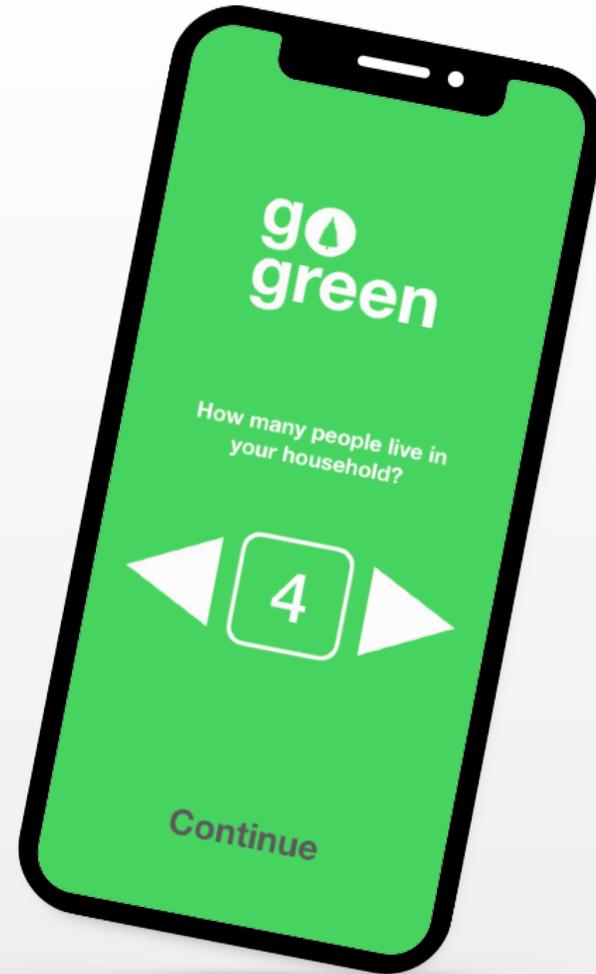
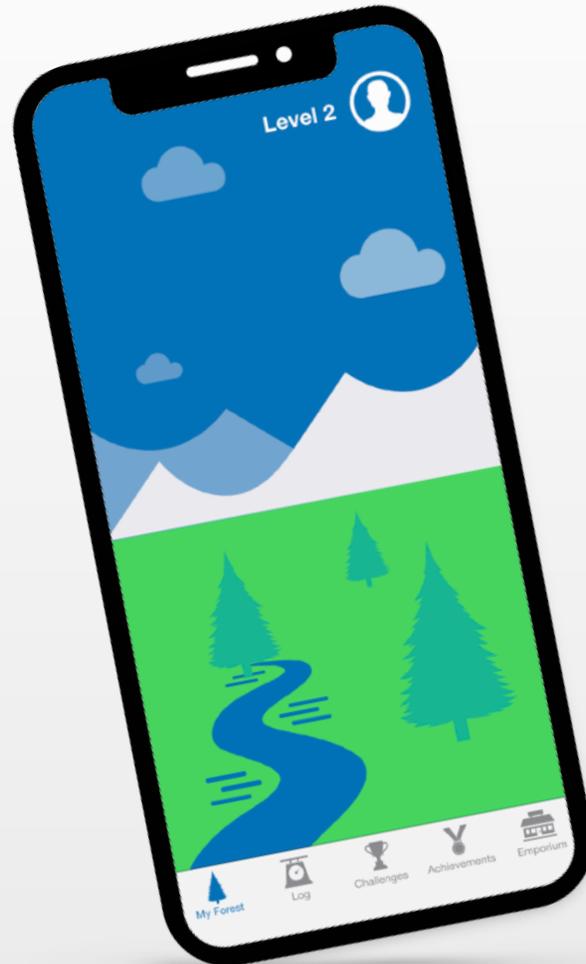




Swift Coding Club

In Swift entwickeln



Projekte zur App-Entwicklung

Projekte

1	SinglePhoto	3
2	QuestionBot	4
3	ChatBot	5
4	ColorMix	6
5	ElementsQuiz	7
6	AnimalSounds	8
7	RockPaperScissors	9
8	MemeMaker	10

Willkommen beim Swift Coding Club!

