

STI2D	AP4.1 : découverte de la programmation d'une application Android à l'aide du logiciel "App Inventor"	
Option SIN Terminale		

Informatique

Durée prévue : 6h.

Problématique : réalisation d'une application Android (smartphone, tablette, ...)

Compétences visées	Savoirs associés
Être capable de comprendre et de modifier une application Android avec App Inventor	

Plan de l'étude	Remarque
I. Introduction	
II. Première application : du texte	
III. Deuxième application : des images, du son et du temps	
IV. Utilisation de fonctions mathématiques	
V. Gestion de plusieurs écrans (pages)	
VI. Sauvegarde de données	
VII. Les variables et les boutons "case à cocher"	
VIII. En bonus	

Logiciels	Matériels
App Inventor	Ordinateur

Mode de distribution	Format numérique
----------------------	------------------



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 à condition de posséder un compte Gmail.

La programmation est réalisée sans taper une seule ligne de code, mais simplement en associant et en paramétrant des briques logicielles toutes faites (langage 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 objets 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 Google "Android Studio" qui nécessite de connaître la programmation en Java) mais les possibilités sont plus limitées.

Vérifions que tout fonctionne avant de nous lancer dans une application :

Pour lancer le logiciel il faut, dans l'ordre :

- lancer "aiStarter" :



cela doit ouvrir la fenêtre suivante :

```
aiStarter
Platform = Windows
AppInventor tools located here: "C:\Program Files (x86)"
Bottle server starting up (using WSGIRefServer())...
Listening on http://127.0.0.1:8004/
Hit Ctrl-C to quit.
```

Ne plus toucher à cette fenêtre (la laisser ouverte)

- dans votre navigateur internet aller sur le site du logiciel : <http://ai2.appinventor.mit.edu/>

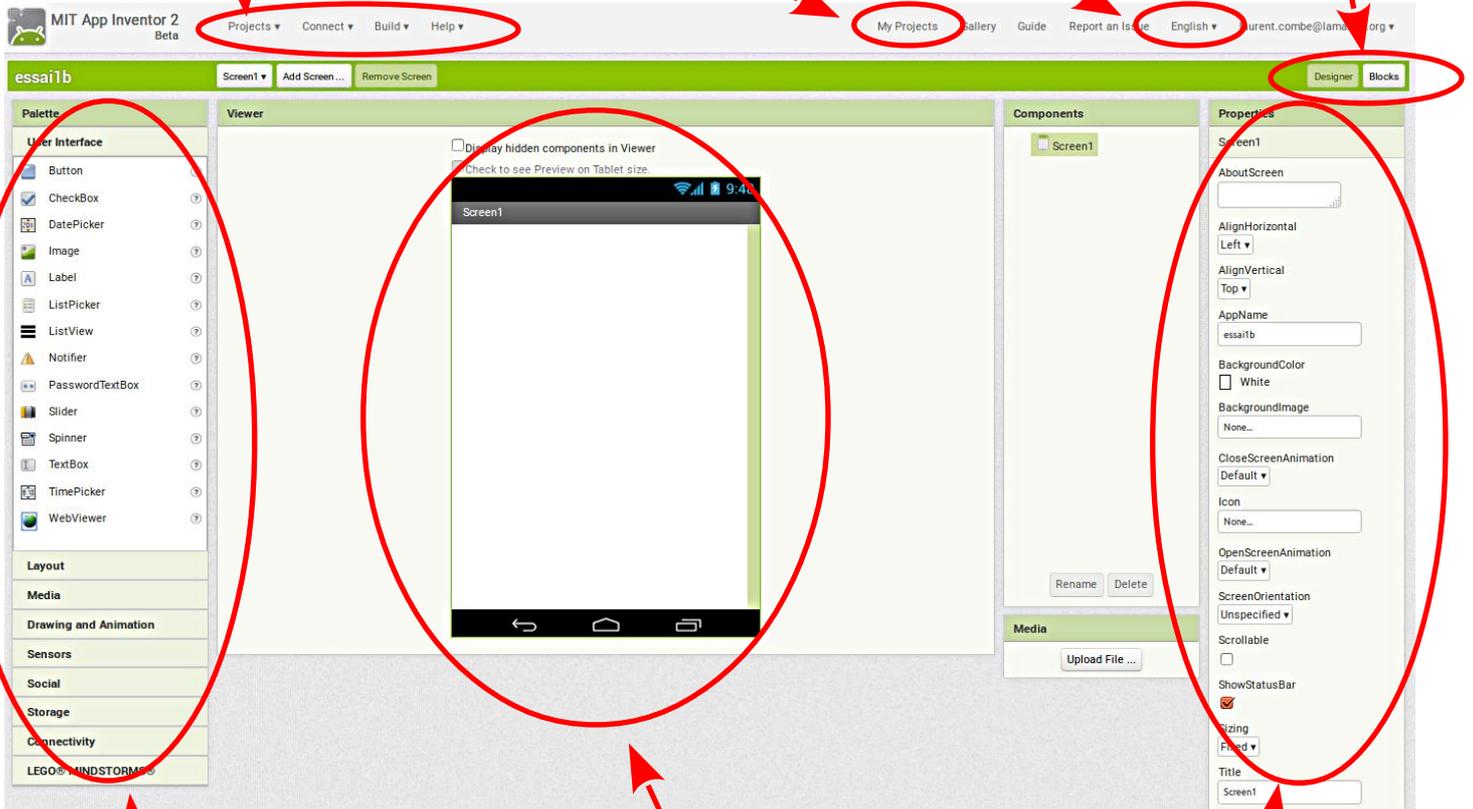
Celui-ci s'ouvre avec une fenêtre comme celle-ci :

