

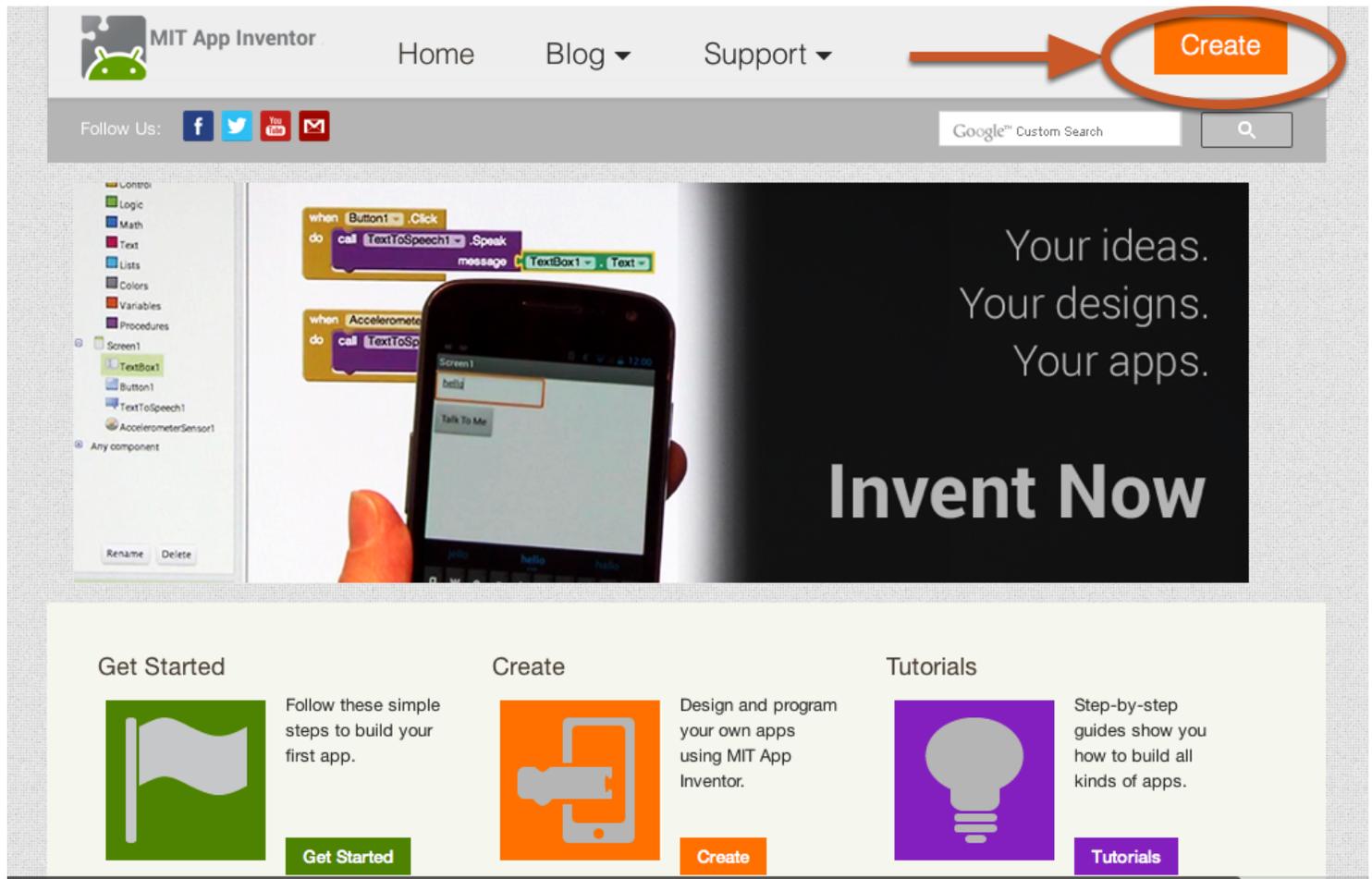
# TalkToMe : votre première appli App Inventor

Ce tutoriel vous aidera à construire une appli parlante qui peut sauvegarder des phrases sur demande.

Connectez-vous sur App Inventor avec le nom d'utilisateur et mot de passe de Gmail (Google).

Allez directement sur [ai2.appinventor.mit.edu](http://ai2.appinventor.mit.edu) ou cliquez sur le bouton orange "Create" du site web d'App Inventor ([appinventor.mit.edu](http://appinventor.mit.edu)).

Utilisez un compte Gmail existant ou le compte Google de l'école pour vous connecter sur [ai2.appinventor.mit.edu](http://ai2.appinventor.mit.edu)



MIT App Inventor

Home Blog Support **Create**

Follow Us: [f](#) [t](#) [y](#) [m](#)

Google™ Custom Search

Your ideas.  
Your designs.  
Your apps.

