

App Design Tutorial

by Phuc Nguyen

1. Meet Old Friends! Pencil and Paper

**“‘Different’ and ‘new’ is relatively easy.
Doing something that's genuinely better is very hard.”**

- Jonathan Ive. SVP Design of Apple Inc.



What is your app?

1. With paper and pencil, jot down what your app does
2. Write out features of your app
3. Cross out any unnecessary features
4. Do a little research to see what is already out there
5. Sketch out rough interface design to visualize



Quiz 101

An App Example

1. What does Quiz do?

Show questions and make users guess answers

2. Jot Down App Features

- Show a question
- Show an answer
- Display ??? while users are thinking
- Play a song while users are waiting
- Make a sound when users get it right
- Animate questions and answers so they look great

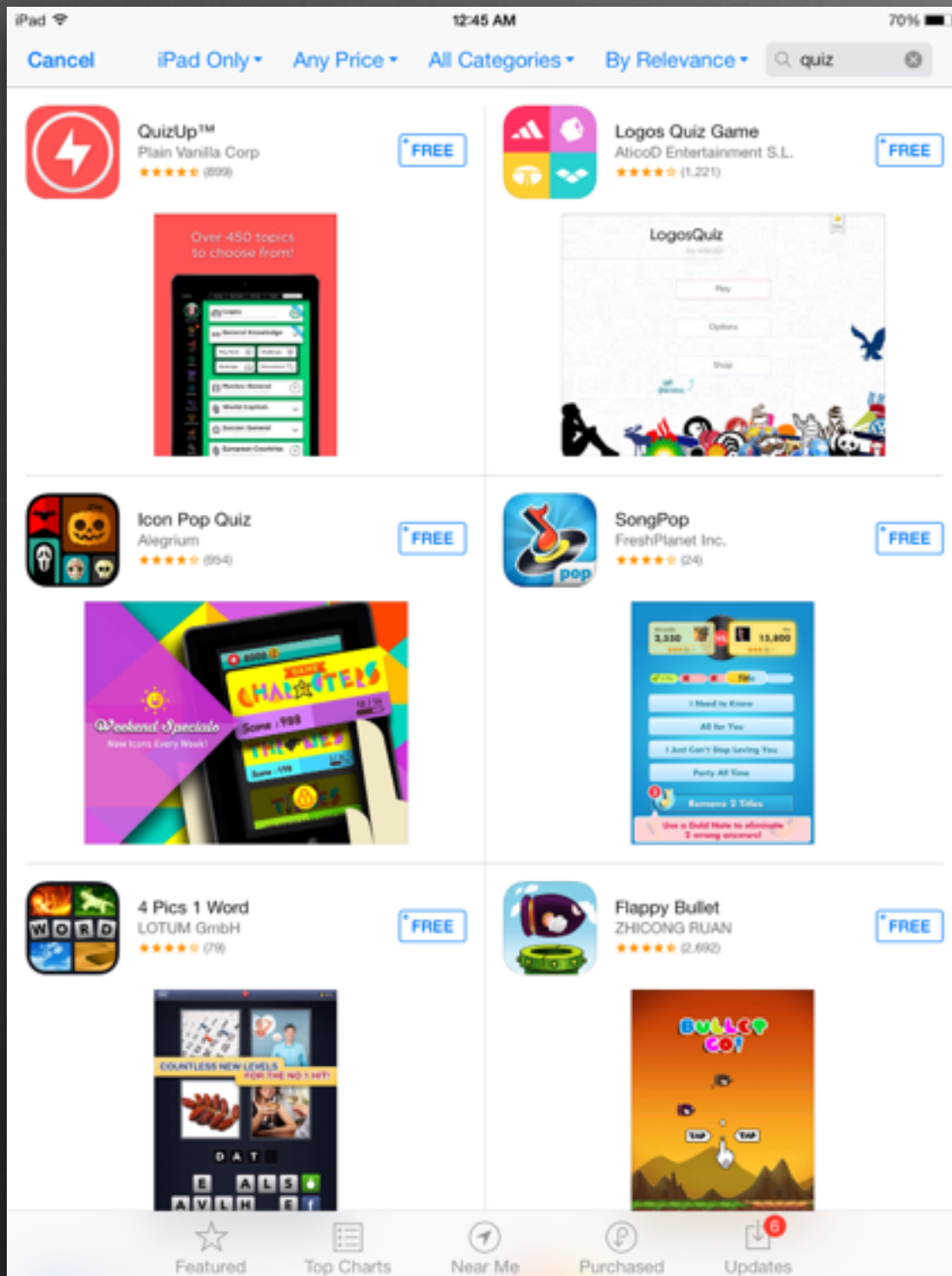
3. Cross out unnecessary features

- Show a question
- Show an answer
- Display ??? while users are thinking
- ~~Play a song while users are waiting~~
- Make a sound when users get it right
- ~~Animate questions and answers so they look great~~



