

# APP INVENTOR



## AppInventor, c'est quoi ?

C'est un logiciel en ligne qui permet de créer des applications pour appareils Android (smartphone ou tablette). L'outil est gratuit et permet de développer sa créativité et ses compétences en programmation.

Ainsi, vous pourrez réaliser vos applications, mais aussi vos objets connectés via une carte arduino ou picaxe par exemple.



Il vous faudra obligatoirement un compte google !

Et un smartphone ou tablette Android pour tester votre application avec l'application MIT AI2 Companion installée



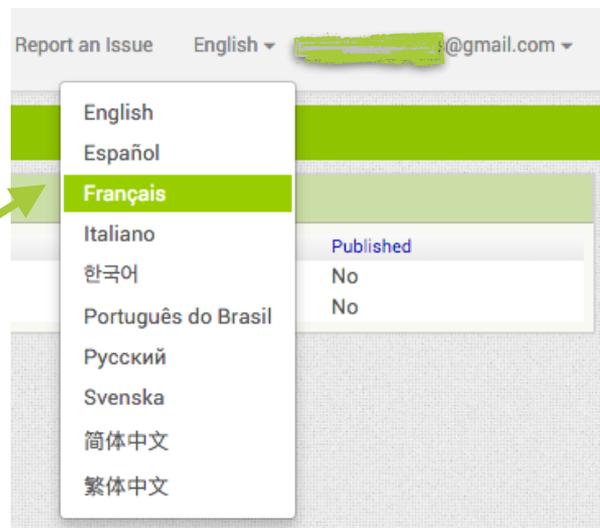
1

Rendez-vous sur le site suivant avec votre compte Google

<http://ai2.appinventor.mit.edu>

Sélectionnez la version Français, cela sera plus simple ...

2



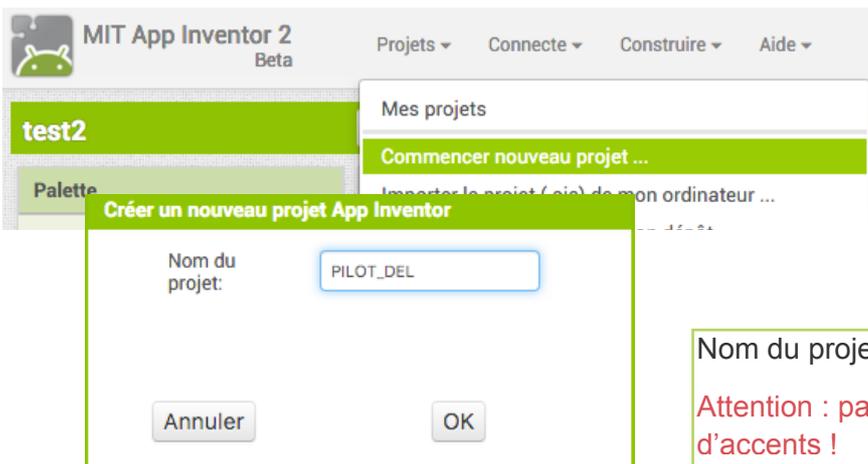
3

Dans le menu Projets > Commencer nouveau projet ...

4

Nom du projet par exemple : PILOT\_DEL

Attention : pas d'espace, pas de nom trop long et pas d'accents !





Notre 1ère application sera très simple !

Elle consiste en un seul bouton de faire apparaître une image d'une lampe allumée



Les outils disponibles

Les composants : les boutons, zones de textes, ... que contient notre application

Affichage pour dessiner l'application « Designer » ou pour programmer son fonctionnement « Blocs »

