



## DÉBUTER AVEC APP INVENTOR

App Inventor est un outil de développement en ligne pour les téléphones et les tablettes sous Android. App Inventor est un OS créé par Google, et concurrent de l'ios d'Apple qui équipe l'iPad, iPodTouch et iPhone.

La plateforme de développement est offerte à tous les utilisateurs possédant un compte Gmail. Elle rappelle certains langages de programmation simplifiés des années 80 et s'inspire des travaux d'une étudiante en thèse au MIT, Ricarose Roque. Le projet a été dirigé par Hal Abelson, professeur au MIT qui a mis au point le portage du langage LOGO sur Apple 2 au début des années 80.



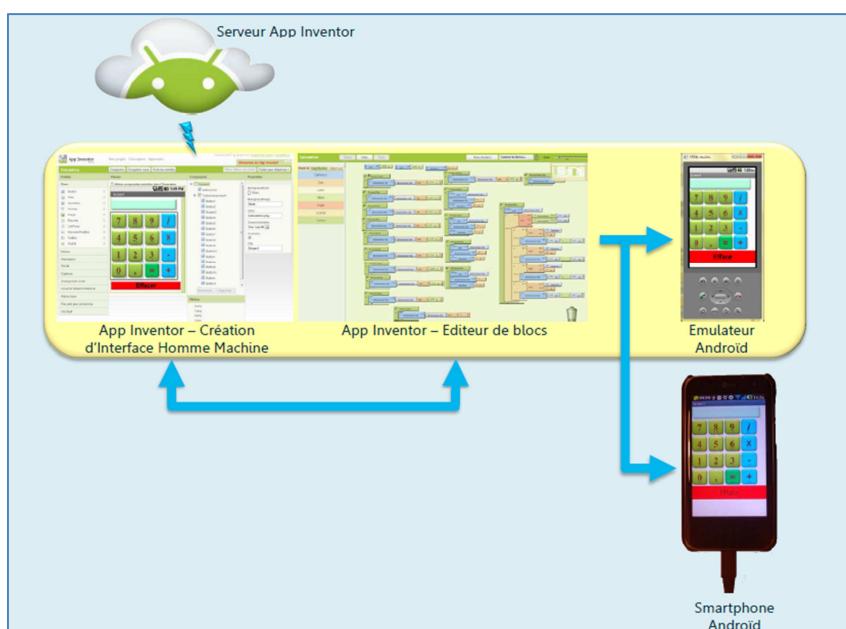
Android App Creator deviendra-t-il le logo des applications mobiles ?

La programmation est réalisée sans taper une seule ligne de code, mais simplement en associant et en paramétrant des briques logicielles toutes faites.

### 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 :

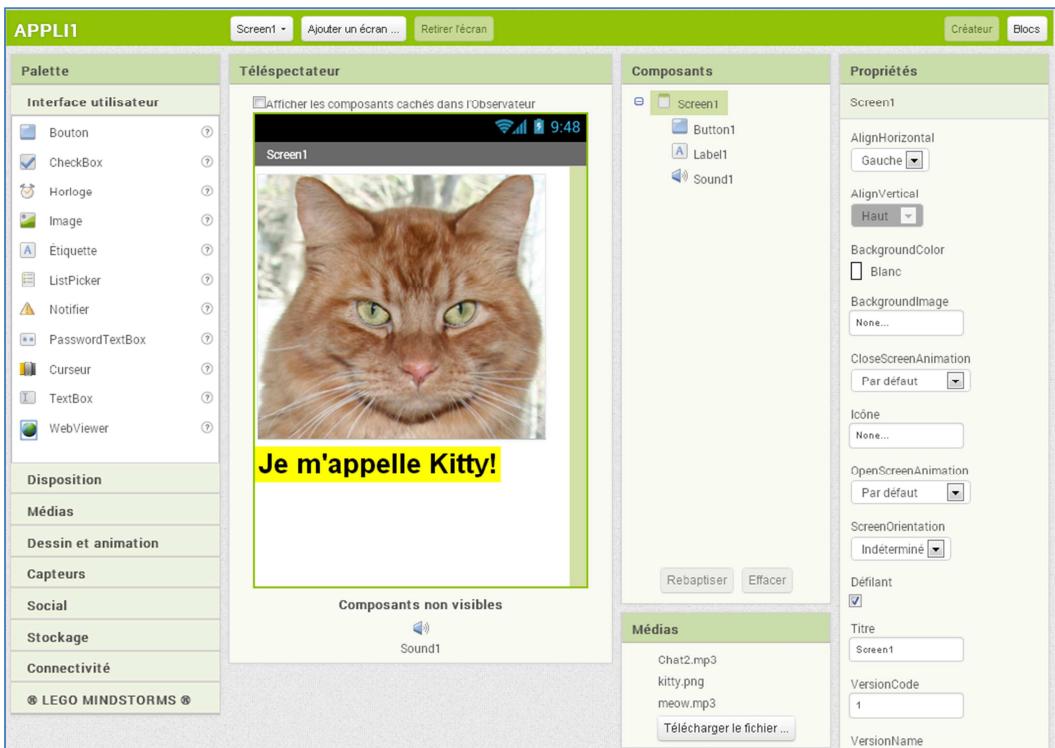


3 fenêtres sont proposées pendant le développement :

- Une pour la création de l'interface homme machine : ce sera l'allure de votre application ;
- Une pour la programmation par elle-même : elle permettra, par l'assemblage de blocs de créer le comportement de l'application ;
- 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.

La connexion d'un terminal réel sous Android permettra ensuite d'y télécharger le programme pour un test réel. Ce terminal pourra aussi bien être un téléphone qu'une tablette, le comportement du programme sera identique.

