

Tutoriel MIT App Inventor

I. Introduction

App Inventor est un outil de développement en ligne pour les téléphones et les tablettes qui fonctionnent avec le système d'exploitation Android. Il faut donc une liaison internet et un navigateur internet comme Mozilla Firefox ou Chrome. App Inventor est une application web open-source à l'origine fournie par Google, et maintenant maintenue par le Massachusetts Institute of Technology (MIT). Cette plate-forme de développement est utilisable par toutes et tous avec un compte Gmail ou sans compte si vous ne possédez pas de gmail .

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

Des connaissances en programmation orientée objet peuvent toutefois être nécessaires. Cependant, on pourra s'affranchir complètement de la connaissance des noms des propriétés et méthodes liées aux objets. En effet des blocs de propriétés, méthodes et événements seront directement proposés dès la création d'un objet sans que l'on s'en rende compte ce qui simplifie la programmation.



La programmation orientée objet ne signifie pas que l'on travaille avec des objets graphiques mais plutôt que l'on manipule des entités informatiques appelés "objets" qui ont des propriétés, une classe et peuvent elles-mêmes avoir des héritiers, qui possèdent des méthodes ...

L'utilisation de App Inventor rend assez facile la réalisation d'applications Android (en tout cas plus facile que l'utilisation du programme "Android Studio" qui nécessite de connaître la programmation en Java) mais les possibilités sont plus limitées.

Première application : du texte

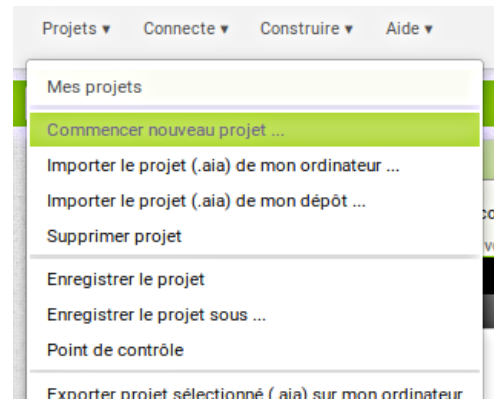
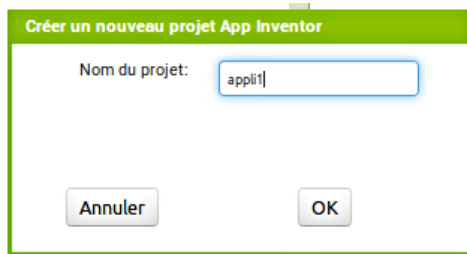
Nous allons réaliser l'application suivante :

Sur l'écran s'affiche un bouton et une zone de Texte à remplir. Quand on appuie sur le bouton, un message apparaît sous le bouton (bonjour) et le nom rentré dans la zone de texte s'affiche sous "bonjour"

Étape 1 : création de l'application :

Allez dans le menu "Projets" puis choisir "Commencer nouveau projet ..."

Choisir un nom de projet puis cliquez sur OK :

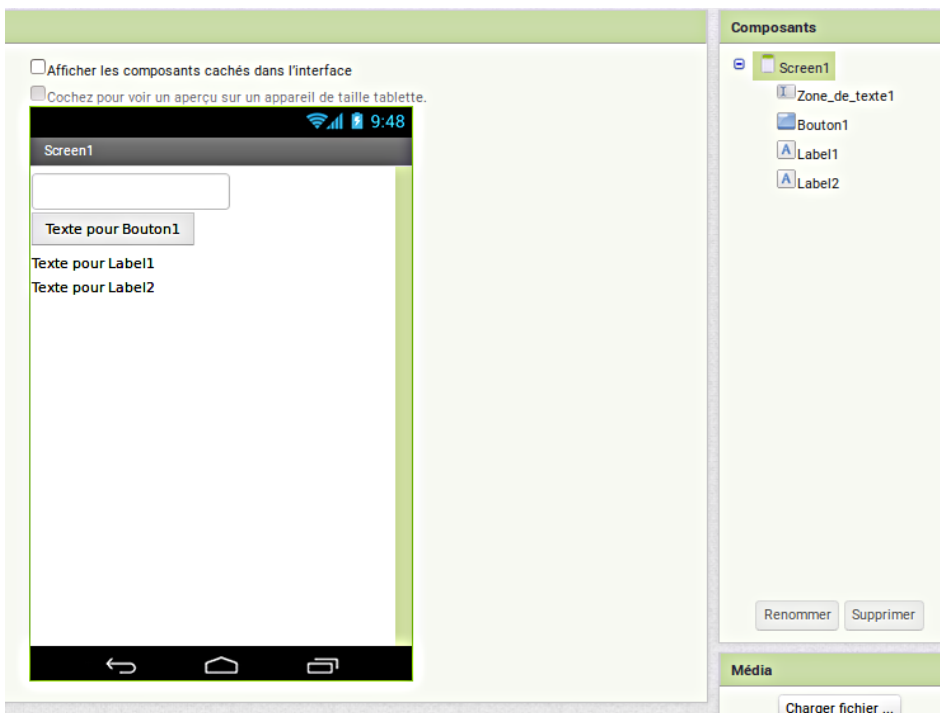


Étape 2: placement des éléments sur l'écran:

Nous avons 4 éléments à placer sur notre écran (interface homme/machine):

- une zone de texte à remplir ('zone de texte')
- un bouton
- 2 zones de texte où écrire ('Label')

Placez les éléments pour obtenir l'écran ci-dessous. Pour placer un élément, cliquez dessus dans le menu de gauche et le déplacer, en gardant le clic enfoncé, à l'endroit où vous voulez le poser (dans le "Screen1" qui est le nom de votre fenêtre d'application)

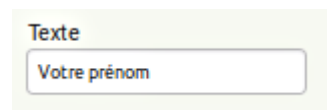


Vous remarquerez que les éléments apparaissent dans 2 fenêtres, dans votre "Screen1" bien évidemment mais aussi dans la fenêtre "composants".