L'aperçu du design de l'application

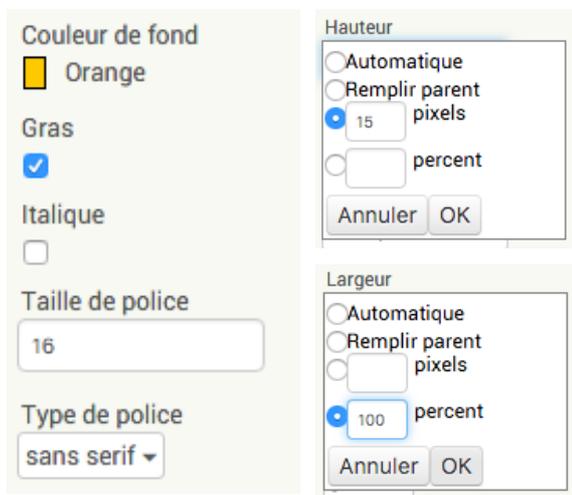
Pour charger des images par exemple

Les outils de mises en forme : Gras, centre, les polices ...



5

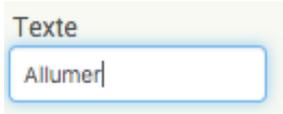
Par glisser/déposer, ajoutez un Label sur le haut de l'écran de la tablette.  
 Il se rajoute automatiquement dans la colonne Composants.  
 Et il est modifiable dans la colonne Propriétés



*Le bandeau du haut fera donc 15 pixel de haut et 100% de la largeur de l'écran*



6

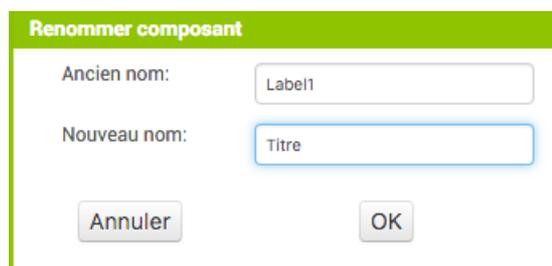


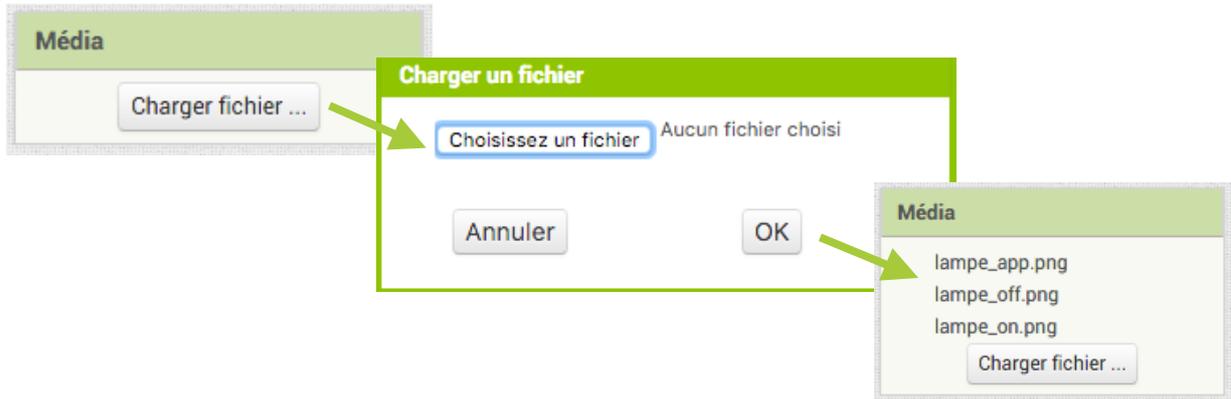
Modifiez les propriétés afin qu'il ressemble à l'exemple

7

Modifiez également le nom du label en « **Titre** » par exemple

Ce qui sera plus facile par la suite, plutôt que d'avoir 15 « labels\_N° » ?





