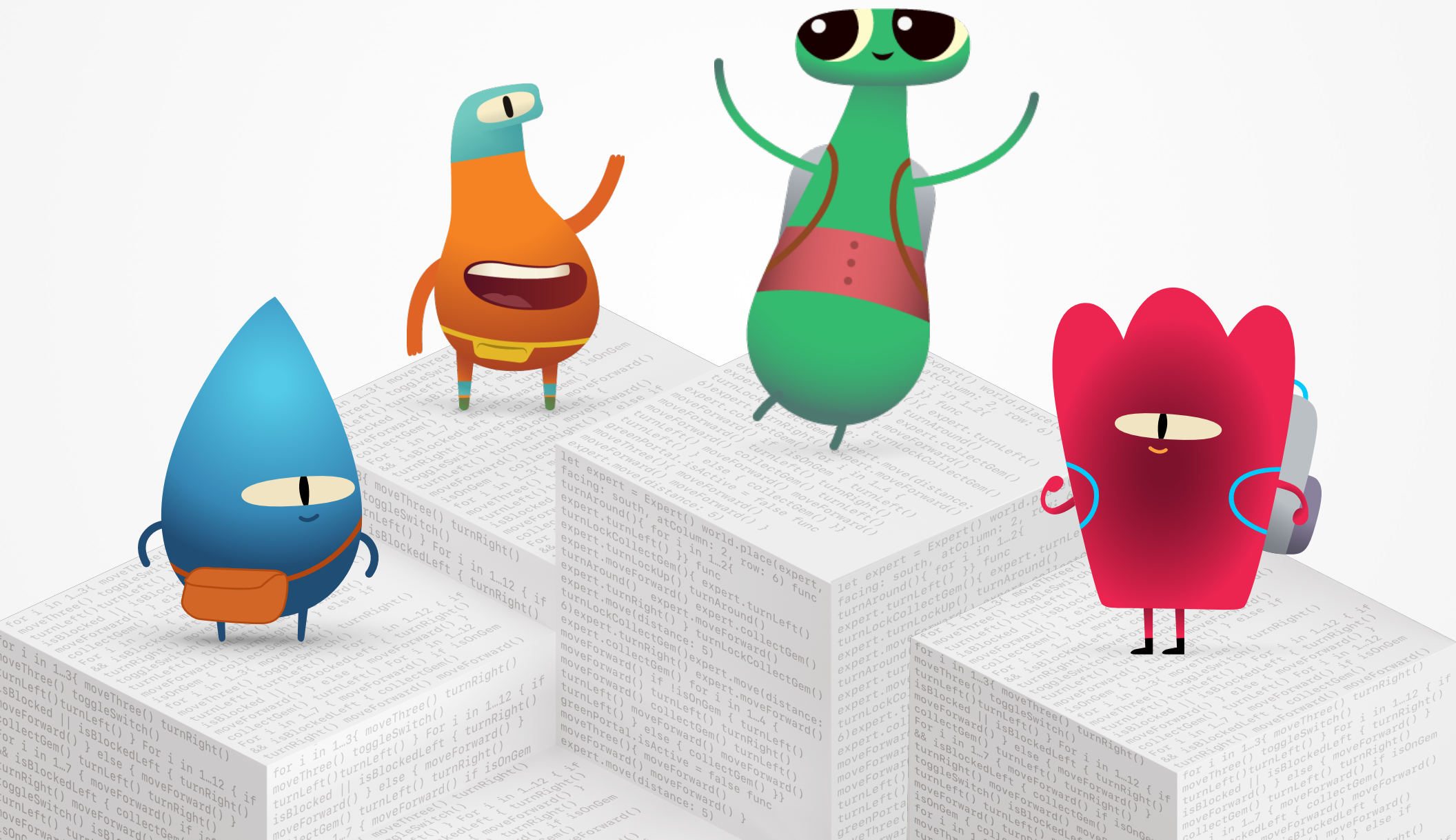




Bienvenue dans le Club de programmation Swift!

La programmation vous enseigne à résoudre des problèmes, à travailler en équipe de façon créative et à donner vie à vos idées.

Les Clubs de programmation Swift sont un moyen amusant d'apprendre à concevoir et à programmer des apps. Les activités conçues pour Swift, le langage de programmation d'Apple, favorisent la collaboration en vous apprenant à programmer et à construire des prototypes. Elles permettent aussi de réfléchir à la manière dont on peut changer le monde autour de soi grâce au code.

Nul besoin d'être enseignant ou spécialiste en programmation pour diriger un Club de programmation Swift. Le matériel pédagogique est conçu pour l'apprentissage individuel, de sorte que vous pouvez apprendre en même temps que les membres de votre club. À la fin, vous pourrez en profiter pour présenter les idées et les réalisations de votre club en organisant une exposition.

Ce guide est divisé en trois sections :



Préparation

Tout ce qu'il faut savoir pour lancer un Club de programmation Swift.



Exploration et mise en pratique

Des modules et des activités pour les rencontres du club.



Célébration

Des ressources pratiques pour planifier et animer un événement.

Ressources pour la programmation

Les Clubs de programmation Swift font appel à diverses ressources d'enseignement. Apple accompagne les programmeurs, de l'apprentissage élémentaire à la conception de véritables apps.



La programmation pour tous | 10 ans et plus

Apprenez les bases de la programmation et du langage Swift avec Swift Playgrounds sur iPad ou Mac. [En savoir plus >](#)



Programmer avec Swift | 14 ans et plus

Découvrez comment développer des apps avec Xcode sur Mac. [En savoir plus >](#)



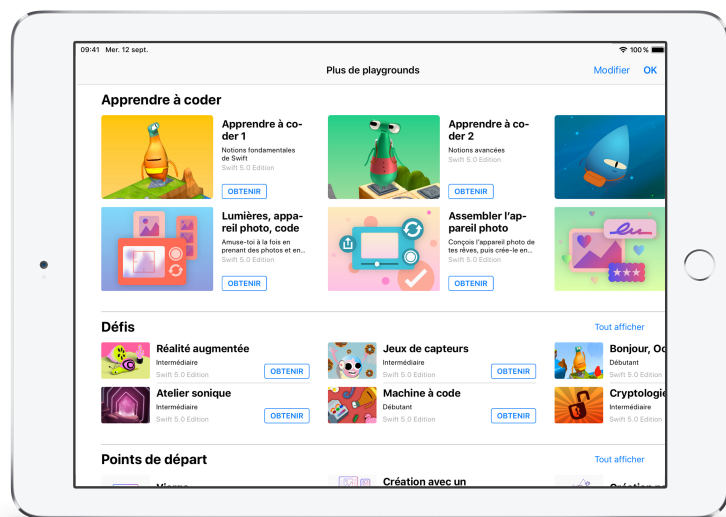
Préparation

1. Explorez les ressources La programmation pour tous

La programmation pour tous initie les participants au monde de la programmation avec des puzzles interactifs, des personnages amusants et des activités captivantes. Les ressources présentées ci-dessous vous aideront à planifier les activités de votre club.

Swift Playgrounds, c'est une app gratuite qui rend l'apprentissage de Swift amusant et interactif. On y trouve des leçons intégrées, ainsi que des défis additionnels créés par des développeurs et des éditeurs de contenu chevronnés.

Les guides La programmation pour tous comprennent des activités qui permettent de découvrir les concepts de programmation, de les comprendre grâce à des exemples du quotidien et de les mettre en pratique en résolvant des puzzles dans Swift Playgrounds.