### La fenêtre de la création de l'interface :



### La fenêtre de programmation graphique :



### L'émulateur pour tester le programme :



L'approche graphique ne remplace pas la réflexion et l'analyse préalable nécessaire à tout projet de développement. La logique de programmation événementielle reste également un préalable avant la mise en œuvre d'App Inventor. Par contre, elle a le mérite de masquer toute la complexité imposée par l'apprentissage d'un langage de programmation objet (ici Java), à savoir une syntaxe et des concepts parfois assez subtils. Cet apprentissage restera obligatoire pour aller plus loin notamment dans le domaine d'une spécialisation post-bac.

### **Qu'est-ce que la programmation objet ?**

Attention : on ne fait pas de la programmation objet parce qu'on place des objets graphiques sur une interface. La programmation objet : c'est autre chose. Pour faire simple, c'est créer et manipuler des entités informatiques modélisatrices d'un comportement qui appartiennent à une classe et peuvent elles-mêmes avoir des héritiers, qui possèdent des méthodes... la programmation graphique ici nous met sur le chemin de la programmation objet mais masque ces aspects pour ne laisser la place qu'à l'approche événementielle du programme.

### **Qu'est ce que la programmation événementielle ?**

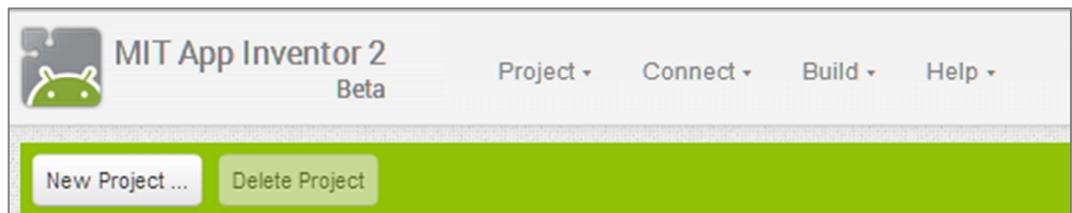
On peut la résumer ainsi : je place un élément sur mon interface (un bouton par exemple) puis je définis son comportement quand il est cliqué par l'utilisateur (événement). Il n'est donc pas nécessaire de tester son état en permanence, c'est lui qui réagit quand se produit l'événement prévu par le programmeur. Encore faut-il prévoir cet événement et surtout le traitement à opérer quand il survient.

## Découvrir APP INVENTOR

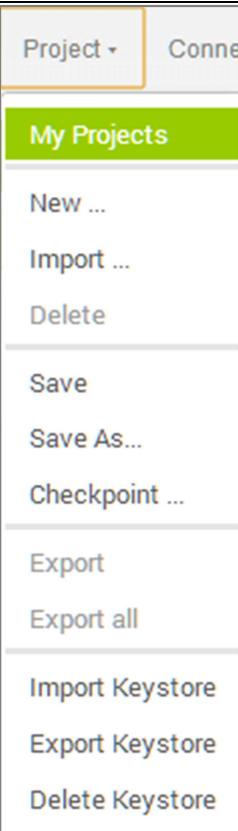
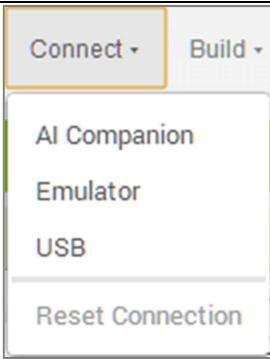
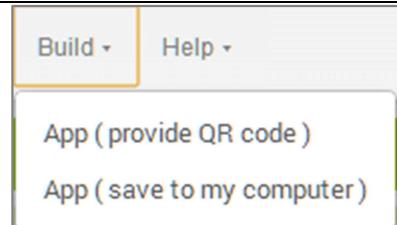
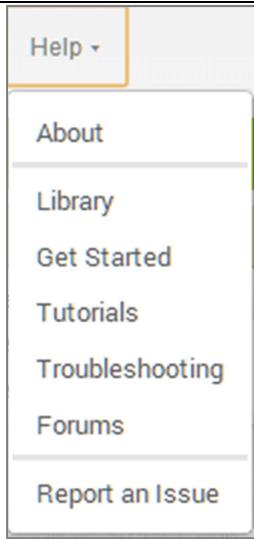
Pour se connecter, il faut un compte Gmail personnel (et être connecté à ce compte).

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

Nous obtenons l'écran suivant :



### Présentation des différents menus :

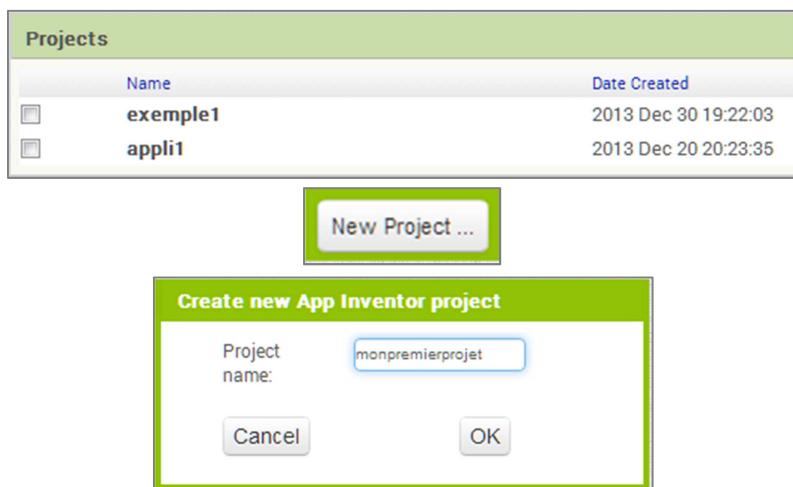
Le menu « Project »	Le menu « Connect »	Le menu « Build »	Le menu « Help »
			
Permet de visualiser les projets réalisés (My Projects), de réaliser un nouveau projet (New), d'importer un projet, de sauvegarder un projet...	Permet de visualiser le résultat du projet soit sur l'émulateur, soit sur le smartphone ou la tablette par l'intermédiaire du Wifi, soit par l'intermédiaire d'un câble USB	Permet de construire le QR code de l'application ou de sauvegarder l'application sur votre ordinateur.	Permet de retrouver un certains nombres de ressources, d'aides, d'exemples... en anglais !