8

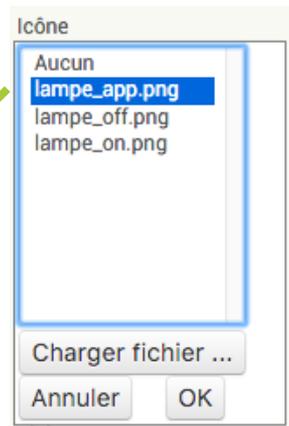
Chargez toutes les images dans la fenêtre « Média »

- L'image de la lampe éteinte : lampe\_off.png
- L'image de la lampe allumée : lampe\_on.png
- L'image du logo de l'application : lampe\_app.png



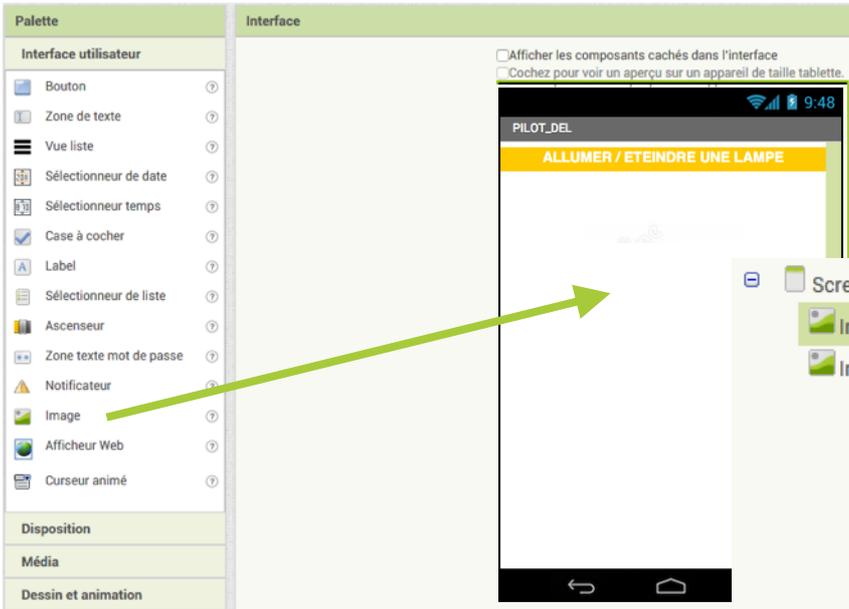
9

Avant de continuer l'application, nous allons améliorer rapidement quelques propriétés d'affichage : Icône de l'app, titre, affichage de l'écran ...

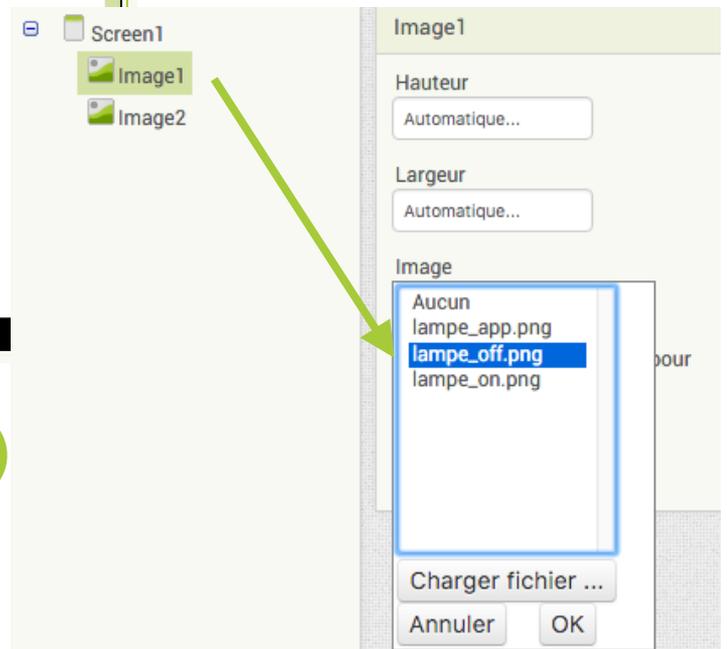


Toutes les propriétés sont dans le bandeau Propriétés de Screen1

Il est possible aussi d'ajouter une image « Charger fichier » qui correspondra à l'icône de l'application



