Nicholas Miller
University of Florida

iOS Topic Questions (67 Questions)

Views (5 questions)

- What properties of a view can be animated?

  Frame, bounds, center, transform, alpha, background color, and content stretch.

- What is the difference between frame and bounds?

  Frame is the location and size in the superview's coordinate system and bounds is the location and size in its own coordinate system.

- How are the typical values used for RGB different than the values used when creating a CGColor (using RGB parameters) and how do you account for this?

  Normally RGB values range from 0-255 each but when you are creating a CGColor the range for each is 0-1. These also have to be defined as type CGFloat. So if we have a generic RGB we have to divide each by 255.

- What view property must be true in order for gesture recognizers to work?

  User interaction must be enabled.

- Give 1-2 examples of when it'd be a good idea to build a custom view.

  When you want to build a particular object (like Tinder's cards) because there is nothing like that in the default object library.

  When you want to add your own animations and flow or create a unique user experience.

Nicholas Miller
University of Florida

Classes (4 Questions)

- What does it mean when a class subclasses another class?

  It inherits all the properties and methods of the parent class. Usually it will have methods of its own that the parent does not have. Inheritance is one of the object-oriented fundamentals (because we can build a base model and add extra features for ease of use).

- Value Types

  For a value type each instance keeps a unique copy of its data, usually defined as a struct, enum, or tuple. Instances are done by declaring a new variable (var x = object).

- Give an example of when we would use “type casting”.

  When we want to convert one primitive type to another. An example is casting a generic tableview cell as a custom tableview cell you have defined.

- Why is casting a UIView to a UIImageView considered “down casting”

  Because UIView is the generic class (base class) and UIImageView is a subclass of UIView (derived class). So we are chopping UIView down to its subclass UIImageView.

Nicholas Miller
University of Florida

View Controllers (5 Questions)

- What are the view controller lifecycle methods?

  They are: viewDidLoad, viewWillAppear, viewDidAppear, viewWillDisappear, viewDidDisappear, and viewDidUnload.

- What is the difference between viewDidLoad & viewDidAppear?

  viewDidLoad is the first thing called as soon as the particular view is launching. It is what happens when the view is ready to go.

  viewDidAppear is called after viewDidLoad and viewWillAppear. This is the method that runs once the view is on the screen (usually an animation goes in here so that it waits until the user can see the view before it runs).

- When adding a child view controller to a container view controller, what are the view controller lifecycle methods you need to call on the child view controller? What about when removing?

  Ultimately every view controller needs to run all of its lifecycle methods.

  Adding a child view controller:

  In order to implement the loading and appearing lifecycle methods we only need to call two methods from the container view. We have to call addChildViewController before adding a child view controller's view as a subview of our view. To notify the child view controller that we finished adding its view to the view hierarchy we have to call didMoveToParentViewController afterwards.

  Removing a child view controller:

  In order to implement the disappearing and unloading lifecycle methods we only need to call two methods from the container view. We first have to notify it we will do this by calling willMoveToParentViewController with nil as the new parent view controller. After we are done removing the child view we have to also remove its view controller from our set of child view controllers by calling its removeFromParentViewController method.

Nicholas Miller
University of Florida

- What is a container view controller? When is it useful?

  Container view controllers are a way to combine the content from multiple view controllers into a single user interface. Container view controllers are most often used to facilitate navigation and to create new user interface types based on existing content. Examples of container view controllers in UIKit include UINavigationController, UITabBarController, and UISplitViewController, all of which facilitate navigation between different parts of your user interface. These allow for better user experience and ease of navigation.

- When should you add a new View Controller to your app?

  Usually you add a new view controller for every "page" in your app. While there are exceptions to improve user experience (a sub view or other type of view to do something quick) they are the center of every page in your app. These view controllers should be different in features and functions.

Nicholas Miller
University of Florida

Navigation (6 Questions)

- What method do we use to pass data to another view controller that we are navigating to?

  If it is done using a segue then we use the method prepareForSegue.

  If not then we make an instance of the class set the public data property we want then present the view controller.

- How do we pass data back when we are returning from another view controller?

  We can pass data back most efficiently by using a closure. A less efficient method would be to set a nsuserdefualt and then grab it after when we go back to our originally view in viewWillAppear.