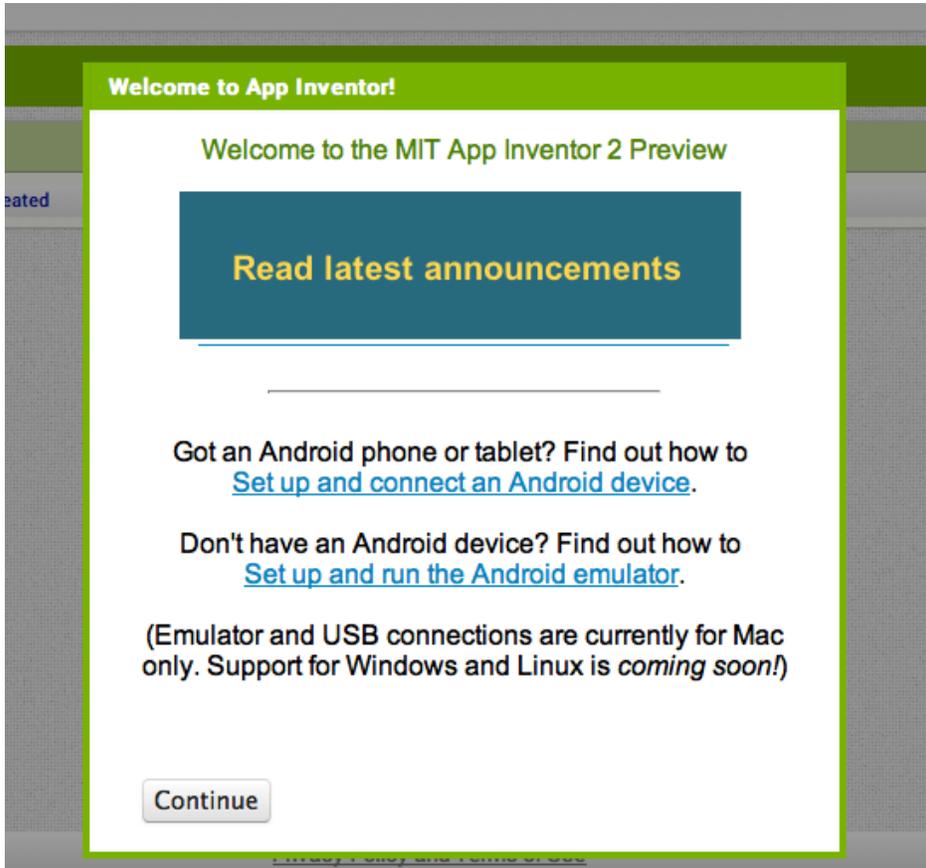
**Invent Now**

**Get Started**  
Follow these simple steps to build your first app.  
**Get Started**

**Create**  
Design and program your own apps using MIT App Inventor.  
**Create**

**Tutorials**  
Step-by-step guides show you how to build all kinds of apps.  
**Tutorials**

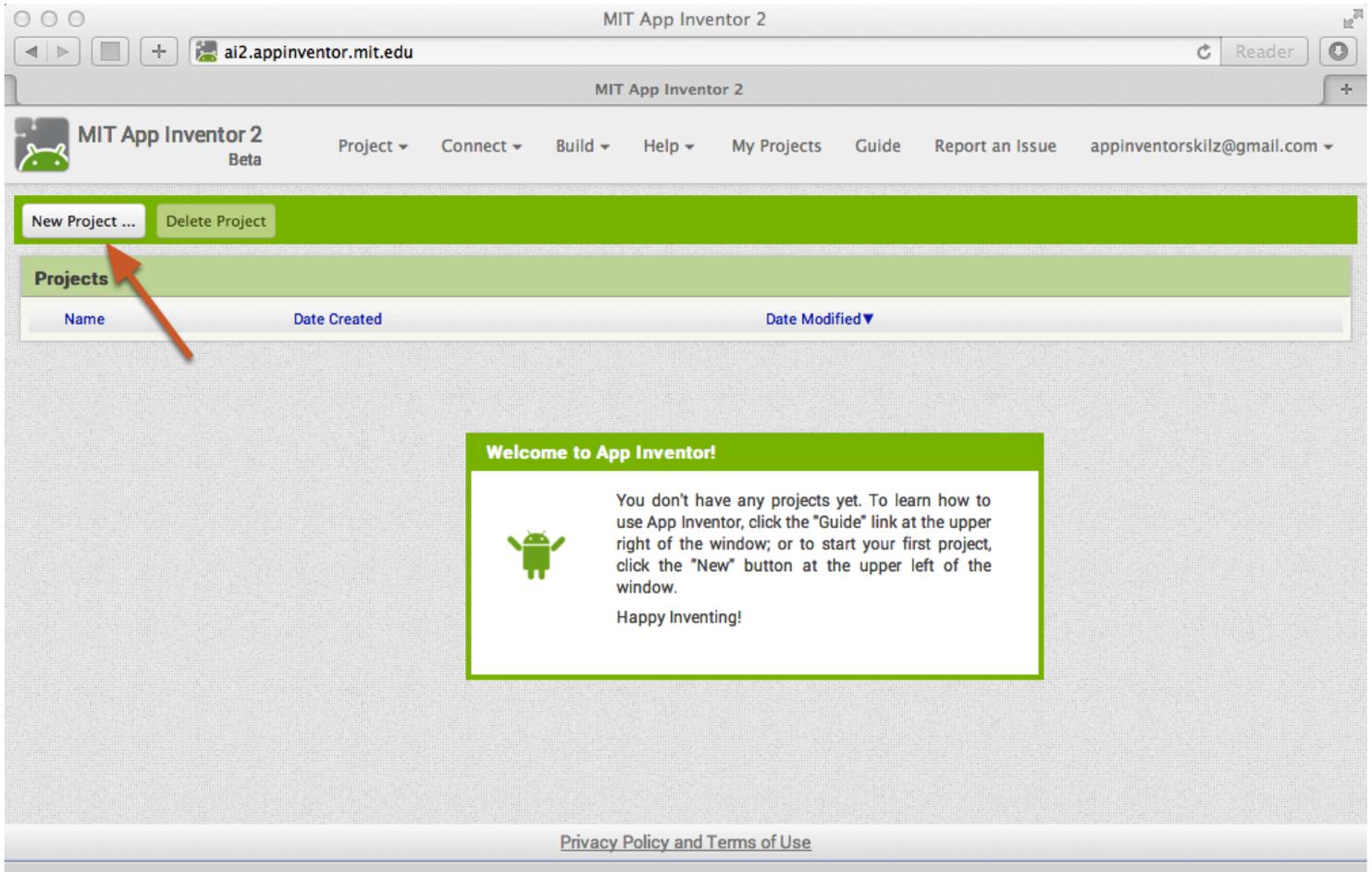
Cliquez sur "Continue" pour sortir de l'écran d'accueil.



**Commencer un nouveau projet.**

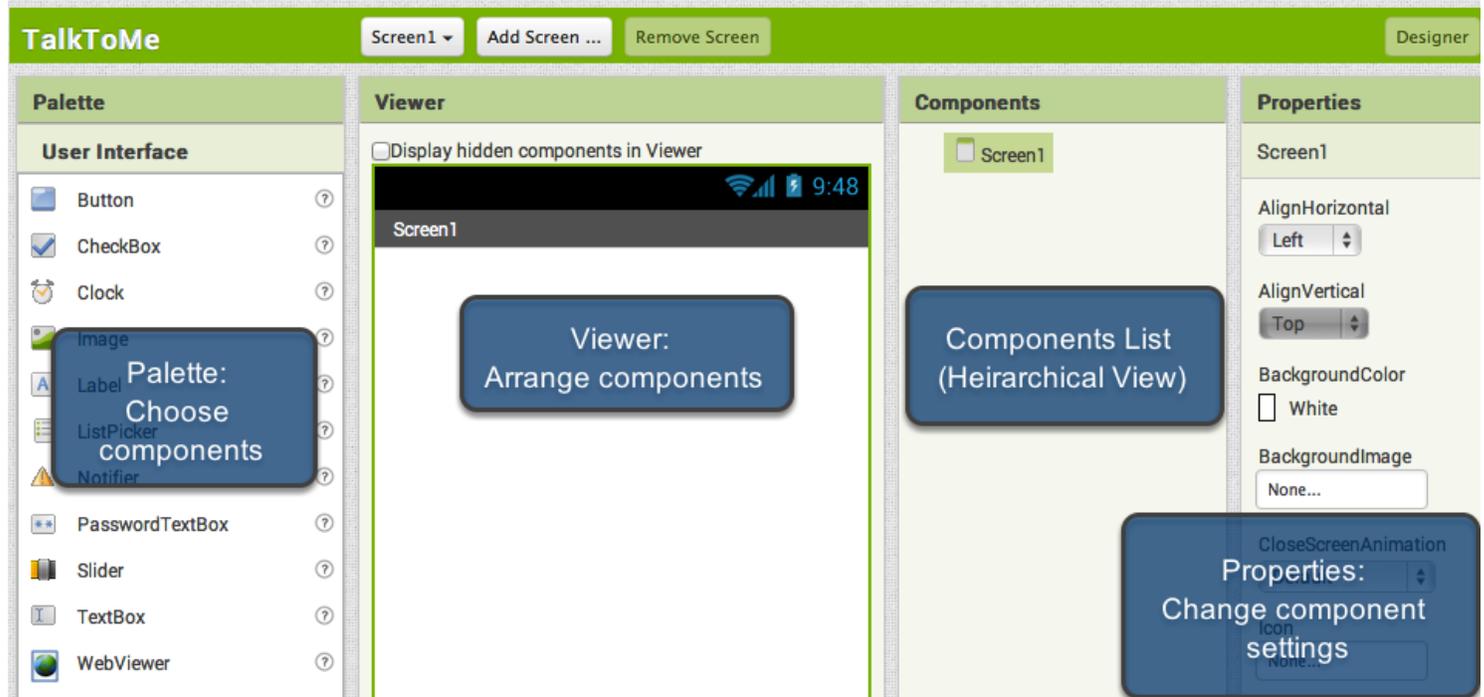
**Nommez ce projet "TalkToMe" (sans espaces !)**

Tapez le nom du projet (les tirets bas sont autorisés mais pas les espaces) et cliquez sur OK.



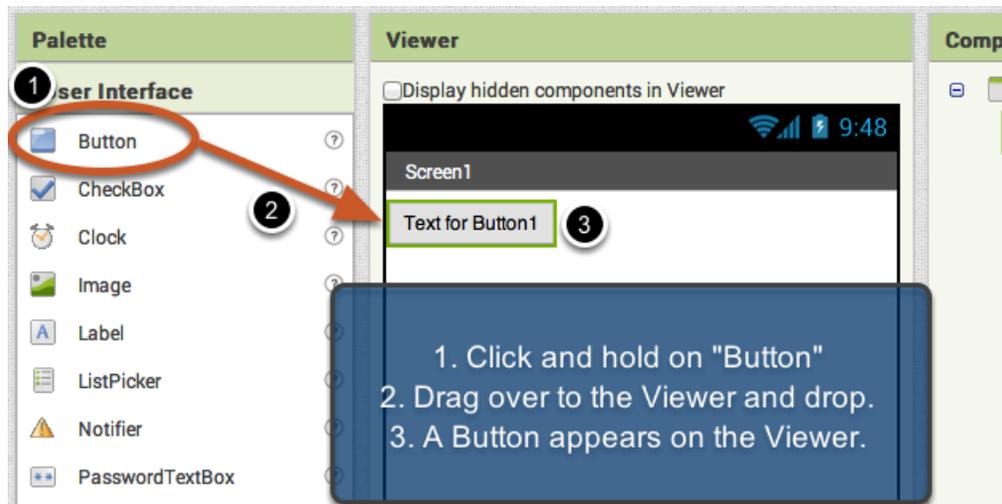
## Vous êtes maintenant dans l'espace Designer où il vous est possible d'afficher l'"user interface" (interface utilisateur) de votre appli.

Avec Design Window, ou "Designer", vous pouvez visualiser l'apparence de votre appli et sélectionner les composants qu'elle doit avoir. Vous pouvez aussi choisir des options pour l'interface utilisateur telles que Buttons, Images, Text Boxes, et des fonctionnalités telles que Text-to-Speech, Sensors et GPS.



