KOTLIN

Développer des applications Android

HUGUES BERSINI

LE LANGAGE RECOMMANDÉ PAR GOOGLE

Coming next

Éditions EYROLLES

Un manuel ultra-pratique sur le langage de programmation le plus novateur depuis Java

Ce manuel d'initiation à Kotlin aborde ce nouveau langage selon une approche résolument concrète. Sans qu'il soit nécessaire d'avoir des connaissances des langages plus anciens, l'ambition de cet ouvrage est de vous apprendre à développer des applications pour les tablettes et smartphones fonctionnant sur Android.

Devenu incontournable depuis que Google l'a déclaré, en 2019, langage officiel pour la création d'applications mobiles Android, Kotlin s'avère entièrement compatible avec Java et il est aussi disponible en open source.

À partir de l'environnement Android Studio, vous pourrez très rapidement créer des animations graphiques (mini-jeux) tout en découvrant de façon ludique les bases de la programmation procédurale et orientée objet.

À qui s'adresse cet ouvrage ?

- **Aux développeurs néophytes et confirmés**
- **À tous ceux qui souhaitent découvrir les coulisses de la programmation moderne**

L'auteur

Membre de l'Académie Royale de Belgique, Hugues Bersini enseigne l'informatique et la programmation aux facultés polytechnique et Solvay de l'Université Libre de Bruxelles, dont il dirige le laboratoire d'Intelligence Artificielle. Il est l'auteur de très nombreuses publications (systèmes complexes, génie logiciel, sciences cognitives et bioinformatique) et de plusieurs ouvrages d'introduction à la programmation, l'intelligence artificielle et les systèmes complexes qui font aujourd'hui autorité dans le monde académique.

Sommaire

Introduction • LE PETIT BILLARD – Démarrer un projet – Premières fonctions Kotlin dans sa version procédurale – Allons-y pour le billard ! – Orienté objet : faire apparaître les balles à l'écran – L'orienté objet, une façon de programmer – Retour à notre petit billard – Ajoutons un peu d'animation à tout cela – Finalisons notre petit billard • UN JEU CANON – Les étapes préliminaires – Ajoutons un peu d'animation – Et maintenant, jouons ! – Finalisons le jeu : Game over

www.editions-eyrolles.com

Coming next

KOTLIN

Développer des applications Android

SUR LE MÊME THÈME

J. Launay. – **Créer des applis avec App Inventor.**
N° 67901, 2020, 118 pages.

B. Barré. – **Concevez des applications mobiles avec React Native.**
N° 67889, 2e édition, 2019, 224 pages.

S. Pittion, B. Siebman. – **Applications mobiles avec Cordova et PhoneGap.**
N° 14052, 2015, 186 pages.

DANS LA COLLECTION BLANCHE

C. Delannoy. – **Le guide complet du langage C.**
N° 67922, 2020, 876 pages.

J. Loriaux, T. Defossez. – **Emailing : développement et intégration.**
N° 67849, 2020, 160 pages.

S. Ringuedé. – **SAS.**
N° 67631, 4e édition, 2019, 688 pages.

C. Blaess. – **Développement système sous Linux.**
N° 67760, 5e édition, 2019, 1080 pages.

T. Parisot. – **Node.js.**
N° 13993, 2018, 472 pages.

C. Pierre de Geyer, J. Pauli, P. Martin, E. Daspet. – **PHP 7 avancé.**
N° 67720, 2e édition, 2018, 736 pages.

H. Wickham, G. Grolemund. – **R pour les data sciences.**
N° 67571, 2018, 496 pages.

F. Provost, T. Fawcett. – **Data science pour l'entreprise.**
N° 67570, 2018, 370 pages.

J. Chokogoue. – **Maîtrisez l'utilisation des technologies Hadoop.**
N° 67478, 2018, 432 pages.

R. Goetter. – **CSS 3 Flexbox.**
N° 14363, 2016, 152 pages.

Retrouvez nos bundles (livres papier + e-book) et livres numériques sur
http://izibook.eyrolles.com

KOTLIN

Développer des applications Android

HUGUES BERSINI

Coming next

Éditions EYROLLES

ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l'autorisation de l'Éditeur ou du Centre Français d'exploitation du droit de copie,

20, rue des Grands Augustins, 75006 Paris.