## Réaliser un premier projet avec APP INVENTOR

Lorsque vous utilisez App Inventor, vous arrivez sur une page présentant les projets App Inventor de votre compte.

Pour créer une application cliquez sur le bouton « NewProject »

Entrez le nom du nouveau projet :



### 1<sup>ère</sup> étape du projet : « Dessiner l'interface »

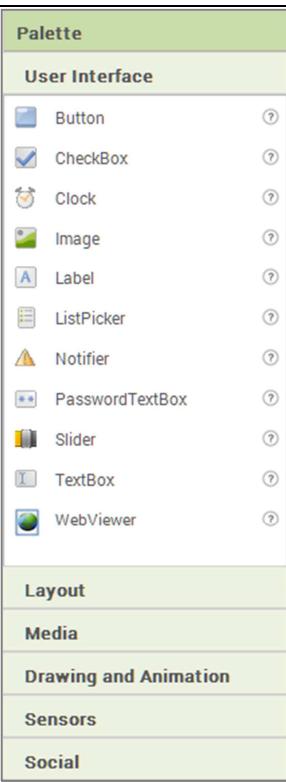
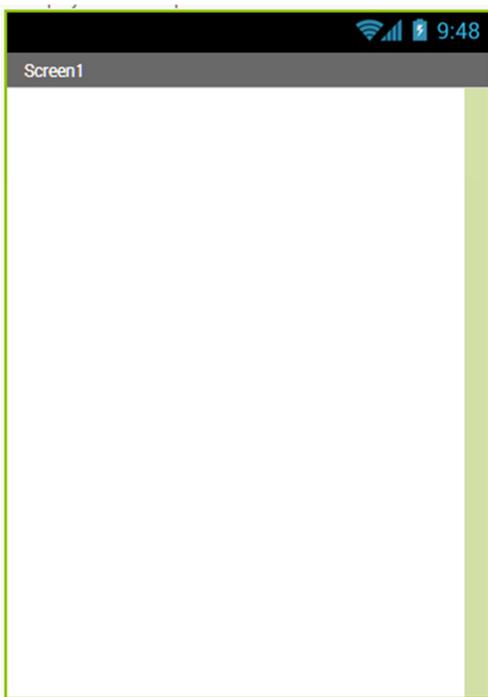
Nous allons créer une application simple qui permettra de détecter un clic sur un bouton pour recopier le contenu d'un texte saisi au clavier.

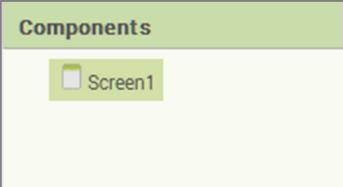
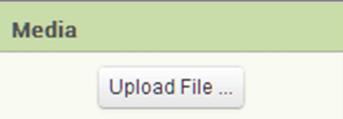
Pour créer une application, la première phase est la création de son interface.

Pour cela, le site affiche un écran de téléphone dans lequel nous pouvons placer, en les faisant glisser, les éléments que nous voulons utiliser.

Il y a des éléments graphiques comme des boutons, des labels... et des éléments non graphiques comme des capteurs (ex. géo localisation...) ou des fonctions permettant d'effectuer des actions : lecteur audio, appareil photo...

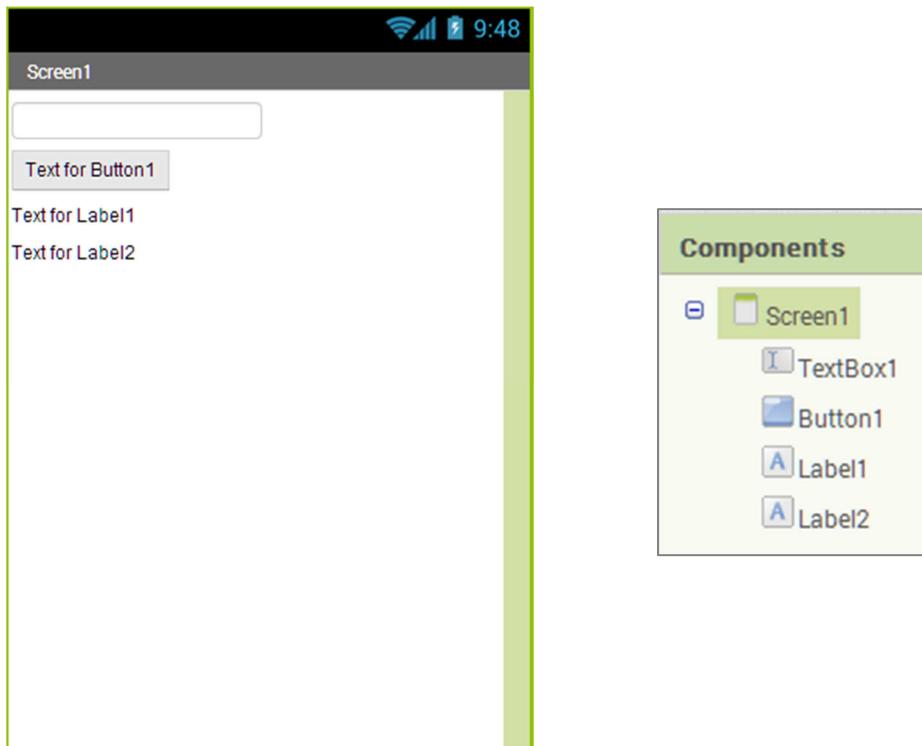
#### L'écran se partage en 4 parties :

La palette contenant tous les éléments qui peuvent être positionnés sur l'écran du téléphone :	L'écran du téléphone lui-même :
	