- What use cases are best suited for show (push) navigation?

  Usually you would use push navigation for any major page in the app or if you want them to be able to move back and forth between pages with ease (back button in the navigation bar).

- What must we include any time we want to use show (push) navigation?

  A navigation controller must be embedded otherwise the app will crash when we try to push the new view controller.

- What use cases are best suited for modal presentation?

  Usually you would use modal presentation for any view controller that has a "done" button. By that I mean any view controller that asks for quick information or is a quick one time page that acts a user extension of the page you are on (page to invite friends and then when you hit done it leaves). It is also used between a login screen and the home screen because you do not want them to be able to go back to the login screen and sign in again.

- When would you want to use a custom view controller transition?

  You would use a custom view controller transition whenever you want to build a unique user experience and design.

Nicholas Miller
University of Florida

Optionals (3 Questions)

- What is a Swift optional and how is it notated?

  An optional is declared with a question mark (var x: String?). It is automatically set to nil in this instance. The importance of optionals is that they state the variable may or may not have a value. So before we unwrap it and assign it to something we can safely check if the variable is not nil.

- Why is force unwrapping optionals dangerous?

  This is dangerous because if the optional value is nil then we are assigning something a nil value and that will crash the app ("Unexpectedly found nil while unwrapping an optional value"). So if we check to make sure it is not nil first then we can force unwrap it inside the if statement, although it is safer to optionally unwrap it (the way to do that is by saying if let x = x. This checks x and if it is not nil it will assign it a value).

- What is optional chaining and when / why would you use this technique?

  Optional chaining is a process for querying and calling properties, methods, and subscripts on an optional that might currently be nil. If the optional contains a value, the property, method, or subscript call succeeds; if the optional is nil, the property, method, or subscript call returns nil. Multiple queries can be chained together, and the entire chain fails gracefully if any link in the chain is nil.

Nicholas Miller
University of Florida

Storyboard connections (4 Questions)

- What is an outlet and when is it used?

  An outlet is a connection from a particular object in a xib, storyboard, or other prototyping file to a reference variable. This variable has all the attributes of that object's type and you can control the object by changing the value of attributes of the reference variable.

- What is an Action and when is it used? Or What is target-action and when is it used?

  An action (linked) is a method that is run when a particular event happens to an object in a xib, storyboard, or other prototyping file (examples are when the value of the object changes or the object is touched).

  A target-action is the same as above except it is a programmatic link instead of a manual link. This is usually when you create an object programmatically you can set a target action for a particular event.

- When creating Actions and Outlets, why is it advisable to drag from the document outline rather than from within the View Controller in the Storyboard?

  You want to drag from the document outline because if you drag from the view controller in storyboard you could be grabbing any object that is above the particular object you want in the view hierarchy. So it is safest to drag from the document outline because you can be sure you are grabbing and linking the right thing.

- How do we associate a Swift File with a View Controller in the Storyboard?

  We can create a new swift file (most common is a cocoa touch class that inherits the property of the view controller class) and then in storyboard click on the identity inspector in the right view pane and then set the class to that proper custom view controller swift file.

Nicholas Miller
University of Florida

Debugging (2 Questions)

- When you run into the exception ... this class is not key value coding-compliant for the key ..., what is the first thing to check?

  First I would make sure that I do not have any broken outlet or action connections because if there is an outlet reference in storyboard but I removed the code for it then the app would crash.

  Second I would make sure that all of the view controllers in the storyboard are set to the proper classes in the indent inspector.

- List at least 2 different reasons why you might encounter the error, “unexpectedly found nil while unwrapping an optional value ”, and explain how you might go about debugging this error?

  One reason is that an optional variable never received a value and you force unwrapped it. Another reason you could get that error is if you are trying to reference a particular class or file that does not exist.

  Examples why it never received a value: database query returned nil, database object attribute is nil after database query for object, attribute in NSDictionary is nil, and typed the wrong table cell reuse identifier.

Nicholas Miller
University of Florida

Scroll Views, Table Views Collection Views (12 Questions)

- When using a UIScrollView, which method in UIScrollViewDelegate would you use to detect when UIScrollView has finished scrolling.

  The method scrollViewDidEndDragging:willDecelerate tells the delegate when dragging ended in the scroll view.

