View Controller Life Cycle

G4ENG

01 AUG 19

Sorry

Agenda

- Life Cycle
- Method

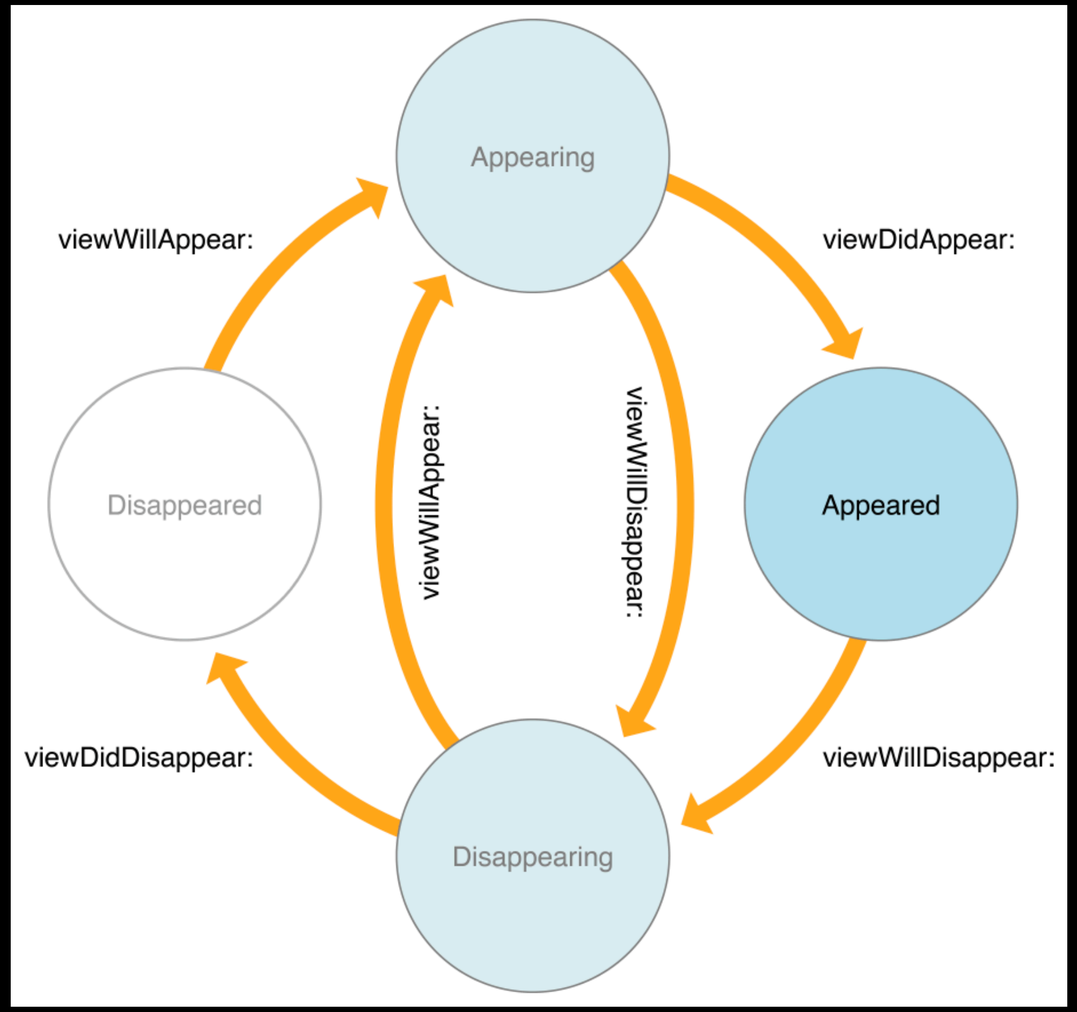
Life Cycle

- So far, the FoodTracker app has a single scene, whose user interface is managed by a single view controller. As you build more complex apps, you'll create more scenes, and will need to manage loading and unloading views as they're moved on and off the screen.
- An object of the UIViewController class (and its subclasses) comes with a set of methods that manage its view hierarchy. iOS automatically calls these methods at appropriate times when a view controller transitions between states. When you create a view controller subclass (like the ViewController class you've been working with), it inherits the methods defined in UIViewController and lets you add your own custom behavior for each method. It's important to understand when the system calls these methods, so you can set up or tear down the views you're displaying at the appropriate step in the process—something you'll need to do later in the lessons.

