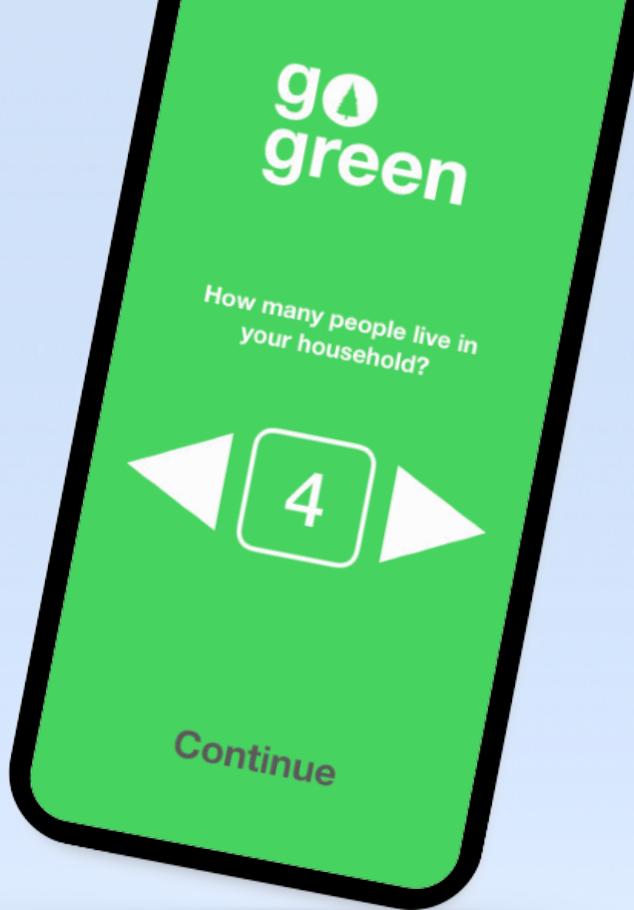




Swift Coding Club



Xcode

Projets de développement d'apps

Projets

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

Bienvenue au Club de programmation Swift !

En rejoignant ce club de programmation, vous faites le premier pas vers la création de vos propres apps. Si vous commencez tout juste à étudier la programmation, ne vous inquiétez pas. Les projets réunis ici vous permettront de comprendre les principaux concepts de programmation et d'acquérir les bases du développement d'apps. Chaque projet vient enrichir le précédent, en mettant l'accent sur différents aspects des apps. Ainsi, à la fin du parcours, vous disposerez des compétences nécessaires à la création de votre propre app.

Pour réaliser ces projets, vous aurez besoin du cours [Introduction au développement d'apps avec Swift](#). Veillez à remplir les conditions préalables pour chacun d'entre eux.