[Télécharger et découvrir Swift Playgrounds >](#)



[Télécharger les ressources La programmation pour tous >](#)



2. Vérifiez si vous avez tout ce qu'il vous faut

Avant la première rencontre, assurez-vous d'avoir les outils suivants :

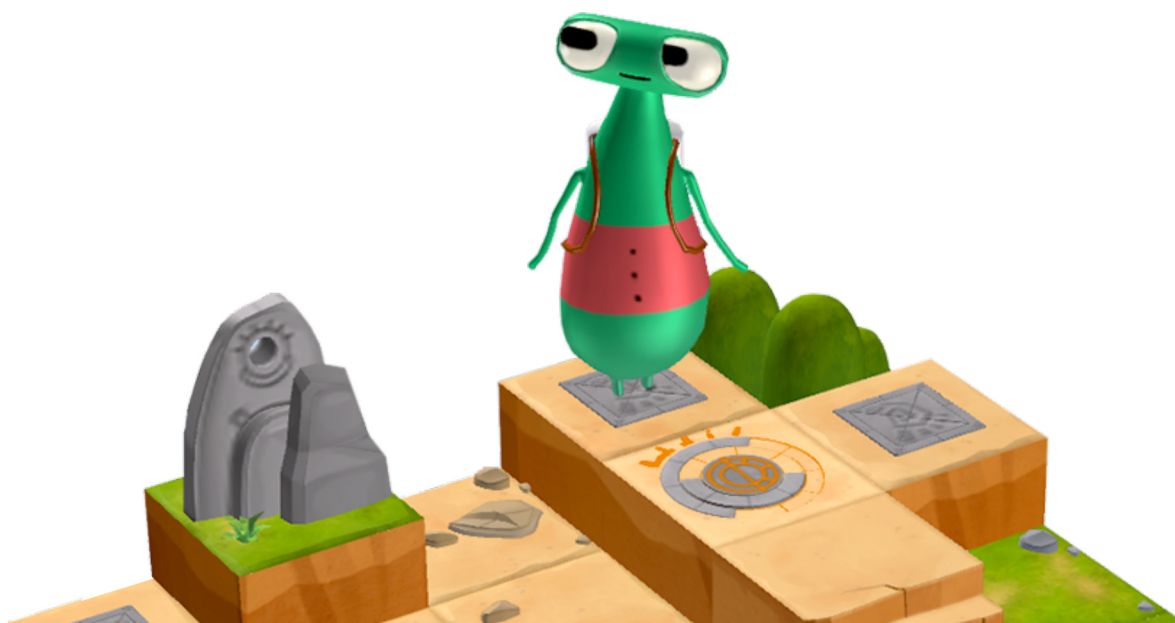
- **iPad ou Mac.** Swift Playgrounds requiert un iPad doté d'iPadOS 13 (ou une version ultérieure) ou un Mac doté de macOS 10.15.3 (ou une version ultérieure). Il est préférable que chaque personne ait son propre appareil, mais il est également possible d'en partager un et de programmer en équipe.
- **App Swift Playgrounds.**
[Télécharger Swift Playgrounds pour iPad >](#)
[Télécharger Swift Playgrounds pour Mac >](#)
- **Guides La programmation pour tous.**
[Télécharger le guide Puzzles de La programmation pour tous >](#)
[Télécharger le guide Aventures de La programmation pour tous >](#) (facultatif)

Consultez [l'assistance Apple](#) pour obtenir de l'aide avec les produits Apple.

3. Faites un plan

Voici quelques aspects à considérer :

- Qui sont les membres de votre club? Quels sont leurs champs d'intérêt? Ont-ils de l'expérience en programmation?
- À quelle fréquence auront lieu les rencontres de votre club? Si vous organisez un camp d'été, combien d'heures réserverez-vous aux activités de programmation?
- Quelles technologies utiliserez-vous?
- Quels sont les objectifs de votre club?





4. Passez le mot

Faites connaître le Club de programmation Swift autour de vous. Voici des idées et des ressources qui vous aideront à recruter des membres :

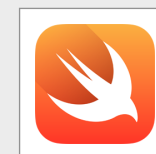
- **Faites la promotion de votre club.** Servez-vous des courriels, des médias sociaux, du web, de dépliants ou du bon vieux bouche-à-oreille pour faire connaître l'existence du club à votre communauté.
- **Organisez une séance d'information.** Demandez aux membres potentiels ce qui les intéresse et le type de projets qu'ils aimeraient créer. Discutez de l'organisation d'un événement et des différentes façons dont les membres pourraient y participer. Vous pouvez aussi diffuser en ligne une courte vidéo de présentation de votre club.

Les éléments suivants vous aideront à promouvoir et à personnaliser votre Club de programmation Swift :

- **Affiches.** [Téléchargez ce modèle gratuit](#), puis personnalisez-le pour créer votre propre affiche. Imprimez-la et distribuez-la, ou faites-en des affiches numériques à diffuser en ligne. Assurez-vous que tous les détails y sont : date, heure et lieu de rencontre, et méthode d'inscription.
- **Autocollants et t-shirts.** Utilisez ces [autocollants du Club de programmation Swift](#) pour promouvoir votre projet. Les t-shirts sont une excellente façon de reconnaître les participants à une exposition d'apps. Téléchargez les [logos du Club de programmation Swift](#) afin de créer des t-shirts pour vos membres.



Affiche



Autocollant



T-shirt



Formez une équipe de responsables. Les choses sont beaucoup plus amusantes et simples à gérer quand on peut avoir de l'aide pour diriger les activités. Lesquels de vos membres ont des qualités de leader?

Pensez à nommer des responsables pour les événements, la programmation, la conception, etc.

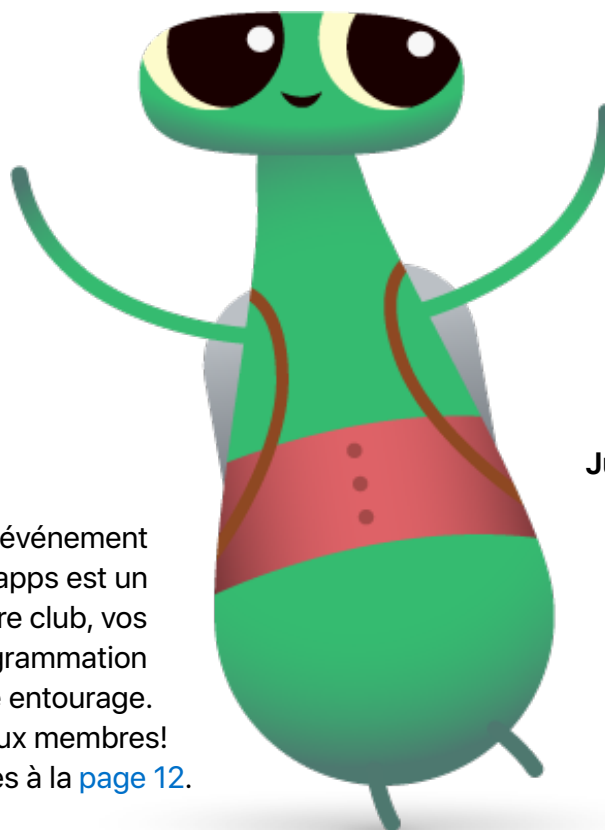
Apprenez les uns des autres.

Les animateurs n'ont pas besoin de tout savoir. Aidez les membres à développer leurs compétences en recherche et en résolution de problèmes, et encouragez-les à s'entraider.

Épatez la galerie. L'organisation d'un événement communautaire ou d'une exposition d'apps est un excellent moyen de faire découvrir votre club, vos idées d'apps et vos compétences en programmation aux enseignants, à vos proches et à votre entourage.

Et vous pourriez même recruter de nouveaux membres!

Trouvez toutes nos astuces à la [page 12](#).



Échangez des idées. Certaines personnes s'intéresseront aux apps de jeux, tandis que d'autres voudront programmer des apps pour aider les gens, se familiariser avec Swift ou faire bouger des robots. Réfléchissez à des façons de collaborer à des projets qui capteront l'intérêt de tous.

Jumelez les membres.