  The method scrollViewDidEndDecelerating tells the delegate that the scroll view has ended decelerating the scrolling movement.

- How might you figure out if the user had scrolled to the bottom of a scroll view?

  You can check the offset. So if the scroll view's content reaches a certain y position (the max y position you set for the scroll view) then you know they hit the bottom as soon as they are greater than or equal to that y position.

- What is cell recycling and why is it important?

  Recycling cells is extremely important because it is more efficient. Recycling the cell is using the old cells and not making a new ones. They change the instance based on the indexpath as the user scrolls.

- How does the Table View know which prototype cell to reuse?

  As you scroll it returns the indexpath for each cell in the current view. This then updates all the reusable cells with the correct content based on those indexpaths.

- How do we access properties in our custom cell class from within the cellForRowAtIndexPath method?

  We crate an instance of that cell and then assign values to its properties.

  (let cell = tableview. dequeueReusableCellWithIdentifier("id") as? UITableViewCell)

  Instead of UITableViewCell we can user our custom cell class. Then we can change the properties of that cell by doing cell.property = value.

- Where does all the logic associated with a particular cell go?

  All of the logic associated with what should be in the cell should be inside the cellForRowAtIndexPath method.

- Why do we avoid naming Image Views and Labels we create in a prototype cell, “imageView” and “label” respectively?

  We avoid these variable names because it can confuse the compiler. The compiler reserves “label” and “imageView” for the specific object type because these are system definitions. If you assign these names then the system will change the actual classes and not the instance variables.

- What is a protocol?

  A protocol is used to declare methods and properties that are independent of any specific class. Protocols can include declarations both instance methods and class methods, as well as properties. Common protocols are data sources (a very popular one is the tableview data source).

- What is UITableViewDataSource and when do you use it?

  The UITableViewDataSource protocol is adopted by an object that mediates the application’s data model for a UITableView object. The data source provides the table-view object with the information it needs to construct and modify a table view. You use it whenever you want to construct a tableview cell’s based on a set of data.

- What is UITableViewDelegate and when do you implement it?

  The UITableViewDelegate protocol provides answers to requests about the layout of the table and about actions the user performs on the tableview. Layout methods include the tableview asking about the height of rows, headers, and footers, what the buttons should look like, etc. Action methods include the user selecting a row and beginning and ending the editing of a row. You implement it so that you can construct the tableview with the right look and have a listener when the user does something to it.

- In which situations is it useful to use a CollectionView vs a TableView?

  These two views are extremely similar. One of the only differences is that the collection view allows cells for multiple columns. So if you have a lot of items to display then you

probably want to show multiple items in one row so the user does not have to scroll too much to see a lot of items. If you have a lot of information you want the user to be aware of then you probably want to do a tableview because you can display a good amount of information about a particular item so the user does not have to click on each cell to read more about the item often.

- If you have an image-intensive iPhone app and want to decrease the lag time when loading images from disk or going through the network, what are some potential approaches to solving this issue?

  You can load the images once and then cache them. Then every time you open the image-intensive section it pulls from the cached memory first before reloading everything. So that way the user cannot see much of a difference except for some images updating.

  The other method is to load a lower resolution of the images and then after that fade in the higher resolutions.

  You could have a loading animation so that the user knows it is loading.

Nicholas Miller
University of Florida

Cocoa Pods (2 Questions)

- What are Cocoa Pods? What's an example of when it can be helpful to use a pod?

  A Cocoa Pod is a third party framework, library, or extension that builds on the existing iOS frameworks and toolsets. They are used for easily accomplishing a task that would normally take more work and coding to do. They also allow for extended and more complicated uikit objects improving the user interface and user experience. One of the most common Cocoa Pods is AFNetworking (allows for all sorts of great url loading and network requests).

- What are the advantages/disadvantages to including Pods in your gitignore file?

  Advantages:

  The source control repo will be smaller and take up less space.

  There won't be any conflicts to deal with when performing source control operations, such as merging branches with different Pod versions ***if no one touches the podfiles (this is commonly an issue)****.*

  As long as the sources for all Pods are available, CocoaPods is generally able to recreate the same installation.

  Disadvantages:

  Pods are extremely common for merge conflicts.