© Éditions Eyrolles, 2020

ISBN : 978-2-212-67968-7

PARTIE 1. INTRODUCTION

POURQUOI CE MANUEL ? 11

QUELQUES PROPOS INTRODUCTIFS À LA PRATIQUE DE LA PROGRAMMATION 23

PARTIE 2. LE PETIT BILLARD

1 DÉMARRER UN PROJET 33

Prendre en main Android Studio 33

Qu'est-ce qu'une activité dans Android ? 37

Votre première exécution sans sommation 42

2 PREMIÈRES FONCTIONS KOTLIN DANS SA VERSION PROCÉDURALE 49

Découpage en fonctions 50

1er exemple : bottledingding 51

2e exemple : calcul de pi 53

3e exemple : inversion d'une chaîne de caractères 55

3 ALLONS-Y POUR LE BILLARD ! 59

4 ORIENTÉ OBJET : FAIRE APPARAÎTRE LES BALLES À L'ÉCRAN 67

Création de la classe Balle 67

Création des objets balles 70

5 L'ORIENTÉ OBJET, UNE FAÇON DE PROGRAMMER 79

La collaboration entre classes et entre objets 79

UML (Unified Modeling Language) 82

Un petit exemple de compte en banque 84

6 RETOUR À NOTRE PETIT BILLARD 93

Réagir au clic 93

Donner des informations 94

7 AJOUTONS UN PEU D'ANIMATION À TOUT CELA 99

Les balles changent leur position… 99

… en permanence 101

Gérer les activités 105

8 FINALISONS NOTRE PETIT BILLARD 111

Ajout des parois 111

Programmons les rebonds 113

Dessiner les parois et animer le tout 117

PARTIE 3. UN JEU CANON

9 LES ÉTAPES PRÉLIMINAIRES 129

Le projet 129

Le canon 131

La balle 138

L'obstacle 141

Les cibles 143

10 AJOUTONS UN PEU D'ANIMATION 149

Déplacer l'obstacle et les cibles 149

Gérer le tir 152

Gérer les interactions de la balle 155

11 ET MAINTENANT, JOUONS ! 161

12 FINALISONS LE JEU : GAME OVER 167

Remise à zéro 167

Partie perdue 168

Partie gagnée 169

Message de fin de partie 170

Ajouter du son 176

Ajouter une icône d'application 179

INDEX 181

PARTIE 01

>INTRODUCTION

INTRODUCTION

POURQUOI CE MANUEL ?

LES DÉFIS

Ce manuel d'initiation à la programmation a l'ambition déraisonnable mais audacieuse (me le reprocherez-vous la dernière page tournée ?) de relever plusieurs défis :

1. Vous initier aux deux pratiques fondamentales de la programmation, *procédurale* et *objet*, en ne supposant aucun prérequis. Ce sont les bases de ces deux pratiques que nous allons découvrir ensemble. Bien sûr, ceux qui possèdent déjà quelques rudiments de programmation sont plus que bienvenus (qu'ils ne se détournent pas immédiatement) et devraient trouver, dans la réalisation de ces animations sur Android, de quoi élargir leurs compétences par des acquis de plus en plus prisés dans l'industrie du logiciel. Comptez le nombre de personnes que vous croisez chaque jour, le nez plaqué sur l'écran de leur smartphone… de futurs clients.
2. Utiliser un langage de programmation des plus modernes et des plus prometteurs, bien qu'encore très peu enseigné : *Kotlin* (une autre île, russe et à 25 km de Saint Pétersbourg, dont tout

le monde jusque-là ignorait l'existence, après celle de Java…[1]). Depuis 2019, il s'est imposé comme le langage de prédilection d'un géant technologique, Google, pour le développement d'applications pour smartphones. Il est devenu le nouveau langage avec lequel il va falloir compter dans les années à venir. Naviguez sur le Web et vous vous convaincrez qu'il a le vent du développement logiciel bien en poupe.