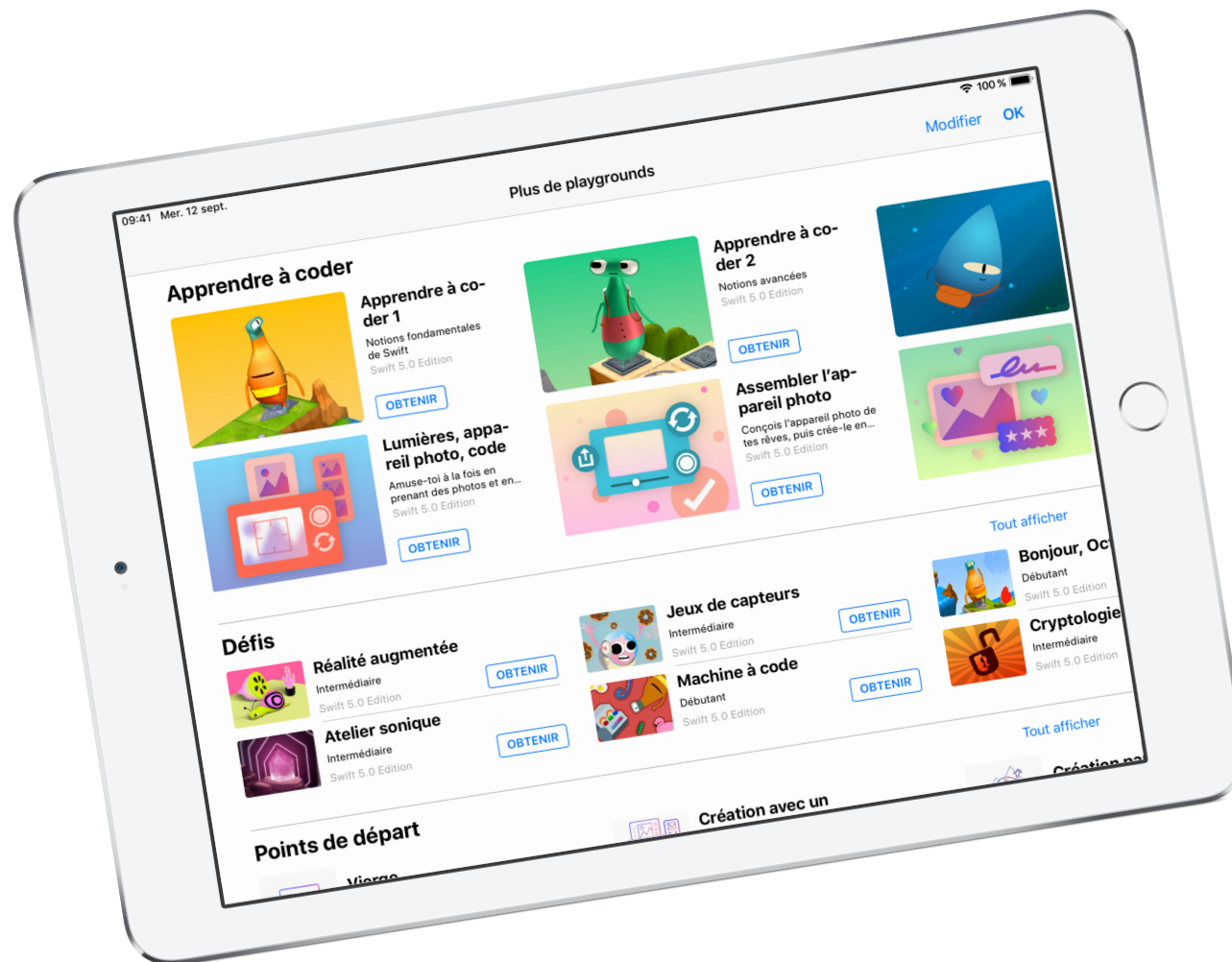
Les personnes plus expérimentées finissent parfois par faire cavalier seul. Encouragez-les à travailler en équipe avec des débutants. On apprend beaucoup quand on enseigne!

Exploration et mise en pratique



1. Explorez Swift Playgrounds

Les ressources du club sont centrées sur Swift Playgrounds, qui propose des leçons intégrées, de même que des défis supplémentaires créés par des développeurs et des éditeurs de contenu chevronnés. Familiarisez-vous d'abord avec le contenu de Swift Playgrounds et les fonctionnalités de l'app.



Fonctionnalités de Swift Playgrounds



Bibliothèque de code

Touchez l'icône dans la barre d'outils pour accéder à cette bibliothèque, puis glissez-déposez des parties de code récurrentes pour en avoir moins à taper.

Outils

Ouvrez ce menu pour réinitialiser une page, prendre une photo, créer un PDF ou enregistrer une vidéo.

Menu de pages

Touchez l'en-tête pour visualiser toutes les pages du playground. Touchez une page ou utilisez les flèches pour naviguer d'une page à l'autre.

Objectif : Utiliser les commandes Swift pour dire à Octet de se déplacer et de collecter une gemme.

Ton personnage, Octet, aime collecter les gemmes, mais il ne peut pas y arriver tout seul. Dans ce premier puzzle, tu devras écrire des **commandes** Swift pour déplacer Octet à travers le puzzle et collecter une gemme.

- 1 Cherche la gemme dans le monde en puzzle.
- 2 Saisis la combinaison correcte de commandes `moveForward()` — *avancer* — et `collectGem()` — *collecter une gemme*.
- 3 Touche Exécuter mon code.

[Touche ici pour saisir ton code](#)

Réglage de la vitesse

Ralentissez ou accélérez la vitesse d'exécution du code.

Mise en évidence du code en cours d'exécution

Sélectionnez Avancer dans mon code pour faire ressortir la ligne de code qui est en train de s'exécuter afin de mieux comprendre son fonctionnement.



Sélection d'un personnage

Touchez le personnage et choisissez-en un autre pour personnaliser votre expérience.

Indice

Cette fonctionnalité offre des pistes pour la résolution des puzzles. Et même si la solution est ultimement dévoilée, vous ne pouvez pas vous contenter de la copier-coller. Pour passer à l'étape suivante, vous devez vous-mêmes franchir les étapes et rédiger le code.

Astuces pour apprendre avec Swift Playgrounds



Examinez d'abord les puzzles. Invitez les membres de votre club à scruter l'univers d'Octet sous tous les angles dans la vue en direct pour avoir un bon aperçu de ce qu'ils doivent faire. Ils peuvent aussi opter pour l'affichage plein écran en touchant la barre entre les deux fenêtres et en la faisant glisser vers la gauche.

Cherchez plusieurs solutions. Chaque puzzle se résout de différentes façons. Si certains membres terminent les puzzles plus tôt, invitez-les à trouver d'autres solutions. Ils développeront leur esprit critique en explorant de nouvelles avenues.

Décomposez les puzzles. Quand les puzzles se corsent, les membres peuvent les décomposer pour mieux réfléchir aux différentes étapes de leur résolution. Ils peuvent aussi se servir de Pages ou de Notes pour planifier et rédiger les étapes avant d'entrer leur code dans l'app.

Nommez des personnes-ressources. Prévoyez un espace au sein duquel les experts du club pourront aider leurs pairs.



Programmez à deux. Demandez aux membres de votre club de se mettre deux par deux sur un iPad ou un Mac. Ils pourront trouver ensemble une solution aux puzzles, puis rédiger le code tour à tour.

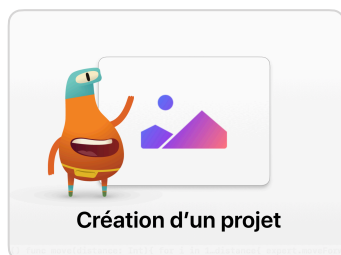
Utilisez les fonctions d'accessibilité. Swift Playgrounds est compatible avec les fonctionnalités d'accessibilité intégrées à macOS et à iPadOS pour que tout le monde puisse apprendre à coder. Il est notamment possible d'inverser les couleurs, d'activer les tons de gris et d'agrandir l'affichage afin d'améliorer la visibilité.



2. Choisissez vos modules

Les ressources du club sont divisées en modules thématiques qui combinent des activités de programmation et de conception créative. Chaque module se compose de 12 séances d'une heure et vise l'acquisition de compétences particulières. Dans les séances Exploration et Exercices, les membres du club découvrent les concepts clés et les mettent en pratique dans les puzzles et les défis de Swift Playgrounds. Et dans les séances Mise en pratique et Application concrète, ils s'intéressent au code en tant qu'outil pour explorer des idées et créer des produits. Les membres du club mettent à profit leurs compétences en programmation et en design pour créer ou concevoir un projet dans Swift Playgrounds en ciblant un public précis.

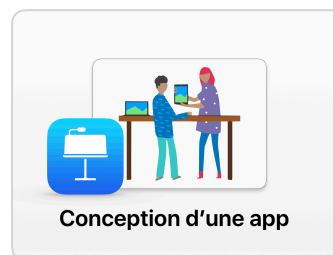
Vous trouverez un guide de l'animateur pour chaque module dans la deuxième partie du présent document. Cliquez sur les liens ci-dessous pour les consulter directement.



Création d'un projet

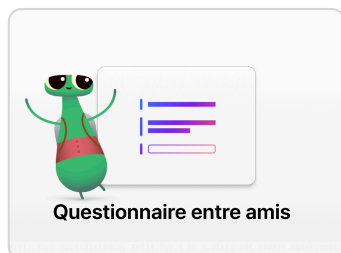
Les membres du club découvrent les bases de la programmation dans Apprendre à coder 1 et 2 de Swift Playgrounds. Ils mettent à profit leurs nouvelles compétences pour créer un playground qui réagit aux touches.