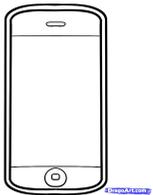
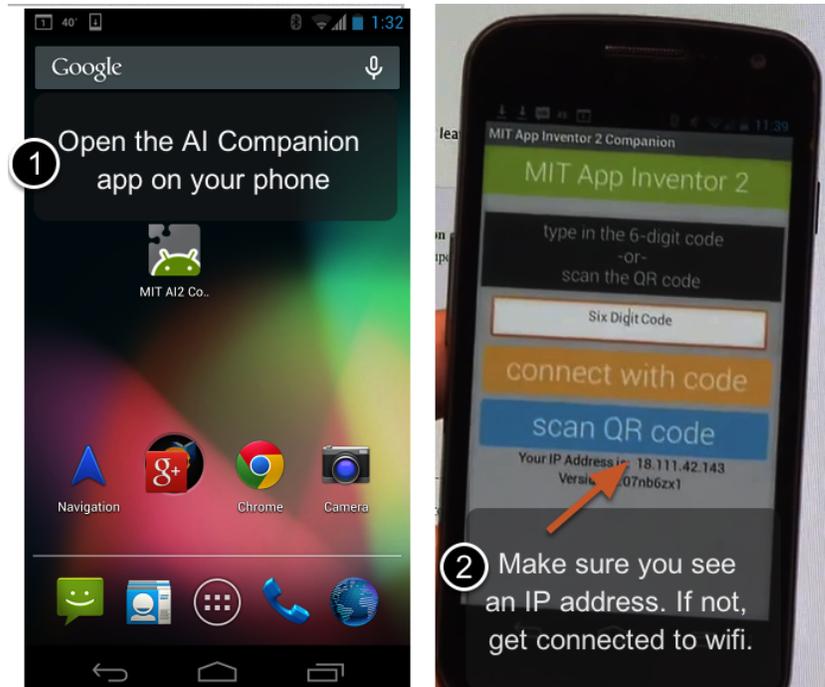
### Ajouter un bouton

Notre projet a besoin d'un bouton. Cliquez sans relâcher sur le mot "Button" dans la palette. Déplacez la souris sur Viewer. Relâchez le bouton et là un nouveau bouton va apparaître sur le Viewer.



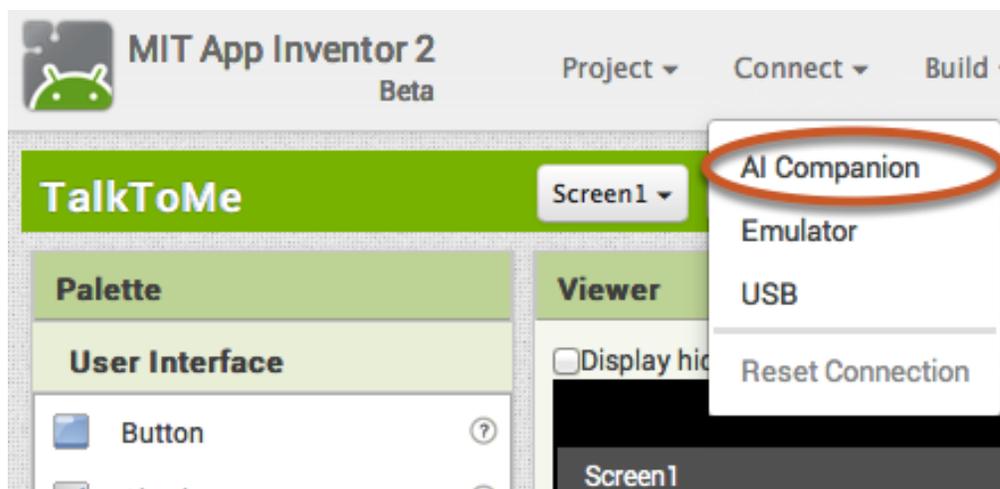
## Démarrez AICompanion sur votre appareil

Depuis votre téléphone ou votre tablette, cliquez sur l'icône qui permettra à MIT AI Companion de démarrer l'appli. NOTEZ : votre téléphone et votre ordinateur doivent être tous les deux connectés sur le même réseau sans fil. Assurez-vous que le WiFi de votre téléphone est activé et que vous êtes connectée sur le réseau sans fil local. Si vous ne pouvez pas vous connecter par WiFi, allez à Setup Instructions sur le site web de l'App Inventor pour voir comment vous connecter avec un câble USB.



## Connectez l'App Inventor sur votre téléphone pour effectuer des tests en temps réel.

Une des fonctions la plus intéressante de App Inventor c'est que vous pouvez visualiser et tester votre appli sur un appareil connecté pendant que vous la construisez. Si vous avez un **téléphone Android ou une tablette**, suivez les étapes ci-dessous.



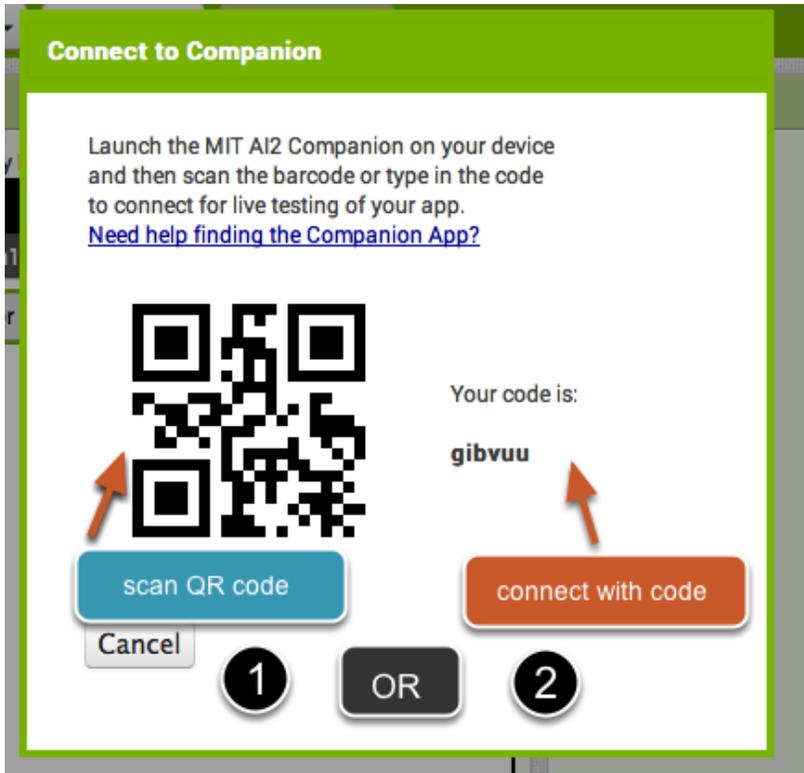
# Obtenez votre Connection Code sur App Inventor et scannez ou tapez-le sur Companion App

Sur le menu Connect, sélectionnez "AI Companion". Vous pouvez vous connecter en :

1 - Scannant le code QR et en cliquant sur «Scan QR code» (#1)

