

Introduction à MIT App Inventor 2

[MIT App Inventor](#) est un environnement de programmation graphique qui vous permet de créer très facilement des applications pour les appareils portables Android.

Les programmes sont construits en imbriquant des blocs (comme dans [Scratch](#)), ce qui permet aux programmeurs débutants de créer des applications fonctionnelles en peu de temps, sans qu'un long apprentissage préalable ne soit nécessaire.



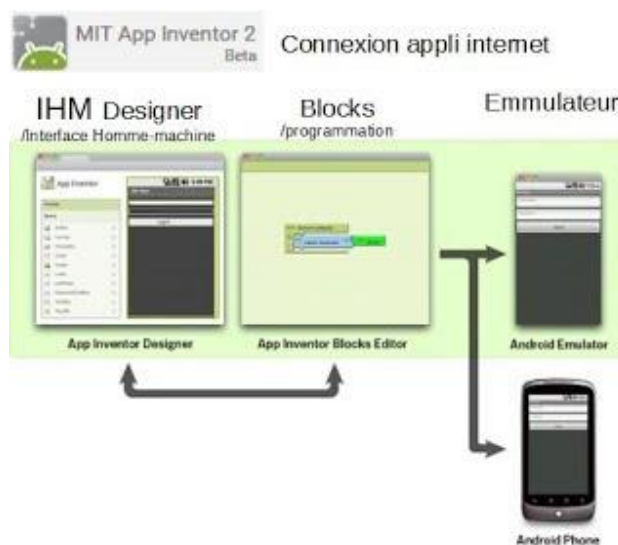
À quoi ressemble un programme App Inventor ?

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 évidemment. Les informations sont stockées sur des serveurs distants.

Le concept d'App Inventor :

Trois fenêtres sont proposées pendant le développement :

- Une pour la création de l'interface homme machine : ce sera l'allure de votre application ; Vous pouvez choisir vos composants : Bouton, texte, image, cadre dessin... et fonctionnalités comme capteur (gps, accéléromètre...) média ...et les paramétrer.
- Une pour la programmation par elle-même : elle permettra, par l'assemblage de blocs de créer le comportement de l'application ; Vous avez des briques à disposition pour traiter du texte, des variables, des comportements mathématiques, contrôler les éléments ...
- Et une pour l'émulateur qui permettra de tester l'application. L'émulateur permet de remplacer un terminal réel pour vérifier le bon fonctionnement du programme (autres solutions: voir annexe).



Accéder au site App Inventor 2

Pour se connecter, il faut un compte Gmail personnel (et être connecté à ce compte).

Pour créer une application, allez sur le site : <http://ai2.appinventor.mit.edu/>

Une fois dans [App Inventor 2](#), demander l'affichage en français (à moins que ce soit déjà le cas): pour ce faire, vous utilisez un menu situé en haut à droite.



Cahier des charges de l'application

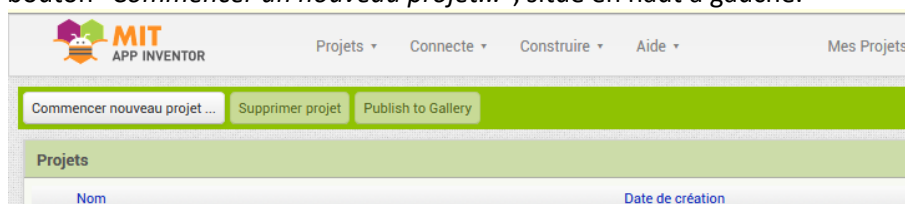
Vous allez créer une application qui calcule le courant qui traverse une résistance, lorsqu'on lui fournit la tension et la valeur de la résistance (application de la loi d'Ohm).

$$U = R \times I \text{ donc } I = U / R$$

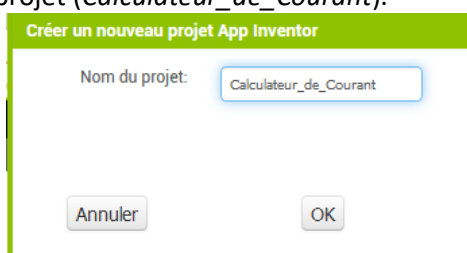
Textuellement, la tension "U" est égale à la valeur de la résistance "R" multipliée par l'intensité "I" du courant électrique.