[Consulter le module >](#)



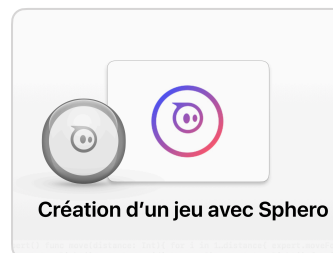
Conception d'une app

Les membres du club collaborent à la création d'une app visant à régler un problème dans leur communauté. Ils prennent part à un processus de conception qui leur enseigne à conceptualiser, à planifier, à prototyper et à évaluer leur propre app. [Consulter le module >](#)



Questionnaire entre amis

Les membres du club misent sur les compétences acquises dans le module Création d'un projet pour s'attaquer à des puzzles avancés dans Apprendre à coder 1 et 2 de Swift Playgrounds. Ils créent un playground qui interagit avec l'utilisateur dans un échange de questions et de réponses. [Consulter le module >](#)



Création d'un jeu avec Sphero

Les membres du club donnent vie à Sphero pour revisiter les classiques des jeux d'arcade. Ensemble, ils examinent et modifient le code du jeu pour créer leur propre expérience. Ils font appel à leurs compétences pour concevoir un jeu à l'aide d'un ou de plusieurs robots Sphero. [Consulter le module >](#)



3. Allez plus loin

Il est également possible d'organiser des séances axées sur les champs d'intérêt des membres. Vous pouvez approfondir les activités de conception et de programmation en proposant diverses expériences, comme l'exploration d'un appareil connecté, la création d'une course à obstacles de drones ou l'invention d'une mission de sauvetage pour un robot.

Pour stimuler le remue-ménages, vous pourriez même inviter des conférenciers ou planifier des sorties éducatives afin d'aider les membres du club à mieux comprendre le public de leur projet et les exigences associées à sa conception.



Célébration



Événement communautaire ou exposition d'apps

Mobilisez la communauté en organisant un événement communautaire ou une exposition d'apps permettant de découvrir comment la programmation peut nous aider à relever les défis du monde d'aujourd'hui. C'est aussi l'occasion parfaite de mettre en valeur le talent des membres de votre club!

1. Planifiez le grand événement. Fixez une date et invitez élèves, enseignants, parents et membres de la communauté à y assister.

Prévoyez du temps pour permettre à chaque équipe de présenter son projet et de répondre à quelques questions. Si vous avez un grand groupe, vous pouvez scinder l'exposition en deux et permettre aux membres d'assister aux présentations de leurs camarades.

Pour terminer en beauté, vous pouvez projeter un diaporama amusant montrant des photos prises tout au long des séances.



2. Décernez des prix de conception. Une compétition amicale peut être une grande source de motivation. Encouragez les membres du club à se surpasser en leur remettant des prix qui récompensent certains mérites en programmation et en conception, par exemple :

- Meilleure ingénierie
- Meilleure innovation
- Meilleur design
- Meilleure présentation

Vous pouvez aussi inciter les spectateurs à participer en décernant un prix du public.



Téléchargez et modifiez ce [certificat](#) pour l'adapter aux différents prix.

3. Recrutez des juges et des mentors. Il peut s'agir d'enseignants ou de membres du personnel, d'élèves ayant de l'expérience en programmation, d'experts en développement ou en conception, de membres de la commission scolaire, de dirigeants locaux ou même de personnes qui pourraient profiter du projet.

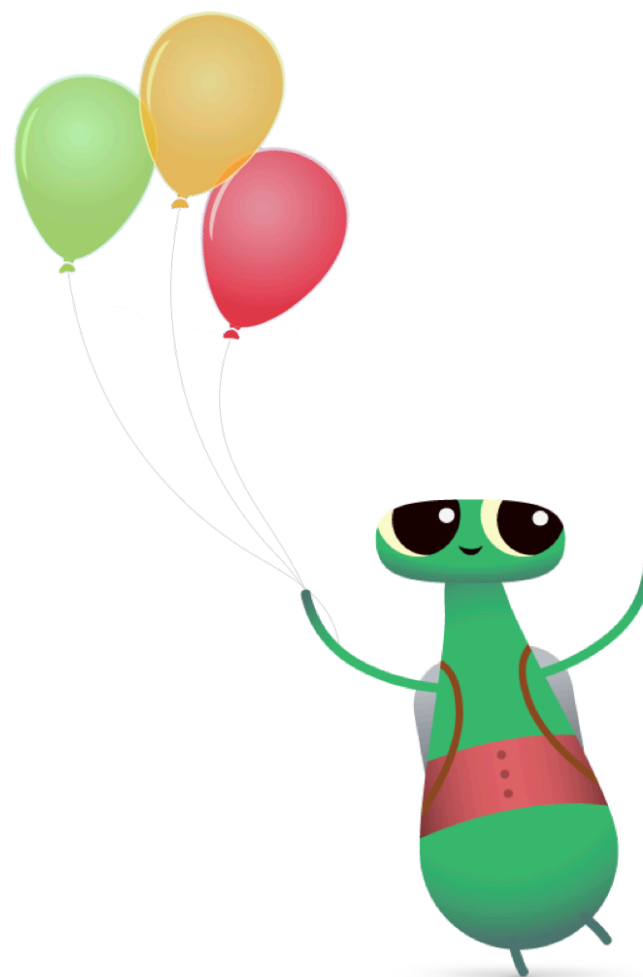
Les juges n'ont pas besoin d'attendre le jour de l'exposition pour rencontrer les membres. Vous pouvez les inviter à venir partager leur expertise pendant la phase de remue-méninges ou de planification des projets.

4. Partagez les présentations et inspirez les autres.

Si le cœur vous en dit, filmez les présentations. Partagez votre enregistrement avec les membres de la communauté et présentez les faits saillants dans une courte vidéo qui incitera d'autres jeunes à se joindre à votre club.

Pour plus de conseils et d'idées, consultez le Guide pour l'exposition d'apps.

[Télécharger le Guide pour l'exposition d'apps >](#)