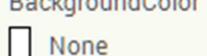
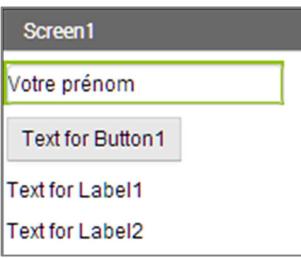
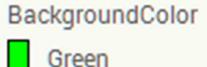
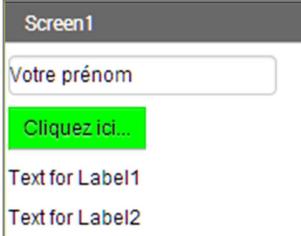
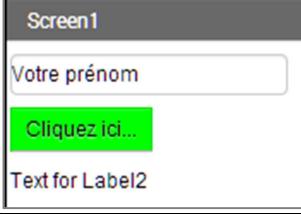
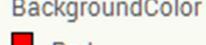
La liste des éléments et des médias utilisés sur l'écran :	Les propriétés des différents éléments utilisés :
 	<p><b>Properties</b></p> <p>Screen1</p> <p>AlignHorizontal: Left</p> <p>AlignVertical: Top</p> <p>BackgroundColor: White</p> <p>BackgroundImage: None...</p> <p>CloseScreenAnimation: Default</p> <p>Icon: None...</p> <p>OpenScreenAnimation: Default</p> <p>ScreenOrientation: Unspecified</p> <p>Scrollable: checked</p> <p>Title: Screen1</p>

Afin de réaliser l'interface, prendre les différents éléments et les faire glisser sur l'écran du téléphone :

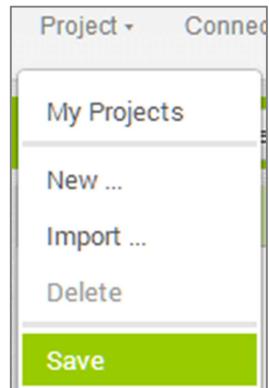
- Une « TextBox » qui permet de saisir du texte,
- Un bouton,
- Deux labels qui permettront d'afficher du texte.



Il faut ensuite définir les propriétés de chaque élément :

Cliquer sur l'élément TextBox1 :	Définir le fond de la zone de texte :	Saisir le texte par défaut :	Le résultat apparaît sur l'écran :
	 BackgroundColor None	 Text Votre prénom	 Screen1 Votre prénom Text for Button1 Text for Label1 Text for Label2
Cliquer sur le bouton :	Définir la couleur du bouton :	Saisir le texte par défaut :	Résultat :
	 BackgroundColor Green	 Text Cliquez ici...	 Screen1 Votre prénom Cliquez ici... Text for Label1 Text for Label2
Cliquer sur le label 1 :	Choisir la couleur du fond du label :	Rendre cette zone invisible :	Résultat :
	 BackgroundColor Red	 Visible hidden	 Screen1 Votre prénom Label1 Text for Label2
Cliquer sur le label 2 :	Choisir la couleur du fond du label :	Rendre cette zone invisible :	Résultat :
	 BackgroundColor Red	 Visible hidden	 Screen1 Votre prénom Label2 Text for Label2

L'interface est prête,  
pensez à enregistrer  
votre travail !



## 2<sup>ème</sup> étape du projet : « Décrire le comportement de l'application »

Une fois l'allure de notre application créée, il est nécessaire de décrire son comportement :

Pour cela, il faut cliquer sur « Blocks » en haut et à droite de la page :

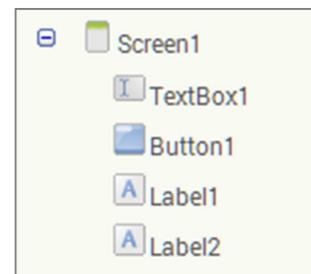
Designer      Blocks

Nous obtenons l'écran suivant :



Sur la gauche, nous avons un système d'onglets. Nous y retrouvons les différents composants que nous avons placé sur l'écran ainsi que des blocks utilitaires « Built in »

Dans l'onglet « Screen1 », nous retrouvons les éléments de l'interface :



Nous allons pouvoir décrire le comportement de chaque élément.

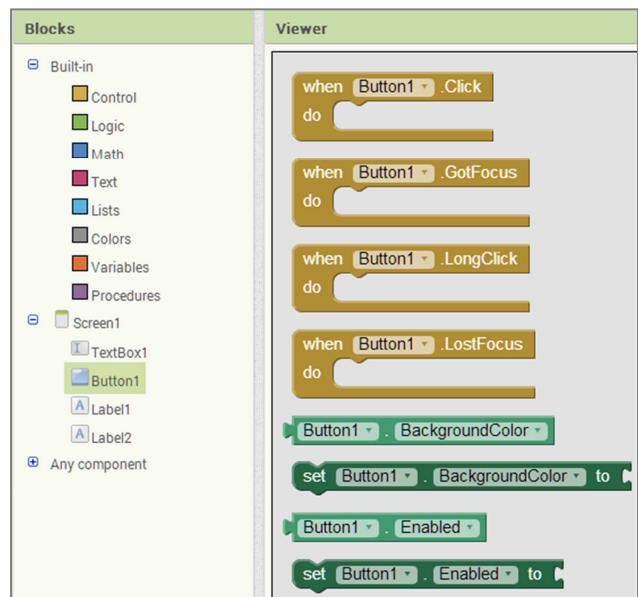
Voici l'algorithme résumant ceci :

QUAND le Bouton1 est cliqué, faire en sorte que :

- La propriété Visible du Label2 soit VRAIE (donc afficher le label2)
- La propriété Visible du Label1 soit VRAIE (donc afficher le label1)
- Le label 1 affiche le texte de la TextBox (donc afficher le texte saisi)

FIN

## Réalisation de l'algorithme à l'aide des blocs :



Cliquer sur l'élément « Button1 » :

Des blocs apparaissent :

Choisir le premier bloc, il restera en position sur la partie « Viewer » de votre écran.