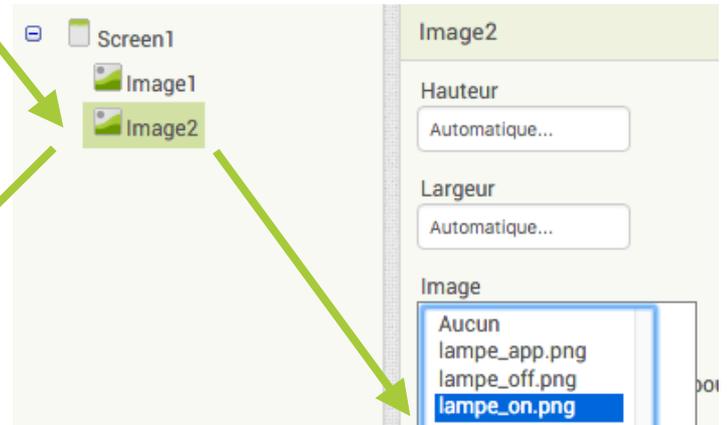
10  
Par glisser/déposer, ajoutez deux images



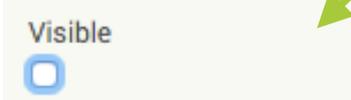
11  
Une image pour la lampe éteinte et une pour la lampe allumée. Indiquez le fichier image pour chacune d'elles



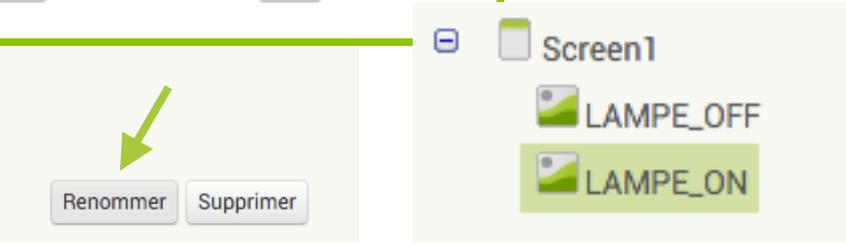
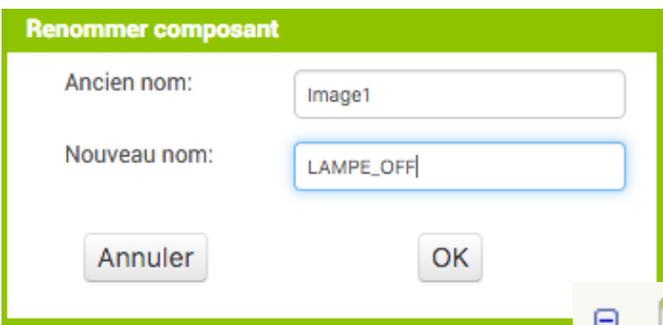
Une seule image sera affichée évidemment ...  
Il faut donc rendre non visible l'image2



12  
Décochez Visible pour l'image2



13  
Pensez à renommer les composants



**Palette**

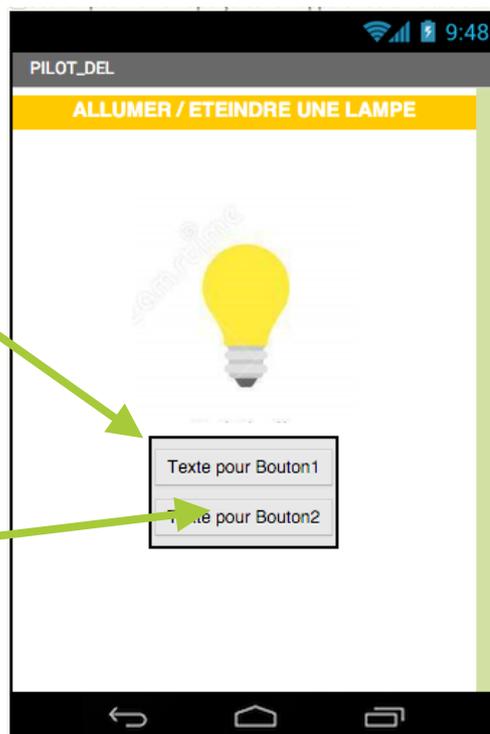
**Interface utilisateur**

**Disposition**

- Arrangement tableau
- Arrangement vertical
- Arrangement horizontal

**Interface utilisateur**

- Bouton



Par glisser/déposer ajoutez un arrangement vertical et 2 boutons

14

- Screen1
  - LAMPE\_OFF
  - LAMPE\_ON
  - Arrangement\_vertical1
    - BP\_ON
    - BP\_OFF