Création du projet

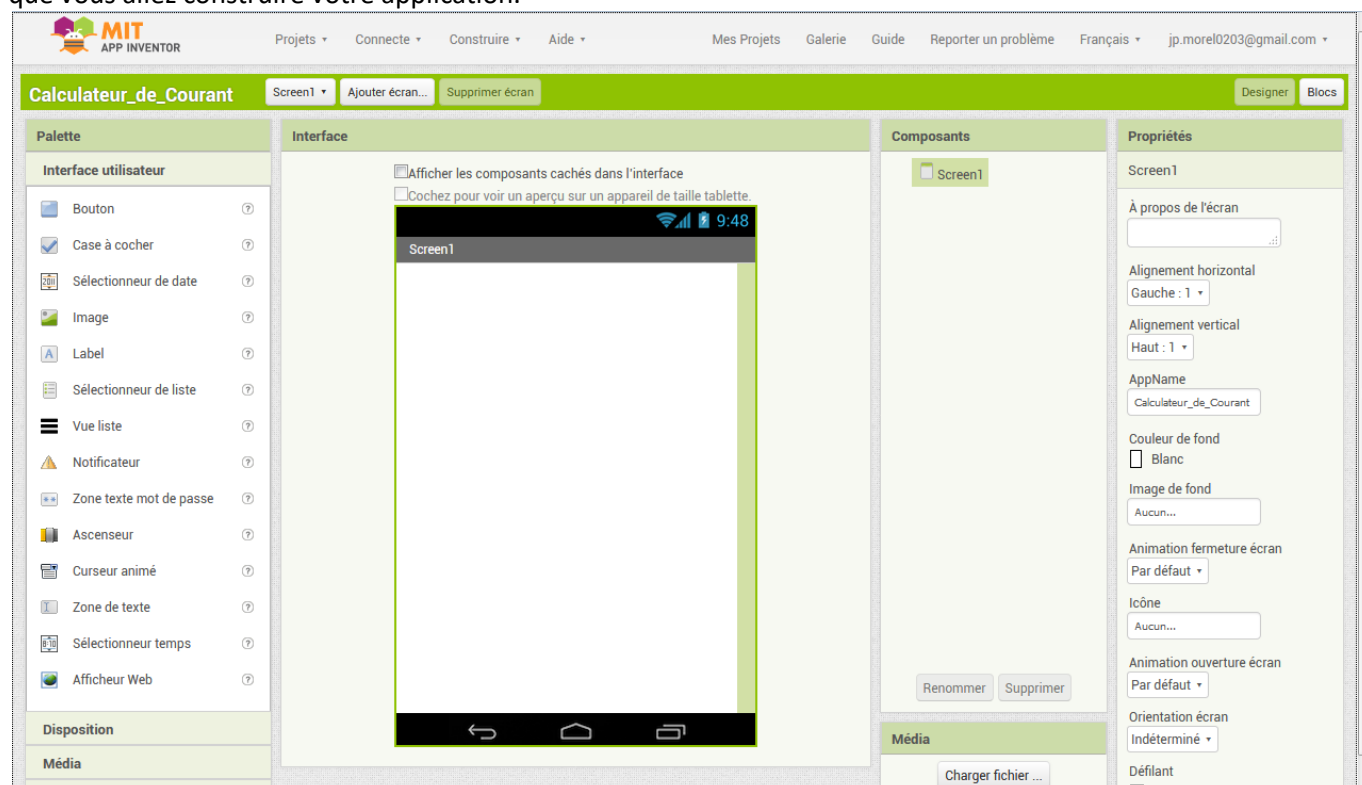
Vous cliquez sur le bouton "Commencer un nouveau projet...", situé en haut à gauche.