Menu principal
(enregistrer, simuler, ...)

Menu pour choisir parmi
ses projets (ils sont
sauvegardés sur les
serveurs du MIT)

Choix de la langue

Menu pour choisir le
mode "designer" (pose
des éléments de
l'interface) et le mode
"blocks" (là où l'on fait le
programme)



Fenêtre contenant les éléments
que l'on peut choisir (bouton,
texte, ...)

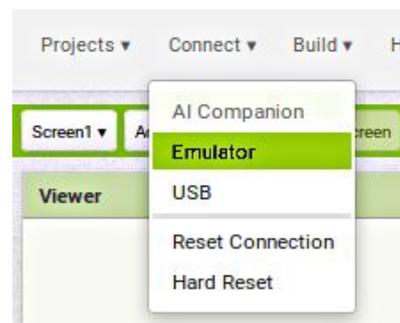
Fenêtre principale représentant
l'écran du téléphone. C'est sur cet
écran que l'on va placer les
éléments qui nous intéressent.

Fenêtre de paramétrage de
l'élément sélectionné

Essayons maintenant la simulation :

Cliquez sur le menu "Connect" puis choisir "Emulator"

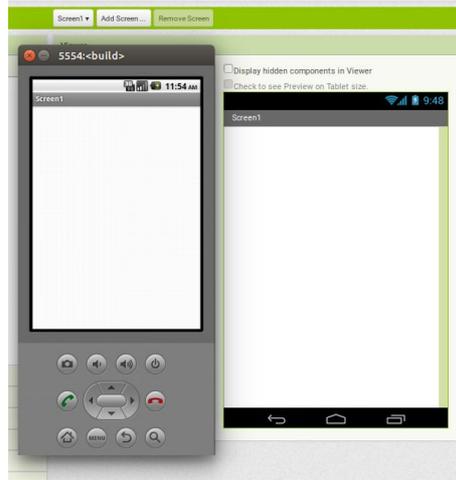
Une simulation de téléphone sous Android doit apparaître.
Attention cela prend pas mal de temps car le programme va
compiler le code sur un serveur du MIT et le renvoyer à votre



ordinateur. On doit obtenir le résultat suivant :

L'écran du téléphone affiche un page vide (normal on n'a pas encore fait de programme!).

Si cela ne fonctionne pas demandez de l'aide au professeur



II. Première application : du texte

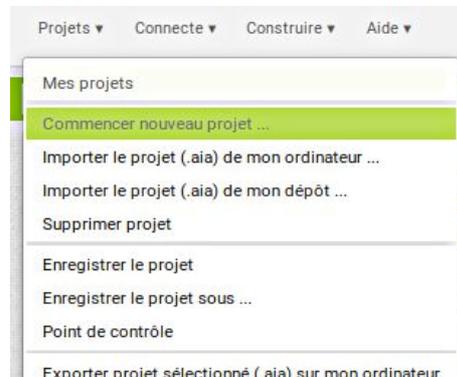
Nous allons réaliser l'application suivante :

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 :