Ce bloc correspond à la partie de l'algorithme :

QUAND le Bouton1 est cliqué, faire en sorte que :

FIN

when **Button1** .Click  
do [ ]

Cliquer ensuite sur le Label2 :

Choisir le bloc « set Label2.Visible to »

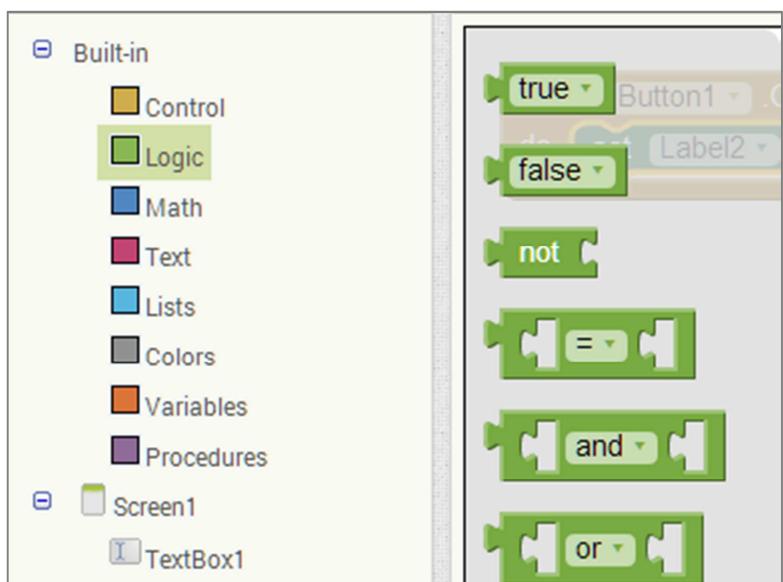
set **Label2** . Visible to [ ]

when **Button1** .Click  
do [set **Label2** . Visible to [ ]]

Positionner le block dans le bloc précédent :

Le label2 deviendra Visible si le clic a eu lieu (donc si VRAI) :

Cliquer sur le bloc « True » dans la liste des blocs logiques (Logic Blocks)



Le bloc reste à l'écran : le positionner à la suite du bloc « Set Label2.Visible to »...  
Le Label2 deviendra Visible si le clic a eu lieu.

when **Button1** .Click  
do [set **Label2** . Visible to [true]]

Procéder de la même façon pour le

label1.

Puis emboîter les blocs dans le bloc

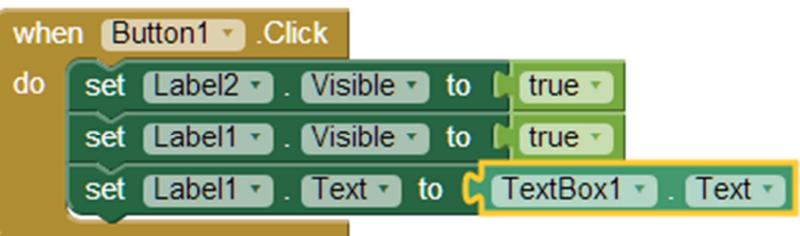
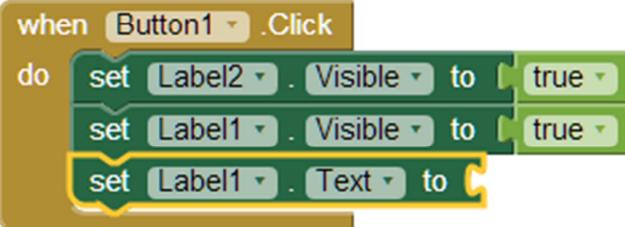
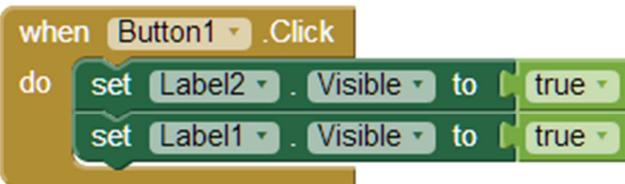
« When Button1.Click do » :

Il est maintenant nécessaire de préciser que dans le label1 il faut afficher le texte saisi dans la TextBox :

Cliquer sur Label1 et choisir le bloc :

« Set Label1.Text to » ;

Positionner correctement le bloc :



Cliquer sur l'élément TextBox1 et choisir le bloc « TextBox1.Text »

Positionner le bloc à la suite du bloc

« Set Label1.Text to » :

L'algorithme a été entièrement réalisé.

Sauvegarder votre programme.

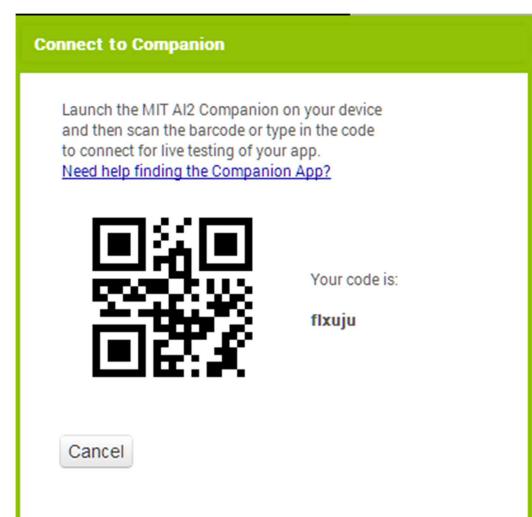
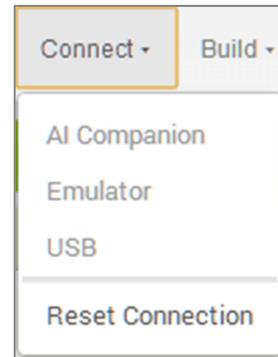
### 3<sup>ème</sup> étape du projet : « Tester l'application sur l'émulateur ou le smartphone »

Afin de tester l'application vous pouvez choisir le test sur :

- L'émulateur : un écran s'affichera sur l'ordinateur,
- AI Companion : dans ce cas, la connexion se fera directement sur le smartphone en wifi,
- USB : la connexion se fera sur le smartphone via un câble USB.

Sur le smartphone, l'application **MIT AI2 Companion** doit être installée à télécharger sur le playStore.