...et vous choisissez un nom pour ce projet (*Calculateur_de_Courant*).

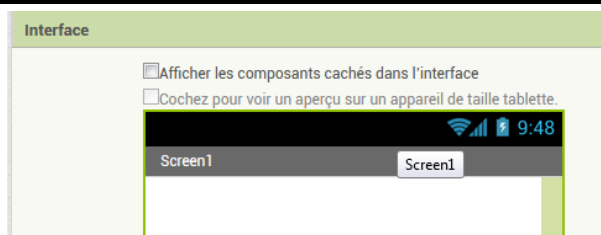


La fenêtre devrait maintenant être divisée en 5 zones: *Palette*, *Interface*, *Composants*, *Média*, *Propriétés*. C'est ici que vous allez construire votre application.

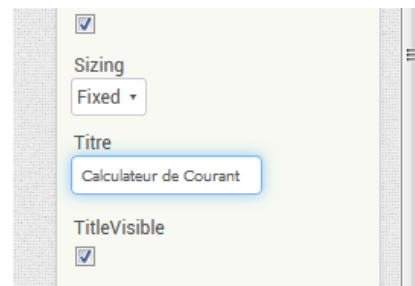


Création de l'interface de notre application

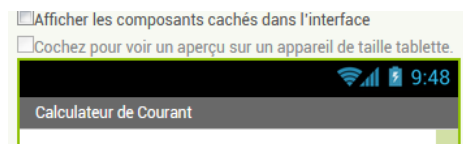
Retour à l'ordinateur, dans MIT App Inventor. La fenêtre principale de notre application est affichée dans la zone "Interface", située en plein milieu de notre écran. Par défaut, cette fenêtre a été baptisée "Screen1", ce qui n'est pas très joli.



Pour changer le titre de la fenêtre, vous allez dans la zone "Propriétés", située complètement à droite. Vous remplacez le titre initial par "Calculateur de Courant".

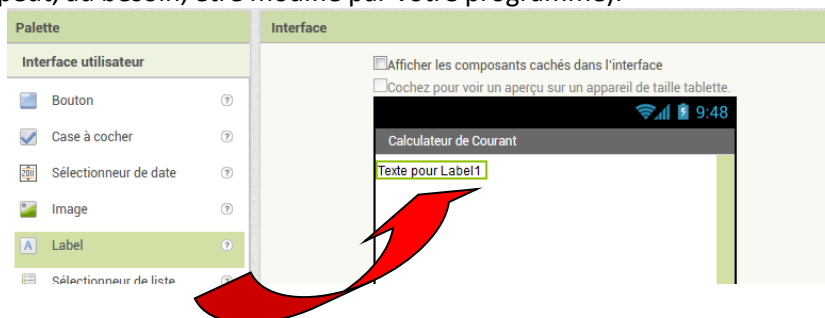


Le titre de la fenêtre devrait instantanément être mis à jour sur la simulation qui se trouve dans la zone "Interface", mais aussi dans le Compagnon qui se trouve sur votre appareil Android.



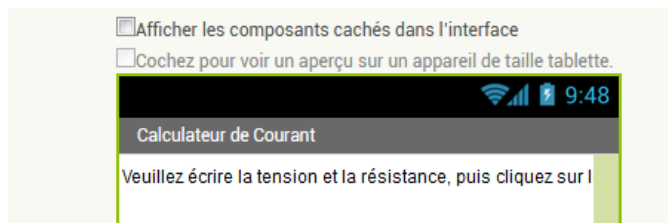
La palette, située à gauche, comporte une liste d'éléments pouvant être disposés dans la fenêtre de votre application: boutons, zones de texte, etc. Pour ajouter un de ces éléments dans la fenêtre, il suffit de le faire glisser de la zone "Palette" vers la zone "Interface" située en plein centre de la fenêtre.

Insérez un élément de type "Label"; il s'agit d'une zone de texte dont le contenu ne peut pas être modifié par l'utilisateur (mais qui peut, au besoin, être modifié par votre programme).

