

APP INVENTOR

Créez votre application Android

FABIEN JONQUIÈRE [1]



Les applications mobiles ont envahi notre quotidien : 30 milliards d'objets connectés sont annoncés pour 2020 ! La génération actuelle de nos élèves, née dans cette nouvelle ère du numérique, s'en approprie les usages naturellement.

Nos pratiques pédagogiques et nos supports de formation doivent prendre en compte cette révolution. Tel est le sens initial du développement de la plate-forme App Inventor. Son objectif est de permettre la réalisation d'applications communicantes en s'affranchissant des syntaxes de programmation, en ne se concentrant que sur l'algorithme. Réaliser une application sous Android devient aussi facile que construire une structure en Lego !

À l'origine, App Inventor a été lancé par Google, fin 2010, pour répondre à un constat. L'idée de ses créateurs était de s'appuyer sur la relation forte des élèves à leur smartphone pour proposer une formation aux concepts informatiques par la création d'applications mobiles. Ainsi est né Google App Inventor.

Son principe se base sur la thèse de Ricarose Roque, ancienne étudiante au MIT (Massachusetts Institute of Technology, voir les « Références » en encadré). Il rappelle certains langages de programmation simplifiés des années quatre-vingts. Hal Abelson, professeur au MIT, cofondateur de la licence Creative Commons, à l'origine du portage du langage Logo (la « géométrie de la tortue » créée par Seymour Papert en 1967) sur Apple 2, au début des années quatre-vingts, a dirigé le projet.

Depuis, le numérique est devenu un enjeu national outre-Atlantique, et Barack Obama s'est adressé en personne aux étudiants dans une vidéo postée sur YouTube le 15 décembre dernier : « Ne vous contentez pas d'acheter un nouveau jeu vidéo, créez-le ! Ne vous contentez pas de télécharger la dernière application, aidez à la concevoir ! Ne vous contentez pas de jouer sur votre smartphone, programmez-le ! » (www.youtube.com/watch?v=6XvmhE1J9PY#t=18).

Le développement d'App Inventor est assuré depuis fin 2011 par le MIT Center for Mobile Learning @ The

mots-clés
programmation,
transmission
d'informations

Media Lab, toujours sous la direction de Hal Abelson. App Inventor 2 vient d'arriver... Je m'appuierai donc ici sur cette nouvelle version, plus aboutie, qui sera nommée simplement App Inventor ou AI.

App Inventor est accessible gratuitement via un navigateur Internet, comme la plupart des applications du MIT.

Le concept d'App Inventor

La programmation se réalise en ligne, à l'aide de son navigateur préféré, et sous l'environnement logiciel de son choix (Mac, Linux ou Windows). Seules contraintes : avoir un compte Gmail pour pouvoir y accéder, et un accès à Internet. Vous n'avez pas forcément besoin d'un téléphone ou d'une tablette pour débiter : vous avez la possibilité de tester vos applications sur votre PC par le biais d'un émulateur Android téléchargeable.

Basé sur une méthode visuelle de programmation de type Scratch (voir en encadré), qui a fait ses preuves même dans les écoles primaires, App Inventor réduit considérablement les obstacles à la création d'applications. C'est un outil visuel dit *drag-and-drop* (glisser-déposer), où vous spécifiez le comportement de l'application en assemblant des « blocs » comme si vous construisiez un puzzle **1**.

Vous souhaitez que votre téléphone vous dise « bonjour », ou réaliser un programme qui déclenche une alarme lors d'un choc ? rien de plus simple **2**.

