Guidelines iOS