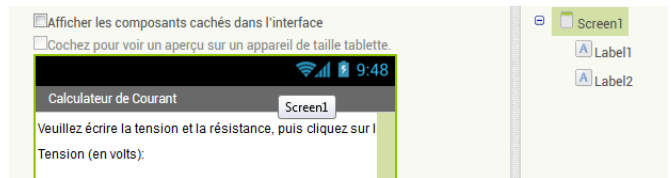


Par défaut, ce label contient le texte "Texte pour Label1". **Modifiez** l'attribut "Texte" dans la zone de "Propriétés" par "Veuillez écrire la tension et la résistance, puis cliquez sur le bouton "Calculer"".

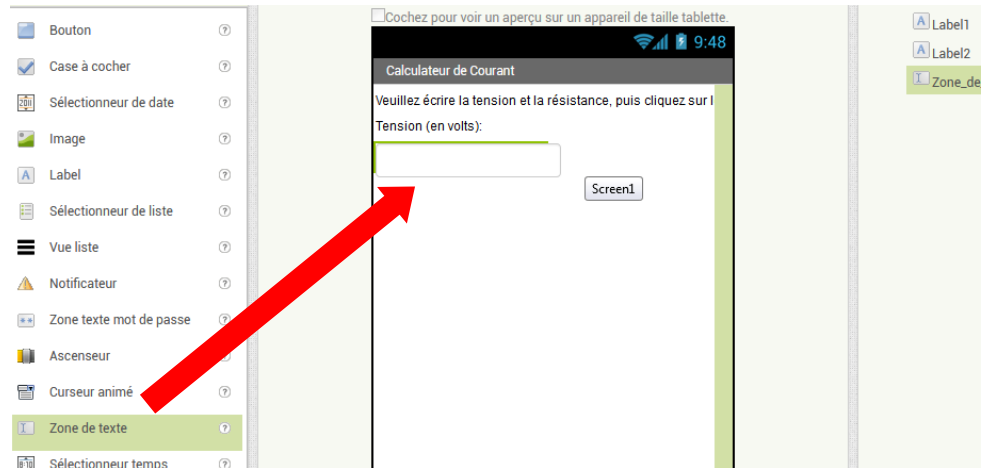
Le texte n'est pas visible au complet dans la zone "Interface" (qui n'est qu'une simulation grossière), mais il est correctement affiché quand on regarde sur l'appareil Android, dans MIT AI2 Companion.



Ajoutez maintenant un deuxième Label, et **remplacez** son contenu par "**Tension (en volts):**"

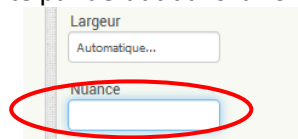


Ensuite, **ajoutez** une zone de texte dans laquelle l'utilisateur écrira la valeur numérique de la tension. Contrairement aux labels, le contenu des zones de texte peut être modifié par l'utilisateur.

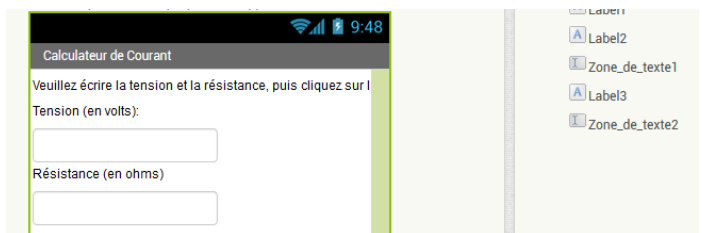


Si vous regardez le résultat sur votre appareil Android, la zone de texte comporte un texte par défaut ("*Nucance pour Zone_de_texte*").