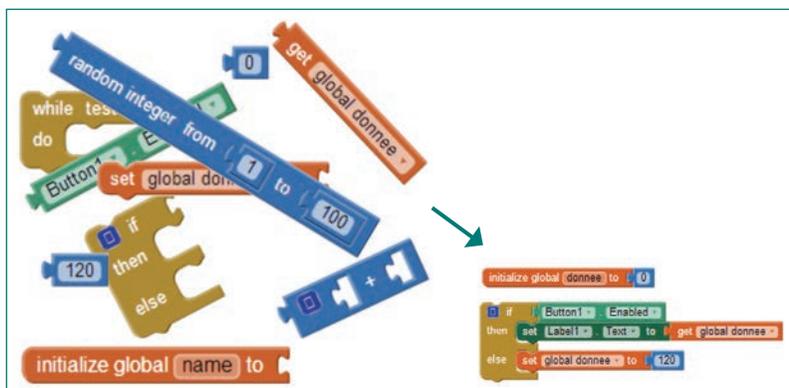
L'interface de développement

Trois fenêtres sont proposées pendant le développement **3** : une pour la création de l'interface, une pour la programmation elle-même, et une pour l'émulateur. Ce dernier pourra avantageusement remplacer un terminal réel pour la vérification du programme, bien que certaines fonctionnalités comme les accéléromètres, disponibles dans la programmation, ne soient pas émulées.

La connexion de votre terminal Android permettra ensuite d'y télécharger le programme pour un test « grandeur nature » sans procéder à son installation proprement dite. Ce terminal pourra aussi bien être un téléphone qu'une tablette haut de gamme, peu importe, le comportement du programme sera identique. Il vous suffit de lancer l'application AICompanion sur votre terminal, à télécharger à l'adresse suivante :

<https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion2>

[1] Professeur agrégé en SII option ingénierie électrique au lycée Chevrolier d'Angers (49) et animateur Éduscol STI « énergie et information ».



1 Un outil drag-and-drop de type puzzle



3 L'organisation de l'interface de développement

puis de scanner le code QR fourni par App Inventor (menu *Connect*). Votre application est alors exécutée sur le terminal pour que vous la testiez. Vos modifications de programmation sont prises en compte en temps réel sur le terminal, fonctionnalité très utile pour le développement.

Une fois mise au point, l'application peut être sauvegardée au format APK (Android Package) ; son installation sera définitive sur votre terminal, et sa transportabilité assurée vers tout support Android (menu *Build*). Vous pouvez aussi créer le code QR de votre application pour en faciliter le téléchargement.

La programmation

App Inventor, ce sont des composants et des blocs rangés dans des tiroirs. Vous programmez en glissant dans le programme un bloc qui contribue à la fonctionnalité que vous voulez construire. Vous n'avez pas à vous soucier des instructions ou à vous référer à un manuel de programmation (enfin, presque pas). Pas de messages d'erreur incompréhensibles, et App Inventor vous évitera nombre de bévues dès la mise en place. Par exemple, si un bloc fonctionnel attend un nombre, vous ne pourrez pas y connecter un texte. À l'inverse, un signal sonore (clic) vous confirme que les éléments sont bien associés. Cela n'élimine pas toutes les erreurs, mais cela aide.

Bien sûr, cette approche graphique ne remplace pas la réflexion et l'analyse préalables nécessaires à tout projet de développement. C'est d'ailleurs l'objectif de nos enseignements ; le langage n'est pas une fin en soi. Par contre, savoir choisir et agencer les bonnes structures algorithmiques est fondamental. Et la logique de la programmation événementielle reste également un préalable à la mise en œuvre d'App Inventor.



Votre smartphone dit « bonjour »



Programme déclenchant une alarme lors d'un choc

2 Des exemples de petites applications

On peut résumer la démarche de programmation ainsi : je place un élément sur mon interface (un bouton par exemple), puis je définis son comportement quand un événement survient. Inutile de tester l'état d'un élément en permanence, donc de définir une boucle de scrutation des variables : il réagit quand se produit l'événement prévu par le programmeur. Par contre, sans événement, vous ne pouvez pas initier un traitement...

App Inventor a aussi le mérite de ne pas nécessiter l'apprentissage d'un langage de programmation objet (ici Java) dans toute sa complexité.

Attention au contresens : la programmation orientée objet (POO), ce n'est pas placer des objets graphiques sur une interface. Pour résumer, un objet est une entité qui regroupe les variables manipulées (champs) et les moyens de traitement de ces variables (méthodes). Procédons par analogie, et prenons l'exemple classique d'un véhicule. Celui-ci dispose de champs (châssis, moteur, suspensions...) et de méthodes destinées à sa conduite (accélérer, freiner, tourner...). Selon les champs définis, vous allez définir une berline, un coupé sport, un monospace qui seront des descendants de l'objet véhicule. Cette notion d'héritage est un concept fort de la POO. Comme un enfant hérite des caractéristiques de ses parents et développe les siennes, un objet peut hériter des caractéristiques de son ancêtre,