Club de programmation Swift

La programmation pour tous

Certificat de réussite

Décerné à

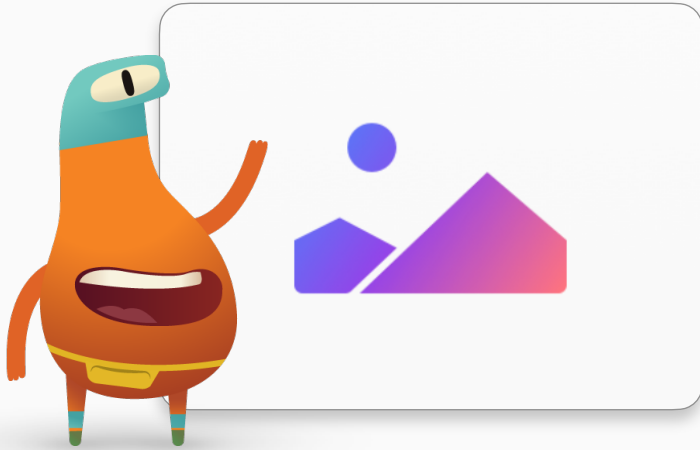
Pour



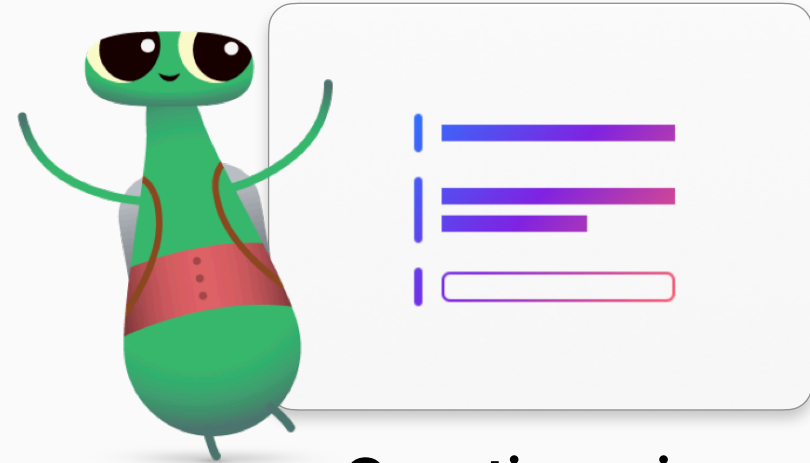
Signature

Date

Modules du Club de programmation Swift



Création d'un projet



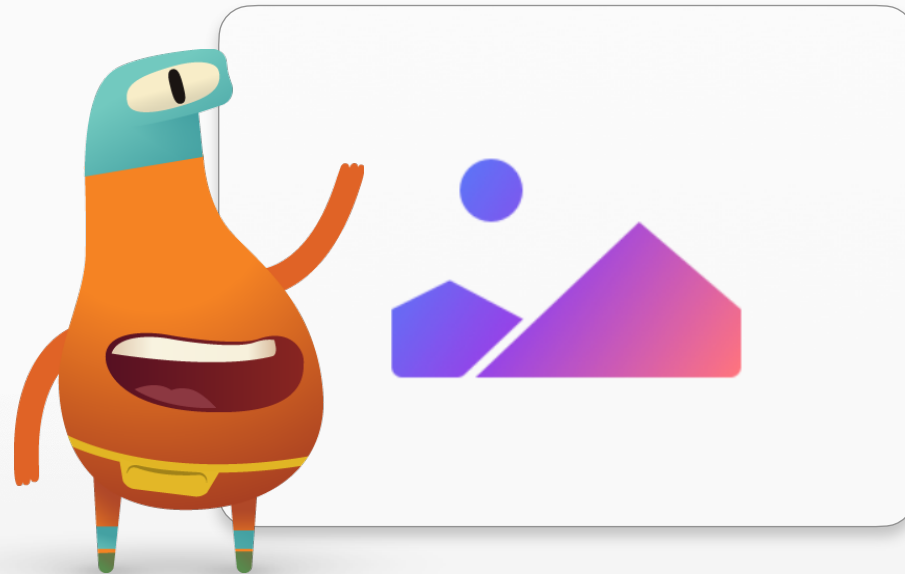
**Questionnaire
entre amis**



Conception d'une app

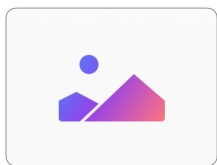


**Création d'un jeu
avec Sphero**



Création d'un projet

```
func show(from actor: Actor) {guard state == .inactive else { return }state = .animatingToPicker// Use the idle for the c
commandSpeed = WorldConfiguration.Actor.idleSpeedoriginalWorldActor = actororiginalActorTransform =
sceneNode.transformactor.reset()actor.sceneNode.runAction(.playSoundEffect(tapSource)) // Reset the existing `pickerActors`.
d).for pickerActor in pickerActors {pickerActor.reset()pickerActor.isInCharacterPicker = true} let result = actor.perform
result.completionHandler = { _ inguard self.state == .animatingToPicker else TriggerDepthOfField(intro: true) { [unowned
adAndDisplayCharacters()}}}func triggerDepthOfField(intro: Bool, completion: CompletionBlock? = nil) {let root = scene?.r
ode = root?.childNode(withName: "camera", recursively: true),let camera = cameraNode.camera else { return } SCNTransaction
saction.animationDuration = CharacterPickerController.fadeDurationcamera.focusDistance = intro ? focusDistanceMax : 0came
de : fStopDefault SCNTransaction.completionBlock = completionSCNTransaction.commit()private func oadAndDisplayCharacters
return } / Setup overlay view & constraoverlayView.alpha = 0overlayView.autoresizingMask = [.flexibleWidth, .flexibleHeig
ldSubview(overlayView).view.bounds // Add a gesture recognizer to detect when character selection happens.let tapGesi
ction: #selector(selectCharacter(:)) overlayView.gestureRecognizers = [tapGesture // Setup overlay scene
```



Création d'un projet

Survol du module

Dans le cadre de ces séances, les membres du club découvrent les bases de la programmation en participant aux amusantes activités du guide Puzzles de La programmation pour tous. Ils s'exercent en s'attaquant à des puzzles dans Apprendre à coder 1 et 2 de Swift Playgrounds et mettent à profit leurs nouvelles compétences pour créer un playground qui réagit aux événements tactiles.

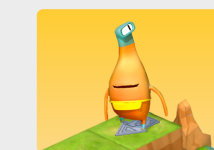
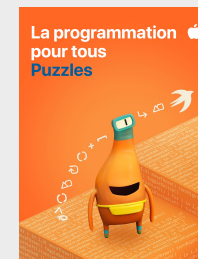
Dans les séances Exploration et Exercices, les membres du club découvrent les concepts clés et les mettent en pratique dans les puzzles et les défis de Swift Playgrounds. Dans les séances Mise en pratique et Application concrète, ils s'intéressent au code en tant qu'outil pour explorer des idées et créer de nouveaux produits. Au terme des séances, vous pouvez organiser un événement communautaire dans le cadre duquel les membres du club présenteront leurs projets.

Pour en savoir plus sur chaque activité, consulter des ressources supplémentaires et découvrir les façons de soutenir ou de stimuler les membres de votre club, jetez un coup d'œil au [guide de l'enseignant Puzzles de La Programmation pour tous](#).

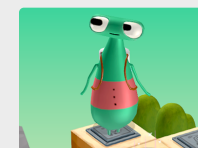
Au programme

- Exploration et Exercices : 6 séances
- Mise en pratique et Application concrète : 6 séances
- Événement communautaire

Ressources



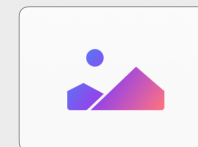
Apprendre à coder 1



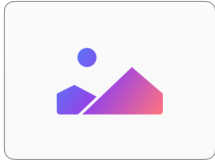
Apprendre à coder 2



Spirales



Formes



Création d'un projet

1 Les commandes

Explorez la notion de commande – une instruction précise donnée à un ordinateur. Apprenez à utiliser des commandes dans une séquence particulière.

Exploration : Regardez l'introduction du chapitre Commandes dans Apprendre à coder 1

Cache-cache (page 3)

Exercices : Tentez de résoudre les puzzles du chapitre Commandes dans Apprendre à coder 1 (pages 4 à 10)

Apprendre à coder 1 Commandes

- Introduction
- Création de commandes
- Ajout d'une nouvelle commande
- Activation d'un interrupteur



2 Les fonctions

Apprenez à créer vos propres commandes en programmant et en appelant des fonctions.

Exploration : Regardez l'introduction du chapitre Fonctions dans Apprendre à coder 1 Origami (page 15)

Exercices : Tentez de résoudre les puzzles du chapitre Fonctions dans Apprendre à coder 1 (pages 16 à 21)

Apprendre à coder 1 Fonctions

- Introduction
- Composition
- Création d'une nouvelle fonction
- Imbrication de patterns



3 Les boucles *for*

Familiarisez-vous avec les boucles *for* et les façons de programmer efficacement avec des fonctions et des boucles.

Exploration : Regardez l'introduction du chapitre Boucles for dans Apprendre à coder 1

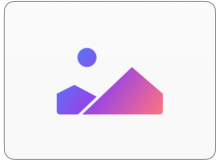
Création d'un motif (page 26)

Exercices : Tentez de résoudre les puzzles du chapitre Boucles for dans Apprendre à coder 1 (pages 27 à 31)

Apprendre à coder 1 Boucles for

- Introduction
- Utilisation des boucles
- Boucler tous les côtés





Création d'un projet

4 Les variables

Découvrez comment les ordinateurs se servent des variables pour stocker des renseignements, et apprenez à programmer avec ces variables.

Exploration : Regardez l'introduction du chapitre Variables dans Apprendre à coder 2 Bot-journaliste (page 36)

Exercices : Tentez de résoudre les puzzles du chapitre Variables dans Apprendre à coder 1 et du défi Spirales (pages 37 à 43)

Apprendre à coder 2 Variables

- Introduction
- Compter



Spirales

- Aperçu
- Hypocycloïdes
- Épicycloïdes
- Hypotrochoïdes
- Ellipses
- Récréation



5 Le code conditionnel

Familiarisez-vous avec la logique booléenne et apprenez à rédiger du code conditionnel.

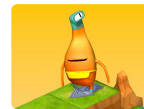
Exploration : Regardez l'introduction du chapitre Code conditionnel dans Apprendre à coder 1

Suivez les instructions (page 49)

Exercices : Tentez de résoudre les puzzles du chapitre Code conditionnel dans Apprendre à coder 1 (pages 50 à 56)

Apprendre à coder 1 Code conditionnel

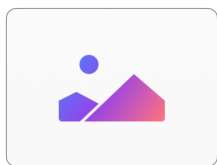
- Introduction
- Recherche d'interrupteurs
- Utilisation d'else if
- Code conditionnel en boucle
- Des fonctions plus intelligentes



6 La conception ciblée

Pensez aux différents points de vue des utilisateurs et à la façon de concevoir des produits en ciblant un public précis.

Application concrète : Adoptez le point de vue de quelqu'un d'autre (page 58)



Création d'un projet

7 Les types et l'initialisation

Apprenez à décrire les types et à les initialiser dans votre code.

Exploration : Regardez l'introduction des chapitres Types et Initialisation dans Apprendre à coder 2

Un produit bien conçu (page 62)

Exercices : Tentez de résoudre les puzzles des chapitres Types et Initialisation dans Apprendre à coder 2 (pages 63 à 66)

Apprendre à coder 2 Types

- Introduction
- Désactivation d'un téléporteur



Initialisation

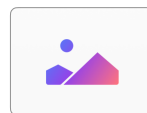
- Introduction
- Initialisation de ton expert
- Instances de différents types

8 Les formes interactives

Ouvrez le Point de départ Formes dans Swift Playgrounds, qui vous permettra de lancer votre projet dans les séances suivantes. Explorez les pages Créer, Toucher et Animer, et déterminez l'utilité et le fonctionnement de chaque section de code. En groupe, dressez la liste des éléments graphiques et des fonctions disponibles dans le Point de départ Formes.