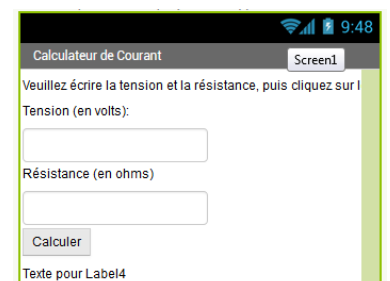
On peut effacer ce texte par défaut dans la zone "Propriétés":



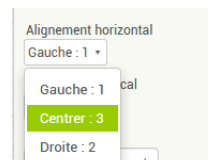
Ajoutez ensuite un 3e label dont le contenu sera "*Résistance (en ohms):*", et une deuxième zone de texte qui permettra à l'utilisateur d'écrire la valeur de la résistance.



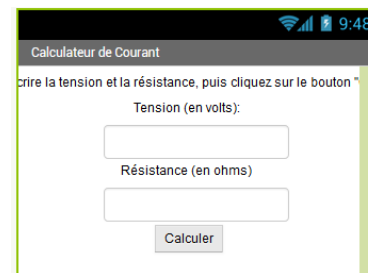
Ajoutez finalement un bouton que l'utilisateur cliquera pour demander le résultat du calcul (**changez** le texte du bouton par "*Calculer*"), et un 4e label, qui contiendra le résultat du calcul (vous pouvez effacer son texte par défaut et le laisser vide pour l'instant).



Par défaut, tous ces éléments sont alignés à gauche de la fenêtre. Si vous cliquez dans le fond de la fenêtre, la zone "Propriétés" affichera à nouveau les propriétés de la fenêtre "Screen1". **Réglez** l'alignement horizontal à "Center: 3".



Votre appareil android affiche maintenant une jolie fenêtre comportant tous les éléments nécessaires. Lorsque vous cliquez sur une zone de texte, c'est déjà possible d'y écrire un nombre. Mais si vous cliquez sur le bouton, évidemment, rien ne se passe.



Construction du programme

Depuis le début, vous étiez en mode "Designer". Vous allez maintenant passer en mode programmation en **cliquant** sur le bouton "Blocs", en haut à droite.



Observez le contenu de la partie gauche de la fenêtre (intitulée "Blocs"): il y a d'abord une liste de catégories (*Contrôle, Logique, Math, etc.*), puis la liste des éléments que vous avez placés dans votre fenêtre (*Label1, Label2, etc.*).

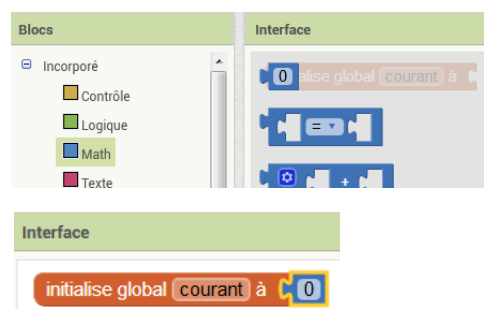
Remarque: en mode "Designer", vous auriez pu modifier chacun de ces noms attribués par défaut aux éléments que vous avez créés. Par exemple, vous auriez pu remplacer le nom "Zone_de_texte1" par "Texte_Tension" ou remplacer "Label4" par "Label_Reponse". Quand le programme est complexe, c'est plus facile de s'y retrouver si vous choisissez des noms plus appropriés.

Commencez par définir une variable, qui contiendra plus tard la valeur du courant. Pour cela, **cliquez** sur la catégorie "Variables".

Cliquez sur le bloc "initialise global nom à" et faites-le glisser dans la zone "Interface". Remplacez "nom" par "courant".

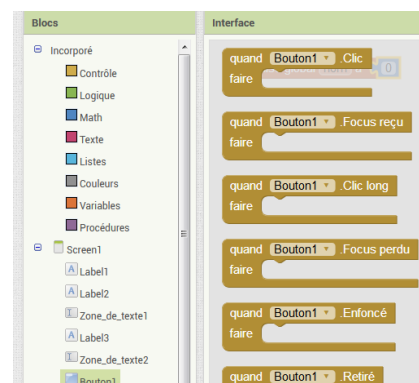


Pour régler sa valeur initiale à zéro, **cliquez** sur la catégorie "Math", et faites glisser le bloc "0" pour l'imbriquer à la droite du bloc précédent.