En ligne

Le site du MIT consacré à App Inventor :

<http://appinventor.mit.edu>

Un guide (en anglais) pour l'utilisation d'App Inventor 2 en classe :

<http://appinventor.mit.edu/explore/ai2/setup-classroom.html>

Un tutoriel (en anglais) :

http://appinventor.mit.edu/explore/tutorialsa06d.html?&field_version_tid

Un tutoriel complet pour faire ses premiers pas :

<http://eduscol.education.fr/sti/sites/eduscol.education.fr/files/ressources/techniques/3396/3396-tutoriel-1er-application-app-inventor.zip>

Pour accéder directement au sommaire du numéro et à tous les liens de cet article :



Retrouvez tous les liens sur <http://eduscol.education.fr/sti/revue-technologie>

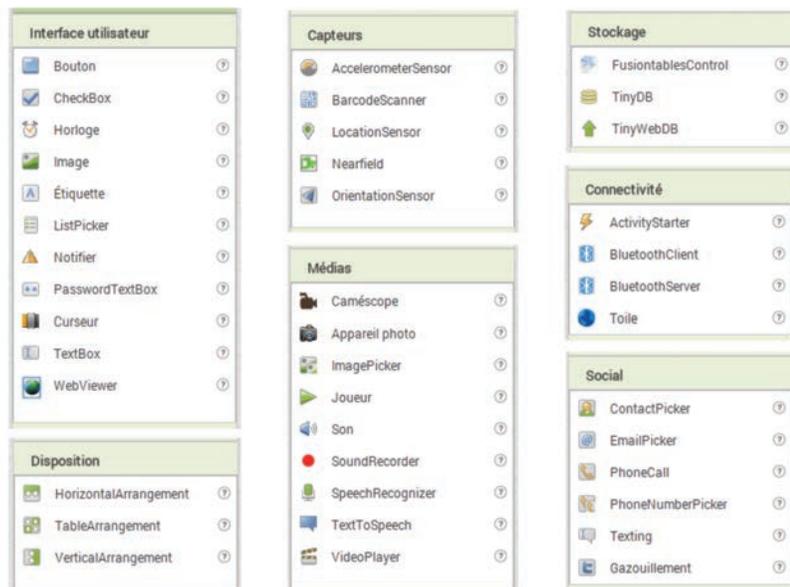
mais aussi en développer de nouvelles, ou bien encore se spécialiser. App Inventor ne nous emmènera pas jusque-là...

Une bibliothèque de fonctions à fort potentiel

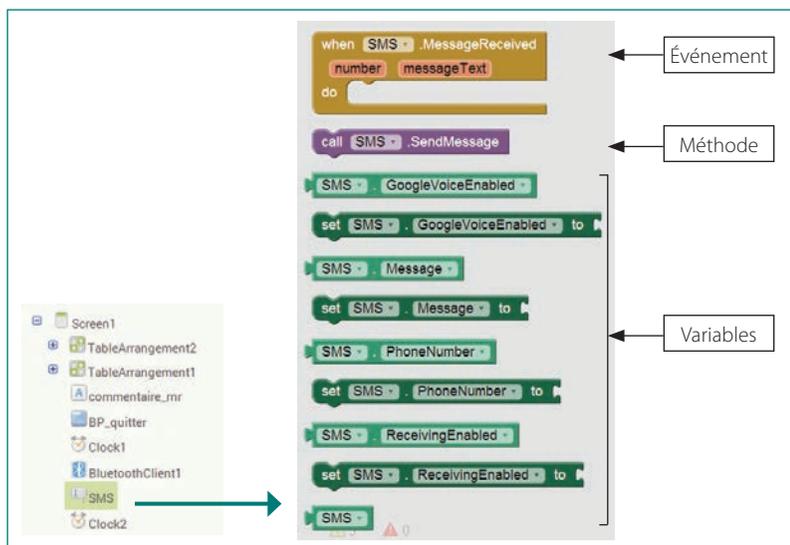
Vous disposez d'une très riche palette d'éléments préprogrammés qui découlent des fonctions inhérentes aux tablettes et smartphones 4. Chaque élément une fois déposé sur votre écran vous offre un choix de blocs fonctionnels 5.