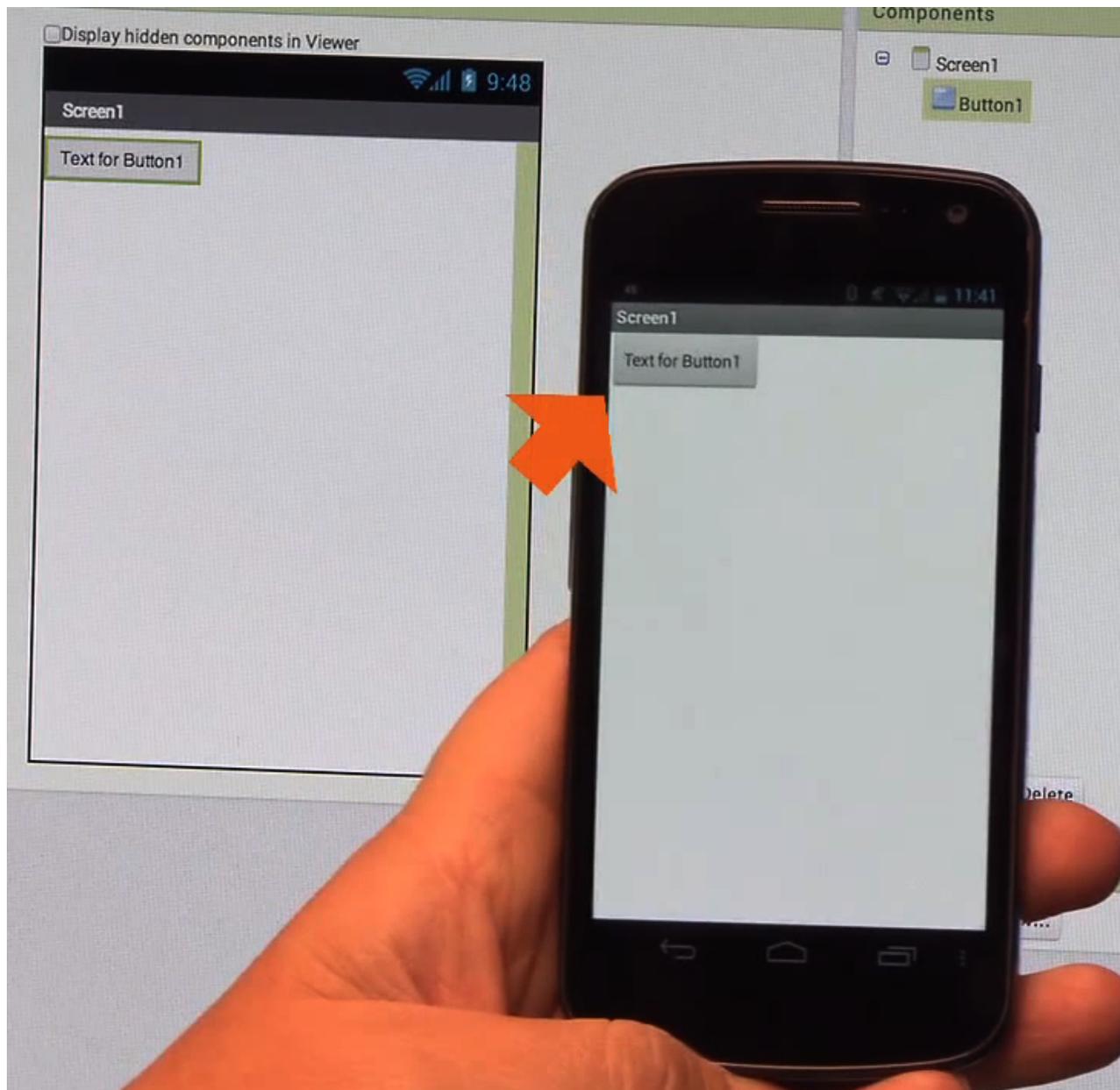
ou

2 - En tapant le code dans la fenêtre de texte et en cliquant sur "Connect with code" (#2)



## Visualisez votre appli sur l'appareil connecté

Vous pouvez vérifier que votre connexion fonctionne si vous arrivez à visualiser votre appli sur l'appareil connecté. Pour le moment votre appli ne possède qu'un seul bouton, par conséquent c'est la seule chose que vous pouvez voir. C'est en progressant sur votre projet que vous verrez votre appli changer sur votre téléphone.



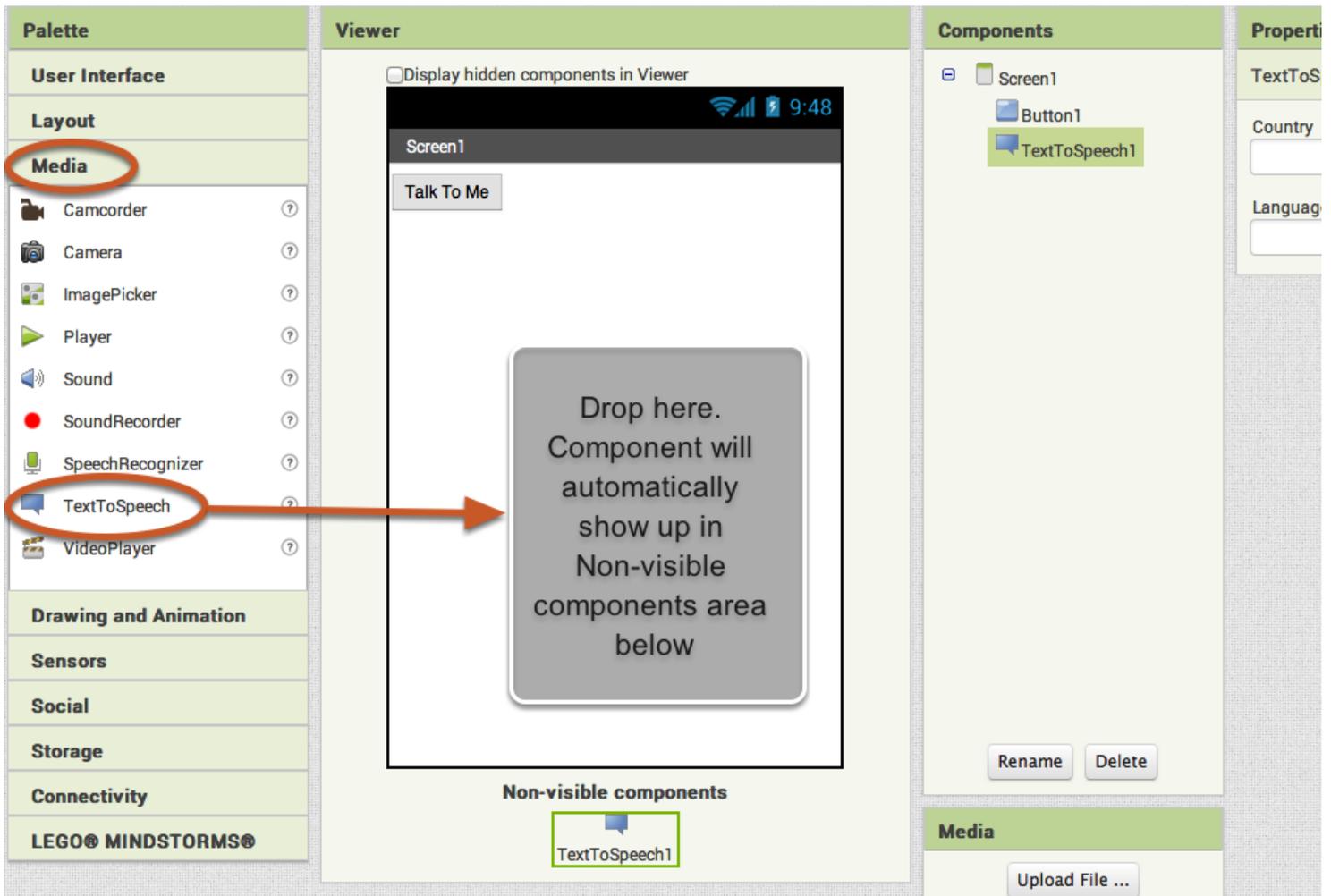
## Changer le texte sur le bouton

Dans le panneau des propriétés, changez le texte pour le bouton. Sélectionnez le texte "Text for button 1", effacez-le et tapez "Talk To Me". Notez que le texte sur le bouton de votre appli change instantanément.

The screenshot displays the App Inventor interface with three main panels: Viewer, Components, and Properties. In the Viewer panel, a mobile app preview shows a screen with a button labeled "Talk To Me". The Components panel shows a tree view with "Screen1" containing "Button1". The Properties panel is open for "Button1", showing various settings. A context menu is overlaid on the "Talk To Me" button in the Viewer, with the "Text" property highlighted. The context menu lists properties: Shape (default), ShowFeedback (checked), Text (Talk To Me), TextAlignment (center), and TextColor (Default). The Properties panel also shows the "Text" property set to "Talk T|".