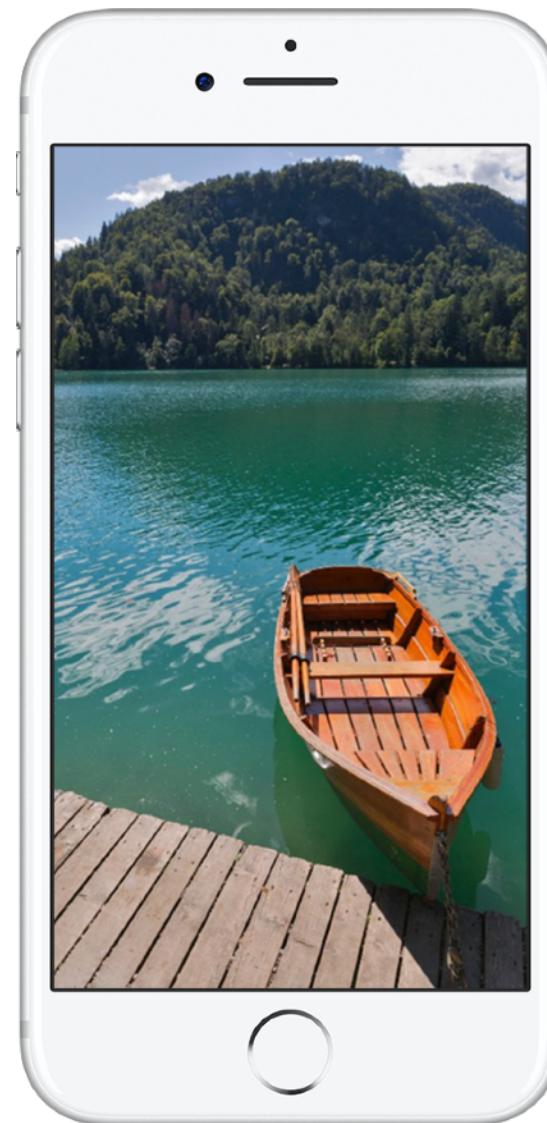
SinglePhoto

Prérequis : Aucun

Durée : 1,5 heure

Référence : [Leçon 5](#) : Première app

Nous allons vous montrer comme il est facile de créer sa toute première app. Avec SinglePhoto, vous allez créer une app qui ne se contente pas d'afficher quelque chose (par exemple, une photo), mais qui réagit lorsqu'un utilisateur touche l'écran. Ce type d'interaction très simple est à la base de bien des apps et se révèlera très utile à mesure que vous développerez vos compétences en matière de programmation et de développement d'apps. Grâce à ce projet, vous allez vous familiariser avec Xcode, Interface Builder et Simulator, et apprendre à les utiliser conjointement pour construire vos propres apps.



QuestionBot

2

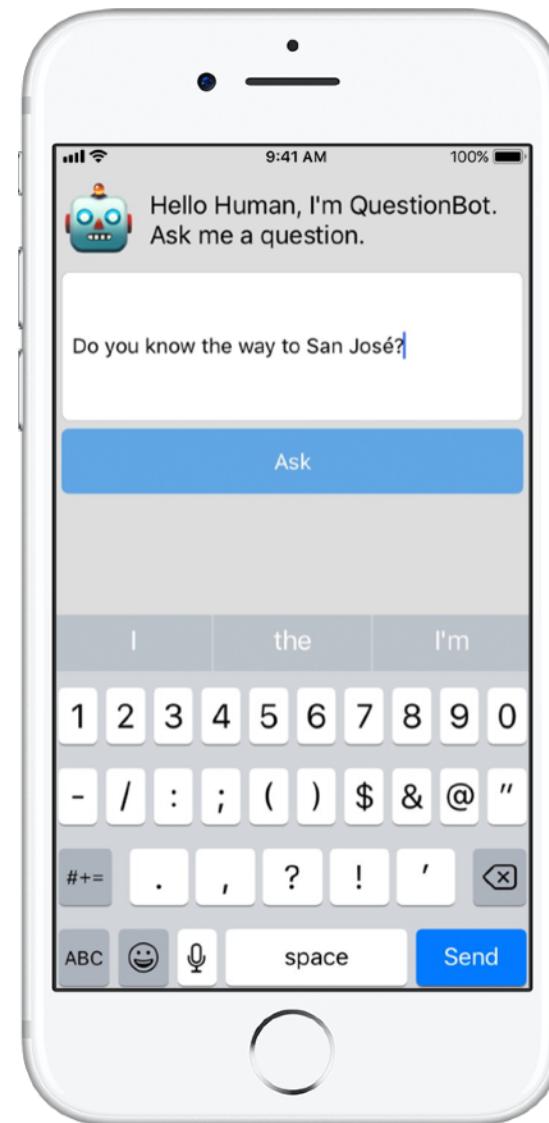
Prérequis :

- App SinglePhoto
- Leçon 2 : Nommage et identificateurs
- Leçon 3 : Chaînes
- Leçon 6 : Fonctions
- Leçon 10 : Paramètres et résultats
- Leçon 11 : Prendre des décisions

Durée : 6,5 heures

Référence : [Leçon 13](#) : QuestionBot

Avez-vous déjà utilisé une app de quiz ? Ou vous êtes-vous déjà demandé comment fonctionne Siri ? De nombreuses apps semblent avoir un « cerveau ». Avec QuestionBot, vous allez créer une app ayant le cerveau d'un robot capable de répondre différemment à différentes questions. Pour ce faire, vous allez apprendre à stocker des valeurs dans des constantes, à représenter du texte en code, à définir des entrées et des sorties, et à faire bien plus encore. Ce projet vous permettra de vous concentrer sur la compréhension du fonctionnement de l'app et de la logique nécessaire pour donner l'impression que votre app a un « cerveau ».



ChatBot

3

Prérequis :

- App QuestionBot
- Leçon 14 : Tableaux et boucles
- Leçon 15 : Définir des structures

Durée : 3,5 heures

Référence : [Leçon 16](#) : QuestionBot 2

Maintenant, rendons notre QuestionBot encore plus intelligent. Vous voulez pouvoir utiliser votre app sans fin. L'idéal serait qu'elle puisse « mémoriser » vos précédentes interactions. Avec ChatBot, vous allez créer une version améliorée de QuestionBot, capable de conserver un historique des messages échangés entre l'utilisateur et le robot. L'app ChatBot étant déjà partiellement construite, vous pouvez vous concentrer sur l'acquisition des différentes compétences nécessaires pour créer la partie de l'app chargée de garder le fil de la conversation.