3. Développer des applications pour cette famille d'appareils la plus utilisée aujourd'hui : les smartphones, essentiellement la catégorie dépendant du système d'exploitation Android de Google, qui représente actuellement un bon 85 % des parts de marché. Par ailleurs, pour les afficionados de la pomme à peine entamée, Kotlin présente de très nombreuses similitudes avec le langage de programmation Swift, très majoritairement utilisé pour la programmation des smartphones Apple. Il est donc possible, en apprenant Kotlin, de faire d'une pierre logicielle deux coups de smartphone.
4. Réaliser des animations graphiques et ludiques s'exécutant sur votre smartphone. L'auteur de ce manuel a croisé dans les transports publics, les bancs de square, les salles de classe, les auditoires universitaires, les chambres d'enfant, partout en fait… tellement de jeunes et de moins jeunes, concentrés fiévreusement sur l'un ou l'autre jeu de leur smartphone, qu'il a décidé de les initier à la programmation intelligente de ces jeux plutôt qu'à leur pratique abrutissante. Le pari ici est qu'il est bien plus stimulant et musclant au plan neuronal de programmer ces jeux, que de seulement s'y adonner. Et, bien évidemment, les programmeurs/*gamers* seront aux anges à la découverte de Kotlin.

1. On attend impatiemment le langage Kos ou le Tremiti…

5. Vous enseigner les bases de ce nouveau langage de programmation sans faire la moindre allusion à tous ceux qui l'ont précédé (pour moi, *geek* vénérable des temps anciens, programmeur sapiens, il s'agit sans doute du défi le plus exigeant[2]) : Simula, C, C++, Smalltalk, Java, Scala, C#, Objective-C, Ruby, Python et beaucoup d'autres, dont il s'est bien évidemment très largement inspiré.

En effet, un nouveau langage s'inscrit systématiquement dans une généalogie dont il hérite de très nombreuses fonctionnalités. Les concepteurs de tout nouveau langage de programmation sont à la recherche d'un Graal inédit, hors de portée, en quête du compromis idéal entre la facilité d'usage, l'économie d'écriture (tout en maintenant la lisibilité), l'efficience (en temps de calcul, en consommation de mémoire), la stabilité (il ne faut pas que le programme plante au beau milieu de son exécution), la réutilisabilité (il doit être facile de reprendre une partie du programme pour un autre usage), l'universalité (il doit s'exécuter indifféremment sur toutes les plates-formes) et bien d'autres objectifs contradictoires encore…

Il est alors très courant et assez naturel que, pour en faciliter la compréhension par les collègues informaticiens et les convaincre que le Graal est en vue, la présentation de cette nouvelle technologie se fasse systématiquement en la comparant à l'existant, afin d'en pointer les innovations ou, au contraire, chez les opposants, d'en dénigrer les tentatives (le Web est saturé d'analyses comparatives, des plus pacifiées aux plus guerrières). Plus un

2. D'autant que j'ai écrit un livre dans la même maison d'édition, faisant la part belle à ces comparaisons entre langages de programmation : *La programmation orientée objet*. Je pratique donc l'exercice, mais le trouve de plus en plus lassant, de surcroît lorsqu'il s'agit d'une initiation.

langage est récent, plus le Web s'en empare et plus, pour les programmeurs aguerris, ces comparaisons avec les prédécesseurs s'imposent comme un exercice de style dont ils raffolent. Ce n'est évidemment pas nécessairement du goût des débutants, qui n'ont, eux, que faire de cette reconnaissance de dette, ces accusations de plagiat et ces guerres de religions technologiques. Et c'est à eux que je m'adresse essentiellement. Donc du Kotlin et uniquement du Kotlin, comme si rien avant n'avait existé, comme si le premier ordinateur programmable datait de l'année passée, comme si la planète ne contenait qu'une seule île.