Formes

- Créer
- Toucher
- Animer



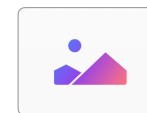
9 La création d'un projet Formes

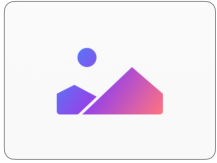
Découvrez comment créer un projet sur la coordination œil-main dans le Point de départ Formes. Peaufinez votre liste d'éléments graphiques et de fonctions.

Mise en pratique : Créez un projet sur la coordination œil-main (page 67)

Formes

- Canevas





Création d'un projet

10 La conception d'un projet

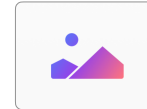
Discutez des projets que vous pourriez créer avec le Point de départ Formes. Songez aux éléments graphiques et aux fonctions disponibles, et déterminez comment ils pourraient répondre aux besoins du public visé. Partagez quelques idées en groupe, puis, en équipe de deux, formulez une idée originale en mettant en lumière l'adéquation entre le projet, son objectif et les besoins du public cible.

11 La création du projet

Toujours en équipe de deux, donnez vie à votre projet en entrant votre code dans la page Canevas du Point de départ Formes. Consultez vos notes de la dernière séance.

Formes

- Canevas

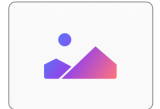


12 L'évaluation du projet

Testez votre playground auprès de vos camarades. Exercez-vous à en expliquer le fonctionnement et à justifier vos décisions pour vous préparer à présenter vos créations lors de l'événement communautaire.

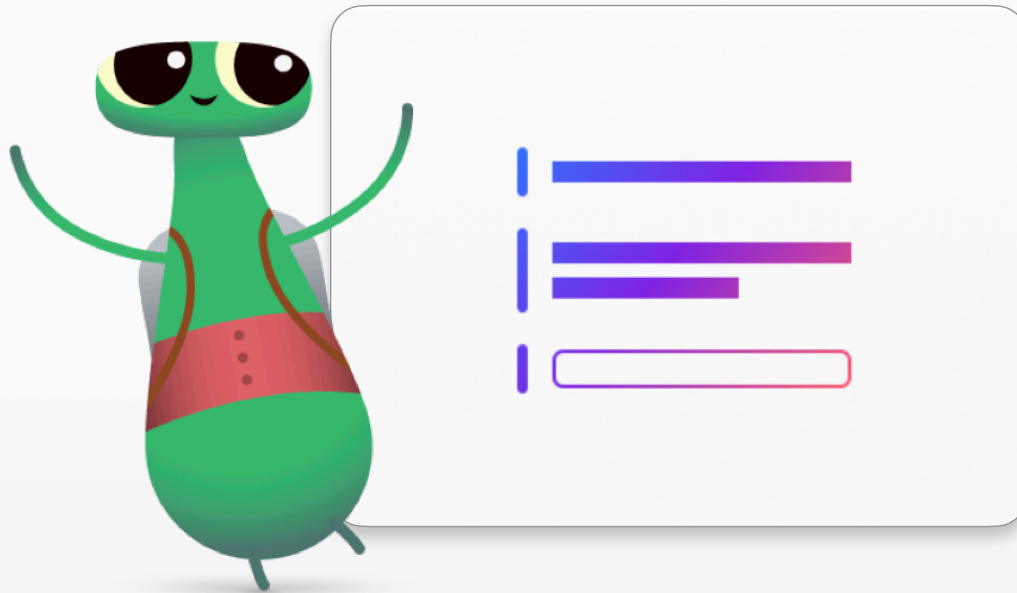
Formes

- Canevas



Événement communautaire

Célébrez les réalisations du club lors d'un événement communautaire où vous pourrez présenter votre projet, expliquer le processus de conception et recueillir les commentaires du public.



Questionnaire entre amis

```
func show(from actor: Actor) {guard state == .inactive else { return }state = .animatingToPicker// Use the idle for the c
commandSpeed = WorldConfiguration.Actor.idleSpeedoriginalWorldActor = actororiginalActorTransform =
sceneNode.transformactor.reset()actor.sceneNode.runAction(.playSoundEffect(tapSource))// Reset the existing `pickerActors`.
d).for pickerActor in pickerActors {pickerActor.reset()pickerActor.isInCharacterPicker = true} let result = actor.perform
result.completionHandler = { _ inguard self.state == .animatingToPicker else TriggerDepthOfField(intro: true) { [unowned
adAndDisplayCharacters()}}}func triggerDepthOfField(intro: Bool, completion: CompletionBlock? = nil) {let root = scene?.r
ode = root?.childNodes(withName: "camera", recursively: true),let camera = cameraNode.camera else { return } SCNTransaction
saction.animationDuration = CharacterPickerController.fadeDurationcamera.focusDistance = intro ? focusDistanceMax : 0came
de : fStopDefault SCNTransaction.completionBlock = completionSCNTransaction.commit()private func oadAndDisplayCharacters
return } / Setup overlay view & constraooverlayView.alpha = 0overlayView.autoresizingMask = [.flexibleWidth, .flexibleHeig
ldSubview(overlayView)view.bounds // Add a gesture recognizer to detect when character selection happens.let tapGesi
ction: #selector(selectCharacter( :))) overlayView.gestureRecognizers = [tapGesture // Setup overlay sc
```



Questionnaire entre amis

Survol du module

Dans ce module, les membres du club développent leurs compétences en participant à des activités plus complexes du guide Puzzles de La programmation pour tous. Ils s'exercent en s'attaquant à des puzzles dans Apprendre à coder 1 et 2 de Swift Playgrounds et mettent à profit leurs compétences avancées pour créer un playground qui interagit avec l'utilisateur dans un échange de questions et de réponses. Avant de commencer, les participants doivent comprendre les notions enseignées dans les chapitres 1 à 6 du guide Puzzles, avoir terminé le module Création d'un projet, ou posséder des connaissances équivalentes.

Dans les séances Exploration et Exercices, les membres du club découvrent les concepts clés et les mettent en pratique dans les puzzles et les défis de Swift Playgrounds. Dans les séances Mise en pratique et Application concrète, ils s'intéressent au code en tant qu'outil pour explorer des idées et créer de nouveaux produits. Au terme des séances, vous pouvez organiser un événement communautaire dans le cadre duquel les membres du club présenteront leurs projets.

Pour en savoir plus sur chaque activité, consulter des ressources supplémentaires et découvrir les façons de soutenir ou de stimuler les membres de votre club, jetez un coup d'œil au [guide de l'enseignant Puzzles de La Programmation pour tous](#).

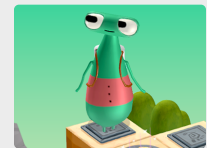
Au programme

-  Exploration et Exercices : 4 séances
-  Mise en pratique et Application concrète : 8 séances
-  Événement communautaire

Ressources



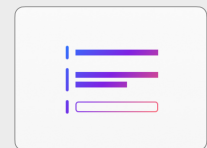
Apprendre à coder 1



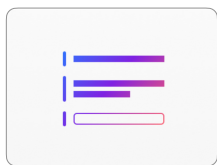
Apprendre à coder 2



Pierre, feuille, ciseaux



Réponses



Questionnaire entre amis

1 Les fonctions avec des paramètres

Découvrez comment donner plus d'information aux ordinateurs en rehaussant la précision des fonctions avec des paramètres.

Exploration : Regardez l'introduction du chapitre Paramètres dans Apprendre à coder 2

La recette du succès (page 71)

Exercices : Tentez de résoudre les puzzles du chapitre Paramètres dans Apprendre à coder 2 (pages 72 à 75)

Apprendre à coder 2 Paramètres

- Introduction
- Avancer



2 La conception d'un jeu

Utilisez le défi Pierre, feuille, ciseaux dans Swift Playgrounds pour créer une version améliorée du jeu.

Mise en pratique : Créez un jeu de roche-papier-ciseaux (page 76)

Pierre, feuille, ciseaux

- Vue d'ensemble
- Personnaliser le jeu
- Ajouter des actions
- Ajouter des actions masquées
- Ajouter des adversaires



3 Les opérateurs logiques

Apprenez à utiliser les opérateurs logiques pour programmer des comportements précis en réponse à certaines conditions.