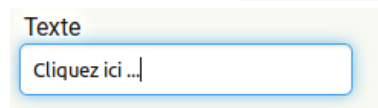
Étape 3: paramétrage des éléments

Le paramétrage se passe dans la fenêtre de droite "propriétés"

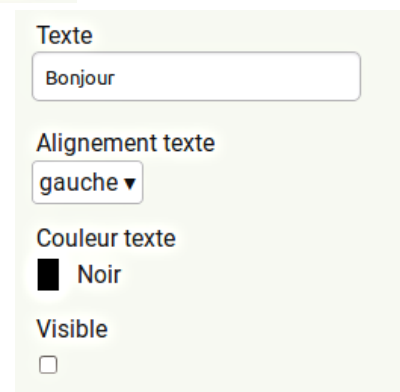
Pour la zone de texte, on va lui mettre un texte par défaut :



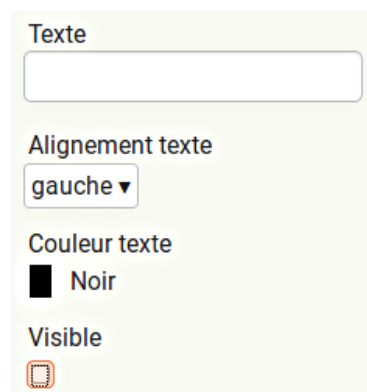
Pour le bouton, on va lui mettre un texte :



Pour le 1er texte (Label1), on va écrire 'bonjour' et le rendre invisible :



Pour le 2ème texte (Label2), on ne va rien écrire et le rendre invisible aussi:



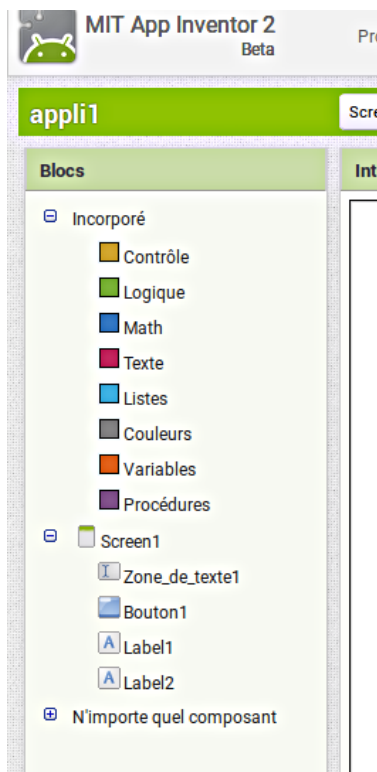
Vous remarquerez que les 2 éléments de texte (label) ont disparu de votre Screen1 mais ils sont toujours présents dans la fenêtre "Composants"



Il faut penser à faire régulièrement des sauvegardes (menu "Projets" puis "enregistrer le projet")

Étape 4: écriture du programme

Comme annoncé en introduction, nous n'allons pas écrire de lignes de codes mais utiliser un mode de programmation graphique. Pour y accéder il faut cliquer sur "Blocs" (en haut à droite).



Les éléments de notre "Screen1" apparaissent bien (en bas à gauche).

Au dessus, ce sont les blocs disponibles classés en grandes fonctions :

Contrôle : les éléments de test, de boucles,

Logique : éléments liés à de la logique, "vrai", "faux", ...

Math : des outils mathématiques (addition, ...)

Text : pour gérer les textes

....



Avant de faire quoi que ce soit, il faut écrire le programme, sous la forme que vous voulez (algorithmes, algorigrammes, ...). Une fois le programme écrit ou pourra le traduire en blocs.

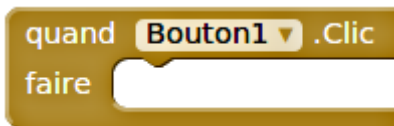
Ici l'algorithme est :

Quand le Bouton1 est cliqué, on fait :

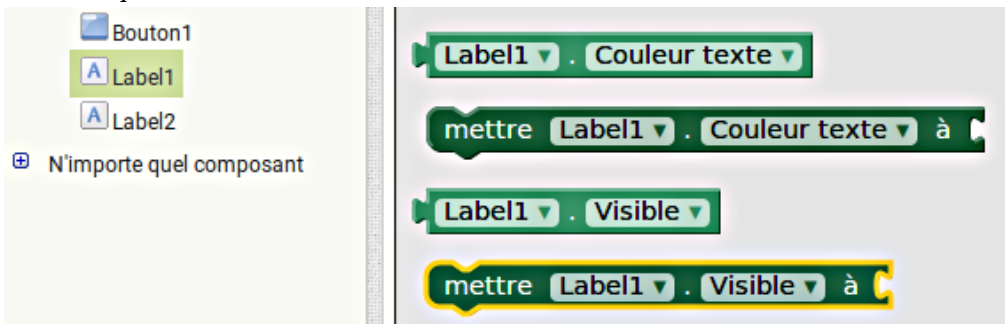
- on rend le label1 visible
- on rend le label2 visible
- on affiche dans le label2 le texte de la zone de texte1

Quand on clique sur le Bouton1, une fenêtre avec toutes les possibilités associées à cette élément apparaissent :

On va choisir :



Si on clique sur "Label1" :



Voilà donc le bloc qui signifie "quand le Bouton1 est cliqué on rend le label1 visible"



A vous de compléter la programmation de façon à obtenir :



Étape 5: essai du programme (de l'application)

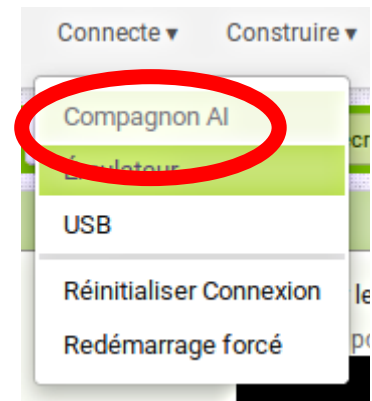
Si l'on possède un téléphone ou une tablette sous Android on peut essayer l'application en réel. Demander à l'enseignant une tablette ou partager un smartphone Android avec un camarade.