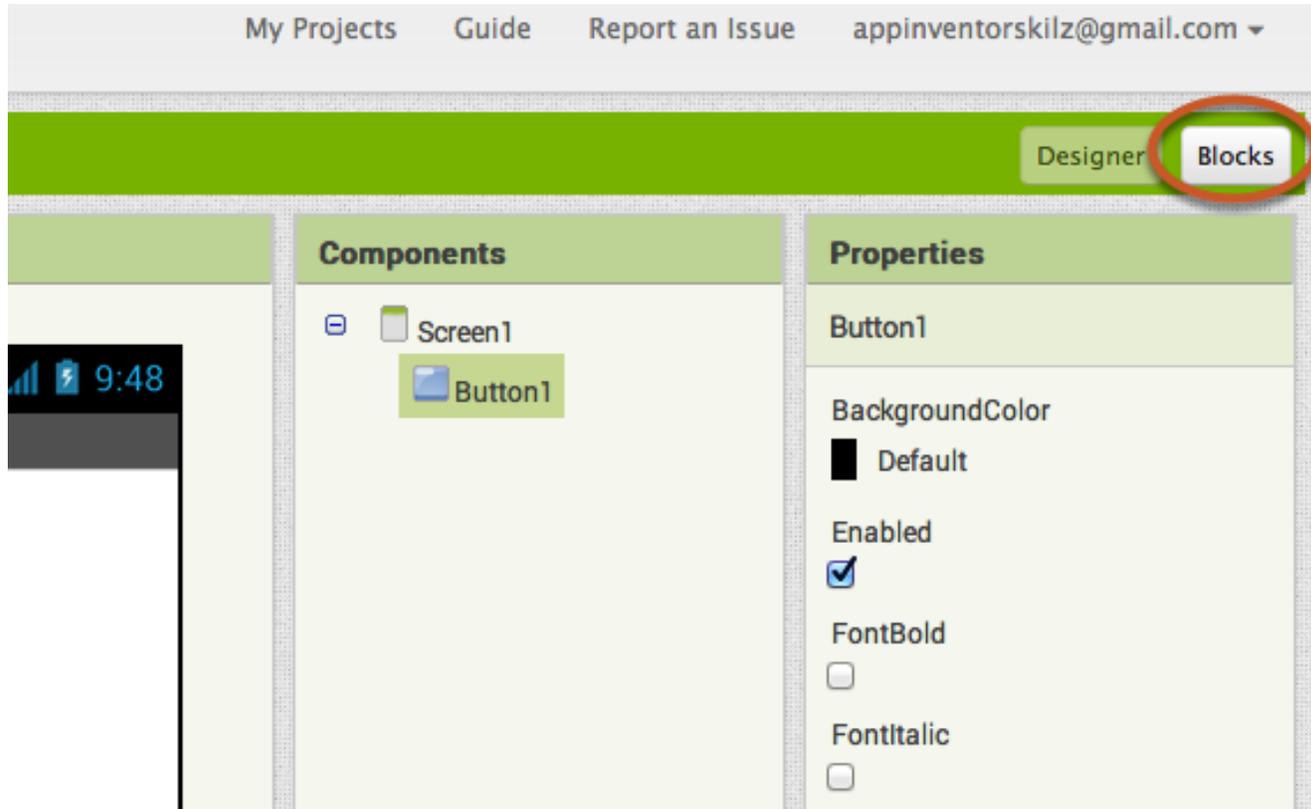
## Ajoutez le composant Text-to-Speech à votre appli

Allez sur le menu Media et faites glisser un composant TextToSpeech. Déposez-le dans le Viewer. Notez qu'elle apparaît maintenant sous la catégorie "Non-visible components" parce c'est quelque chose qui n'apparaîtra pas sur l'interface de l'utilisateur. Il s'agit plutôt d'un outil qui est disponible sur l'appli.



## Passez aux Blocks Editor

Il est temps de dire à votre appli quoi faire ! Cliquez sur "Blocks" pour aller sur Blocks Editor. Jusqu'à présent, nous avons seulement décidé de l'apparence de l'appli. Le Blocks Editor vous permettra de savoir comment l'appli doit se comporter. Les boutons Designer et Blocks sont comme des onglets -- Utilisez-les pour passer d'un espace à l'autre sur App Inventor.



# Le Blocks Editor

Le Blocks Editor vous permet de programmer le comportement de l'appli. Ce sont des blocs intégrés pouvant gérer des choses comme les maths, la logique et le texte. Ci-dessous vous trouverez des blocs associés à chaque composant que contient votre appli. Pour qu'un bloc pour un certain composant apparaisse dans le Blocks Editor, vous devrez dans un premier temps ajouter ce composant dans l'appli via le Designer.

The screenshot shows the MIT App Inventor 2 interface. At the top, the title bar reads 'MIT App Inventor 2 Beta' and 'TalkToMe'. Below the title bar, there are navigation buttons: 'Screen1', 'Add Screen ...', and 'Remove Screen'. On the right side of the title bar, there are buttons for 'Designer' and 'Blocks'. The main interface is divided into two panels: 'Blocks' on the left and 'Viewer' on the right. The 'Blocks' panel is organized into categories: 'Built-in' (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), 'Screen1' (Button1, TextToSpeech1), and 'Any component'. The 'Viewer' panel shows a workspace with a large box containing the text 'Workspace where you assemble the blocks into a program.' and a trash can icon with the text 'Trash for deleting unneeded blocks.' Below the workspace, there are warning indicators (0 yellow and 0 red triangles) and a 'Show Warnings' button. The interface also includes 'Rename' and 'Delete' buttons at the bottom left.

## Créer un bouton événement clic

Cliquez sur le menu Button1. Cliquez sans relâcher sur le bloc When Button1.Click do. Déplacez-le jusqu'à l'espace de travail et déposez-le. C'est ce bloc qui va gérer ce qui se passe quand on clique sur le bouton de votre appli. On l'appelle le "Event Handler" (le gestionnaire d'évènements). Tous les Event Handlers sont de couleur brun/marron dans AI2. Les Event Handlers sont déclenchés quand un événement est initié par l'utilisateur (par exemple, cliquer sur un bouton).



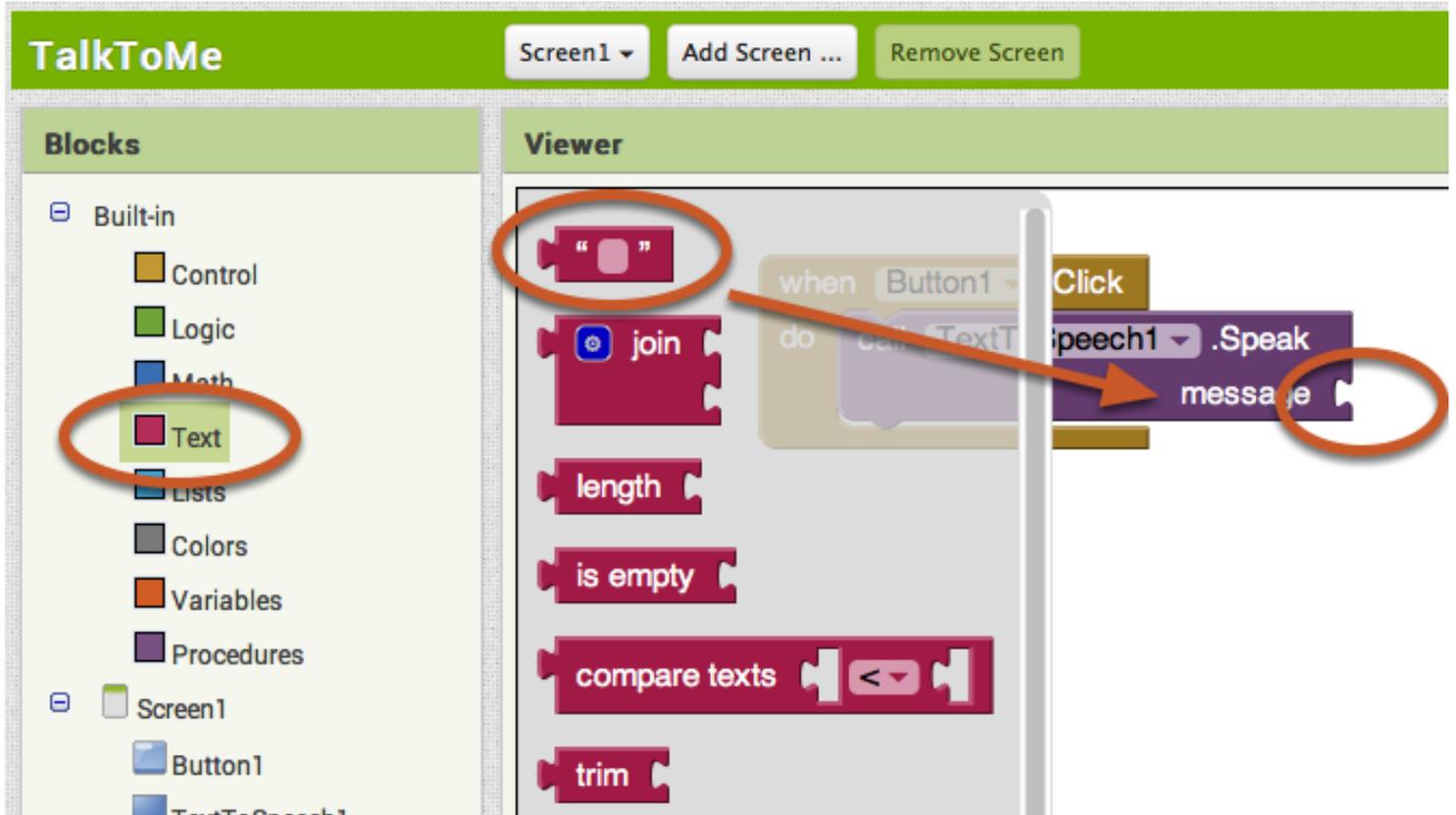
# Programmer l'action TextToSpeech

Cliquez sur le menu TextToSpeech. Déplacez le bloc call TextToSpeech1.Speak sur l'espace de travail et déposez-le. Nous appelons ce bloc violet "procédure" dans App Inventor. C'est grâce à cette procédure que votre téléphone peut parler. Parce que la fonction se situe à l'intérieur de Button.Click, elle se mettra en route dès que le bouton de votre appli sera cliqué. Les procédures doivent être appelées sur un événement (par exemple, le bouton événement clic).