Renommez les boutons par BP\_ON et BP\_OFF

15

En utilisant les propriétés des boutons, obtenez l'écran ci-contre

16

**Largueur**

- Automatique
- Remplir parent
- 100 pixels
- percent

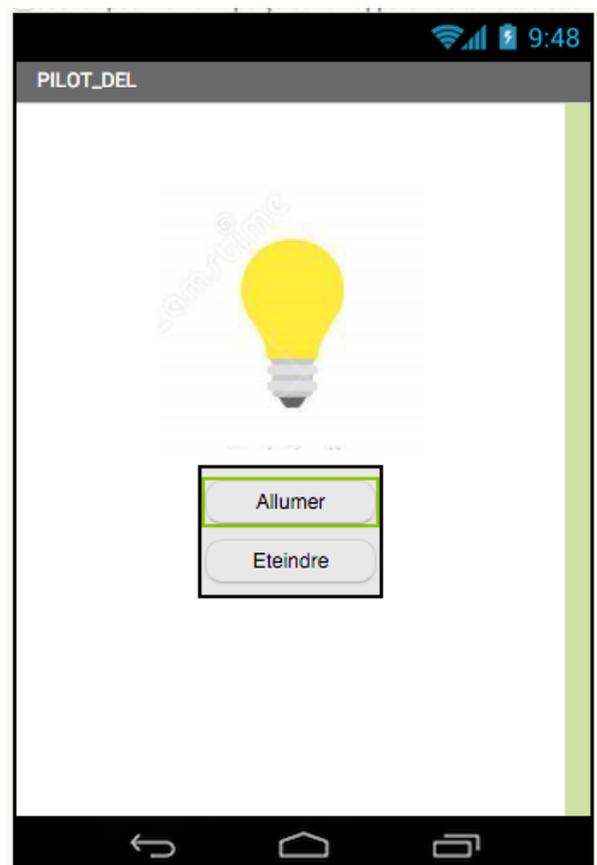
Annuler OK

**Texte**

Allumer|



Les boutons feront donc 100 pixels de long

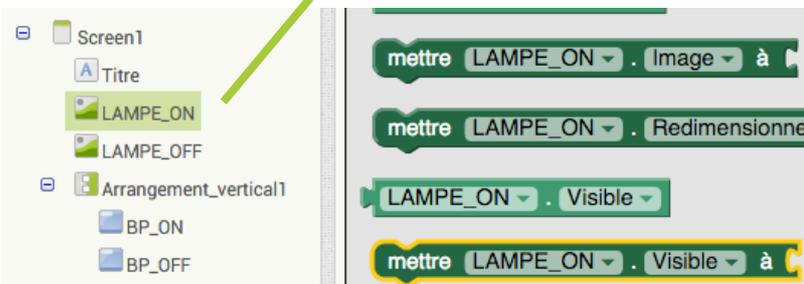
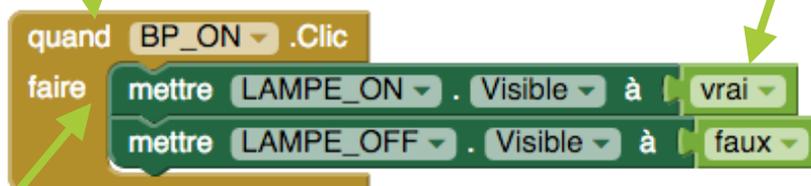
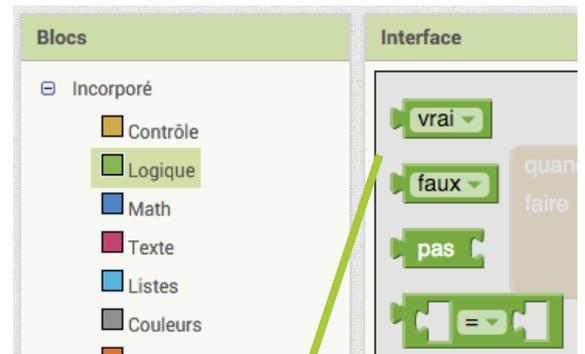
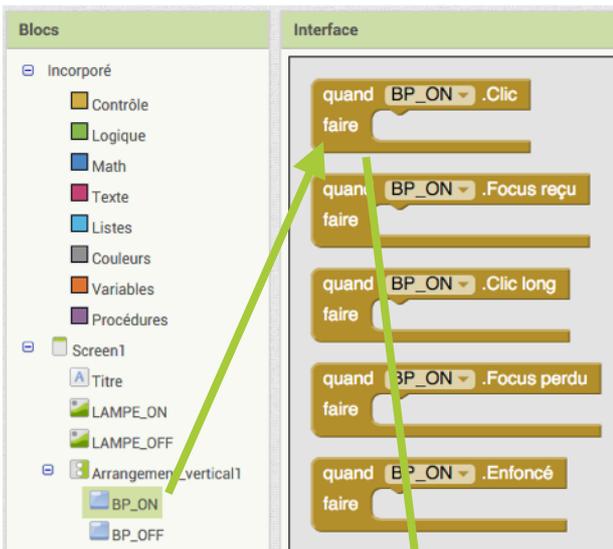
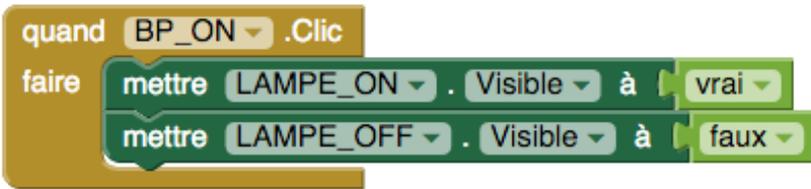


Passons maintenant sur la programmation des boutons et donc de l'affichage des images. Rendez-vous dans Blocs