Maybe

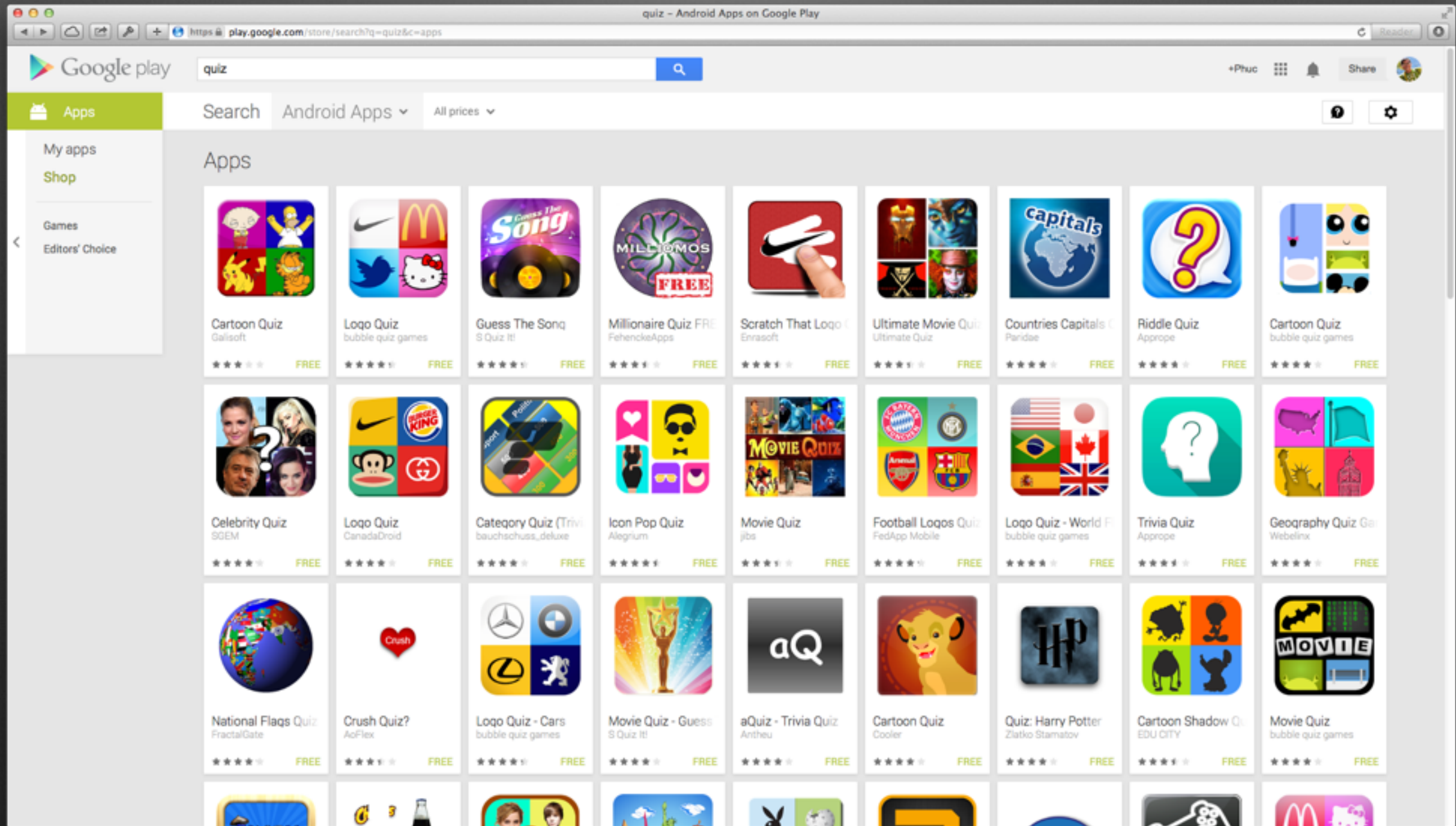
4. Do Research



iOS App Store



Google Play - Android Store



5. Sketch out rough interface

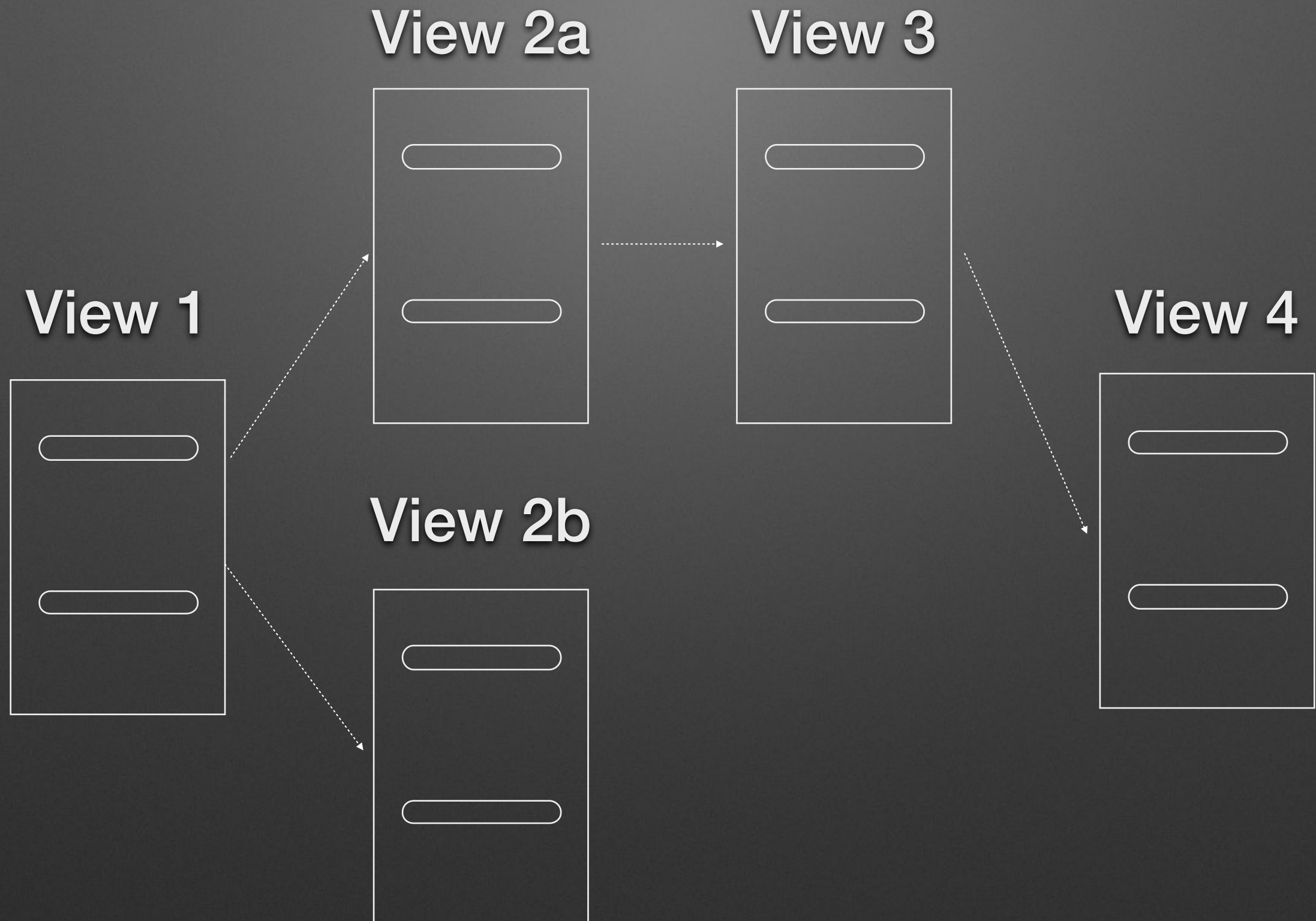