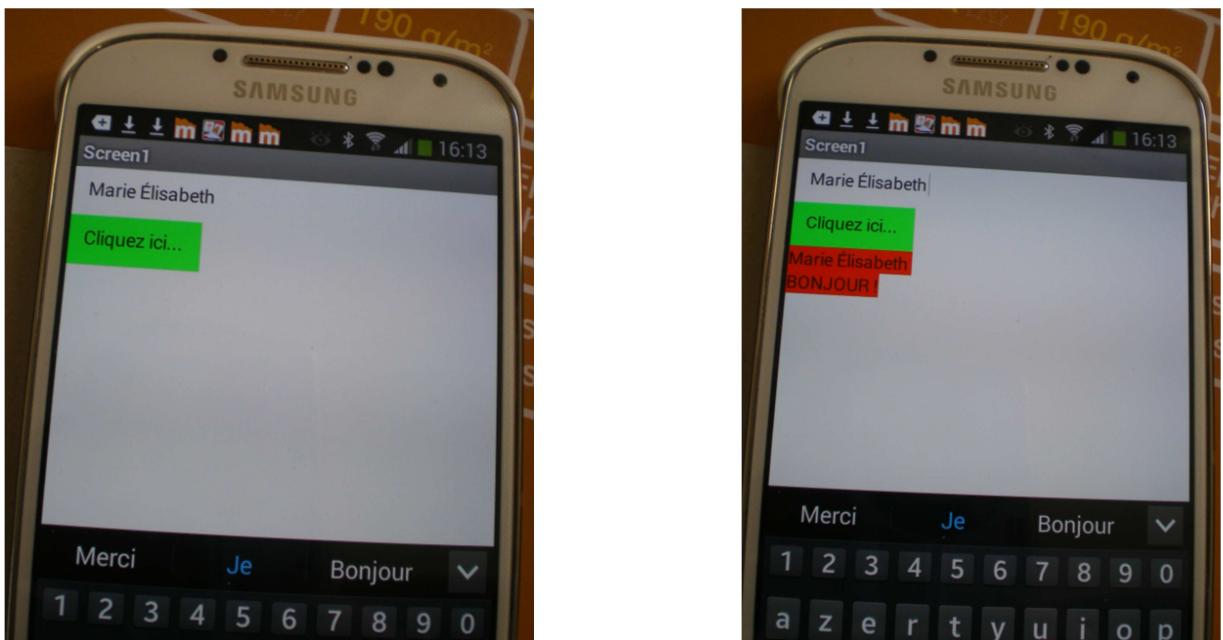
Sur le PC, l'application MIT **Emulator in AI2** doit être installée  
<http://appinventor.mit.edu/explore/ai2/windows.html>



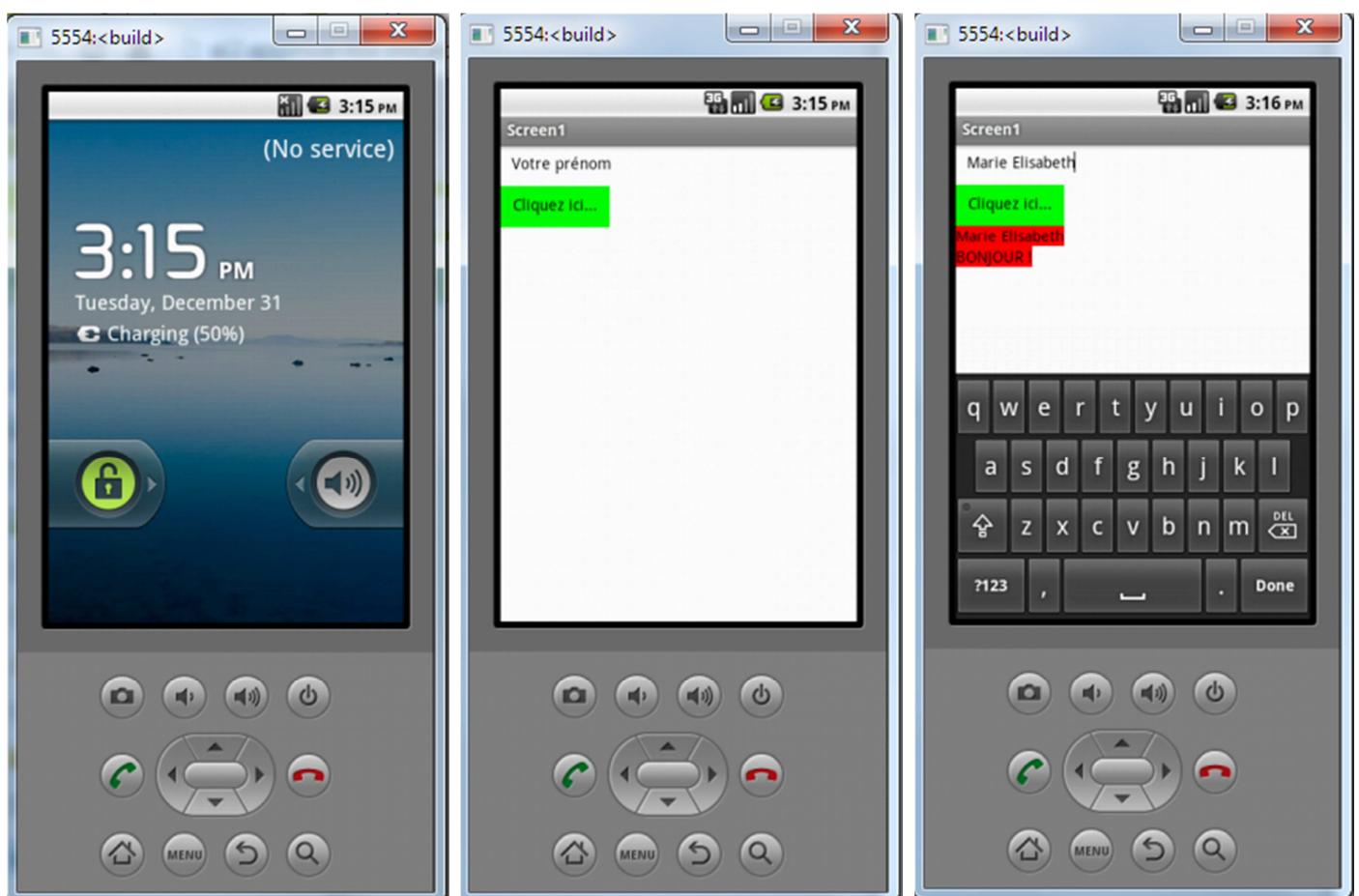
En choisissant AI Companion : un Code de 6 caractères est généré ainsi qu'un QR Code.