Un petit mot de Java malgré tout, le papa du jeune Kotlin (et ce sera le seul, je vous le promets) ! Jusque-là, la programmation sur Android s'effectuait essentiellement en Java. Il était donc indispensable pour les ingénieurs de Google, en cas d'innovation, de rester très proches de l'existant pour ne pas égarer les programmeurs, de maintenir cet existant et de permettre la cohabitation des deux langages ; obligations tenues. On parle dans ce cas de rétrocompatibilité. L'exercice comparatif se fait donc le plus fréquemment avec et au détriment de Java. Dans leur version exécutable, c'est-à-dire lors du passage aux instructions élémentaires qu'exécutera le processeur de votre smartphone, les deux langages convergent complètement, deviennent indissociables. Ils peuvent idéalement se mêler l'un à l'autre et une partie de code Kotlin peut, sans problème, faire appel à une partie de code Java. En fait, Kotlin refait un détour par Java dès qu'il s'agit de passer à l'exécution du programme. La raison pour laquelle Kotlin a de très beaux jours devant lui est que Google a décidé de l'adopter pour sa simplicité, sa concision et sa fiabilité d'usage. Et ce que Google veut…

LES IMPASSES

Ce manuel fera également de nombreuses impasses, dont il est plus qu'honnête d'avertir le lecteur. Il n'est qu'une modeste initiation à la programmation et nullement une exploration en profondeur du système d'exploitation Android, indispensable pourtant à l'exécution des programmes expliqués dans ses pages. Afin de développer et d'exécuter les applications qui vont suivre, il est nécessaire d'installer le logiciel gratuit Android Studio, qu'il est facile de se procurer sur le site web *https://developer.android.com/studio* et qui tourne sur tous les ordinateurs (Windows, Linux ou macOS). Cet environnement de développement est idéal pour la programmation Kotlin sur Android, car il a été principalement conçu pour cela. Il intègre en son sein des émulateurs de la plupart des smartphones (toutes les générations successives de ce système d'exploitation et tous les appareils qui l'intègrent), ce qui vous permet de réaliser et tester la bonne exécution de vos applications, sans obligatoirement avoir votre smartphone à portée de main.

Toutefois, cette exécution d'un émulateur de smartphones sur votre ordinateur exige une quantité considérable de mémoire RAM ainsi qu'un processeur assez récent. Vous serez souvent surpris par la lenteur à l'exécution des émulateurs et même par des « plantages » intempestifs. Il est grandement recommandé, dès lors, de vite relier votre smartphone à votre ordinateur (via un connecteur USB), afin de tester et d'exécuter vos applications directement sur l'appareil (l'émulateur n'a donc plus de raison d'être), le même destiné finalement à les héberger. Quelques ajustements et manipulations s'imposeront sur votre smartphone afin d'autoriser pareille installation et prise de contrôle.

Rassurez-vous, le Web regorge de tutoriels, de « Youtubeurs » et de sites qui vous indiqueront la marche à suivre.

Afin que ce manuel reste léger et vise à l'essentiel, c'est-à-dire les bases de la programmation, Android Studio sera également très peu décortiqué. Il s'agit pourtant d'un environnement de développement puissant, un des plus sophistiqués à ce jour (conçu par les mêmes concepteurs que Kotlin) et dont l'utilisation efficace requiert une formation à proprement parler, mais que ce manuel n'a pas vocation à donner. Il va donc falloir faire sans (ou avec l'ami Google) et ne découvrir ici que le juste nécessaire pour écrire et lancer nos applications.

LES DEUX ANIMATIONS

Deux animations graphiques seront réalisées dans ce manuel et décrites de la première à la dernière ligne de code. Je les ai nommées *lepetitbillard* et *unjeucanon*.

La première n'est pas un jeu à proprement parler et s'avère assez élémentaire conceptuellement.