ColorMix

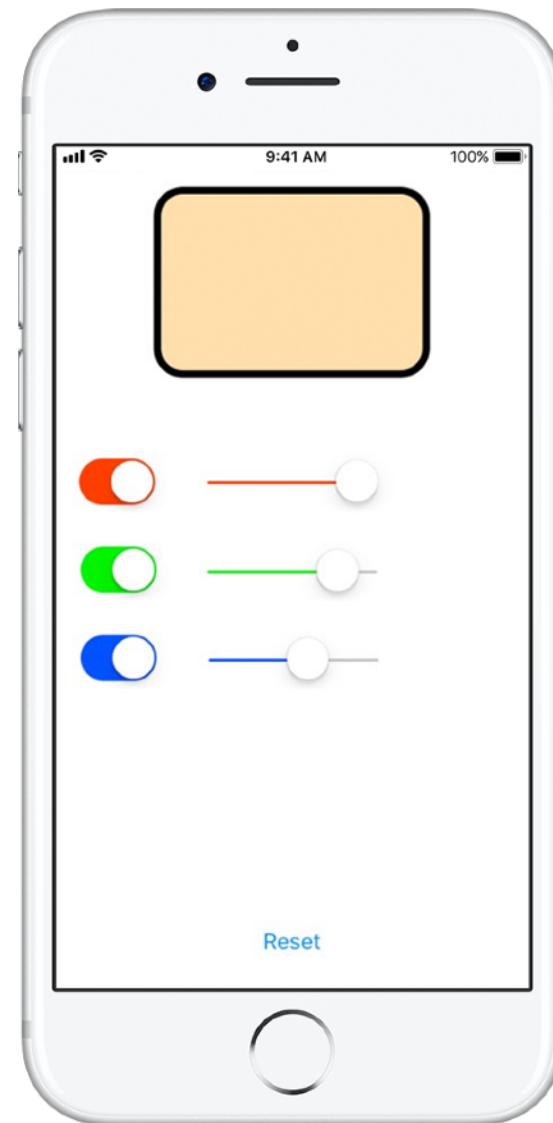
4

Prérequis : App ChatBot

Durée : 1,5 heure

Référence : [Leçon 17](#) : Actions et outlets

Réfléchissez à l'interface utilisateur. Jusqu'à présent, vous avez créé des apps dans lesquelles un utilisateur appuie sur des boutons et saisit du texte. Avec ColorMix, vous allez également découvrir comment ajouter des interrupteurs et des curseurs à l'interface utilisateur. Mais surtout, vous allez apprendre à relier ces éléments d'interface visuels à du code Swift, afin que tous fonctionnent comme vous le souhaitez. Le résultat, ce sera ColorMix, une app capable de générer des couleurs en mélangeant du rouge, du vert et du bleu.