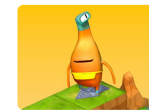
Exploration : Regardez l'introduction du chapitre Opérateurs logiques dans Apprendre à coder 1

Suivez les instructions (partie 2) (page 81)

Exercices : Tentez de résoudre les puzzles du chapitre Opérateurs logiques dans Apprendre à coder 1 (pages 82 à 85)

Apprendre à coder 1 Opérateurs logiques

- Introduction
- Utilisation de l'opérateur PAS
- Vérifier ceci ET cela
- Vérifier ceci OU cela





Questionnaire entre amis

4 La création d'un questionnaire

Dans le Point de départ Réponses de Swift Playgrounds, créez un questionnaire en combinant vos connaissances en matière d'opérateurs, de conditions, de variables, de fonctions et de paramètres.

Mise en pratique : Créez un questionnaire (page 86)

5 La conception d'un questionnaire

En vous basant sur le Point de départ Réponses, imaginez votre propre projet de questionnaire. Déterminez l'objectif du questionnaire, explorez les divers formats d'apps, ciblez un public et formulez votre idée.

6 Les boucles *while*

Familiarisez-vous avec les boucles *while*, qui servent à répéter une portion de code tant qu'une condition donnée est vraie.

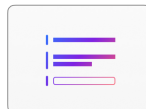
Exploration : Regardez l'introduction du chapitre Boucles while dans Apprendre à coder 1

Les règles du jeu (page 90)

Exercices : Tentez de résoudre les puzzles du chapitre Boucles while dans Apprendre à coder 1 (pages 91 à 94)

Réponses

- Texte
- Types



Apprendre à coder 1 Boucles while

- Introduction
- Exécution du code pendant...
- Boucles while plus intelligentes
- Imbrication de boucles





Questionnaire entre amis

7 Mise au point du questionnaire

Améliorez le questionnaire original en y intégrant différents modes à l'aide de boucles *while*. Ces compétences vous serviront lors des séances suivantes, quand vous rédigerez le code pour votre propre projet.

Mise en pratique : Peaufinez votre questionnaire (page 95)

8 Les tableaux et le réusinage

Dans cette séance, les membres du club acquerront de nouvelles compétences techniques en apprenant à se servir de tableaux, puis les mettront à profit pour réusiner le code.

Exploration : Regardez l'introduction du chapitre Tableaux dans Apprendre à coder 2 Évaluation (page 99)

Exercices : Tentez de résoudre les puzzles du chapitre Tableaux dans Apprendre à coder 2 (pages 100 à 105)

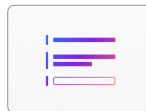
9 L'ajout de listes de choix au questionnaire

Intégrez des listes de choix à votre questionnaire, et réfléchissez aux autres projets que vous pourriez créer avec des listes de choix.

Mise en pratique : Ajoutez des listes de choix à votre questionnaire (page 106)

Réponses

- Texte
- Types



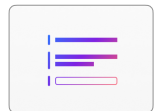
Apprendre à coder 2 Tableaux

- Introduction
- Stockage de données
- Exploration des itérations
- Empilement de blocs
- Mise dans l'ordre
- Corriger des erreurs d'index hors limites



Réponses

- Texte
- Types





Questionnaire entre amis

10 La conception d'un nouveau projet

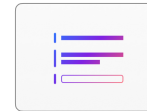
Discutez des projets que vous pourriez créer avec le Point de départ Réponses. Explorez les idées en groupe, puis travaillez individuellement pour en choisir une, fixer un objectif et cibler le public, et réaliser une maquette.

11 La création du projet

Créez votre propre projet dans le Point de départ Réponses. Laissez-vous guider par la maquette définie à la séance précédente.

Réponses

- Texte
- Types

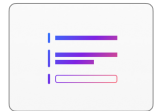


12 L'évaluation du projet

Testez votre playground auprès de vos camarades. Exercez-vous à en expliquer le fonctionnement et à justifier vos décisions pour vous préparer à présenter vos créations lors de l'événement communautaire.

Réponses

- Texte
- Types



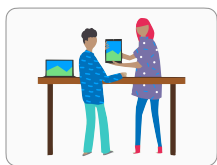
Événement communautaire

Célébrez les réalisations du club lors d'un événement communautaire où vous pourrez présenter votre projet, expliquer le processus de conception et recueillir les commentaires du public.



Conception d'une app

```
func show(from actor: Actor) {guard state == .inactive else { return }state = .animatingToPicker// Use the idle for the c
commandSpeed = WorldConfiguration.Actor.idleSpeedoriginalWorldActor = actororiginalActorTransform =
scnNode.transformactor.reset()actor.scnNode.runAction(.playSoundEffect(tapSource)) // Reset the existing `pickerActors`.
d).for pickerActor in pickerActors {pickerActor.reset()pickerActor.isInCharacterPicker = true} let result = actor.perform
result.completionHandler = { _ inguard self.state == .animatingToPicker else TriggerDepthOfField(intro: true) { [unowned
adAndDisplayCharacters()}}}func triggerDepthOfField(intro: Bool, completion: CompletionBlock? = nil) {let root = scene?.r
ode = root?.childNode(withName: "camera", recursively: true),let camera = cameraNode.camera else { return } SCNTransaction
saction.animationDuration = CharacterPickerController.fadeDurationcamera.focusDistance = intro ? focusDistanceMax : 0came
de : fStopDefault SCNTransaction.completionBlock = completionSCNTransaction.commit()private func loadAndDisplayCharacters
return } / Setup overlay view & constraooverlayView.alpha = 0overlayView.autoresizingMask = [.flexibleWidth, .flexibleHeig
ldSubview(overlayView)view.bounds // Add a gesture recognizer to detect when character selection happens.let tapGesti
ction: #selector(selectCharacter(:)) overlayView.gestureRecognizers = [tapGesture // Setup overlay sc
```



Conception d'une app

Survol du module

Dans ce module, les membres du club collaborent à la création d'une app visant à régler un problème dans leur communauté. Ils suivent un processus de conception qui consiste à trouver des idées, planifier leur projet, créer un prototype fonctionnel dans Keynote, puis évaluer l'app. Chaque équipe crée ensuite une démonstration vidéo qui relate le processus et présente l'app.

Le processus de conception est illustré dans un [journal de bord de conception](#), ce qui aide les membres du club à assurer le suivi de leurs idées tout au long du cycle de conception. Il s'agit de décrire le processus pour améliorer le projet d'app par itération. Les membres pourront aussi s'y reporter et s'en servir comme point de départ à l'avenir.

Au terme de ce module, organisez une exposition d'apps pour célébrer l'ingéniosité des membres du club. Pour obtenir des astuces et des ressources pour planifier votre événement, téléchargez le [Guide pour l'exposition d'apps](#).

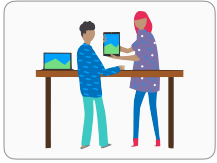
Au programme

- Remue-méninges : 3 séances
- Planification : 2 séances
- Prototypage : 4 séances
- Évaluation : 2 séances
- Présentation : 1 séance
- Exposition d'apps

Ressources



**Journal de bord de
conception d'app**



Conception d'une app

1–3

Remue-méninges

Explorez diverses pistes, déterminez l'utilité et le public cible de votre app, et sélectionnez des idées.

Remue-méninges

- Utilité
- Idées
- Public cible
- Sélection
- Répétition



4–5

Planification

Pensez à la façon dont vous utiliserez les fonctionnalités d'iPadOS dans votre app, et réfléchissez aux éléments essentiels d'une bonne interface utilisateur (IU).

Planification

- Interface et expérience utilisateur
- Fonctionnalités iPadOS
- Design



6–9

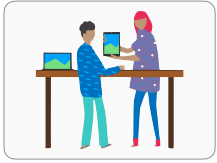
Prototypage

Concevez l'interface utilisateur de votre app, planifiez les écrans, et créez un prototype fonctionnel dans Keynote.

Prototypage

- Conception
- Maquette
- Construction





Conception d'une app

10-11 Évaluation

Mettez votre prototype à l'essai auprès de vos camarades et des membres de votre communauté, puis apportez des modifications en fonction des commentaires reçus.