Au niveau du Bouton BP\_ON :  
(si bouton cliqué)

Rendre visible l'image LAMPE\_ON  
et  
Ne pas afficher l'image LAMPE\_OFF



```

quand BP_ON .Clic
faire
mettre LAMPE_ON . Visible à vrai
mettre LAMPE_OFF . Visible à faux

```

```

quand BP_OFF .Clic
faire
mettre LAMPE_ON . Visible à faux
mettre LAMPE_OFF . Visible à vrai

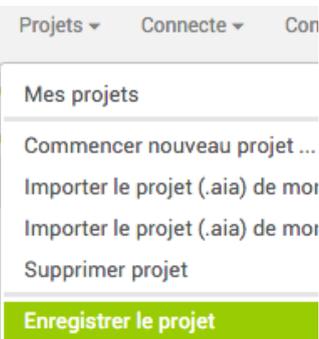
```

18

Le bouton BP\_On étant programmé, il faut aussi décrire le fonctionnement du bouton BP\_OFF comme ci-contre

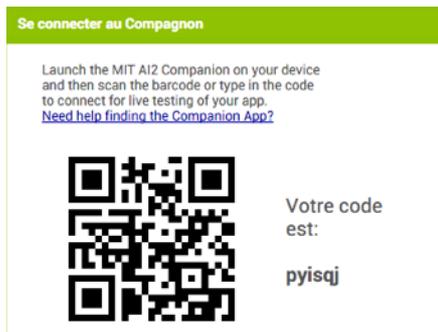
19

Il est peut être temps d'enregistrer le projet ...