Lorsque Daniel Finnegan, élève de David Wolber, un des membres du projet, a produit en 2010 le simple mais puissant No Texting While Driving (« pas de texto en conduisant »), les médias se sont emparés du phénomène AI, dont la notoriété fut dès lors assurée. Cette petite application répond automatiquement aux SMS reçus lorsque vous conduisez en envoyant le message *No texting while driving* 6. Elle est téléchargeable sur Google Play :

<https://play.google.com/store/apps/details?id=com.driveon&hl=fr>



4 Les fonctionnalités proposées



5 Un exemple de blocs fonctionnels proposés

Pourtant, son auteur, étudiant d'une filière littéraire de l'université de San Francisco, n'avait aucune prédisposition pour l'informatique. Il s'était inscrit par curiosité au module « Programmer vos applications Android ». App Inventor lui a permis de *prototyper* une idée.

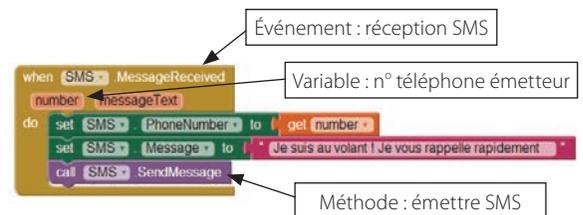
Le concept de prototypage rapide d'applications informatiques était dès lors admis. Comme le souligne Clive Thompson sur le site du célèbre magazine informatique *Wired*, « l'histoire de Finnegan illustre un point fort : il est temps pour la programmation informatique de se démocratiser » (voir les « Références » en encadré).

L'intérêt pédagogique

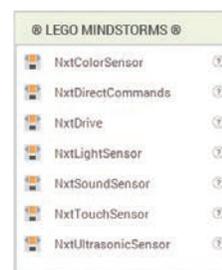
App Inventor est un outil conçu pour l'enseignement. Son objectif est favoriser la créativité tout en permettant l'acquisition des structures algorithmiques de base, mais en s'affranchissant du code. C'est en cela qu'il est un outil à prendre en considération pour l'enseignement en prébac. À l'instar des imprimantes 3D, il permet de prototyper une application rapidement. Sa méthode de programmation par blocs, de type Scratch, permet de s'appuyer sur les acquis des élèves de seconde et de favoriser les projets en lien avec le collègue de mathématiques en SI.

Bien sûr, ce n'est pas un outil de développement. Les fonctions sont en nombre limité, et le code généré n'est certainement pas optimisé. Évidemment, pour les formations postbac, les étudiants devront s'approprier les langages de programmation spécifiques à Android et les outils de développement professionnels. Pour cela, vous trouverez sur le portail STI d'Éducol une excellente ressource d'Alain Menu, enseignant en BTS IRIS au lycée La Fayette de Champagne-sur-Seine (académie de Créteil) :

http://eduscol.education.fr/sti/ressources_pedagogiques/initiation-au-developpement-sur-cibles-android



6 L'application No Texting While Driving francisée



7 Les fonctions relatives aux briques Mindstorms NXT

Mesure de distance par ultrasons avec la brique Lego Mindstorms NXT

La compatibilité d'App Inventor avec les briques Lego Mindstorms NXT ouvre, pour l'enseignement CIT en classe de 2^{de}, un potentiel de créativité en robotique et domotique. Vous disposez d'une palette de fonctions permettant d'exploiter tous les principaux périphériques de la brique NXT **7**.

Il s'agit ici de faire évoluer une canne pour aveugle en y ajoutant une fonction de détection des obstacles en hauteur : si un obstacle en hauteur est détecté à moins de 80 cm, le smartphone vibrera. Ce type de signal tactile est traité par le cerveau en un temps bien plus court qu'une information sonore qu'il doit discriminer dans un ensemble de sons.

Les éléments constituant la chaîne d'information sont donnés en **8**, l'algorithme et sa transcription en **9**. La connexion Bluetooth n'est pas détaillée ici, vous en trouverez le graphe du programme sur le lien suivant :

http://eduscol.education.fr/sti/ressources_techniques/app-inventor-realiser-une-application-pour-lego-nxt

Vous trouverez aussi sur ce lien un tutoriel pour réaliser un robot NXT piloté par smartphone ou tablette, et bien d'autres applications sont envisageables.