Question here

Show Question

Answer here

Show Answer

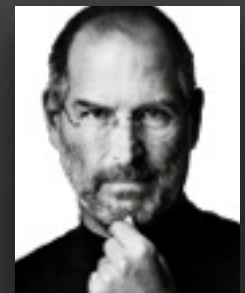
5. Storyboarding



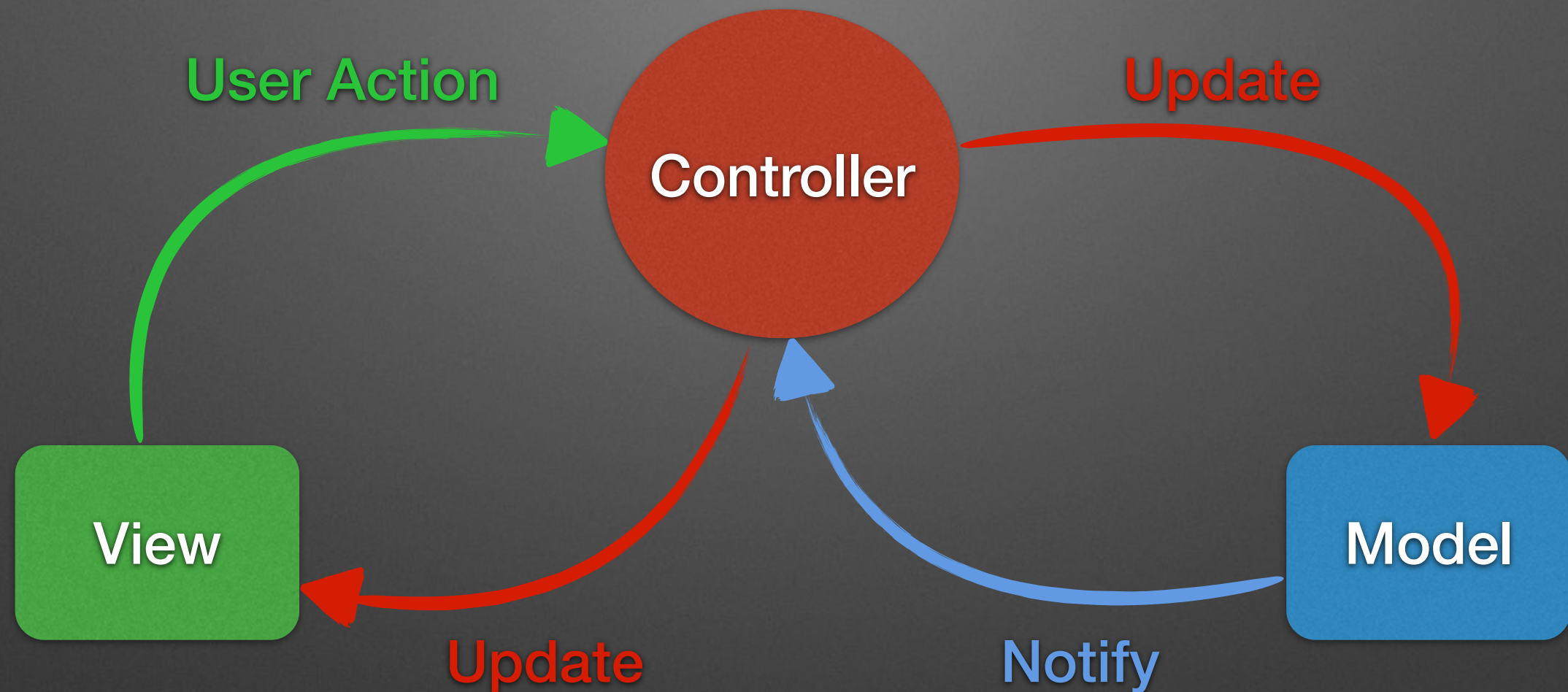
2. Software Fashion - MVC

**“Some people think design means how it looks.
If you dig deeper, it’s really how it works.”**

- Steve Jobs. Co-founder and CEO of Apple Inc.