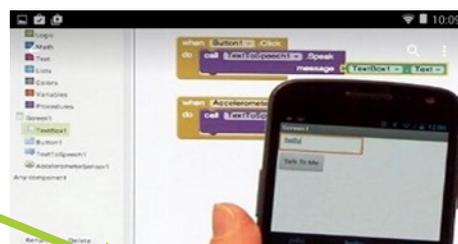
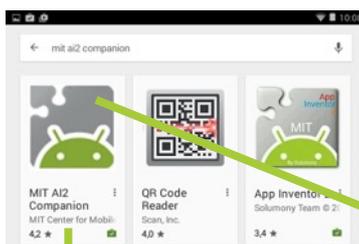
20

Notre application est terminée, il faut maintenant la tester sur notre tablette Android pour cela, menu Connecte > Compagnon AI  
Un QR Code est généré ainsi qu'un code



21

Sur votre tablette, évidemment vous venez d'installer l'application MIT AI2 Companion

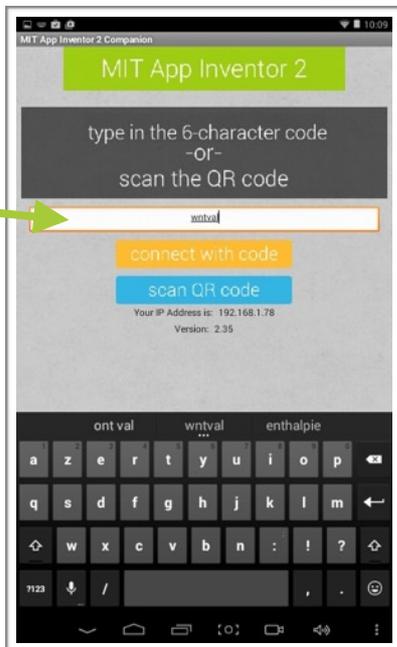


Se connecter au Compagnon

Launch the MIT AI2 Companion on your device and then scan the barcode or type in the code to connect for live testing of your app. [Need help finding the Companion App?](#)



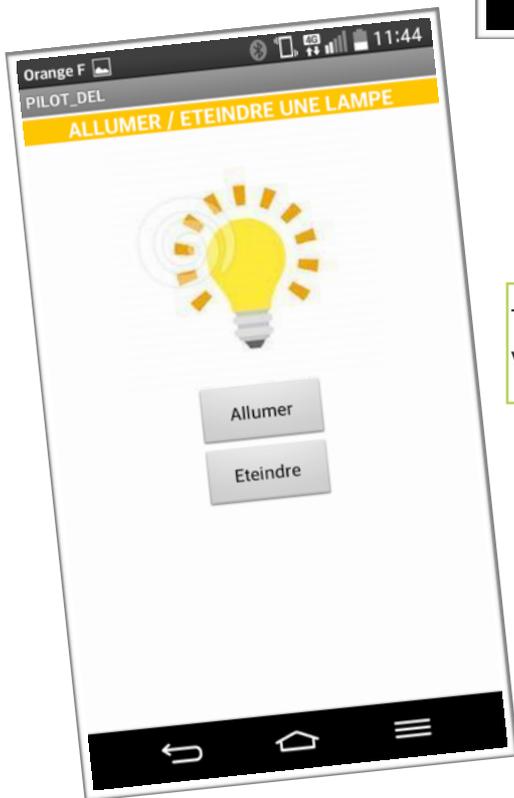
Votre code est: pyisqj




Une fois ce code indiqué, l'application est synchronisée avec le logiciel en ligne. Toutes modifications sur le logiciel sont donc immédiatement prises en compte sur la tablette

22

Une fois sur l'application, indiquez le code et **Connect with code**  
L'application se lance ensuite automatiquement



23

Terminé ! Vous pouvez tester votre application ...

Passons maintenant à l'installation complète de l'application

Construire ▾ Aide ▾ Mes Projets

App ( Donnez le code QR pour fichier .apk )

App ( enregistrer .apk sur mon ordinateur )

MaTereApplication Progress Bar

20%

Compiling part 1

Lien code à barre pour MaTereApplication



OK

Note: ce code à barre n'est valide que 2 heures. Regardez la FAQ pour voir des informations sur comment partager votre application avec les autres.

24

Pour générer l'application : Menu Construire > App (Donner le Code QR)



25

Depuis l'application Compagnon ou depuis n'importe quelle application de lecteur de QR Code, lancez la lecture du QR code précédemment généré depuis le logiciel

26

Terminez en suivant les indications d'installation ...