8 La chaîne d'information pour la canne d'aveugle

L'algorithme	Sa transcription
Début Se connecter à la brique NXT Si connexion établie Lire capteur ultrason Si distance < à 80 cm Faire vibrer le smartphone Fin	

9 L'application pour la canne d'aveugle (connexion Bluetooth non détaillée)



10 La chaîne d'information pour la commande d'une lampe à distance

Commande à distance d'une lampe avec une carte Arduino

La capacité d'App Inventor à communiquer en mode Bluetooth l'ouvre à tout système communicant, notamment les cartes Arduino mises en œuvre en STI2D. Cela offre un potentiel de développement d'objets communicants, et l'opportunité d'une autre approche des IHM (interfaces homme-machine).

Des réalisations en ce sens ont déjà été remarquées lors des Olympiades de sciences de l'ingénieur 2013, comme le robot inspecteur de câbles du lycée François-Mauriac à Andrézieux-Bouthéon (académie de Lyon) :

http://eduscol.education.fr/sti/ressources_pedagogiques/robot-inspecteur-de-cables

À noter aussi le travail mené par Jean-Marc Séguret sur le pilotage en Wi-Fi de l'aspirateur Roomba à partir d'un smartphone sous Android :

<http://sti.tice.ac-orleans-tours.fr/spip2/spip.php?article950>

Ici, il s'agit de commander une lampe (ou tout autre actionneur) par SMS. Par un téléphone tiers, vous envoyez par SMS « On » ou « Off » à votre smartphone. Celui-ci transmet les consignes via la liaison Bluetooth à la carte Arduino.

Les éléments constituant la chaîne d'information sont donnés en **10**, le programme Arduino – je vous laisse le décoder tant il est limpide ! – et celui de l'application App Inventor en **11** (la connexion Bluetooth n'est pas détaillée ici).

Vous trouverez sur Éduscol STI, dans la ressource « App Inventor : exemples d'IHM Android pour carte Arduino », les programmes sources ainsi que les fiches explicatives de plusieurs versions de cette application, dont une avec la commande vocale de la lampe :

http://eduscol.education.fr/sti/ressources_techniques/app-inventor-exemples-dihm-android-pour-carte-arduino

Références

ROQUE (R.), *OpenBlocks: An Extendable Framework for Graphical Block Programming Systems*, MIT, 2007, en ligne :

<http://dspace.mit.edu/bitstream/handle/1721.1/41550/220927290.pdf>

TARAUD (F.), « Un outil de programmation puissant adapté aux débutants », en ligne sur le site STI de l'académie d'Orléans-Tours :

<http://sti.tice.ac-orleans-tours.fr/spip2/spip.php?article905>

THOMPSON (C.) « Clive Thompson on Coding for the Masses », 2010, en ligne sur le site de *Wired* (en anglais) :

www.wired.com/magazine/2010/11/st_thompson_wereallcoders

WOLBER (D.), ABELSON (H.), SPERTUS (E.), LOONEY (L.), *App Inventor : Create Your Own Android Apps*, O'Reilly Media, 2011, en ligne (en anglais) :

http://cs.usfca.edu/~wolber/appinventor/appinv_0331.pdf

Leur complexité est volontairement progressive pour une première prise en main :

- Paramétrer les connexions Bluetooth (App Inventor et Arduino)
- Commande de la lampe par 2 boutons sur le smartphone
- Commande vocale de la lampe
- Moduler la luminosité de la lampe à partir d'un curseur sur le smartphone
- Répondre et émettre un SMS
- Commande de la lampe par envoi de SMS au smartphone
- Commande par SMS et envoi de SMS en retour avec consommation de la lampe
- Graphismes pour suivre l'évolution d'une grandeur physique

Ces exemples ont pour objectif de vous faire découvrir le potentiel d'App Inventor. De mon point de vue, la seule limite est notre imagination (enseignants et élèves)...