Indem du diesem Programmierclub beigetreten bist, hast du schon den ersten Schritt zur Entwicklung deiner ersten Apps gemacht. Mach dir keine Sorgen, falls du gerade erst damit beginnst, Programmieren zu lernen. Diese Projekte führen dich in die Grundlagen von Programmierkonzepten und der App-Entwicklung ein. Jedes Projekt baut auf dem vorherigen auf und konzentriert sich auf einen anderen Aspekt von Apps, sodass du am Ende die erforderlichen Fähigkeiten hast, um deine eigene App zu entwickeln.

Du brauchst den Kurs [Einführung in die App-Entwicklung mit Swift](#), um diese Projekte abzuschließen. Achte darauf, immer zuerst alle Voraussetzungen für das jeweilige Projekt abzuschließen.



SinglePhoto

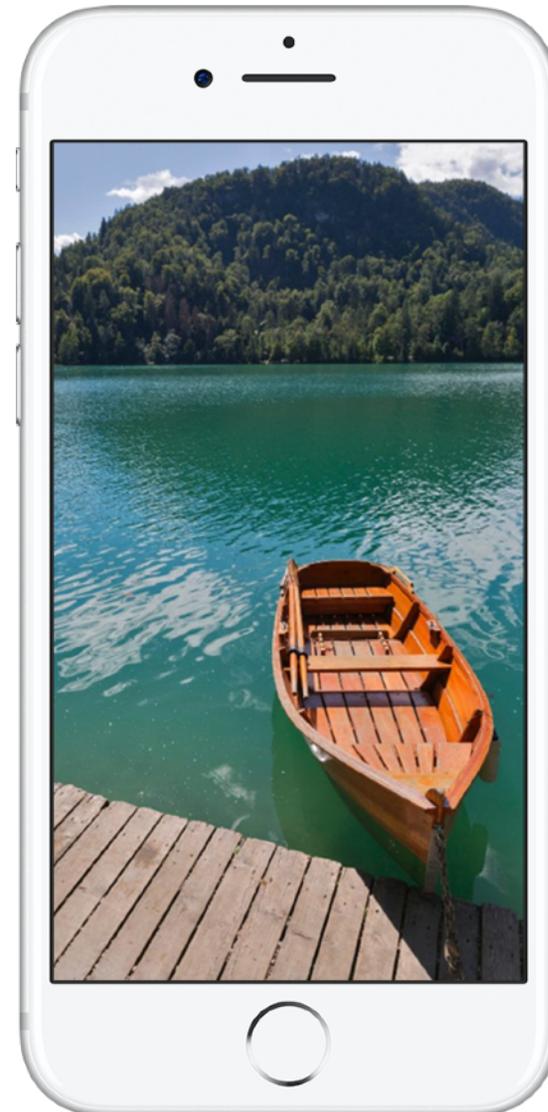
1

Voraussetzungen: Keine

Zeit: 1,5 Stunden

Gehe zu: [Lektion 5](#): Die erste App

Finde heraus, wie einfach es ist, deine allererste App zu erstellen. Mit SinglePhoto erstellst du eine App, die nicht nur etwas zeigt (z. B. ein Foto), sondern auch reagiert, wenn ein Benutzer auf das Display tippt. Diese einfache Interaktion ist die Basis von vielen Apps und wird noch oft nützlich sein, während du mehr über das Programmieren und die App-Entwicklung lernst. Außerdem machst du dich in diesem Projekt mit Xcode, dem Interface Builder und dem Simulator vertraut und lernst, wie du sie zusammen einsetzt, um eigene Apps zu entwickeln.