Il faut télécharger l'application MIT AI2 Companion sur le Store

Si vous n'avez pas d'autres options, vous pouvez utiliser le simulateur (là il faut être patient, très patient !). **à éviter**

Cliquer dans le menu « Connecte » puis Compagnon AI

Scanner le QR code depuis l'application MIT AI2 Companion



Vérifier que votre application fonctionne et faire valider par le professeur.

III. Deuxième application : des images, du son et du temps

Version N°1 :

Cette deuxième application va afficher une image/bouton. Quand on cliquera dessus un son sera émis et un texte s'affichera. Vous disposez de 2 fichiers dans l'espace de travail (gong.mp3 et photo_StE.jpg)

1. Avant de faire le paramétrage de l'interface et la programmation des blocs, écrire l'algorithme du programme précédant (cela doit décrire de manière intelligible ce que réalise le programme.

Faire valider par le professeur

Description de l'interface :

Vous allez donc insérer un bouton, du son (menu "média" puis "son") et un texte (Label)

Pour le paramétrage (propriétés) :

Bouton1 (Bouton)

▼ Appearance

Couleur de fond [?]
 Par défaut

Gras [?]

Italique [?]

Taille de police [?]
30

Type de police [?]
Par défaut...

Hauteur [?]
Automatique...

Largeur [?]
100 percent...

Image [?]
logo-st-erembert.png...

Forme [?]
par défaut ▾

Montrer réaction [?]

Texte [?]
Cliquez...