ElementsQuiz

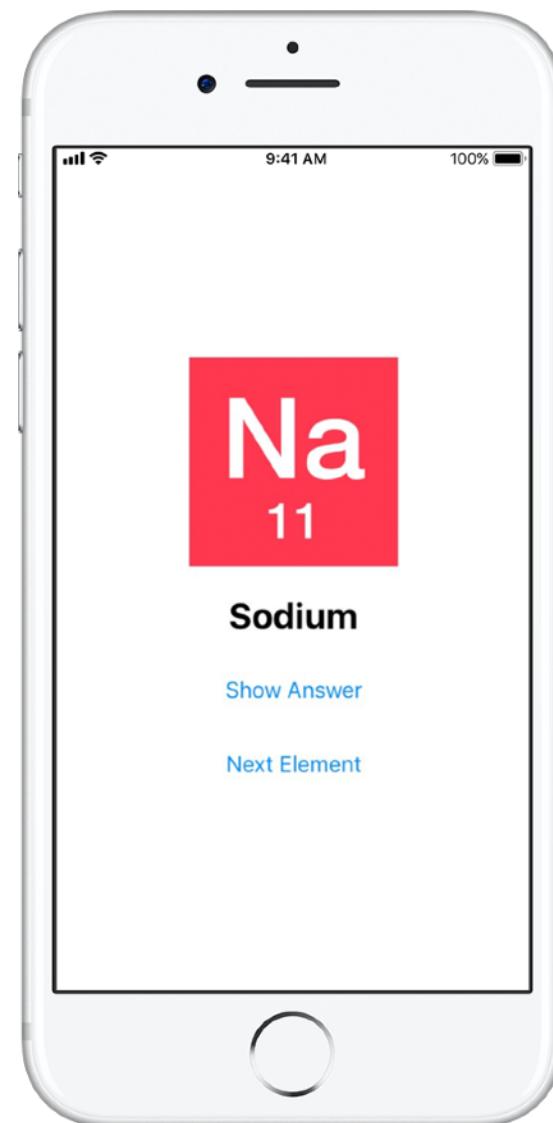
5

Prérequis : App ColorMix

Durée : 2 heures

Référence : [Leçon 18](#) : Interfaces utilisateur adaptatives

La plupart des gens téléchargent des apps pour résoudre un problème particulier : par exemple, pour s'organiser, pour faire leurs comptes ou pour trouver leur chemin. Dans ElementsQuiz, vous allez créer une app qui aide les élèves à mémoriser les éléments du tableau périodique. Vous utiliserez les vues en pile pour afficher plusieurs vues à l'écran, lesquelles se positionneront différemment en fonction de la taille de votre écran.



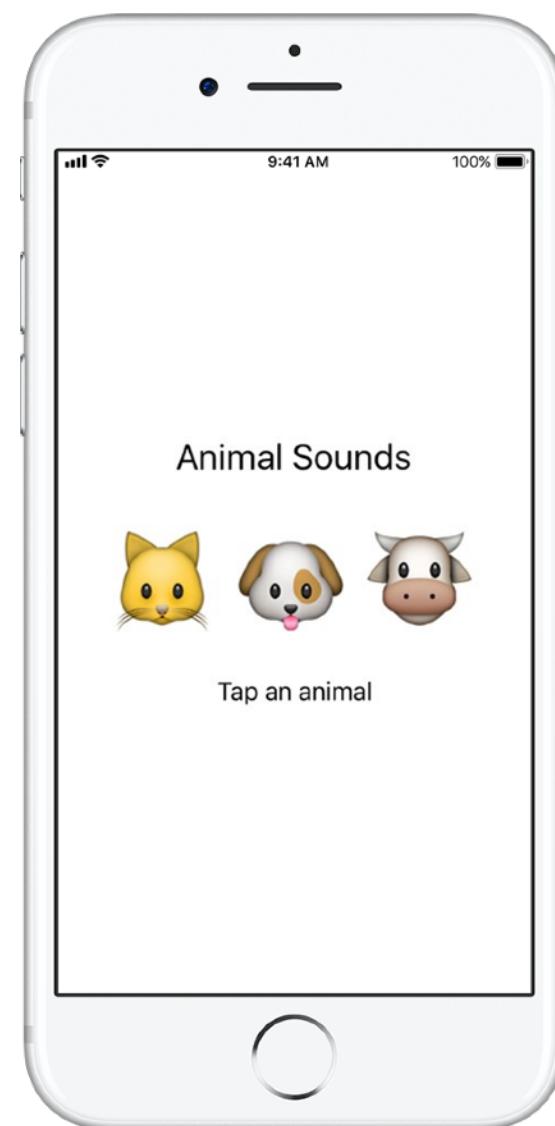
AnimalSounds

Prérequis : App ElementsQuiz

Durée : 1 heure

Référence : [Leçon 18](#) : Interfaces utilisateur adaptatives

À mesure que vos apps se complexifient, vous allez peut-être vous demander : « Faut-il que j'écrive tout ce code moi-même ? Quelqu'un n'a-t-il pas déjà écrit ce code ? » Dans AnimalSounds, vous allez réutiliser du code se chargeant de la lecture audio, que vous apprendrez à intégrer à votre app. Ensuite, vous utiliserez des vues en pile au sein d'autres vues en pile pour construire l'interface utilisateur la plus sophistiquée que vous ayez jamais créée.



RockPaperScissors

7

Prérequis :

- App AnimalSounds
- Leçon 19 : Énumérations

Durée : 1,5 heure

Référence : [Leçon 20](#) : Projet final

Bien souvent, vous aurez besoin de proposer à l'utilisateur une liste d'options, comme des itinéraires pour un déplacement ou des types de cuisine. Dans ce projet, vous allez apprendre à créer les trois seules options possibles pour un jeu Pierre, feuille, ciseaux, afin que l'utilisateur puisse jouer sans fin contre l'ordinateur. Vous apprendrez, en cours de route, à générer des nombres aléatoires pour que l'ordinateur sélectionne une option au hasard.



MemeMaker

Prérequis :

- App AnimalSounds
- Leçon 19 : Énumérations

Durée : 1,5 heure

Référence : [Leçon 20](#) : Projet final

Un mème internet est généralement une image ou une photo encadrée d'un court texte. Dans MemeMaker, vous allez apprendre à utiliser les contrôles segmentés pour afficher différents sous-titres ou légendes au-dessus et au-dessous d'une image. Comme les contrôles sont indépendants les uns des autres, vous pouvez mélanger les éléments du texte pour créer des combinaisons originales.