Évaluation

- Observation
- Entrevue



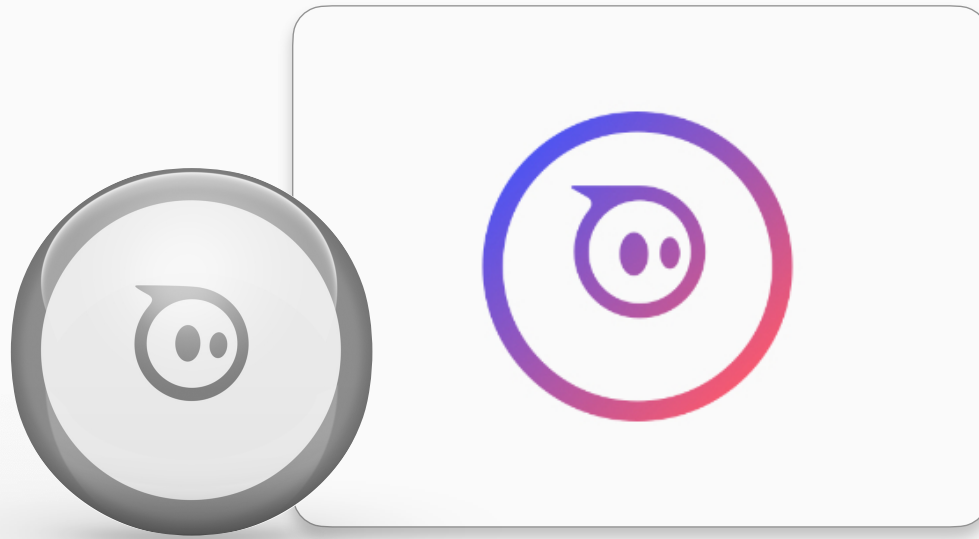
12 Présentation de l'app

Préparez un exposé ou une vidéo de trois minutes décrivant le fonctionnement de votre app et le problème qu'elle vise à régler.



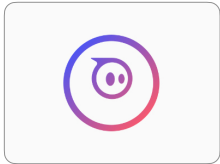
Exposition

Partagez les prototypes et les présentations d'apps du club avec la communauté dans le cadre d'une exposition d'apps. Inspirez-vous du [Guide pour l'exposition d'apps](#) pour organiser l'événement.



Création d'un jeu avec Sphero

```
func show(from actor: Actor) {guard state == .inactive else { return }state = .animatingToPicker// Use the idle for the c
commandSpeed = WorldConfiguration.Actor.idleSpeedoriginalWorldActor = actororiginalActorTransform =
cnNode.transformactor.reset()actor.scnNode.runAction(.playSoundEffect(tapSource)) // Reset the existing `pickerActors`.
d).for pickerActor in pickerActors {pickerActor.reset()pickerActor.isInCharacterPicker = true} let result = actor.perform
result.completionHandler = { _ inguard self.state == .animatingToPicker else TriggerDepthOfField(intro: true) { [unowned
adAndDisplayCharacters()}}}func triggerDepthOfField(intro: Bool, completion: CompletionBlock? = nil) {let root = scene?.r
ode = root?.childNode(withName: "camera", recursively: true),let camera = cameraNode.camera else { return } SCNTransaction
saction.animationDuration = CharacterPickerController.fadeDurationcamera.focusDistance = intro ? focusDistanceMax : 0came
de : fStopDefault SCNTransaction.completionBlock = completionSCNTransaction.commit()private func oadAndDisplayCharacters
return } / Setup overlay view & constraooverlayView.alpha = 0overlayView.autoresizingMask = [.flexibleWidth, .flexibleHeig
ldSubview(overlayView)view.bounds // Add a gesture recognizer to detect when character selection happens.let tapGesti
ction: #selector(selectCharacter(:)) overlayView.gestureRecognizers = [tapGesture // Setup overlay sc
```



Création d'un jeu avec Sphero

Survol du module

Dans ce module, les membres du club utilisent Swift Playgrounds pour donner vie à Sphero et revisiter les classiques des jeux d'arcade. Pour ce faire, vous devez disposer d'au moins un robot Sphero par équipe de deux participants.

Les membres du club explorent les données recueillies par Sphero et la façon dont elles peuvent faciliter la création de jeux interactifs. Ensemble, ils se familiarisent avec le code requis pour créer le jeu, puis modifient le code pour façonner une expérience unique.

Ensuite, les membres du club mettent en pratique leurs connaissances pour concevoir un jeu à l'aide d'un ou de plusieurs robots Sphero. Lors d'un événement communautaire, ils invitent le public à examiner leurs jeux ou à en faire l'essai, et expliquent les décisions qu'ils ont prises en matière de conception et de programmation.

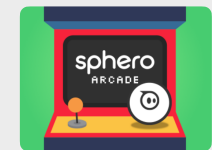
Au programme

- Sphero Pong : 3 séances
- Sphero Bop It : 2 séances
- Sphero Pac-Man : 2 séances
- Conception d'un jeu : 5 séances
- Événement communautaire

Ressources



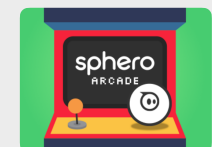
Mini-robot Sphero
(un par équipe de deux)



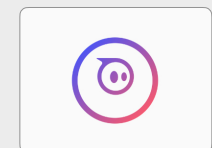
Sphero Arcade 1



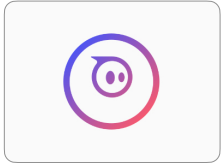
Sphero Arcade 2



Sphero Arcade 3



Modèle Sphero



Création d'un jeu avec Sphero

1 Sphero Pong

Explorez Sphero Arcade 1 dans Swift Playgrounds. Apprenez à animer Sphero, puis ouvrez la page Pong Original et jouez au jeu en équipe de deux. Déterminez le code dont vous aurez besoin pour créer le jeu, formulez des idées, puis annotez avec du pseudocode.

2-3 Sphero Pong

En petits groupes, créez une version « dans le monde réel » du jeu Sphero Pong, où vos pieds serviront de raquettes. Au terme de la séance 3, examinez le code dont vous croyiez avoir besoin pour créer le jeu, et déterminez s'il est suffisant.

4-5 Sphero Bop It

En équipe de deux, recréez le jeu Bop It avec Sphero. Explorez Sphero Arcade 2 pour apprendre à programmer chaque mouvement et rendre le jeu plus difficile. Modifiez le code pour créer vos propres mouvements, et réfléchissez au code dont vous pourriez avoir besoin pour associer l'interface visuelle du playground à Sphero.

Sphero Arcade 1

- Introduction
- Rouler
- Viser
- Direction
- Collisions
- Pong Original



Sphero Arcade 1

- Préparation dans le monde réel
- Angle de rebond
- Allers-retours
- Compter les points
- Remporter la partie
- Jouer au jeu



Sphero Arcade 2

- Présentation
- Toucher
- Lance
- Tourniquet
- Secouer
- Rendre le jeu aléatoire
- Niveaux de difficulté
- Jouer au jeu





Création d'un jeu avec Sphero

6-7 Sphero Pac-Man

En équipe de deux, recréez le jeu Pac-Man avec Sphero. Explorez Sphero Arcade 3 pour apprendre à programmer Sphero pour l'utiliser comme manette, à attribuer des points et à créer des ennemis. Modifiez le jeu pour augmenter le niveau de difficulté, et déterminez si du code supplémentaire est requis pour créer tous les éléments de l'interface visuelle.

Sphero Arcade 3

- Présentation
- Commandes simples
- Score
- Super pac-gommes
- Ennemis simples
- Ennemi avancé
- Jouer au jeu

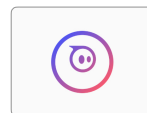


8-9 Labyrinthe

Programmez Sphero pour lui faire parcourir un labyrinthe dans le playground Modèle Sphero. Tracez un labyrinthe simple, puis reproduisez-le avec du ruban adhésif. Orientez Sphero avec le modèle Guidage et programmez-le pour qu'il parcoure le labyrinthe dans la page Modèle. Vous pouvez travailler en petits groupes, ou créer un seul labyrinthe et organiser une course pour déterminer quel robot terminera le plus rapidement et le plus facilement.

Modèle Sphero

- Modèle
- Guidage

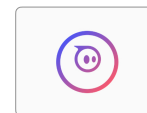


10-12 Conception d'un jeu

Faites un remue-méninges pour trouver des idées pour concevoir votre propre jeu avec Sphero. Il peut s'agir d'une version physique d'un jeu d'arcade, d'une course à obstacles ou même d'un jeu faisant appel à plus d'un robot Sphero. Dessinez et planifiez votre jeu, puis créez un projet de playground dans Modèle Sphero. N'oubliez pas de décortiquer les divers éléments de votre jeu et d'ajouter des commentaires dans votre code pour expliquer votre raisonnement.

Modèle Sphero

- Modèle
- Guidage



Événement communautaire

Célébrez les réalisations du club lors d'un événement communautaire où vous pourrez présenter votre projet, expliquer le processus de conception et recueillir les commentaires du public.