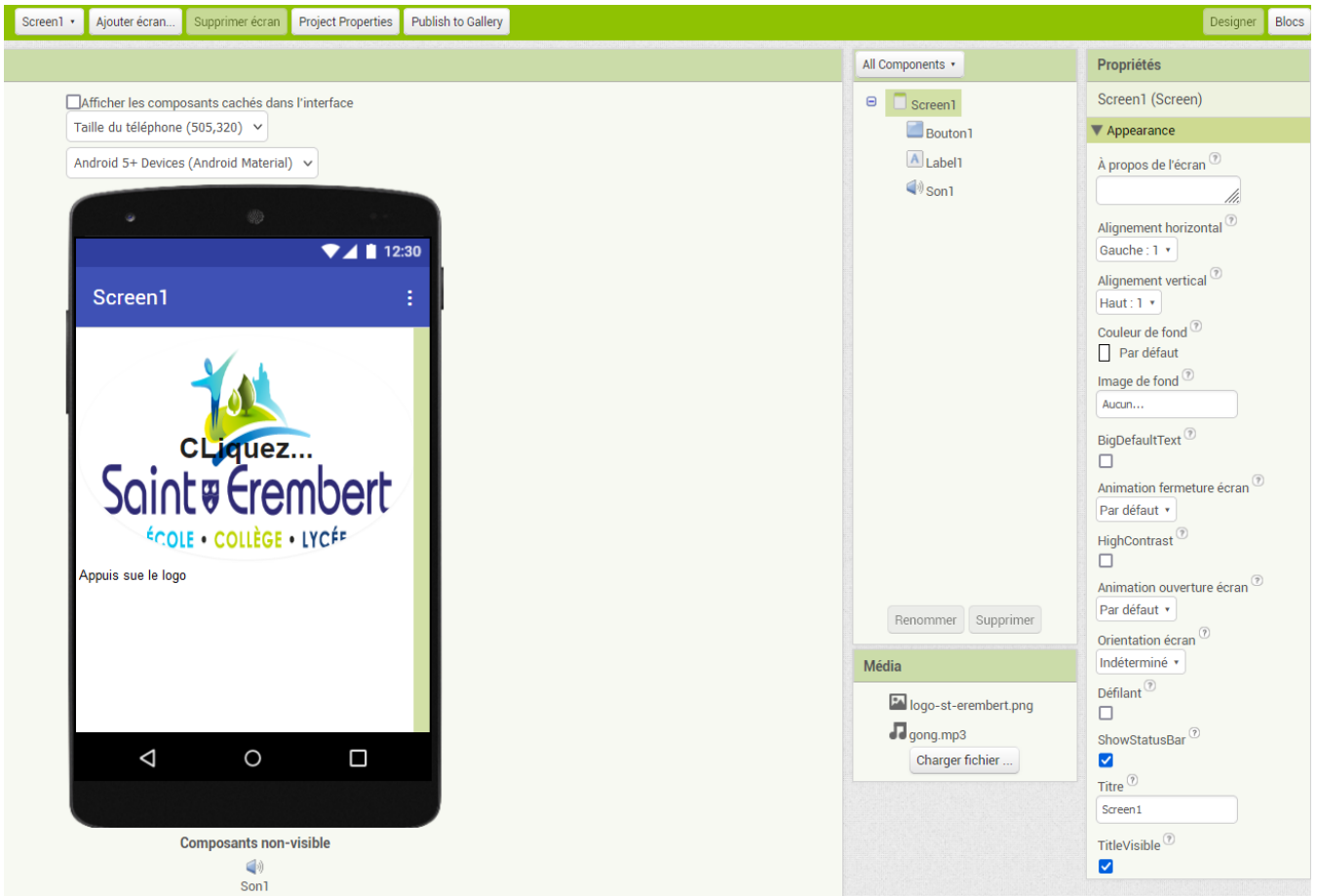
Alignement texte [?]
centre : 1 ▾

Couleur texte [?]
 Par défaut

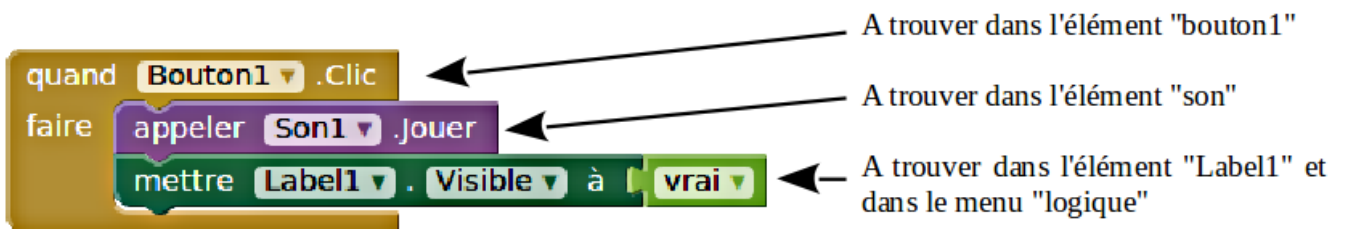
Visible [?]

Pour le Label vous y mettez le texte : "Appuis sur le logo" et le rendez invisible.

Pour le paramétrage du son vous allez ajouter le fichier son ("Propriétés" puis "Source"):



Maintenant le programme à réaliser :



2. Faites le programme et vérifiez que votre application fonctionne (copies d'écran à fournir)

Version N°2 : amélioration et rajout d'une temporisation

On veut que le texte disparaisse au bout de 4s.

3. Avant de faire la programmation des blocs, écrire l'algorithme du programme précédant (cela doit décrire de manière intelligible ce que réalise le programme)

Nous rajoutons maintenant l'élément "Horloge" (dans le menu Capteur) avec comme paramétrage : 4000 (le temps est en millisecondes = 4000 ms = 4s)

Propriétés
Horloge1
TimerAlwaysFires <input checked="" type="checkbox"/>
ChronomètreActivé <input type="checkbox"/>
IntervalleChronomètre 4000

Le programme sera le suivant :

```
quand Bouton1 .Clic
faire
  appeler Son1 .Jouer
  mettre Label1 . Visible à vrai
  mettre Horloge1 . ChronomètreActivé à vrai
```

```
quand Horloge1 .Chronomètre
faire
  mettre Label1 . Visible à faux
```

4. Faites le programme et testez-le (copies d'écran à fournir). Quand tout fonctionne faites valider par le professeur.

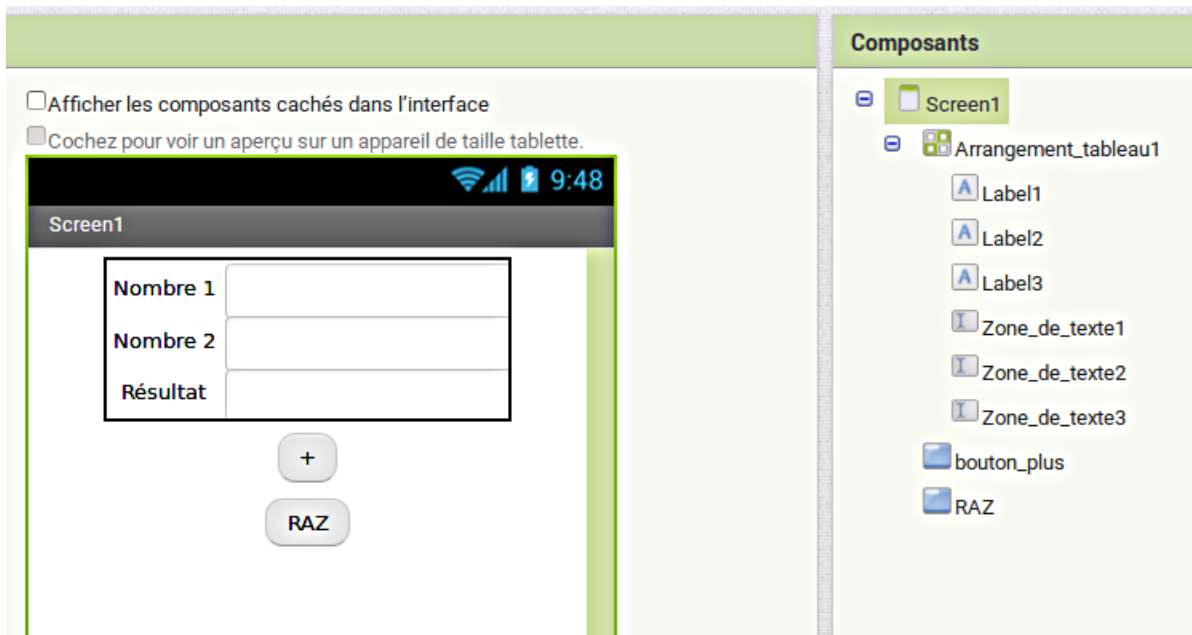
IV. Utilisation de fonctions mathématiques

Nous allons maintenant réaliser une application qui fait l'addition de deux nombres.