La pérennité d'App Inventor

Le développement d'AI était en sommeil depuis sa version initiale de 2011, et il était légitime de s'interroger sur son avenir. Mais, comme je l'annonçais en introduction, le MIT en a publié une nouvelle version en ce début d'année. Ses fonctionnalités sont très similaires, mais le design des blocs la rapproche d'avantage de Scratch. Surtout, le lancement de l'éditeur de programme est instantané, plus d'attente de l'exécution de Java. Et vous travaillez dans une seule et même fenêtre de votre navigateur.

Mais App Inventor version bêta (maintenant appelé App Inventor classique) est toujours là. Bien que très proches, AI classique et AI 2 n'ont pas le même noyau, et les projets réalisés avec la première version ne peuvent être chargés dans la seconde... Le MIT prévoit de fournir un outil pour aider à leur transfert de l'une à l'autre, mais il ne sera pas disponible avant quelques mois. J'ai personnellement pu retranscrire sans difficultés toutes mes applications de la version classique « à la main », et tout a fonctionné du premier coup...

Donc, n'ayez aucune inquiétude, vous pouvez dès à présent vous emparer de cet outil, son suivi semble assuré.

En conclusion

Comme le dit Hal Abelson lui-même, « l'essentiel est que vous appreniez par la création ». C'est aussi l'esprit des réformes mises en place ces dernières années en prébac. App Inventor est une opportunité pour initier des projets en phase avec la génération des *digital natives*, et donner envie aux élèves de poursuivre leurs études. Dès le départ, les initiateurs du projet l'ont constaté, App Inventor transforme les élèves de consommateurs en créateurs : ils adorent créer des applications pour leur téléphone ! ■

Remerciements

Mes remerciements à Frédéric TARAUD, Jean-Marc SÉGURET, Laurent FERRAPIE, Thomas DUFOUR, Jacques RIOT et Rodolphe PIVERT.

Programme Arduino	Application App Inventor
<pre>//stocker dans w la valeur reçue par Bluetooth if (Serial.available()) { w=Serial.read(); // commande du relais if (w>0) { digitalWrite(RELAIS,HIGH); } else { digitalWrite(RELAIS,LOW); } }</pre>	

11 La programmation pour la commande d'une lampe à distance (connexion Bluetooth non détaillée)

Scratch, la programmation mise en pièces

Scratch, dont l'idée a germé en janvier 2003, est un logiciel conçu pour développer la pensée créative, le raisonnement et le travail en équipe. Par une approche ludique de l'algorithmique, il permet d'initier les élèves dès l'âge de 8 ans à des concepts fondamentaux en mathématiques et en informatique. Le slogan de Scratch est « Imagine, programme, partage ! ». Le partage est en effet un fondement de la pédagogie de Scratch.

Depuis 2007, le site *Scratched* fédère une communauté en ligne où l'on peut échanger des ressources, partager des expériences, discuter et rencontrer des formateurs Scratch :

<http://scratched.media.mit.edu>

Son succès est international. À tel point que Bill Gates et Mark Zuckerberg ont soutenu le projet "Hour of code", événement international d'initiation à la programmation s'appuyant sur le langage Scratch (et comprenant le jeu Lightbot que nous vous présentons dans le « Techno mag » du dernier numéro) :

<http://code.org>

Scratch est un projet du groupe Lifelong Kindergarten au MIT Media Lab. Il est d'accès libre, sous licence GPL :

<http://scratch.mit.edu>

Depuis, il a essaimé sur de nombreuses applications de programmation :

Ardublock pour les cartes Arduino :

www.semageek.com/arduino-presentation-et-traduction-en-francais-de-ardublock

Enchanting pour les briques Lego Mindstorms NXT :

<https://launchpad.net/enchanting/+download>

RoboCom Basic (héritier de la tortue Logo pour smartphone) :

<https://play.google.com/store/apps/details?id=com.CubesTeam.RoboCom&hl=fr>

et bien sûr App Inventor.

L'enseignement de l'algorithmique est au programme de l'enseignement des mathématiques en classe de 2^{de}. Éduscol maths propose un document « Ressources pour la classe de seconde » consacré à l'algorithmique qui fait référence à Scratch comme support de programmation :

http://cache.media.eduscol.education.fr/file/Programmes/17/8/Doc_ress_algo_v25_109178.pdf

Vous y trouverez de nombreux exemples d'exercices. N'hésitez pas en à parler avec vos collègues mathématiciens !