Il suffit de saisir le code ou de scanner le QR Code pour que l'application soit visible sur le smartphone et vous pouvez ensuite la tester :

L'application démarre sur le Smartphone :



Sur l'émulateur : un smartphone apparaît à l'écran. Il faut le déverrouiller, puis l'application se lance automatiquement. Tester l'application comme sur un téléphone normal.



## Enregistrer votre premier projet sur votre ordinateur :

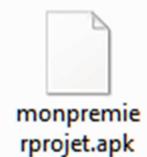
Dans l'onglet « Build » Choisir “App (save to my computer)”



La compilation de votre application s'exécute :

Vous obtenez un fichier APK qui sera sauvegarder dans votre dossier téléchargement :

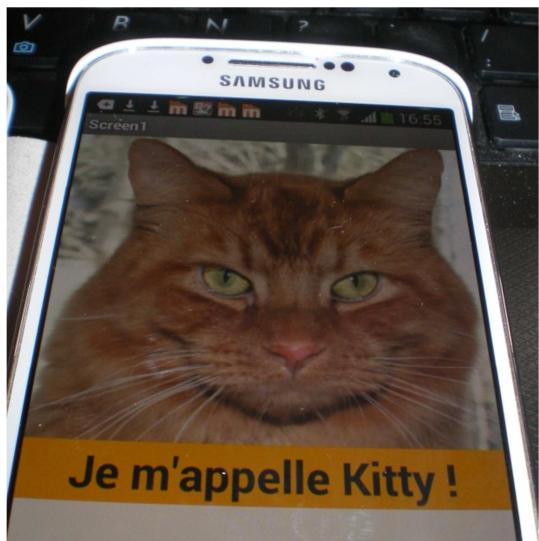
Application compilé : monpremierprojet.apk est prêt pour être installé sur un smartphone Android.



## Réaliser un deuxième projet avec APP INVENTOR

Réalisation d'une deuxième petite application qui va permettre d'afficher l'image d'un chat sur l'écran du smartphone.

Cette image sera en fait un bouton, lorsque l'on cliquera sur le bouton, nous entendrons un miaulement du chat.

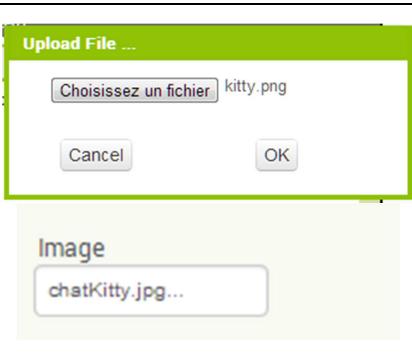
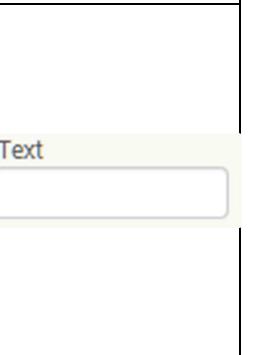


### Réalisation de l'interface :

Sur l'écran (Add screen) ajouter un bouton et un label (étiquette).



Définir les propriétés du bouton :

Choisir une image en cliquant sur « Upload File »	Rechercher le fichier kitty.png dans vos dossiers et demander le fichier sélectionné dans la propriété : Image	Définir la taille de l'image : choisir comme largeur et hauteur : Fill Parent qui permet de s'adapter à la taille de l'écran.	Supprimer ensuite le texte qui reste sur l'image.
			

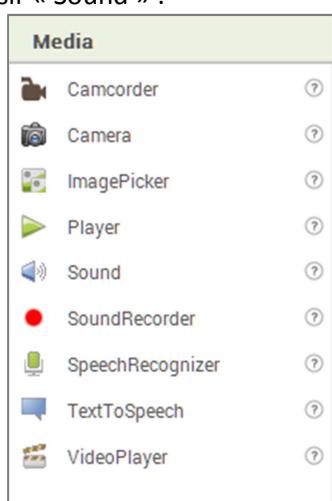
Mettre en forme le texte :



BackgroundColor	Orange
FontBold	<input checked="" type="checkbox"/>
FontItalic	<input type="checkbox"/>
FontSize	30
FontTypeface	default
Text	Je m'appelle Kitty !
TextAlignment	center
TextColor	Black
Visible	showing
Width	Fill parent...
Height	Fill parent...

Ajouter le miaulement du chat :

Dans média choisir « Sound » :



Positionner le son sous l'écran :



Rechercher le son :		
<p>Source</p> <p>None kitty.png</p> <p>Upload File ...</p> <p>Cancel OK</p>	<p>Upload File ...</p> <p>Choisissez un fichier meow.mp3</p> <p>Cancel OK</p>	<p>Sound1</p> <p>MinimumInterval 500</p> <p>Source meow.mp3...</p>

## Décrire le comportement de l'application :

Passer en mode « blocks ».



Indiquer que lorsque l'on clique sur le bouton,  
Le son est joué et que celui-ci dure 500 millisecondes.