QuestionBot

2

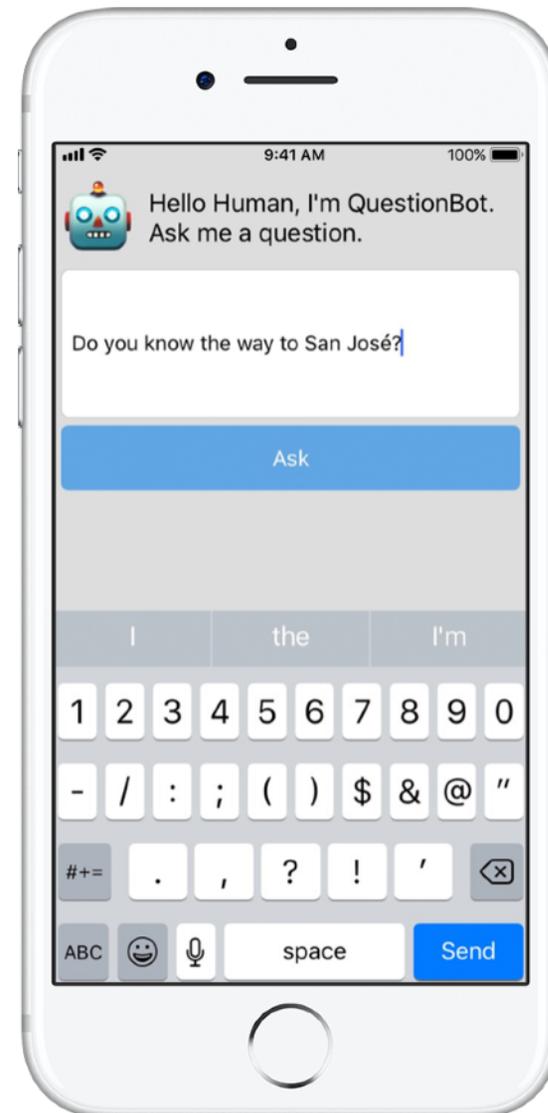
Voraussetzungen:

- SinglePhoto App
- Lektion 2: Namensgebung und Identifier
- Lektion 3: Strings
- Lektion 6: Funktionen
- Lektion 10: Parameter und Ergebnisse
- Lektion 11: Entscheidungen treffen

Zeit: 3,5 Stunden

Gehe zu: [Lektion 13: QuestionBot](#)

Hast du je eine Quiz-App verwendet oder dich gefragt, wie Siri funktioniert? Viele Apps scheinen ihr eigenes „Gehirn“ zu haben. Mit QuestionBot erstellst du eine App, die das Gehirn eines Bots enthält, der auf verschiedene Fragen unterschiedlich antwortet. Zu diesem Zweck lernst du, wie du Werte in Konstanten speicherst, Text in Code darstellst, Ein- und Ausgaben definierst und mehr. Im Mittelpunkt des Projekts steht, zu verstehen, wie die App funktioniert und welche Logik gebraucht wird, damit es so scheint, als habe deine App ein „Gehirn“.



Voraussetzungen:

- QuestionBot App
- Lektion 14: Datenfelder und Schleifen
- Lektion 15: Strukturen definieren

Zeit: 3,5 Stunden

Gehe zu: [Lektion 16](#): QuestionBot 2

Jetzt machen wir QuestionBot noch intelligenter. Du willst deine App wieder und wieder verwenden und es wäre hilfreich, wenn sie sich an deine vergangenen Interaktionen „erinnern“ könnte. Mit ChatBot baust du eine verbesserte Version von QuestionBot, die den Gesprächsverlauf zwischen Benutzer und Bot speichert. ChatBot ist schon teilweise erstellt. Du kannst dich also darauf konzentrieren, die nötigen Fertigkeiten zu erlernen, um die Teile der App zu erstellen, die die Unterhaltung speichern.