1. Avant de faire le paramétrage de l'interface et la programmation des blocs, écrire l'algorithme du programme précédent (cela doit décrire de manière intelligible ce que réalise le programme).

Description de l'interface :

Il y aura 3 textes, 3 zones de textes et 2 boutons :



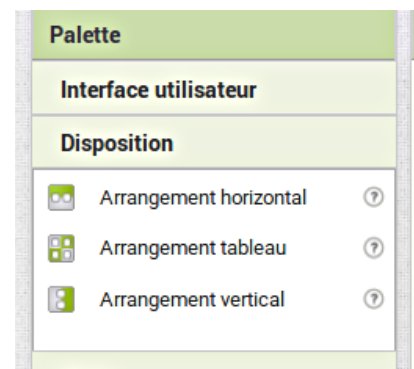
Pour placer nos différents éléments correctement les uns par rapport aux autres nous allons les mettre dans un tableau :

Label1	Zone de texte 1
Label2	Zone de texte 2
Label3	Zone de texte 3

Le tableau se trouve dans le menu de gauche "Disposition" et s'appelle "arrangement tableau"

On choisira donc 2 colonnes et 3 lignes (dans le menu "propriétés")

Pour les zones de texte il faudra préciser "nombres uniquement" et pour la 3ème zone de texte ne pas l'activer (en effet elle ne doit pas pouvoir être modifiée manuellement).





Il peut être très utile de renommer les éléments utilisés afin de rendre leur utilisation plus facile dans la partie programmation.

Ainsi vous renommerez les 2 boutons par "bouton_plus" et "RAZ" (ce qui est quand même plus intelligible que "Bouton1" ou "Bouton2"!).

Le programme sera le suivant :

```
quand Screen1 .Initialise
faire
mettre Zone_de_texte1 . Texte à " "
mettre Zone_de_texte2 . Texte à " "
mettre Zone_de_texte3 . Texte à " "

quand bouton_plus .Clic
faire
mettre Zone_de_texte3 . Texte à [ Zone_de_texte1 . Texte + Zone_de_texte2 . Texte ]

quand RAZ .Clic
faire
mettre Zone_de_texte1 . Texte à " "
mettre Zone_de_texte2 . Texte à " "
mettre Zone_de_texte3 . Texte à " "
```

A prendre dans le menu de droite qui s'appelle "Math"



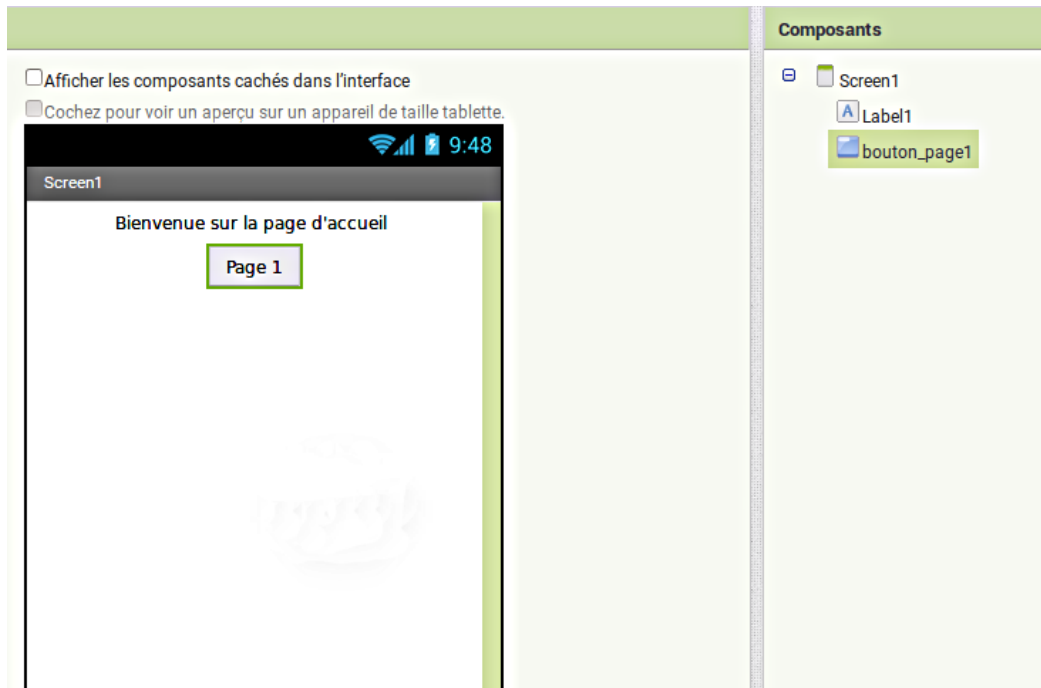
2. Faites le programme et testez-le (copies d'écran à fournir). Quand tout fonctionne faites valider par le professeur.

V. Gestion de plusieurs écrans (pages)

Nous allons maintenant découvrir comment avoir plusieurs écrans (pages) dans notre applications et comment passer de l'une à l'autre

