

Kapitel 1

Wie Sie Swift lernen

Programmieren von Computern gilt im Allgemeinen als kompliziert, schwierig und aufwendig zu lernen. In einem gewissen Sinne ist das alles richtig, aber es gibt noch einen anderen wichtigen Punkt: Computer zu programmieren kann vor allem richtig Spaß machen!

1.1 Grafische Oberflächen auf Apple-Systemen

Apple-Rechner machen Spaß! Sie haben ein sehr ausgereiftes und bewährtes Betriebssystem und die Apple-Ingenieure haben sehr viel Zeit und Mühe darauf verwendet, ihre Produkte einfach und angenehm bedienbar zu machen. Für die Entwickler bedeutet das manchmal gewisse Einschränkungen in dem, was sie machen dürfen. Aber genauso profitieren Sie auch von einer einheitlichen und schönen grafischen Oberfläche, wenn Sie neue Programme schreiben.

Ein wesentliches Hilfsmittel ist eine vollständige Entwicklungsumgebung, mit der Sie programmieren und arbeiten können. Diese enthält auch einen grafischen Editor, in dem Sie Benutzeroberflächen ähnlich wie in einem Grafikprogramm visuell arrangieren können. Er ist ein wichtiger Bestandteil von Xcode, der Entwicklungsumgebung von Apple für iPhone, iPad, die Apple Watch und auch MacOS X.

Dann gibt es die Programmiersprache. Aber es gibt noch mehr: Stellt ein iPhone beispielsweise eine Tabelle dar, so muss ein Programm die Tabelle mit Inhalt füllen und ebenfalls darauf reagieren, wenn der Benutzer einzelne Tabellenzeilen auswählt. Um eine Sache muss sich der Programmierer des Programms allerdings nicht kümmern: die Tabelle selbst darzustellen, sie zu scrollen, wenn ein Benutzer mit dem Finger nach oben oder nach unten fährt, und die Berührungen zu analysieren. Diese Arbeit wird vom Betriebssystem übernommen.

Dadurch nimmt das Betriebssystem einem Programmierer viel Arbeit ab. Diese Funktionen gehören nicht zur Programmiersprache selbst, die Programmiersprache bestimmt nur die Art und Weise, wie ein Programmierer seine Anweisungen formuliert. Daher hat diese Funktionalität unter MacOS X einen eigenen Namen, man spricht dort von *Cocoa*. Cocoa ist der Name aller Funktionalitäten, die das Betriebssystem einem Pro-

grammierer in der Programmiersprache Swift bietet. Auf dem iPhone gibt es weitere Funktionen, zum Beispiel die Gestensteuerung, die unter MacOS X nicht vorhanden sind. Daher haben diese Funktionen auf dem iPhone einen etwas anderen Namen, er lautet *Cocoa Touch*. Wenn Sie Swift beherrschen, können Sie Cocoa und Cocoa Touch benutzen, um Programme sowohl für MacOS X als auch das iPhone, das iPad und sogar die Apple Watch zu schreiben.

Dieses Buch vermittelt nicht nur die Grundlagen von Swift, Cocoa und Cocoa Touch, sondern auch die Konzepte und Prinzipien, nach denen diese funktionieren. Alle heute gängigen Systeme beruhen auf ähnlichen Prinzipien, wobei Cocoa und Cocoa Touch die bewährtesten und ausgereiftesten Systeme auf dem Markt sind. Dadurch können Sie mit einem geringen zusätzlichen Lernaufwand auch auf die Entwicklung anderer Systeme umsteigen, wenn Sie dies später einmal wünschen sollten. Andererseits bin ich mir sicher, dass Sie in naher Zukunft von den Apple-Lösungen überzeugt sein werden und dann gar nicht mehr umsteigen wollen.

1.2 Die Geschichte von Swift

Swift ist eine sehr junge Programmiersprache. Bevor sie im Sommer 2014 vorgestellt wurde, war die Programmiersprache Objective-C die Sprache der Wahl für alle Apple-Systeme. Sie war ausgereift und bewährt, hatte allerdings gewisse Schwächen und war nur schwer für Konzepte wie die *funktionale Programmierung* zu erweitern.