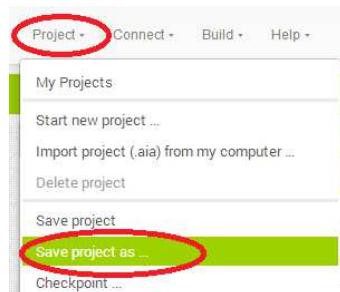
## Tester l'application :

Tester l'application sur le smartphone ou dans l'émulateur :



## Enregistrer l'application :

Project / Save project as ... / KittySon.apk



## Réaliser un troisième projet avec APP INVENTOR : MaCalculette

Réalisation d'une petite calculatrice 4 opérations :



### 1<sup>ère</sup> étape : réaliser l'interface Homme-Machine

Cette interface se compose :

- D'un premier tableau contenant 2 colonnes et 3 lignes.
- Chaque ligne contenant un « libellé et une zone de saisie.
- D'un deuxième tableau contenant 1 ligne et 4 colonnes, chaque cellule du tableau contient un bouton avec le signe de l'opération.
- Un dernier bouton est ajouté pour remettre à zéro les valeurs de la calculatrice.



#### Création d'un premier tableau :

- Choisir la forme du tableau dans la palette « Layout » / « TableArrangement » :



- Indiquer ensuite le nombre de lignes et de colonnes dans les propriétés :

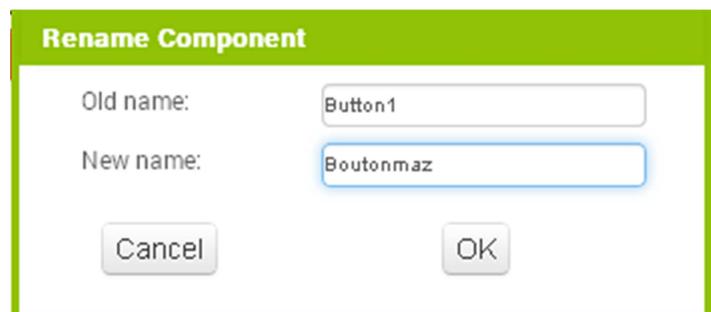


### Renommer les objets utilisés dans l'interface :

Il est important de renommer les objets utilisés pour l'élaboration de l'interface, cela permettra ensuite de mieux les reconnaître lorsque nous les utiliserons dans la partie programmation événementielle.

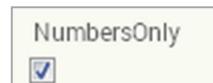
Pour renommer un objet :

- Cliquer sur l'objet à renommer, exemple Button1, puis cliquer sur « Rename » dans la partie « Components »
- L'ancien nom de l'élément s'affiche, saisir le nouveau nom puis cliquer sur « OK »



### Préciser que les valeurs saisies doivent être obligatoirement des valeurs numériques :

Afin d'effectuer les calculs, les valeurs saisies dans les zones de texte (TextBox) doivent obligatoirement être des nombres.



Il faut le préciser dans les propriétés (zone de saisie des deux nombres et zone d'affichage du résultat).

Pour cela, cocher la case « NumbersOnly »

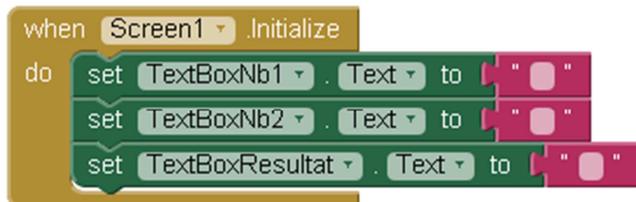


### La liste des éléments utilisés dans l'interface :

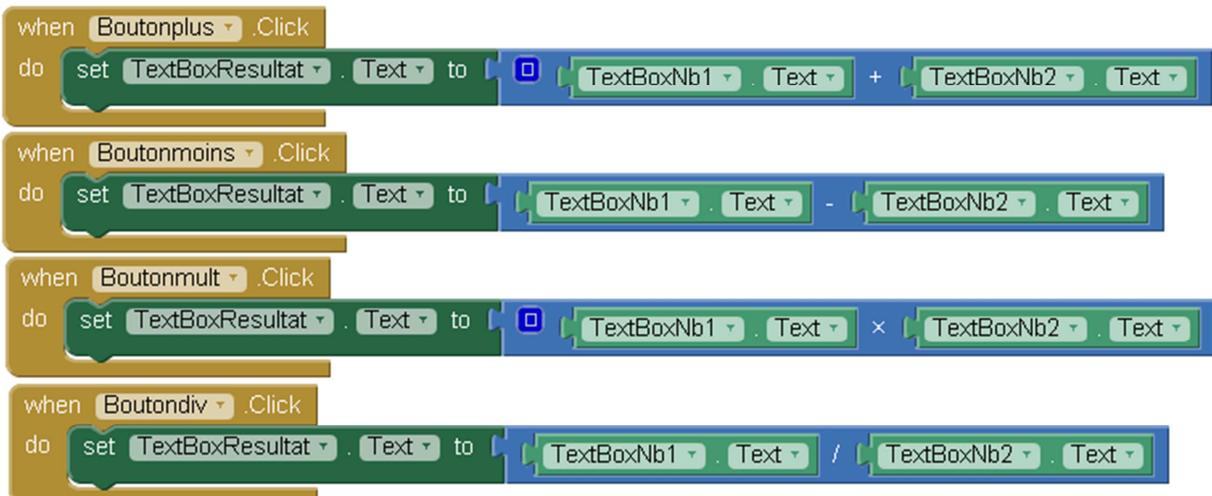
## 2<sup>ème</sup> étape : la programmation événementielle de l'application calculatrice

3 grandes parties dans cet algorithme :

- L'initialisation de l'écran : au départ les libellés (labels) s'affichent mais les zones de texte sont vides (TextBox).



- Lorsque l'on clique sur l'un des signes d'opération : le calcul doit être effectué et affiché dans la zone de texte Résultat.



- Enfin lorsque l'on clique sur le bouton « Mise à zéro », les zones de saisies doivent être effacées :



## 3<sup>ème</sup> étape : Tester l'application sur l'émulateur ou le smartphone...

(cf. photo de départ)

**Amélioration possible :** Ajouter des images dans les boutons (image d'un plus, d'un moins, d'un signe de multiplication, d'un signe de division, et d'un « reset ») par exemple.