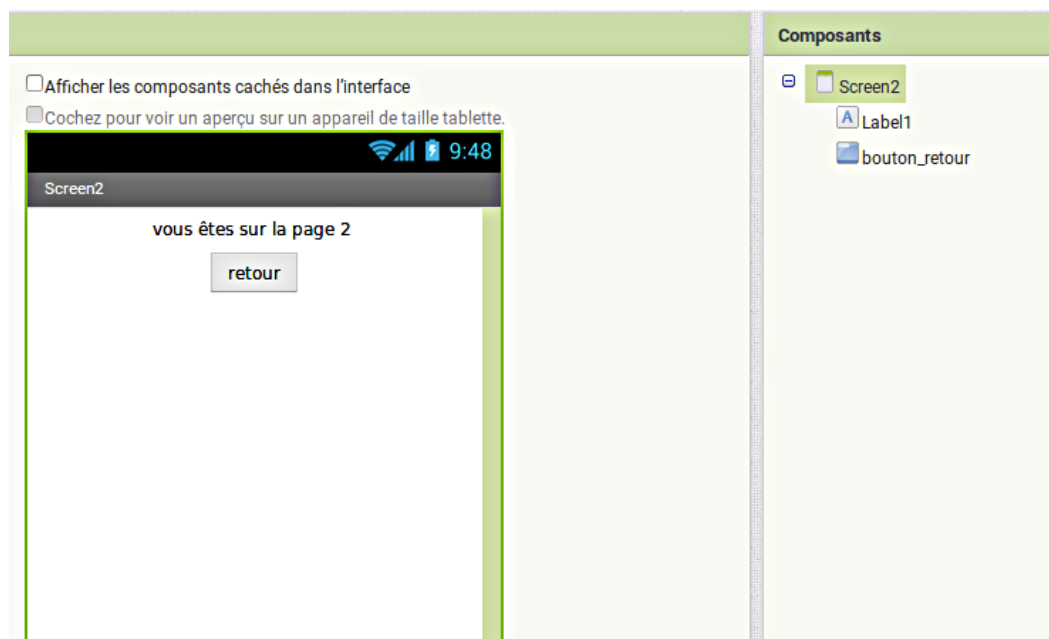
Créez une nouvelle application et paramétrez le "Screen1" de la façon suivante :

un texte et
un bouton:

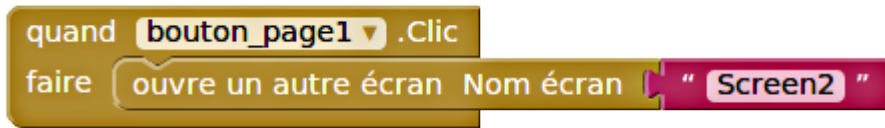


Maintenant vous allez créer un nouvel écran en cliquant sur le bouton "Ajouter écran ..." et le paramétrer de la façon suivante :

un texte et
un bouton :



Comment faire la programmation du bouton ? Voici un exemple pour le "Screen1" :



Faites la programmation de l'interface et du programme et testez votre application. Quand tout fonctionne faites valider par le professeur.

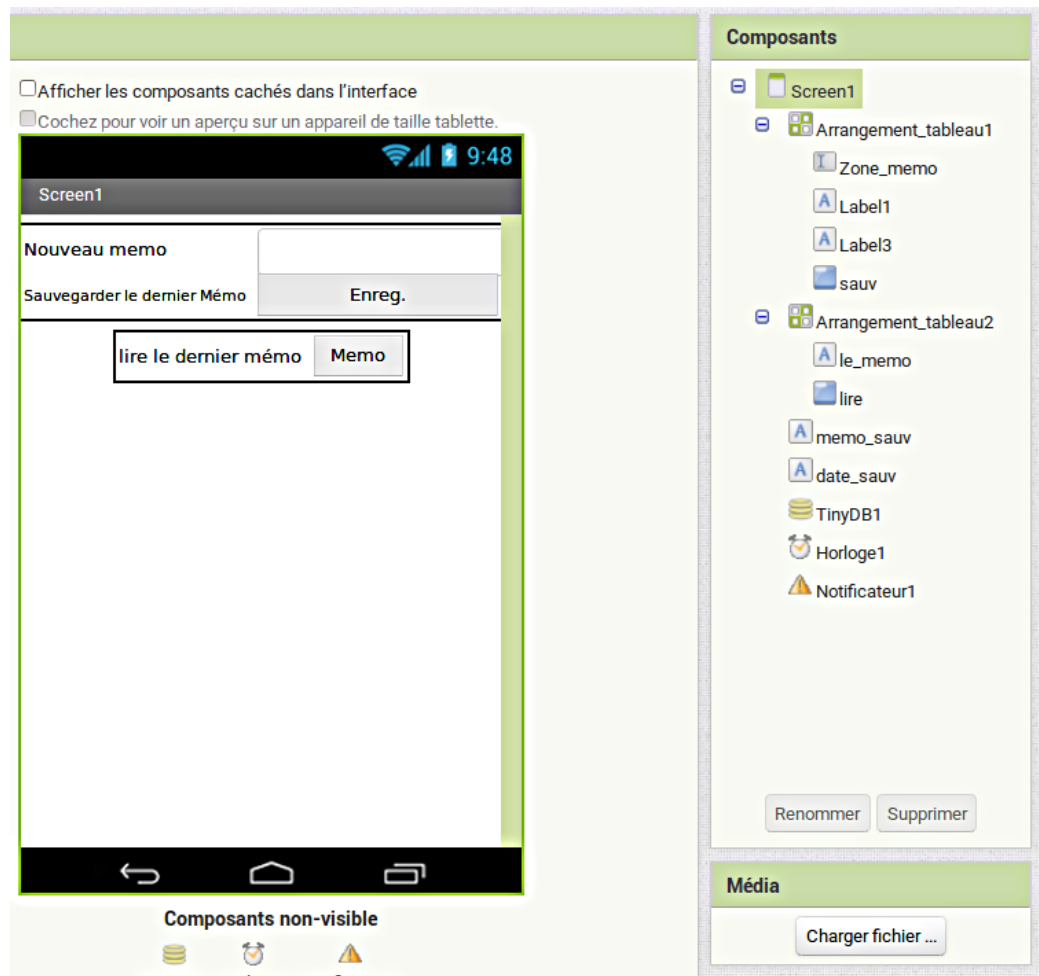
VI. Sauvegarde de données

Vous allez maintenant créer une nouvelle application dont le but est de mémoriser quelque chose que nous appellerons un "mémo" (petite phrase) dans une base de données située dans le téléphone.