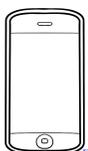
The screenshot shows the MIT App Inventor 2 Beta interface. The top navigation bar includes 'Project', 'Connect', 'Build', and 'Help' menus, along with 'My Projects', 'Guide', and 'Report an Issue' links. The main workspace is titled 'TalkToMe' and contains a 'Screen1' tab with 'Add Screen ...' and 'Remove Screen' buttons. On the left, the 'Blocks' palette is visible, with the 'TextToSpeech1' block under the 'Procedures' category circled in red and labeled with a '1'. In the center, the 'Viewer' pane shows a code block for 'when TextToSpeech1 .AfterSpeaking' with a 'result' block and a 'do' block containing a 'call TextToSpeech1 .Speak message' block, which is circled in red and labeled with a '2'. Below this, there are blocks for 'TextToSpeech1 . Country', 'set TextToSpeech1 . Country to', and 'TextToSpeech1 . Language'. On the right, a 'when Button1 .Click' block is circled in red and labeled with a '3', containing a 'do' block with a 'call TextToSpeech1 .Speak message' block. An orange arrow points from the 'call TextToSpeech1 .Speak message' block in the 'when Button1 .Click' block to the 'call TextToSpeech1 .Speak message' block in the 'when TextToSpeech1 .AfterSpeaking' block.

## Rédiger un message sur le bloc TextToSpeech.Speak

C'est presque fini ! Il ne vous reste plus qu'à informer le bloc TextToSpeech.Speak sur ce qu'il doit dire. Pour ce faire, cliquez sur le menu Text, faites glisser un bloc de texte et branchez-le dans la prise étiquetée "message".



Cliquez sur le bloc texte et tapez "Félicitations ! Vous venez de construire votre première appli." (Vous pouvez bien entendu choisir la phrase que vous préférez, ça n'est là qu'une suggestion).



### Maintenant, testez !

Allez sur l'appareil connecté et cliquez sur le bouton. Assurez-vous que le volume soit suffisamment haut ! Vous devriez entendre le téléphone prononcer votre phrase tout haut.

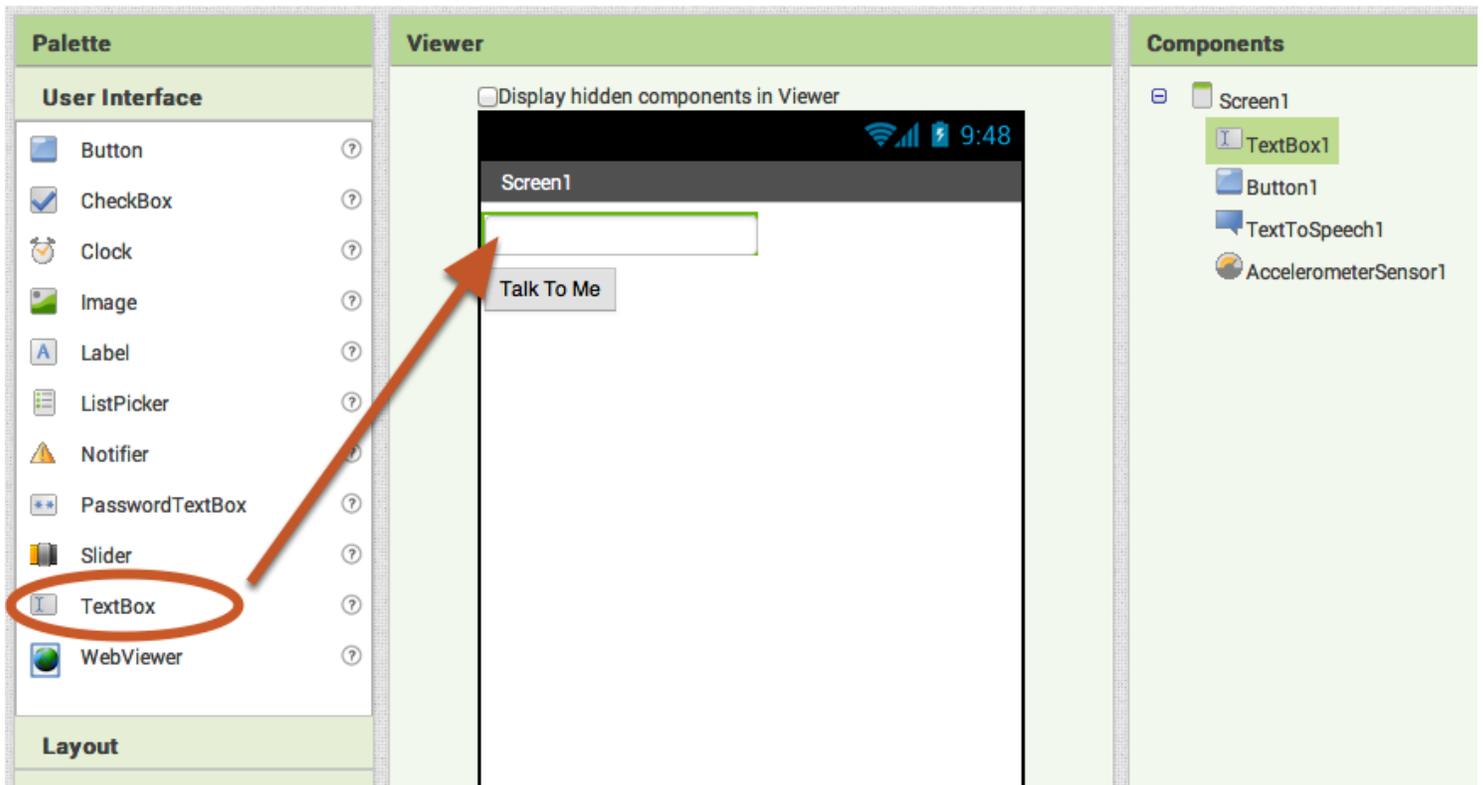
## Retournons sur l'onglet Designer

Maintenant, faites dire à votre téléphone ce que vous souhaitez ! Pour cela, retournez sur l'onglet Designer. Cliquez sur "Designer" en haut à droite.



## Ajouter une Text Box à votre interface utilisateur.

Depuis le menu User Interface, faites glisser une TextBox et déposez-la au-dessus du bouton qui est déjà sur l'écran.



## Sélectionner le texte qui est tapé dans la TextBox.

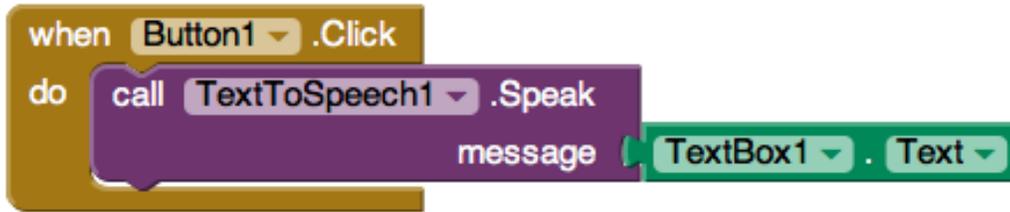
Sélectionnez la propriété du texte dans TextBox1. Les blocs vert clair dans le menu de la TextBox1 sont appelés les "getters" et les blocs vert foncé pour les composants de la TextBox1 sont appelés les "setters". On les appelle "getters" et "setters" car ils vous permettent de sélectionner ou régler la valeur qui est stockée pour cette partie du composant qui utilise ces blocs.

Vous voulez que votre appli exprime à haute voix ce qui se trouve dans le bloc TextBox1.Text. (c.a.d. quoi que ce soit qui est tapé dans la text box) Faites glisser le bloc getter TextBox1.text.