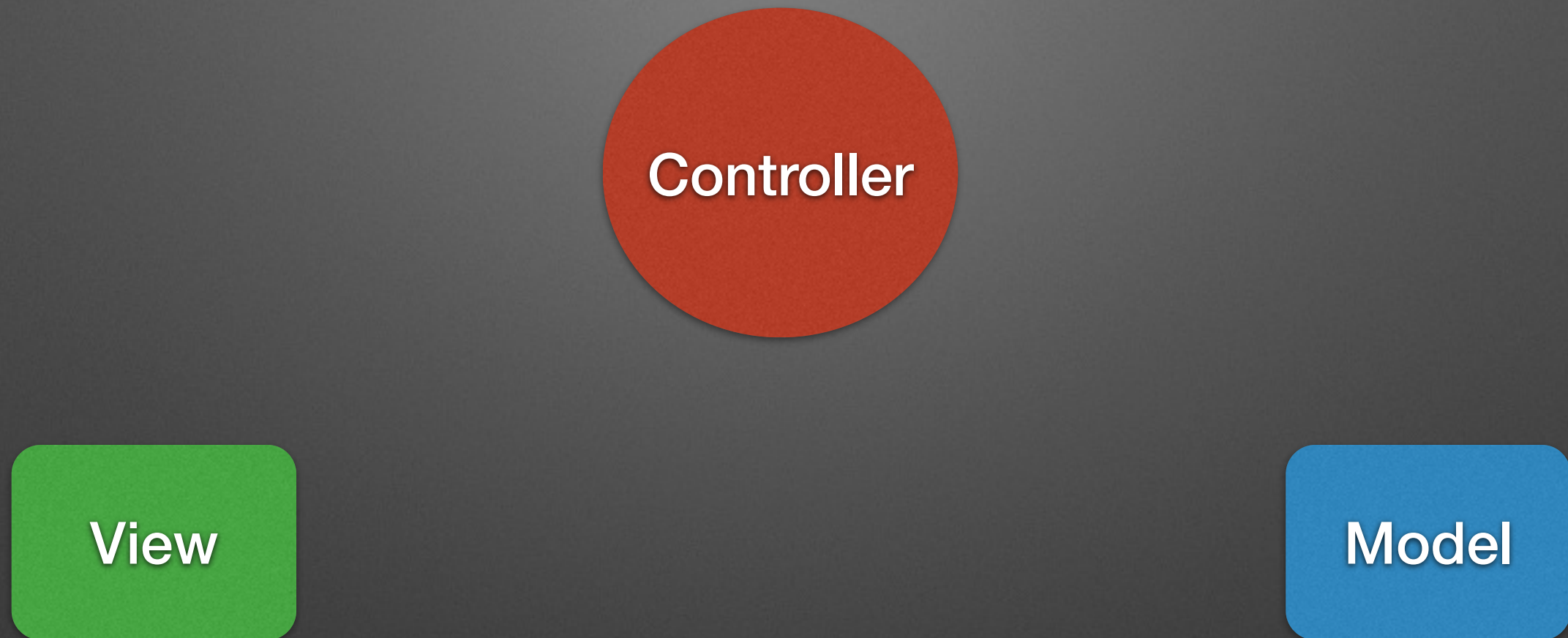


Model-View-Controller



Design Pattern

Model-View-Controller



Model-View-Controller



Controller

Controller Object learns about changes in View and updates Model and vice versa



View

View Objects can be seen by users. They know how to draw themselves and respond to user actions



Model

Model Objects encapsulate data and define methods to manipulate data



Quiz 101

An App Example



View Controller

Controller

- Show Question Button
- Show Answer Button

View

- List of questions
- List of answers

Model

3. The Making Of A Software Designer

“By failing to prepare, you are preparing to fail.”