L'interface doit ressembler à :

et doit contenir :

- 5 labels
- une zone de texte
- 2 boutons
- une base de données (TinyDB)
- une horloge
- un notificateur

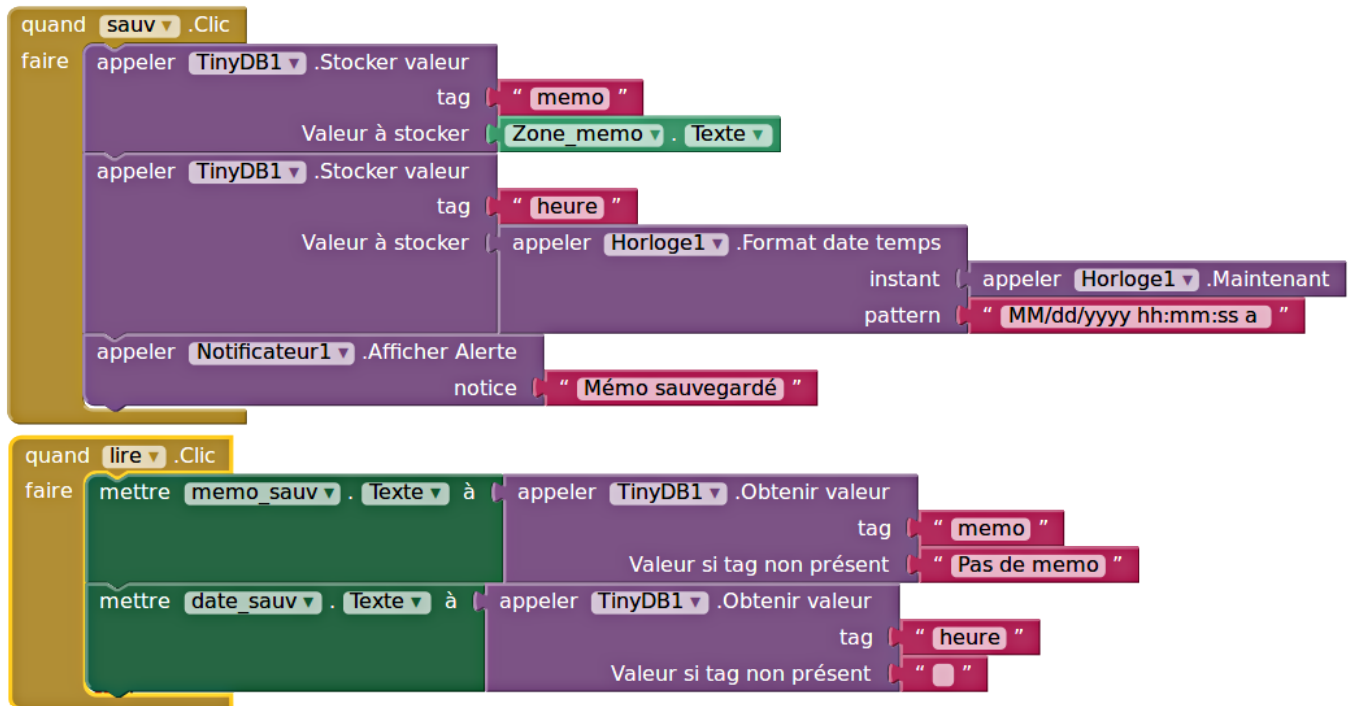


Le fonctionnement :

Quand on rentre une phrase à mémoriser et que l'on appuie sur le bouton "Enreg.", cette phrase (mémo) est sauvegardée dans la base de données (avec la date et l'heure de l'enregistrement) et un message signalant l'enregistrement s'affiche (notification) sur fond rouge.

Si on appuie sur le bouton "Memo", le dernier mémo s'affiche ainsi que le jour et la date d'enregistrement

Le programme sera le suivant :



1. Avant de faire le paramétrage de l'interface et la programmation des blocs, écrire l'algorithme du programme précédant (cela doit décrire de manière intelligible ce que réalise le programme).

2. Faites le programme et testez-le (copies d'écran à fournir). Quand tout fonctionne faites valider par le professeur.

VII. Les variables et les boutons "case à cocher"

Version 1 :

Nous allons maintenant créer une nouvelle application dont le but est de mémoriser le choix d'une langue et, suivant ce choix, d'afficher des pages différentes.