Dans la zone interface, votre programme a maintenant cette allure:

Vous allez maintenant établir ce qui se produira lors d'un clic sur le bouton. Pour ce faire, **cliquez** sur "Bouton1", et **choisissez** la première option: "quand Bouton1 Clic faire".



Que vous faites glisser dans la zone "Interface":



Que devez-vous faire lors d'un clic sur le bouton? Tout d'abord, une mise à jour de la variable courant, dont la valeur est calculée à partir du contenu des zones de texte.

Donc retour à la catégorie "Variables", dans laquelle on choisit le bloc "mettre à":



Ce bloc doit être **inséré** à l'intérieur du bloc "quand bouton1 clic". De plus, vous **remplacez** "nom" par "global courant", qui est le nom de notre variable.

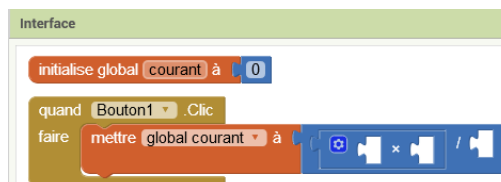


Le calcul du courant en milliampères (unités de mesure plus pratiques que les ampères) est égal à $1000 * \text{tension} / \text{résistance}$.

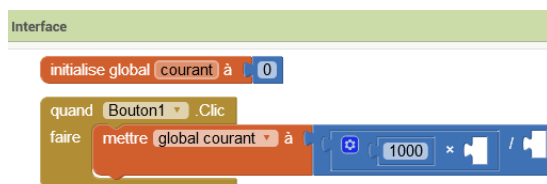
Les opérations mathématiques se trouvent, sans surprise, dans la catégorie mathématique. **Commencer** par la division:



...dans laquelle vous **imbriquez** une multiplication:



Pour la valeur numérique "1000" (qui permet de convertir les ampères en milliampères), on retourne dans la catégorie mathématique, on choisit encore une fois le bloc "0" situé complètement en haut, et on l'imbrique dans le premier espace libre de votre opération mathématique. On modifie sa valeur pour 1000:



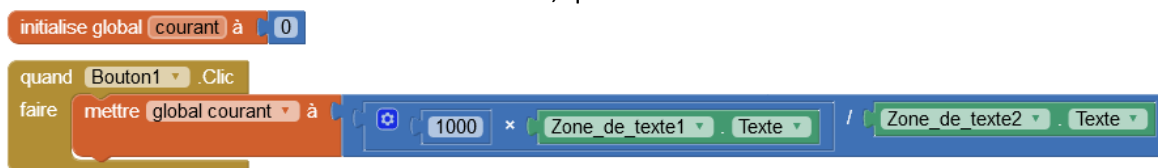
Le prochain élément de l'équation sera la tension, qui se trouve dans la "Zone_de_texte1". Dans la liste des blocs, vous cliquez donc sur "Zone_de_texte1", qui vous donne accès aux blocs qui concernent cet élément.



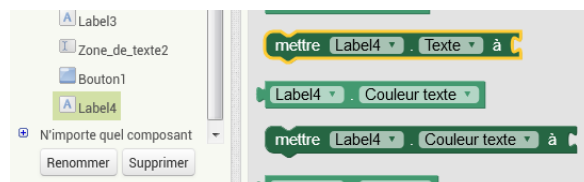
Dans la liste, vous choisissez le bloc "Zone_de_texte1.Texte": il s'agit du texte qui a été entré par l'utilisateur dans la zone de texte numéro 1. On fait glisser ce bloc à l'intérieur de votre équation mathématique.



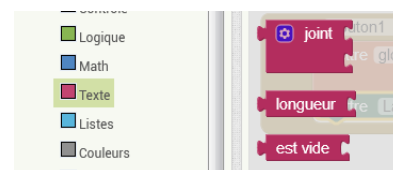
On fait la même chose avec la zone de texte numéro 2, qui contient la valeur de la résistance:



Mais on ne peut pas se contenter de calculer la valeur du courant, il faut aussi l'afficher dans votre label numéro 4. Dans la liste des blocs, vous cliquez sur "Label4" et vous choisissez le bloc "mettre Label4.Texte à".



