

Übung 3: Modal-Views

Das Ziel dieser Übung ist es, das von Apple für iOS extensiv verwendete Model-View-Controller-Muster anhand der Klasse `UIViewController` kennen zu lernen und anzuwenden. Dazu wird eine kleine Demo-iPhone-App entwickelt, welche die Verwendung von xib- und Storyboard-Dateien zeigt. Konkret verwendet diese Übung modale View-Übergänge, um per Knopfdruck Übergänge auf andere Views zu ermöglichen. Sie sehen wie sich diese Übergänge programmatisch oder im Interface Builder (d.h. im Storyboard) umsetzen lassen.



1. Neues Projekt „ModalDemo“ inkl. MainView

Erstellen sie in Xcode ein neues Projekt „ModalDemo“ fürs iPhone, wählen sie „Single View Application“. Xcode generiert damit automatisch ein Projekt mit diversen vorgegebenen Dateien. Unter anderem wurde von Xcode eine Datei „Main.storyboard“ generiert, diese steht für unser Storyboard, in welcher auch unsere Hauptview (= die Initiale App-View) ist. Setzen sie auf diese im Interface Builder einen grossen Titel „MainView“, eine Hintergrundfarbe und platzieren sie unten einen Knopf „show second view“. Definieren sie in der Klasse `ViewController` eine `IBAction`-Methode `showSecondView` und verbinden sie diese mit dem entsprechenden Knopf.

2. Eine zweite View: `SecondViewController` inkl. xib-Datei

Generieren sie eine neue Klasse `SecondViewController`. Klicken sie dazu mit rechts auf den „ModalDemo“-Ordner im Projekt-Navigator und wählen sie „New File...“. Wählen sie dann „iOS“ - „Cocoa Touch Class“ und dann „subclass of `UIViewController`“ und „Also create XIB file“. Präparieren sie im Interface Builder diese neue View (d.h. die Datei `SecondViewController.xib`) wie gewünscht mit Titel, Farbe und zurück-Knopf. Verbinden sie den zurück-Knopf mit einer in der Klasse `SecondViewController` dafür definierten `IBAction`-Methode `backButtonPressed`.

3. Actions für `SecondView` implementieren

Beim Drücken von „show second view“ auf der `MainView` soll nun natürlich auch die `SecondView` angezeigt werden. Die Klasse `ViewController` und die dazu gehörige View (aus `Main.storyboard`) werden automatisch geladen, da diese den Startbildschirm von unserem Programm darstellen. Die neu erstellte `SecondView` müssen wir jedoch selber instanziiieren und initialisieren. Legen sie dazu in der Klasse `ViewController` ein `let`-Property `secondViewController` als Referenztyp auf `SecondViewController` an und initialisieren Sie dieses gleich mit einer Instanz von dieser Klasse. Probieren sie ebenfalls die verschiedenen Styles vom Property `modalTransitionStyle` dieser Instanz von einem `SecondViewController` aus. Die `IBAction`-Methode `showSecondViewButtonPressed` soll nun also die `SecondView` mittels modalem Übergang anzeigen. Implementieren Sie dies mit Hilfe der Methode `presentViewController:animated:completion:` oder mit Hilfe der

Methode `show(_sender:)`, welche beide in der Klasse `UIViewController` (und somit auch in unserer Klasse `ViewController`) implementiert sind. Was ist der Unterschied zwischen diesen beiden Methoden? Analog dazu soll die Modal-View beim Drücken von „zurück“ auf der SecondView auch wieder verschwinden. Implementieren Sie dazu in der Klasse `SecondViewController` eine `IBAction`-Methode `backButtonPressed`, in welcher dann dazu die Methode `dismiss(animated:completion:)` der Klasse `UIViewController` verwendet wird. Damit können Sie modal eine zweite View anzeigen und wieder verschwinden lassen. Hinweis: Lesen Sie die API-Doku vom Property `presentingViewController` und der Instanz-Methode `dismiss(animated:completion:)` der Klasse `UIViewController`. – Was für ein Mechanismus wurde da offensichtlich eingebaut?

4. Anzeigen-Zähler auf der MainView

Nun wollen wir auf der MainView anzeigen, wie oft diese bisher angezeigt wurde. Fügen Sie dazu der Klasse `ViewController` ein Property `appearanceCounter` hinzu, mit dessen Hilfe werden wir uns die Anzahl „Erscheinungen“ dieser View als Ganzzahl merken. Fügen Sie der MainView ein Label hinzu und verbinden Sie dieses als `IBOutlet`-Property `counterLabel` mit dem `ViewController`. Jedes mal, wenn unsere MainView neu angezeigt wird, wird vom Framework im entsprechenden Controller (also hier in der Klasse `ViewController`) die Methode `viewWillAppear:` aufgerufen. Erhöhen Sie also in dieser Methode den Wert von `appearanceCounter` und aktualisieren Sie entsprechend den Text, welcher vom `counterLabel` dargestellt wird.

5. Dritte View inkl. Übergänge im Storyboard: StoryBoardView

Zum Schluss sollen Sie einen dritten neuen Bildschirm direkt in der Datei `Main.storyboard` erstellen. Ziehen Sie also im Interface Builder aus der „Object Library“ eine weitere `ViewController`-Instanz auf ihr Storyboard. Fügen Sie auf der MainView einen Knopf „show storyboard view“ ein und verbinden Sie diesen im Interface Builder als Sequen mit dem gewünschten neuen `ViewController` (Maus: ctrl-Drag vom Knopf auf die neue `StoryboardView`). Bringen Sie auf der `StoryboardView` zwei zurück-Knöpfe an, einen für die Verwendung vom „sequen unwind“-Mechanismus (Titel: „back with Sequen unwind“) und den anderen (Titel: „back with Custom `IBAction`“) für eine selber implementierte `IBAction`-Methode und unter der Verwendung der oben gesehenen Methode `dismiss(animated:completion:)` der Klasse `UIViewController`. Hinweis: Sie können hier der Einfachheit halber für die beiden Views im Storyboard (also für die MainView und `StoryboardView`) dieselbe `ViewController`-Klasse (nämlich `ViewController`) verwenden. Dies können Sie im Interface Builder im „Identity Manager“ festlegen, resp. kontrollieren, siehe Bild rechts (roter Pfeil).