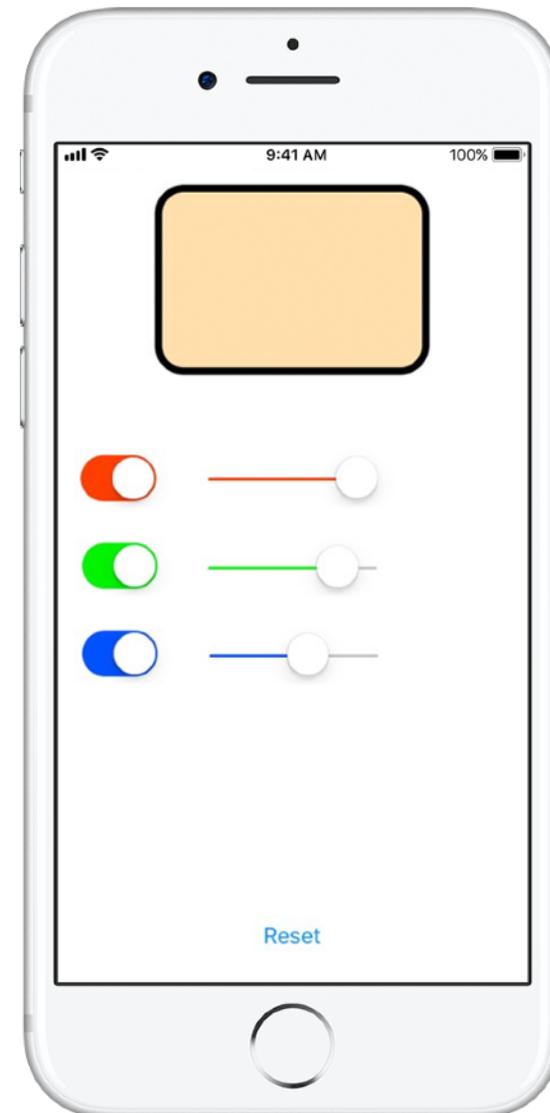


Voraussetzungen: ChatBot App

Zeit: 1,5 Stunden

Gehe zu: [Lektion 17](#): Aktionen und Outlets

Denke über die Benutzeroberfläche (User Interface, UI) nach. Bisher hast du Apps erstellt, bei denen der Benutzer auf Tasten tippen und Text eingeben kann. Mit ColorMix lernst du nun, wie du Schalter und Regler zur UI hinzufügst. Noch wichtiger ist, dass du lernst, wie du diese visuellen UI-Elemente mit Swift Code verbindest, damit sie so funktionieren, wie du es willst. Am Ende hast du ColorMix erstellt, eine App, die Farben erzeugt, indem sie Rot, Grün und Blau mischt.

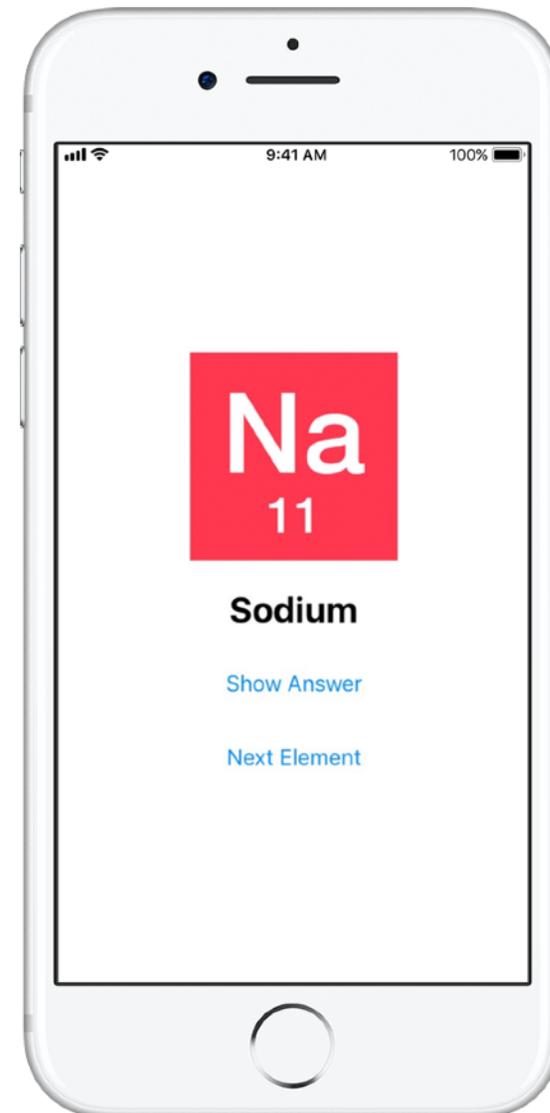


Voraussetzungen: ColorMix App

Zeit: 2 Stunden

Gehe zu: [Lektion 18](#): Adaptive Benutzeroberflächen

Die meisten Leute laden Apps, um damit ein konkretes Problem zu lösen – z. B. um sich besser zu organisieren, ihre Finanzen zu verwalten oder Wegbeschreibungen zu erhalten. Mit ElementsQuiz erstellst du eine App, die Schülern hilft, sich die Elemente des Periodensystems zu merken. Du wirst Stapelansichten verwenden, um einfach mehrere Ansichten gleichzeitig auf dem Display zu positionieren, wobei sich die Positionen an viele verschiedene Bildschirmgrößen anpassen können.