Le label numéro 4 contiendra la valeur numérique du courant, mais aussi les unités de mesures (pour que l'utilisateur sache que le résultat est affiché en milliampères). Pour ce faire, vous utiliserez un bloc "joint" dans la catégorie texte:

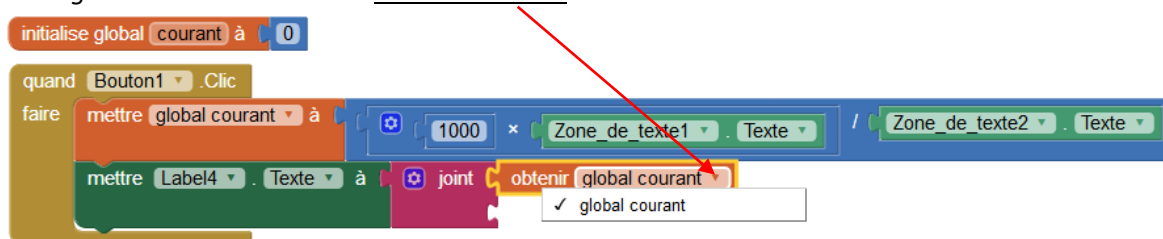


Dans la zone "Interface", ce bloc "joint" doit être imbriqué dans le bloc "mettre Label4.Texte à".

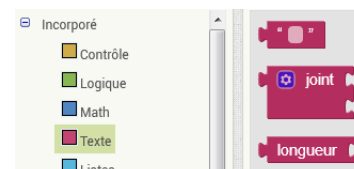


Retour à la catégorie variable pour prendre un bloc "obtenir"

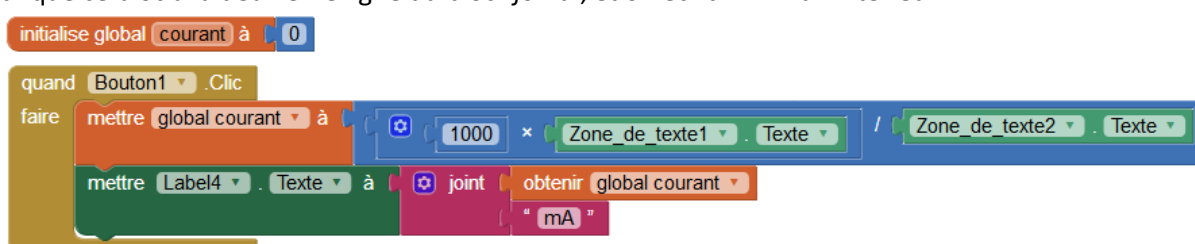
Ce bloc "obtenir" représente la valeur numérique stockée dans la variable "courant", à la condition de sélectionner "global courant" dans son menu déroulant:



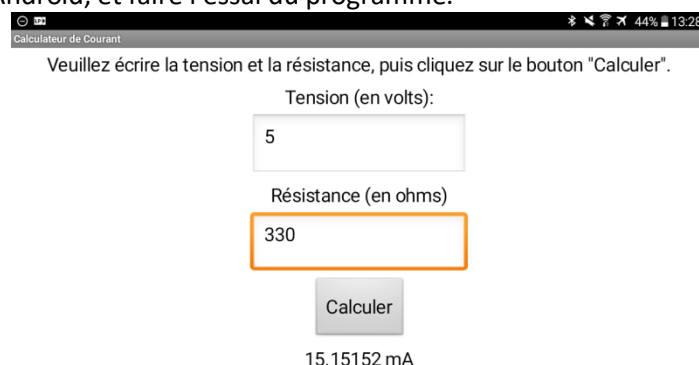
Pour les unités de mesure, on retourne dans la catégorie "Texte", mais cette fois on choisit le premier bloc en haut de la liste (" "):