The image shows the App Inventor interface. On the left is the 'Blocks' panel, which is organized into categories: Built-in (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), Screen1 (containing TextBox1, Button1, TextToSpeech1, and AccelerometerSensor1), and Any component. On the right is the 'Viewer' panel, which displays a list of property blocks for 'TextBox1'. The blocks are: 'set TextBox1 . Height to', 'TextBox1 . Hint', 'set TextBox1 . Hint to', 'TextBox1 . MultiLine', 'set TextBox1 . MultiLine to', 'TextBox1 . NumbersOnly', 'set TextBox1 . NumbersOnly to', 'TextBox1 . Text', 'set TextBox1 . Text to', and 'TextBox1 . TextColor'. The 'TextBox1 . Text' block is circled in orange. A red box highlights the scroll bar on the right side of the Viewer panel, and an orange arrow points from the 'Text' block to the scroll bar.

## Régler le bouton événement clic pour que le texte dans la Text Box soit prononcé.

Effacez le message "félicitations..." de la text box et branchez le bloc TextBox1.Text. Vous pouvez retirer le bloc texte rose en le glissant dans la corbeille en bas à droite de l'espace de travail.



" Congratulations! You've made your

## Sauvegardons le texte comme une variable.

Le texte que l'appli prononce est maintenant une variable, c'est-à-dire qu'il peut changer en fonction de l'usage que l'on fait de l'appli. Par conséquent, nous pouvons sauvegarder le texte dans une variable. Nous pouvons nommer cette variable en cliquant sur la partie du bloc appelée "name" après l'avoir fait glisser. Choisissez un nom significatif pour vos variables. Aussi, si vous devez apporter des modifications sur votre appli dans 3 mois, il sera plus facile de retrouver la variable en question. Appelons cette variable "textToSpeak".

Pour commencer, chaque variable doit posséder une valeur. Comme cette variable sera utilisée pour stocker du texte, nous commencerons avec un texte vierge.

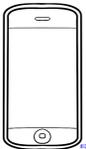
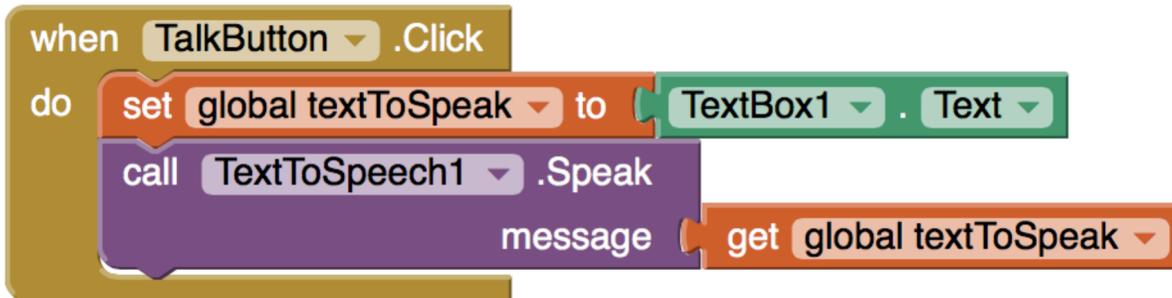
Comment utilise-t-on une variable ? La mission originale du message était la suivante :

**Message → TextBox1.Text**

Puisque nous allons maintenant utiliser la variable textToSpeak pour le message, nous devons remplacer TextBox1.Text avec la variable et attribuer la valeur de TextBox1.Text à la variable. À chaque fois qu'on clique sur le bouton, la valeur de la variable est mise à jour et le message correct passe à TextToSpeech.

**textToSpeak → TextBox1.Text**

**Message → textToSpeak**



## Testez votre appli !

La fonctionnalité doit rester la même. Que se passe-t-il si vous inversez l'ordre des blocs "set" et "call" ? Essayez et constatez par vous-même ! Décrivez ce qui se passe et pourquoi.

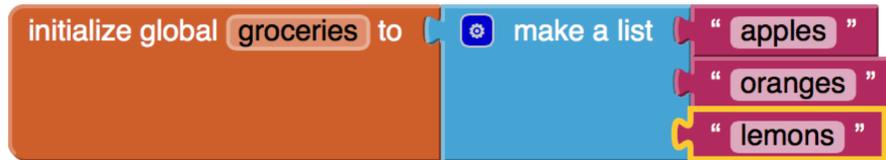
## Sauvegardez toutes les phrases que vous avez entrées.

Comment allez-vous sauvegarder ce que nous ordonnons de dire à l'appli ? Il faut trouver un moyen pour "stocker" le texte que nous tapons dans la TextBox. Voici comment vous y prendre avec une liste.

### Listes

Les listes sont exactement ce à quoi vous pensez. Elles vous permettent de garder la trace de certains articles. Vous pouvez, par exemple, suivre les articles à acheter pour vos courses dans une liste. Appelons cette liste "courses".

1. Pommes
2. Oranges
3. Citrons



Chaque article est "indexé", cela veut dire que vous pouvez demander l'article numéro 1 de votre liste, et vous obtiendrez des "pommes". Si vous demandez l'article numéro 2, vous aurez des "oranges"... et ainsi de suite.

Dans App Inventor, à chaque fois que vous ajoutez un article à cette liste, vous l'annexez à la liste, ou vous rajoutez un autre article à la fin de la liste. Par exemple, si vous ajoutez "bananes", voici à quoi ressemblera votre liste :

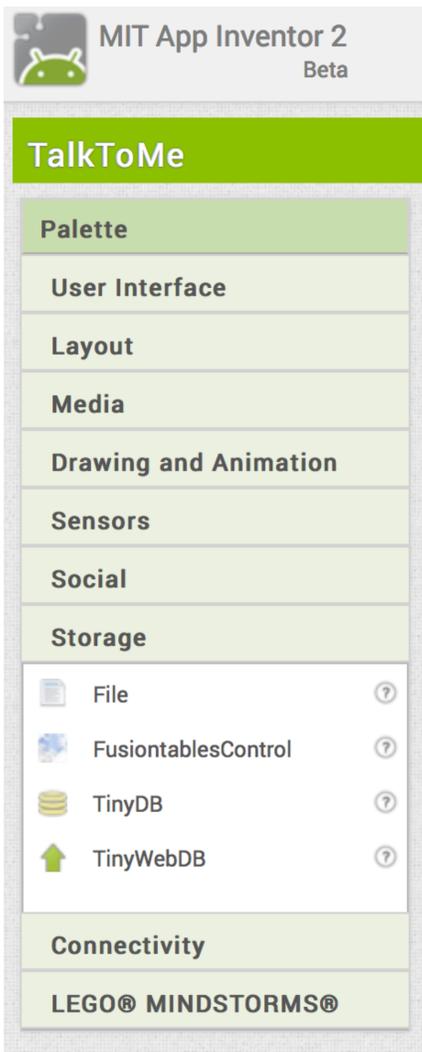
1. Pommes
2. Oranges
3. Citrons
4. Bananes



Ajouter des articles à votre liste se fait sans discrimination. Si un article existe déjà dans votre liste, vous pouvez quand même ajouter un duplicata. Si je voulais ajouter "oranges" à la liste, voici à quoi le résultat final ressemblera :

1. Pommes
2. Oranges
3. Citrons
4. Bananes
5. Oranges

Par conséquent, demander l'article numéro 2 vous retournera "oranges" mais demander l'article numéro 5 vous retournera également "oranges". Afin d'éviter cela, vérifiez dans un premier temps si l'article "is in list?" (est dans la liste ?). Ce bloc retournera "true" (vrai) si c'est le cas et "false" (faux) dans le cas contraire.



**ASTUCE** : si vous voulez plus d'information sur un composant, vous pouvez cliquer sur le point d'interrogation à côté du composant dans la palette.