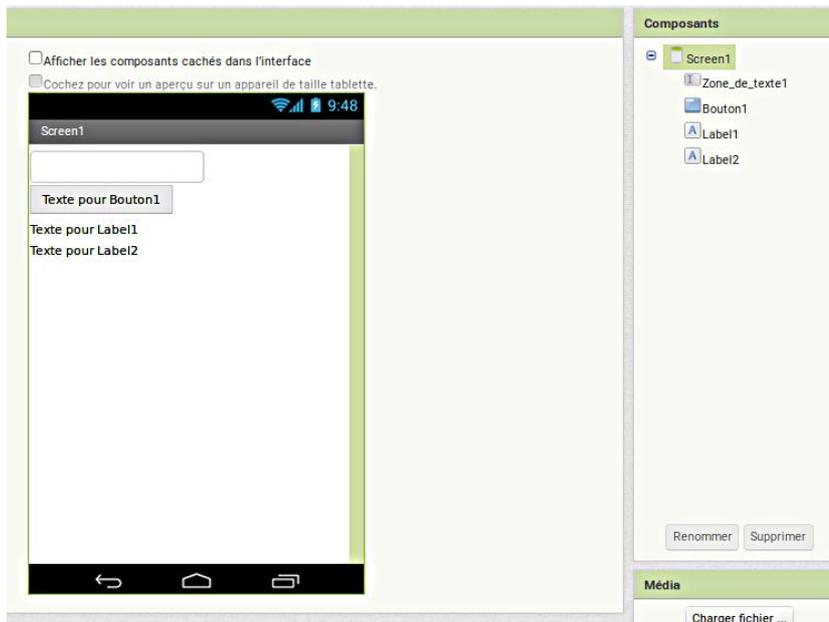


Étape 2: placement des éléments sur l'écran:

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".

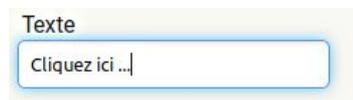
Étape 3: paramétrage des éléments

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

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



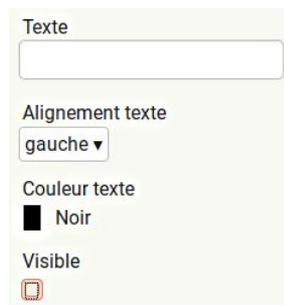
Pour le bouton, on va lui mettre un texte :



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



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



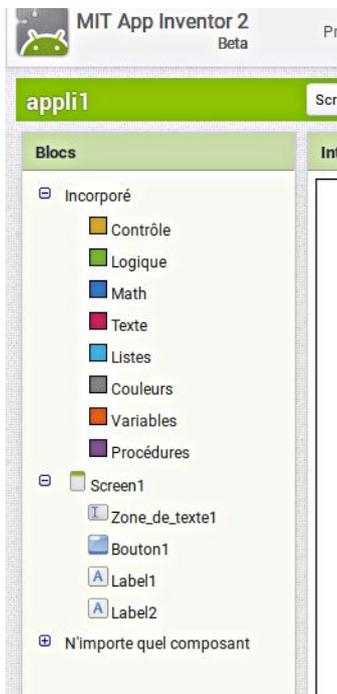
Vous remarquerez que les 2 éléments de texte (label) ont disparu de votre Screen1 mais ils sont toujours présents dans la fenêtre "Composants"



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 :

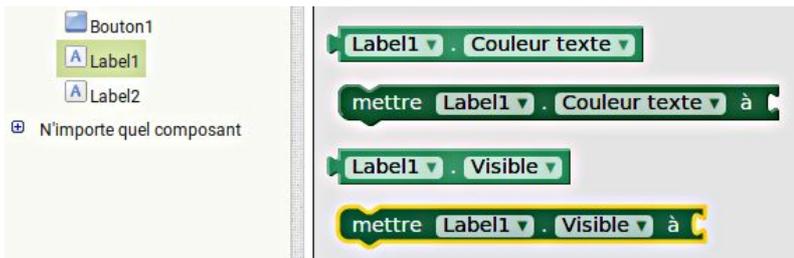
- on rend le label1 visible
- on rend le label2 visible
- on affiche dans le label2 le texte de la zone de texte1

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"



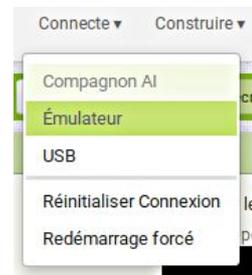
A vous de compléter la programmation de façon à obtenir :



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 (et ça fonctionne bien mieux que la simulation)

Sinon on utilise la simulation (là il faut être patient!).



Vérifier que votre application fonctionne (copie d'écran à fournir)



Si vous voulez enregistrer votre application sur votre ordinateur , allez sans l'onglet "Construire" puis choisissez "App (enregistrer .apk sur mon ordinateur)"
La compilation de votre application s'exécute et vous obtenez un fichier APK qui s'enregistre dans votre dossier téléchargement.