On imbrique ce bloc à la deuxième ligne du bloc "joint", et on écrit "mA" à l'intérieur:



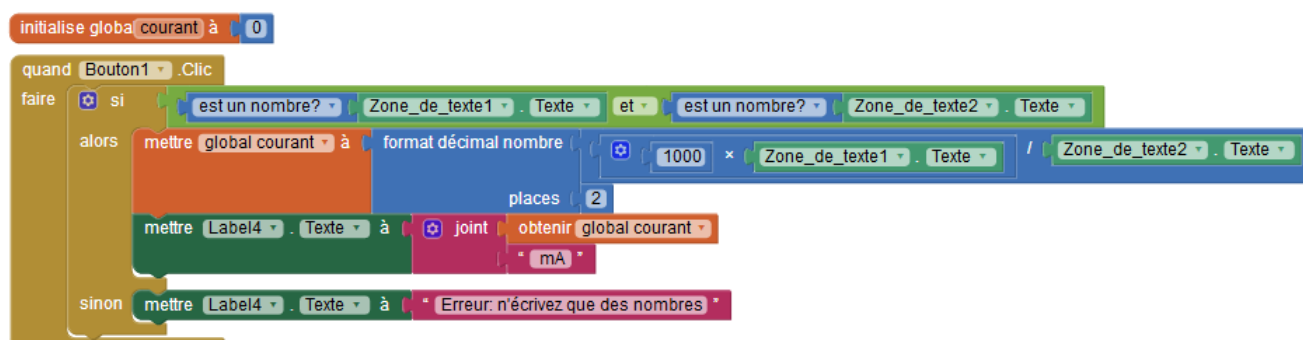
Voilà, Vous avez maintenant un programme fonctionnel! On peut retourner sur le programme MIT AI2 Companion, sur l'appareil Android, et faire l'essai du programme.



Quelques améliorations

Votre programme affiche le résultat avec un nombre de décimales inutilement grand. De plus, il ne vérifie pas si les zones de texte contiennent des valeurs numériques avant d'utiliser leur contenu dans un calcul.

Le programme ci-dessous est un tout petit peu plus évolué: avant d'effectuer les calculs, il vérifie si le contenu des deux zones de texte est bel et bien un nombre. Si ce n'est pas le cas, il affiche un message d'erreur. De plus, le calcul est inséré à l'intérieur d'un bloc "format décimal nombre" réglé à deux décimales, ce qui rendra le résultat affiché plus lisible.

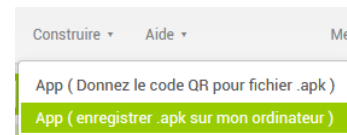


Modifiez la construction de votre programme comme ci-dessus. A vous de rechercher les blocs nécessaires parmi la liste de catégories (*Contrôle, Logique, Math, etc.*).

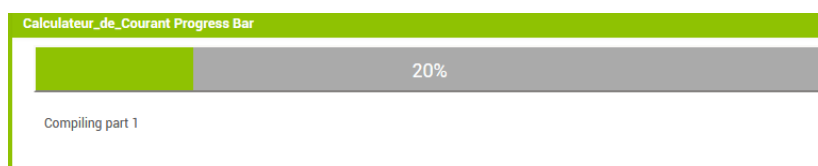
Finalisation de l'application: le fichier apk

Vous avez construit un code fonctionnel, mais il ne s'exécute qu'à l'intérieur de MIT AI2 Companion...n'y aurait-il pas moyen d'en faire une véritable application autonome?

Pour ce faire, vous choisissez "App (enregistrer .apk sur mon ordinateur)" dans le menu "Construire".



Cette opération peut prendre quelques secondes, malgré la simplicité de votre programme.



Votre appareil android doit être réglé pour accepter les applis "de source inconnue". Ce réglage se trouve dans "Paramètres", "Personnel", "Sécurité", "Sources inconnues".

Il ne reste plus qu'à transférer le fichier sur votre appareil android: lorsque vous ouvrez le fichier, android vous propose d'installer l'appli.