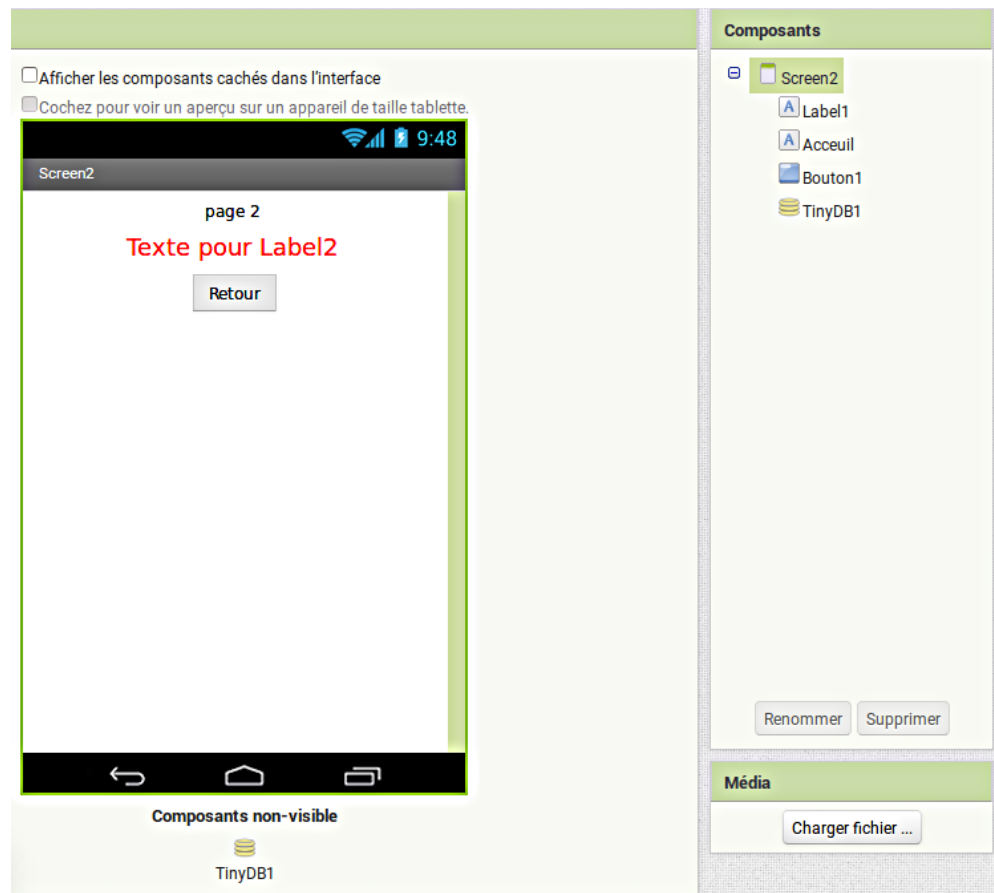
Nous allons utiliser les **variables globales** : "global" signifie qu'elles existent et sont disponibles sur toute la page (par contre elles ne sont pas globales à plusieurs pages) à l'inverse des variables locales qui ne fonctionnent qu'à l'intérieur d'un bloc.

L'interface homme/machine pour la page 1 (Screen1) sera la suivante :

- 2 cases à cocher
- 2 textes (labels)
- un bouton
- un notificateur (pour la version 2)
- une base de données

L'interface homme/machine pour la page 2 (Screen2) sera la suivante :

- 2 textes (labels)
- un bouton
- une base de données



Le programme pour la page 1 (Screen1) sera :

```

initialise global choix_fait à faux
quand Screen1 .Initialise
faire mettre Label2 . Texte à " Choisir la langue "

```

```
quand Case_anglais .Changé
faire
  si Case_anglais . Vérifié
  alors
    si Case_français . Vérifié
    alors
      mettre Label2 . Texte à " Anglais et français "
      mettre global choix_fait à vrai
    sinon
      mettre Label2 . Texte à " Anglais "
      mettre global choix_fait à vrai
  sinon
    si Case_français . Vérifié
    alors
      mettre Label2 . Texte à " Français "
      mettre global choix_fait à vrai
    sinon
      mettre Label2 . Texte à " Choisir la langue "
      mettre global choix_fait à faux
```

```
quand Case_français .Changé
faire
  si Case_anglais . Vérifié
  alors
    si Case_français . Vérifié
    alors
      mettre Label2 . Texte à " Anglais et français "
      mettre global choix_fait à vrai
    sinon
      mettre Label2 . Texte à " Anglais "
      mettre global choix_fait à vrai
  sinon
    si Case_français . Vérifié
    alors
      mettre Label2 . Texte à " Français "
      mettre global choix_fait à vrai
    sinon
      mettre Label2 . Texte à " Choisir la langue "
      mettre global choix_fait à faux
```

```

quand validation .Clic
faire
  si obtenir global choix_fait
  alors
    appeler TinyDB1 .Stocker valeur
      tag " langue "
      Valeur à stocker Label2 . Texte
    ouvre un autre écran Nom écran " Screen2 "
  mettre Label2 . Texte à " Il faut choisir la langue "

```

Le programme pour la page 2 (Screen2) sera :

```

initialise global choix_langue à " "

quand Screen2 .Initialise
faire
  mettre global choix_langue à appeler TinyDB1 .Obtenir valeur
    tag " langue "
    Valeur si tag non présent " pas de langue "
  si obtenir global choix_langue = " Anglais "
  alors mettre Accueil . Texte à " Welcome "
  si obtenir global choix_langue = " Français "
  alors mettre Accueil . Texte à " bienvenue "
  si obtenir global choix_langue = " Anglais et français "
  alors mettre Accueil . Texte à " Welcome | Bienvenue "

quand Bouton1 .Clic
faire
  ouvre un autre écran Nom écran " Screen1 "

```



```

Case_anglais . Vérifié

```

Ce bloc "logique" renvoie "vrai" si la case a été validée et "faux" s'il elle n'est pas validée

1. Avant de faire le paramétrage de l'interface et la programmation des blocs, écrire l'algorithme des programmes précédents (cela doit décrire de manière intelligible ce que réalise le programme).
2. En se référant à ce que vous avez vu dans d'autres langages (php, javascript, arduino, python) , donner les types des 2 variables des programmes (Screen1 et Screen2).
3. Faites le programme et testez-le (copies d'écran à fournir). Quand tout fonctionne faites valider par le professeur.

Version 2 :

Amélioration à faire:

- sur la page 1 (Screen1) quand on clique et qu'aucun choix n'a été fait, rajouter un message éphémère (notification), en blanc sur fond rouge, pour dire "il faut choisir une langue"

4. Faites le programme et testez-le (copies d'écran à fournir). Quand tout fonctionne faites valider par le professeur.