Die erste Version von Swift hatte viele dieser Punkte verbessert. Allerdings hatte sich herausgestellt, dass sie noch in einigen Punkten verbessert werden musste. So war sie in bestimmten Situationen nur schwer zu verwenden oder sie hatte keine Möglichkeit, bestimmte Fehler zu behandeln. Deswegen hat Swift im Sommer 2015 ein Update bekommen, das nicht mehr kompatibel zur älteren Version war. Dies war zugegebenermaßen eine schlechte Nachricht für alle Entwickler, die schon frühzeitig auf Swift umgestiegen sind.

Andererseits bedeutet es auch, dass nun ein guter Zeitpunkt gekommen ist, um Swift zu erlernen: Die Sprache ist ausgereifter, aber immer noch leichter zu erlernen als Objective-C. Trotzdem greift sie auf Apple-Systemen auf die ausgereifte grafische Oberfläche und das Betriebssystem zurück, genauso wie Objective-C es tut.

In der Praxis können Sie jederzeit Swift und Objective-C mischen. Das ist ein bisschen umständlich, ermöglicht Ihnen jedoch, auch ältere Programme und Bibliotheken weiterzuverwenden. Neue Projekte werden oftmals als reine Swift-Programme gestartet. Allerdings ist es noch nicht der Fall, dass Swift Objective-C verdrängt hat oder in naher Zukunft verdrängen wird. Mittelfristig wird Apple allerdings auf Swift setzen und gerade für den Neueinstieg in die Programmierung ist diese Sprache bestens geeignet!

1.3 Was Sie hier programmieren werden

Sie werden in diesem Buch die Grundlagen der Programmierung in der Programmiersprache Swift und der Entwicklungsumgebung Xcode erlernen. Dadurch können Sie sowohl iPhone, iPad als auch MacOS X und die Apple Watch programmieren. Einige Konzepte funktionieren auf allen Plattformen gleich, aber andere sind auch recht unterschiedlich. Daher zeige ich Ihnen Beispiele für alle diese Plattformen.

Abbildung 1.1 zeigt einige der Programme, die Sie in späteren Kapiteln schreiben werden: eine App zum Erzeugen von Passwörtern und ein einfaches Spiel, die beide auf dem iPhone laufen. Die Passwort-App werden Sie anschließend auch auf das iPad, auf MacOS X und auf die Apple Watch portieren.

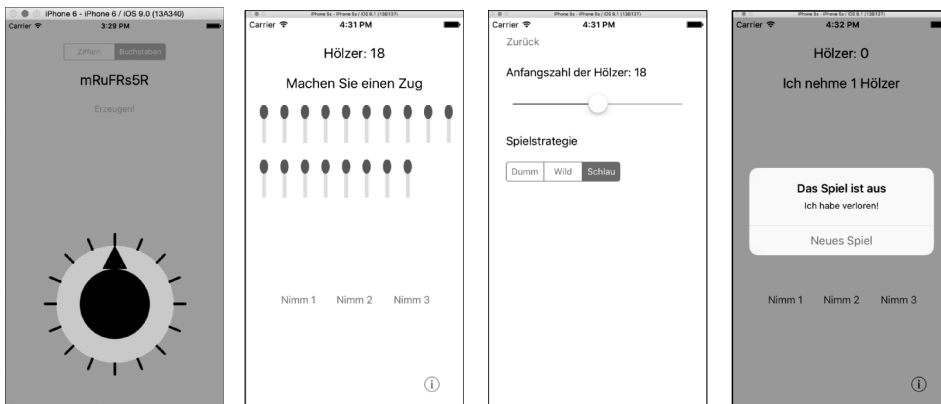


Abbildung 1.1 Beispiele der Apps, die Sie für das iPhone entwickeln werden

1.4 Wenn Sie bereits Objective-C kennen

Swift unterscheidet sich in einigen Aspekten von Objective-C. Insbesondere verfügt Swift über die Möglichkeit, interaktiv in so genannten Playgrounds zu arbeiten, die ich in Abschnitt 3.1 einführe. Datentypen unter Swift sind sicherer zu verwenden, weil ihre Prüfung in den meisten Fällen strenger ist und man sie nicht so einfach »missbrauchen« kann wie unter Objective-C. Auf Datentypen gehe ich allgemein in Abschnitt 3.3 ein.