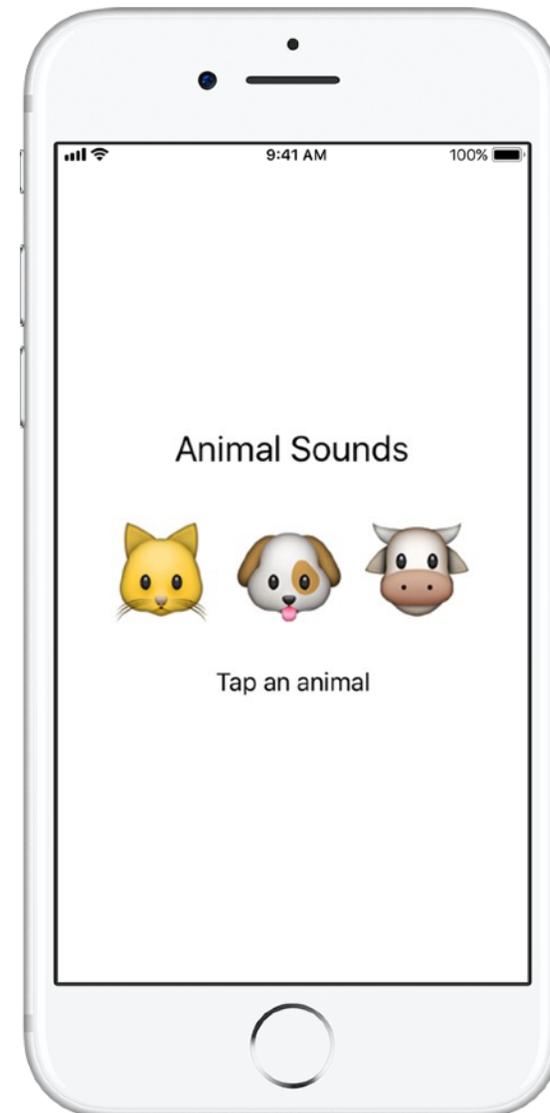


Voraussetzungen: ElementQuiz App

Zeit: 1 Stunde

Gehe zu: [Lektion 18](#): Adaptive Benutzeroberflächen

Da die Apps, die du erstellst, immer komplexer werden, fragst du dich vielleicht, ob du all diesen Code selbst schreiben musst. Hat das noch niemand vor dir gemacht? In AnimalSounds verwendest du bereits geschriebenen Code, der komplizierte Aufgaben wie das Abspielen von Audio für dich übernimmt, und lernst, wie du diesen Code in deine App integrierst. Dann verwendest du Stapelansichten in anderen Stapelansichten, um deine bisher anspruchsvollste Benutzeroberfläche zu erstellen.



RockPaperScissors

7

Voraussetzungen:

- AnimalSounds App
- Lektion 19: Aufzählungen

Zeit: 1,5 Stunden

Gehe zu: [Lektion 20](#): Abschlussprojekt

Du wirst Benutzern oft eine Liste mit mehreren Möglichkeiten zur Auswahl anzeigen müssen, z. B. Reiseziele oder verschiedene Gerichte. In diesem Projekt lernst du, wie du die drei Optionen für das Spiel „Schere, Stein, Papier“ erstellst, damit die Benutzer beliebig lange gegen den Computer spielen können. Dabei lernst du auch, wie du Zufallsnummern generierst, damit der Computer eine zufällige Wahl treffen kann.



Voraussetzungen:

- AnimalSounds App
- Lektion 19: Aufzählungen

Zeit: 1,5 Stunden

Gehe zu: [Lektion 20](#): Abschlussprojekt

Ein Internetmeme besteht meistens aus einem Bild mit Text oben und unten. Mit MemeMaker lernst du, segmentierte Steuerelemente zu verwenden, um verschiedene Phrasen über und unter einem Bild anzuzeigen. Und da die Steuerelemente unabhängig voneinander sind, kannst du den Text mischen, um eigene Kombinationen zu erstellen.