**OK ! Sauvegardez les phrases dans votre appli. Mettez en oeuvre les blocs nécessaires pour sauvegarder toutes vos phrases.**

**Testez votre appli ! Est-ce que la fonctionnalité à changé ? Pouvez-vous dire si les valeurs ont été sauvegardées ?**

---

---

---

---

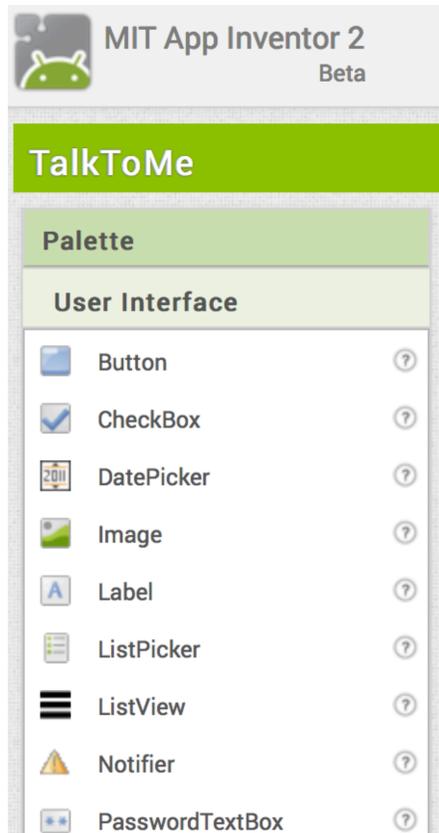
## Assurons-nous que notre sauvegarde fonctionne en ajoutant une fonction de visualisation.

Nous pouvons visualiser ce qu'il y a dans la liste de deux manières : Labels et ListViews. Explorons l'option ListView.

ListView doit être ajoutée depuis le Designer depuis la palette "User Interface", puisque cela permet à l'utilisateur de visualiser quelque chose.

Le composant ListView possède une propriété "Elements". C'est comme cela que le composant ListView va stocker une liste (tel que décrit en page 19) d'articles qu'il va afficher. En continuant avec l'exemple de liste de courses, si nous réglons ListView1.Element à "courses", la liste définie ici va être affichée par ce composant.

```
set ListView1 . Elements to get global groceries
```



En quoi le bloc TalkToMe pour ListView diffère-t-il de ceux ci-dessus ?

Le composant ListView possède également une propriété appelée "Visible". Régler cette propriété sur "false", permet à la liste d'être cachée à l'utilisateur (invisible sur l'écran). Le composant ListView sera visible par défaut (réglée sur True) si vous ne modifiez pas son statut.

```
set ListView1 . Visible to  
ListView1 . Visible
```

**Vous avez déjà ajouté une liste à vos blocs afin de sauvegarder toutes vos phrases. Comment allez-vous utiliser cela avec le composant ListView de façon à voir si toutes les phrases sont correctement sauvegardées ?**

## Ajoutons une fonction de sauvegarde sélective sur notre appli.

Il se peut que vous vouliez que votre téléphone prononce certaines phrases plusieurs fois, mais devoir retaper la même phrase chaque fois que vous souhaitez l'utiliser peut être laborieux. Nous avons déjà ajouté un moyen de sauvegarder toutes les phrases que vous avez entrées ; mais que faire si vous voulez n'en sauvegarder qu'une partie ? Comment construiriez-vous une telle fonction de sauvegarde ?

**Voici quelques composants qui peuvent être utiles :**

Button	ListView	ListPicker	TinyDB
--------	----------	------------	--------

Voici des blocs qui peuvent être utiles :

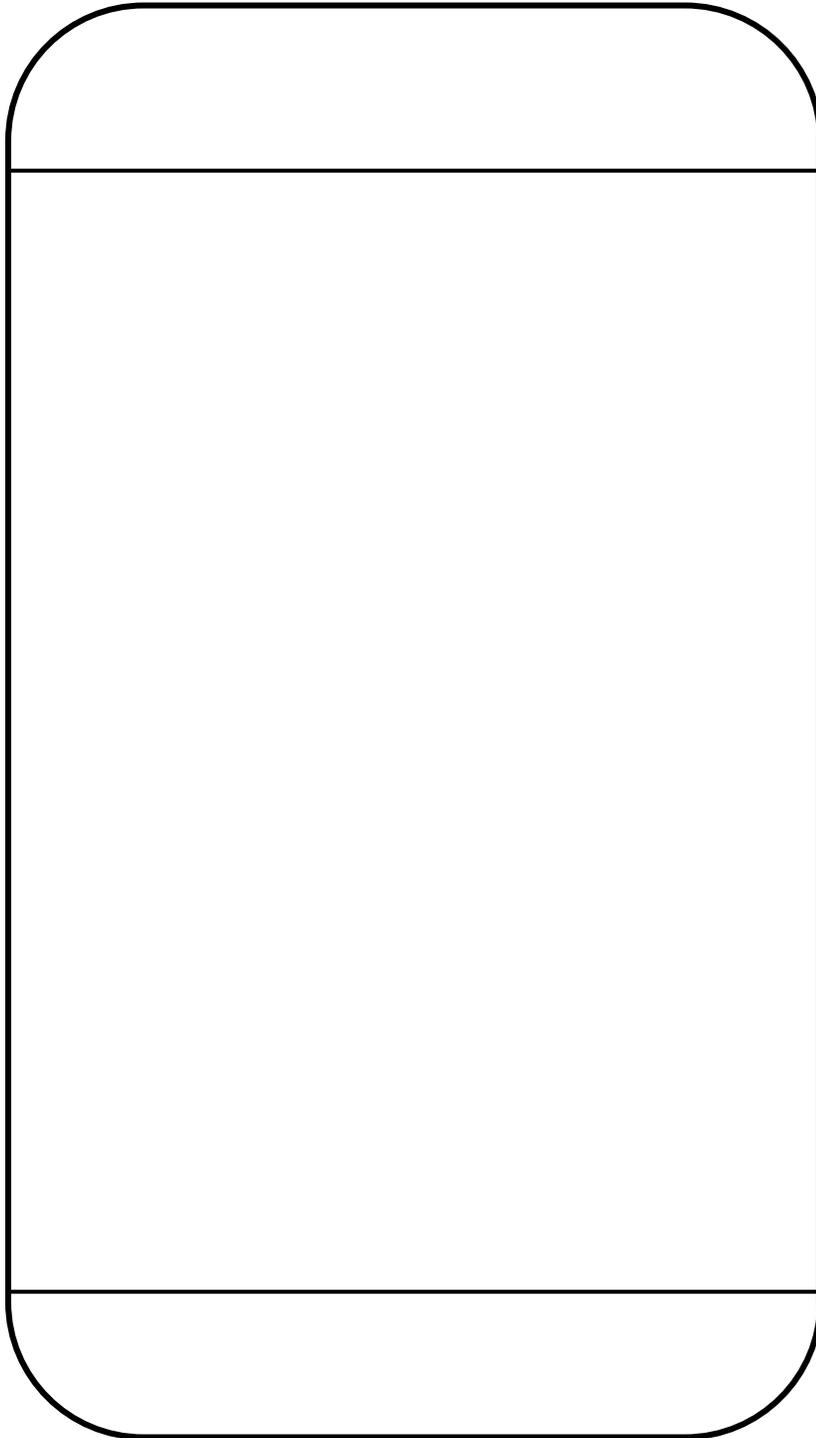
Lists	Procedures	Logic (true ou false)
-------	------------	-----------------------

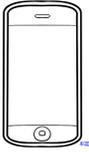
**Commencer par lister tout composant et bloc supplémentaire que vous pourriez utiliser.**

---

---

**Puis, faites un schéma de ce à quoi votre écran ressemblera après avoir ajouté le(s) composant(s) supplémentaire(s).**





**Il est temps de coder et de tester ! Comment pouvez-vous vérifier que vous sauvegardez seulement les phrases que vous voulez ?**

---

---

---

**Que se passe-t-il si vous essayez de sauvegarder les mêmes phrases maintenant ? Comment faire pour éviter ce qui vient de se produire ? (indice : cherchez autour des blocs "Lists" et "Control").**

---

---

---

**Dans quelle mesure trouvez-vous cette appli utile à ce stade ? Comment pouvez-vous utiliser d'autres composants dans App Inventor ?**

---

---

**Essayons d'ajouter le composant YandexTranslator. En quoi votre appli changerait-elle si vous ajoutiez ceci ? (C'est à vous de décider ; vous êtes le concepteur !)**

**Commencez par réaliser un prototype (schéma de l'interface de l'écran) ici, et n'oubliez pas que vous trouverez plus d'information sur le composant en cliquant sur le point d'interrogation à côté de lui dans la palette.**



## Conclusions de ce tutoriel.

À la fin de ce tutoriel, vous devriez être capable de :

- Explorer les onglets "design" et "blocks" d'App Inventor afin de construire ce dont vous avez besoin.
- Tester vos fonctions : c'est génial de pouvoir bâtir une fonction mais encore faut-il qu'elle marche correctement. Toujours penser comment vous pourriez vérifier une deuxième fois que ce que vous faites est bien ce que vous pensez faire.
- Progresser pas à pas : les fonctions sont souvent construites l'unes après l'autre. Ainsi, si un élément de votre projet casse, vous trouverez plus facilement où peut-être le problème.