Gerade fortgeschrittene Entwickler werden sich über die eleganten Möglichkeiten der funktionalen Programmierung freuen, die ich im fortgeschrittenen Abschnitt 10.6.1 vorstelle. Die Speicherverwaltung von Swift ist ein wenig ausgefeilter und wird in Kapitel 9 ausführlich beschrieben.

In großen Teilen ist die Entwicklung grafischer Apps in Swift und Objective-C allerdings ähnlich: Für das Zusammenstellen der grafischen Elemente unter Xcode verwenden Sie die gleichen Ausrichtungsregeln und Verknüpfungen. Diese stelle ich zum ersten Mal in Abschnitt 2.4 vor, sie werden Ihnen allerdings in allen Kapiteln mit grafischen Elementen wieder begegnen.

1.5 Was Sie mitbringen müssen

Um mit diesem Buch arbeiten zu können, sollten Sie mit einem Mac umgehen können und die Bedienung und Benutzerführung – wie beispielsweise Dateiverwaltung, Farbauswahl und Zeichensätze etc. – der Plattform verstehen und anwenden können. Programmierkenntnisse in anderen Sprachen sind von Vorteil, aber nicht zwingend erforderlich.

Um den Beispielen in diesem Buch folgen zu können, benötigen Sie einen Mac mit MacOS X ab Version 10.10 aufwärts. Einige Beispiele sind für ein iPhone und ein iPad gedacht. Aber diese werden zunächst nur auf dem Mac mit dem Simulator ausgeführt. Dieser simuliert ein iPhone oder ein iPad auf dem Mac, sieht so ähnlich aus und verhält sich ähnlich wie »echte« Geräte. Ich zeige Ihnen auch, wie Sie Ihre Apps auf einem richtigen Gerät laufen lassen können.

Sie brauchen außerdem das bereits genannte Programm *Xcode* von Apple. Dieses enthält alles, um Programme zu schreiben und auszuführen. Dies bekommen Sie kostenlos von Apple, das ist gleich zu Anfang in Kapitel 2 beschrieben.

Ich verwende in diesem Buch gelegentlich englischsprachige Fachbegriffe. Wo es möglich und sinnvoll ist, gebe ich auch deutsche Übersetzungen an und versuche, diese zu verwenden. Da Xcode – wie generell die meisten Programmiersprachen – allerdings durchgehend englischsprachig ist, gibt es für viele Ausdrücke keine gängigen und verständlichen deutschen Übersetzungen. Deswegen verwende ich dort die englischsprachigen Begriffe, wo die deutschen missverständlich oder unüblich wären.

1.6 Wie Sie mit diesem Buch Erfolg haben

Dieses Buch ist hauptsächlich ein Lehrbuch, kein Nachschlagewerk. Die einzelnen Kapitel bauen aufeinander auf. Das bedeutet, dass Sie versuchen sollten, die Kapitel in der richtigen Reihenfolge zu bearbeiten. Auch empfehle ich Ihnen, alle Beispiele auszuprobieren und selbst ein bisschen zu experimentieren. Dadurch lernen Sie erfahrungsgemäß am besten.

Die Codebeispiele, die im Buch beschrieben werden, können Sie auch von den Webseiten <http://www.downloads.wrox-press.de> und <http://swift-sprachkurs.de> herunterladen.