– *Benjamin Franklin.*



Use Cases

- A **use case** is a method to identify program requirements
- A **requirement** is usually **a goal** that users want to achieve
- Use **NOUN + VERB** to describe an **interaction** between users and a program



Use Cases

USER

QUIZ PROGRAM

Click	
	Display
	Display ???
Click	
	Display
	If the user is right, with a If the user is wrong, subtract a point

Identify Classes & Objects

- **Look for NOUNS** to identify a set of classes
- Pay attention to **classes that are required** for other classes to work properly
- Example classes: Foundational classes, system interfaces, users, events, interactions

Identify responsibilities of classes & objects

- **Look for Verbs** in the requirements
- **1** responsibility = **1** class

Relationships between classes

- **Inheritance, is-a** relationship.
 - Ex: Sentence (superclass) -> Question (subclass)
- **Aggregation, has-a** relationship
 - Ex: College -> Department -> Faculty, Student
- **Dependency, depend-on** relationship
 - Ex: Student -> Courses, Teachers, College

QUIZ 101 Objects

In this case, we use Objects from System classes

- Questions

- NSArray (in iOS)
- ArrayList (in Android)

- Answers

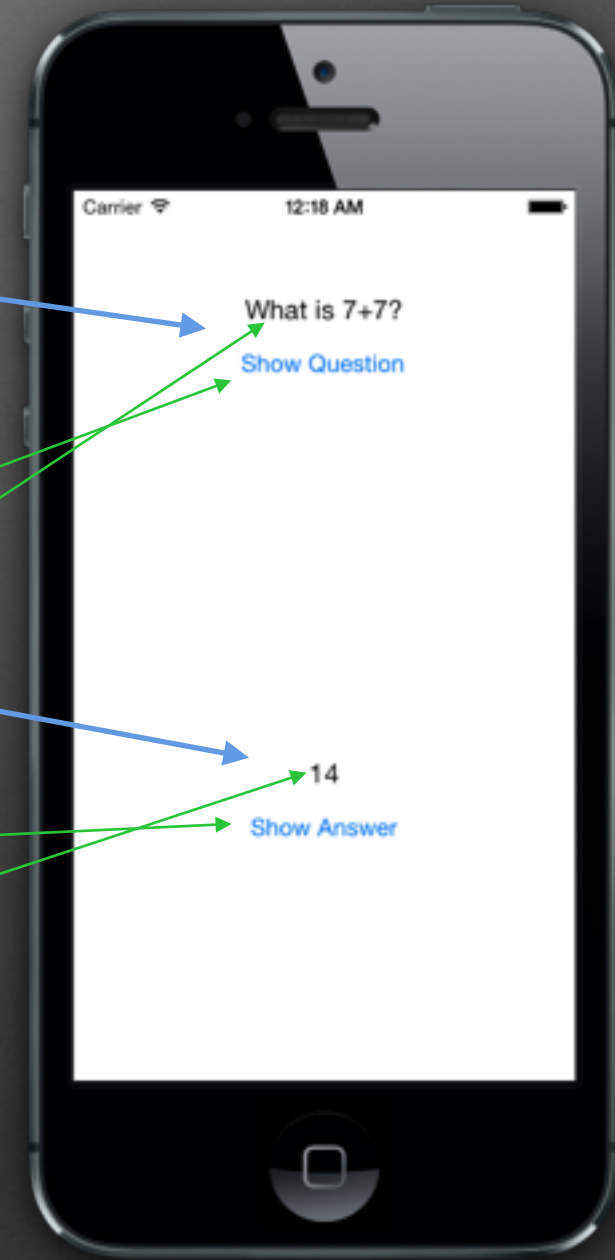
- NSArray (in iOS)
- ArrayList (in Android)

- 2 buttons

- UIButton (in iOS)
- Button (in Android)

- 2 labels

- UIButton (in iOS)
- Button (in Android)





UIViewController or
Activity

Controller

- UIButton or Button
- UIButton or Button

View

- NSArray or ArrayList
- NSArray or ArrayList

Model

That's it! Time to code

Click [here](#) to return to Tutorials page