  If the pod is in the podfile and not installed then you can get an error.

  Dependencies on a pod that was removed or not installed will result in an app failure. The other issue is that anyone can mess with the podfile and accidently change it.

Nicholas Miller
University of Florida

App Architecture (3 Questions)

- Describe the differing responsibilities of each component of the Model-View-Controller design pattern.

    Model

    Model objects encapsulate the data specific to an application and define the logic and computation that manipulate and process that data. A model object can have to-one and to-many relationships with other model objects.

    View

    A view object is an object in an application that users can see. A view object knows how to draw itself and can respond to user actions. A major purpose of view objects is to display data from the application's model objects and to enable the editing of that data.

    Controller

    A controller object acts as an intermediary between one or more of an application's view objects and one or more of its model objects. Controller objects are thus a conduit through which view objects learn about changes in model objects and vice versa. Controller objects can also perform setup and coordinating tasks for an application and manage the life cycles of other objects.

- Provide an example of MVC implementation.

    The controller tells the view what to have. The view shows the uibutton and the uilabel. The view notifies the controller when the button is touched. The controller calls the model connected to that button. The model then tells the controller what the label text change is. The controller then tells the view what the label text should be. The process repeats.

- What's a singleton? Give an example of when you would want to use it. Or Describe a use case where using a shared instance make the most sense?

    A singleton class returns the same instance no matter how many times an application requests it. A typical class permits callers to create as many instances of the class as they want, whereas with a singleton class, there can be only one instance of the class

per process. A singleton object provides a global point of access to the resources of its class. Singletons are used in situations where this single point of control is desirable, such as with classes that offer some general service or resource. We saw this with our twitter app. When we make a shared instance everything can only use that one instance.

Nicholas Miller
University of Florida

Auto Layout (4 Questions)

- What is the difference between Auto Layout and Auto Resizing in iOS?

    Auto Layout (The preferred and newer method).

        Depends on the constraints of views. A constraint (an instance of NSLayoutConstraint) is much more sophisticated than the "autoresizingMask" it's a full-fledged object with numeric values, and can describe a relationship between any two views (not just a subview and its superview).

        This can change the position and size of the object.

    Auto Resizing (The old way).

        A matter of conceptually assigning a subview "springs and struts." A spring can stretch; a strut can't. Springs and struts can be assigned internally or externally. Thus you can specify (using internal springs and struts) whether and how the view can be resized, and (using external springs and struts) whether and how the view can be repositioned.

        This usually changes the size of the object.

- What is the difference between content hugging priority & content compression resistance priority?

    Hugging Priority is when the object and content does not want to grow. Content Compression Resistance Priority is when the object and content does not want to shrink.

- What is intrinsic content size?

    Intrinsic content size is the minimum space needed to express the full view content without squeezing or clipping that data.

- How would you animate views with AutoLayout?

    You need to set a constant constraint between the views. You should set one of two views to have a higher priority so that as you change it the other view is forced to change instead of the view you are changing.

Nicholas Miller
University of Florida

APIs (4 Questions)

- What does OAuth allow for?

  OAuth is an open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications.

- When would you want to use Parse?

  Whenever you want a database (with MongoDB as the backend) and the developer tools that Parse provides. This happens whenever you want your data to appear on more than one device. It also allows for push notifications.

- What data type do we typically get back from APIs?

  The data type we usually get back from APIs is a JSON File (set as a NSDictionary).

- What method can we use to dig into nested dictionaries?

  The method is called ”Object for Key”.

  If your NSDictionary is called data and does not have this method then you can call data[“attribute_1”] to get the NSArray or value related to that attribute name (“attribute_1”).

Nicholas Miller
University of Florida

General Application (3 Questions)

- What does the NSUserDefaults class provide to you as the app developer?

  NSUserDefaults allows you to store certain values based on a key into the application's memory. So you can retrieve this value based on that key at any point in the application.

- List & explain the different types of iOS Application States.

  Not Running- The application has not been launched or was running but was terminated by the system.

  Inactive- The app is running in the foreground but is currently not receiving events. An app usually stays in this state only briefly as it transitions to a different state.

  Active- The app is running in the foreground and is receiving events. This is the normal mode for foreground apps.