Grundsätzlich empfehle ich Ihnen allerdings, die Beispiele Schritt für Schritt selbst nachzuprogrammieren. Denn sie enthalten nicht nur die fertigen Programmtexte, sondern auch die eigentlichen Arbeitsschritte, die typischerweise unter Xcode erforderlich sind, um bestimmte Ergebnisse zu erzielen. In späteren Kapiteln werde ich voraussetzen, dass Sie beispielsweise die Ausrichtungsregeln der grafischen Oberfläche selbstständig festlegen können, ohne dass ich die dafür notwendigen Schritte wiederholen muss.

Am Ende der meisten Kapitel finden Sie ein Unterkapitel »Fortgeschrittenes«. Dort finden Sie zum weiterführende Anmerkungen zu den Themen, die Sie kennengelernt haben. Ich gebe auch Hinweise und Stichworte zu Themen, die über dieses Buch hinausgehen und mit denen Sie sich beschäftigen können, wenn Sie bei einem Thema weiter in die Tiefe gehen möchten. Dieses Unterkapitel ist für das Verständnis der folgenden Kapitel nicht zwingend notwendig und Sie können es beim ersten Lesen überspringen.

Außerdem finden Sie am Ende der meisten Kapitel Übungsaufgaben. Diese sollten Sie nicht überspringen, sondern mit ihnen experimentieren. Sie sind einerseits dazu gedacht, Sie anzuregen, neue Dinge auszuprobieren und zu bearbeiten. Andererseits geht es aber auch darum, dass Sie bereits erlernte Arbeitsschritte nochmals wiederholen. Dadurch geht Ihnen die Arbeit leichter von der Hand und Sie lernen einfacher und besser.

Die Übungen haben unterschiedliche Schwierigkeitsgrade. Einige werden Sie möglicherweise beim ersten Lesen nicht lösen. Wenn Sie daher eine Übung nicht auf Anhieb schaffen, empfehle ich Ihnen, erstmal weiterzulesen und später das Kapitel nochmals durchzuarbeiten.

Die Lösungen der meisten Übungsaufgaben finden Sie in Anhang A. Einige Übungen in den späteren Kapiteln sind allerdings zu komplex und erfordern ein eigenes Projekt. Diese können Sie wiederum auf der oben genannten Webseite finden.

Für ein Kapitel sollten Sie etwa zwei bis drei Stunden investieren, um es nachzuprogrammieren. Dadurch lernen Sie, wie Sie die Werkzeuge benutzen, insbesondere auch, wie Sie die Entwicklungsumgebung richtig einsetzen. Für die Übungen sollten Sie zusätzlich etwa ein bis vier Stunden Zeit investieren, in späteren Kapiteln unter Umständen auch mehr. Ich empfehle Ihnen hierbei, anfangs nicht mehr als eine Stunde pro Tag zu lesen. Die besten Ergebnisse erzielen Sie, wenn Sie kleine Schritte machen und auch gelegentlich die älteren Kapitel wiederholen. Wenn Sie diesen Ratschlägen folgen, beherrschen Sie am Ende die Grundlagen des Programmierens und können sich viele Dinge selbstständig erarbeiten.

Es ist sehr wichtig, dass Sie Spaß daran haben, Programme zu schreiben und Probleme zu lösen. In der Praxis verbringen Sie die meiste Zeit nicht mit dem Schreiben von Programmen, sondern mit dem Lesen von Programmtexten anderer Entwickler sowie mit der Fehlersuche – sowohl in Ihrem eigenen Programm als auch in anderer Leute Programmtexten. Dies macht auch in der Gruppe viel Spaß. Darum treffen Sie sich mit

1.6 | Wie Sie Swift lernen

anderen Leuten, nehmen Sie an *CocoaHeads*-Meetings teil und fragen Sie erfolgreiche Entwickler, wie diese bei ihren Programmen vorgegangen sind und vorgehen. CocoaHeads sind eine Gruppe von Interessierten, die sich jeden Monat zu Themen rund um MacOS und iOS-Entwicklung treffen. Diese Treffen finden in allen größeren Städten statt. Die Treffpunkte in Ihrer Nähe finden Sie unter <http://cocoaheads.org>.