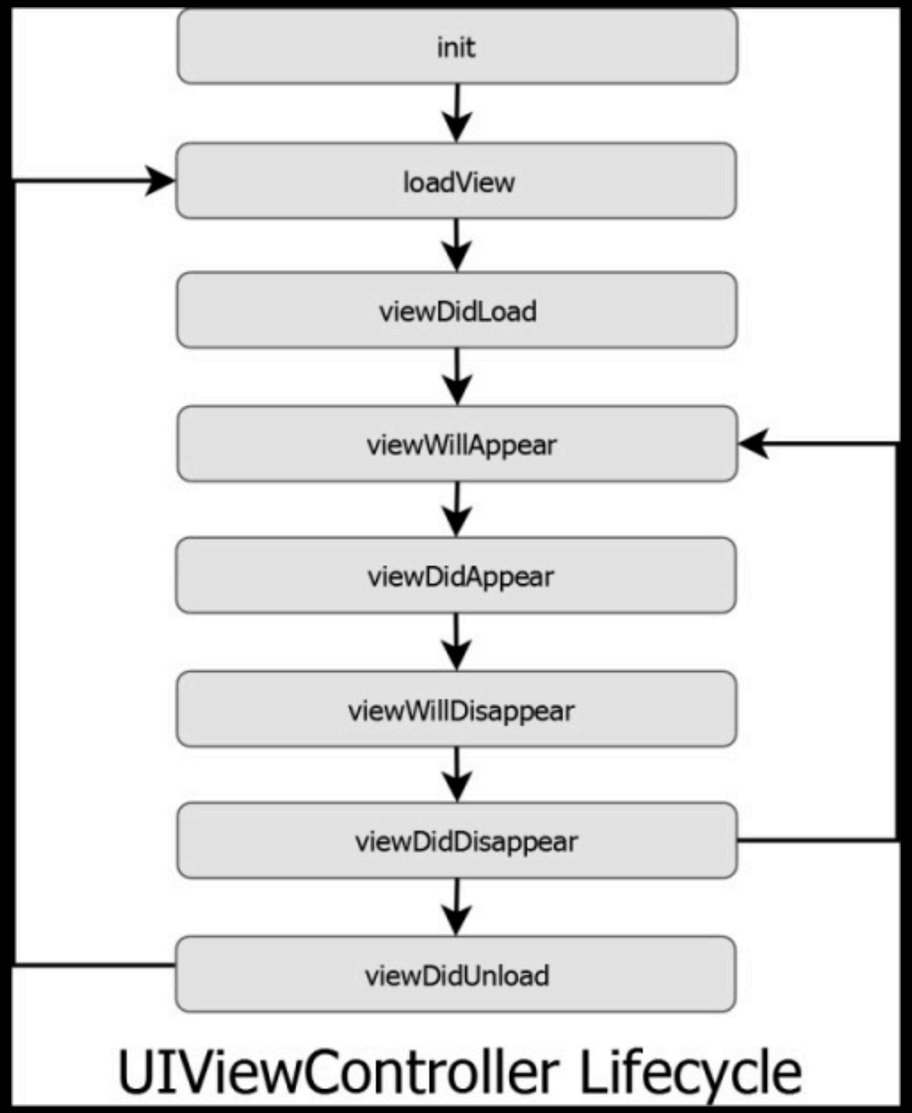
Life Cycle

- 지금까지 FoodTracker 앱에는 단일 장면이 있으며이 장면의 사용자 인터페이스는 단일보기 컨트롤러로 관리됩니다. 보다 복잡한 응용 프로그램을 만들면서 더 많은 장면을 만들고 화면을 이동하거나 화면 밖으로 이동하면보기로드 및 언로드를 관리해야합니다.
- UIViewController 클래스 (및 해당 하위 클래스)의 객체에는 뷰 계층 구조를 관리하는 일련의 메서드가 있습니다. iOS는보기 컨트롤러가 상태간에 전환 할 때 적절한 시간에 자동으로이 메소드를 호출합니다. 뷰 컨트롤러 하위 클래스 (작업 한 ViewController 클래스와 같은)를 만들면 UIViewController에 정의 된 메서드를 상속하며 각 메서드에 대한 사용자 지정 동작을 추가 할 수 있습니다. 시스템이 이러한 메소드를 호출 할 때를 이해하는 것이 중요합니다. 따라서 프로세스의 적절한 단계에서 표시하고있는 뷰를 설정하거나 해체 할 수 있습니다.이 과정은 나중에 강의에서 수행해야합니다.

Life Cycle


Appearing
viewWillAppear:
viewDidAppear:
Disappeared
viewWillAppear:
viewWillDisappear:
Appeared
viewDidDisappear:
viewWillDisappear:
Disappearing

Life Cycle



출처: https://zeddios.tistory.com/43 [ZeddiOS]

Method

- `viewDidLoad()`—Called when the view controller's content view (the top of its view hierarchy) is created and loaded from a storyboard. The view controller's outlets are guaranteed to have valid values by the time this method is called. Use this method to perform any additional setup required by your view controller.
- Typically, iOS calls `viewDidLoad()` only once, when its content view is first created; however, the content view is not necessarily created when the controller is first instantiated. Instead, it is lazily created the first time the system or any code accesses the controller's `view` property.

Method

- `viewWillAppear()`—Called just before the view controller's content view is added to the app's view hierarchy. Use this method to trigger any operations that need to occur before the content view is presented onscreen. Despite the name, just because the system calls this method, it does not guarantee that the content view will become visible. The view may be obscured by other views or hidden. This method simply indicates that the content view is about to be added to the app's view hierarchy.

Method

- `viewDidAppear()`—Called just after the view controller's content view has been added to the app's view hierarchy. Use this method to trigger any operations that need to occur as soon as the view is presented onscreen, such as fetching data or showing an animation. Despite the name, just because the system calls this method, it does not guarantee that the content view is visible. The view may be obscured by other views or hidden. This method simply indicates that the content view has been added to the app's view hierarchy.

Method

- `viewWillDisappear()`—Called just before the view controller's content view is removed from the app's view hierarchy. Use this method to perform cleanup tasks like committing changes or resigning the first responder status. Despite the name, the system does not call this method just because the content view will be hidden or obscured. This method is only called when the content view is about to be removed from the app's view hierarchy.

Method

- `viewDidDisappear()`—Called just after the view controller's content view has been removed from the app's view hierarchy. Use this method to perform additional teardown activities. Despite the name, the system does not call this method just because the content view has become hidden or obscured. This method is only called when the content view has been removed from the app's view hierarchy.

Q & A