  Background- The app is in the background and executing code. Most apps enter this state briefly on their way to being suspended.

  Suspended- The app is in the background but is not executing code. The system moves apps to this state automatically and does not notify them before doing so. While suspended, an app remains in memory but does not execute any code. In addition, an app being launched directly into the background enters this state instead of the inactive state.

- What is the very first method that is called when an app launches and what file is it found in?

  The first method that is called when the application launches application:willFinishLaunchingWithOptions and the second method called is application:didFinishLaunchingWithOptions and it is found in the AppDelegate.

Nicholas Miller
University of Florida

Closures (4 Questions)

- What is a closure?

  Closures are self-contained blocks of functionality that can be passed around and used in your code. Closures can capture and store references to any constants and variables from the context in which they are defined. This is known as closing over those constants and variables. Swift handles all of the memory management of capturing for you.

- How are closures helpful when passing data obtained from a network request?

  We can wait for the completion block of the network request (a closure). Once it is complete we can do the code we need to accomplish. We have to wait for the completion block because network requests are asynchronous.

- Why do we need to use “self” inside the body of closures?

  We use self to get the global variable and not the local variable inside the closure.

  We also pass self into the closure like below. This is so that we can have a reference to the base view controller class.

  ```
  ViewController.value = {[weak self]

      (data: [Type]) in

      self.value = value

  }
  ```

- Why does the UIView.animateWithDuration method utilize a closure?

  The method animateWithDuration utilizes a closure because it needs to know how to animate the object and what to do once the animation is complete.

Nicholas Miller
University of Florida

Design (4 Questions)

- What are 1-2 examples of a good use of animation in some of the apps you use?

  Uber has a really cool splash screen while the app loads from launch.

  The default weather app makes great use of animations. The screen matches the current weather (so you can see the rain, lightning, sun, snow, and fog).

- When would you use AVFoundation instead of UIImagePickerController?

  AVFoundation is used for audio, video, and animation. Its sub classes are Core Audio, Core Media, and Core Animation.

  UIImagePickerController is used to take an image using the phone's camera or select an image from the phone's photo library.

- Choose an app that you use often and figure out all the different ways it leverages the device frameworks (i.e. camera, push notifications, maps, location, etc).

  Flixster (Movie App)

    Flixster calls their movie API in viewDidLoad and returns the movies based on a parameter. Segmentedcontrols allow the user to change the movies returned. It then takes the results and sets the contents of the their custom table view cell. It loads a lower resolution image first and the fades in the higher resolution image. If you touch a table view cell it passes the particular movie to the detail view controller by the prepareForSegue method. This then presents the detail of the movie in its own page. You can watch a video by clicking the video button. This uses AFNetowrking to load the video and audio and present it. It uses core location to find nearby theatres based on core location. The information in the detail view controller has a scroll view so you can see a lot of information without having to go to a new view controller. It is a pretty basic idea and they execute it very well.

- What's the difference between using storyboards, xib's, and programmatic views in your app? What are the pros and cons of each one?

  Storyboard

Pros: Can easily prototype the application from start to finish.

Cons: A lot of merge conflicts when multiple people are working on storyboard.

Xibs

Pros: Allows for easy prototyping for each page or view in your application.

Cons: Can be tricky with autolayout and setting the right sizes for table view cells

Programmatic

Pros: Very little merge conflicts and no outlet or action reference errors.

Cons: Takes much more time to create each view controller and harder to keep track of all of the object button references and methods.

Nicholas Miller
University of Florida

Networking (2 Questions)

- What is the difference between NSURLConnection and NSURLSession?

  An NSURLConnection object lets you load the contents of a URL by providing a URL request object. The interface for NSURLConnection is sparse, providing only the controls to start and cancel asynchronous loads of a URL request.

  The NSURLSession class and related classes provide an API for downloading content. This API provides a rich set of delegate methods for supporting authentication and gives your app the ability to perform background downloads when your app is not running or, in iOS, while your app is suspended.

- Now that there is NSURLSession, do we still need AFNetworking? Why or why not?

  NSURLSession is a replacement for NSURLConnection introduced in iOS 7. AFNetworking extends NSURLSession to pave over some of the rough spots, and maximize its usefulness. So we still need AFNetworking.