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 (**gong.mp3** et **photo_lamache.jpg**)

Description de l'interface :

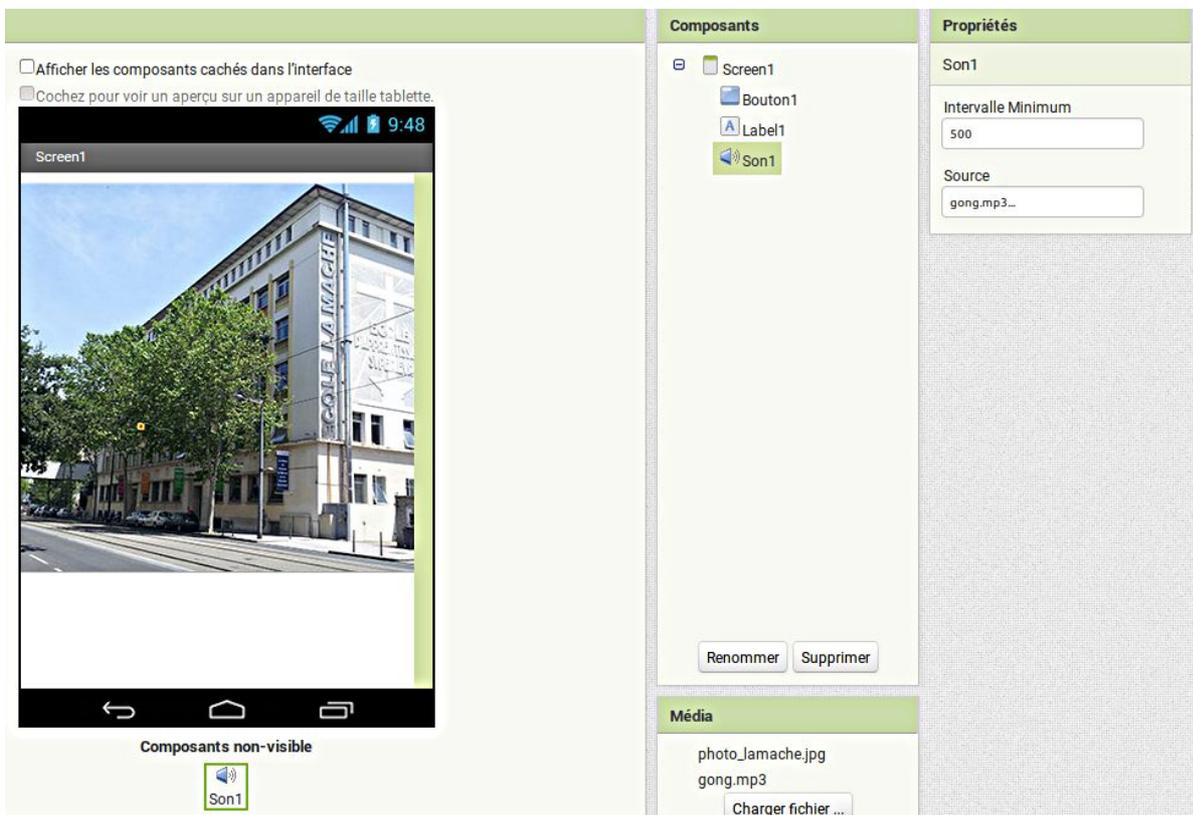
Vous allez donc insérer un bouton, du son (menu "média" puis "son") et un texte (Label)

Pour le paramétrage (propriétés) du bouton vous allez ajouter l'image :



Pour le Label vous y mettez le texte : "Bienvenue au Lycée LaMache" et le rendez invisible.

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



Maintenant le programme à réaliser :



← A trouver dans l'élément "bouton1"

← A trouver dans l'élément "son"

← A trouver dans l'élément "Label1" et dans le menu "logique"

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

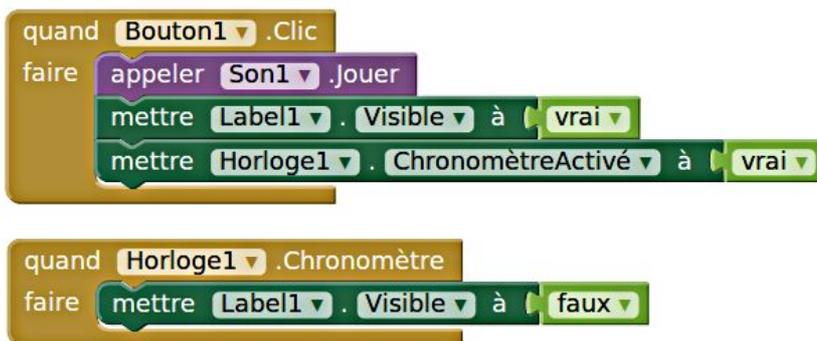
Version N°2 : amélioration et rajout d'une temporisation

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



Vous mettrez aussi l'image à une taille adaptée à notre écran (H:55 %, L:60%)

Le programme sera le suivant :



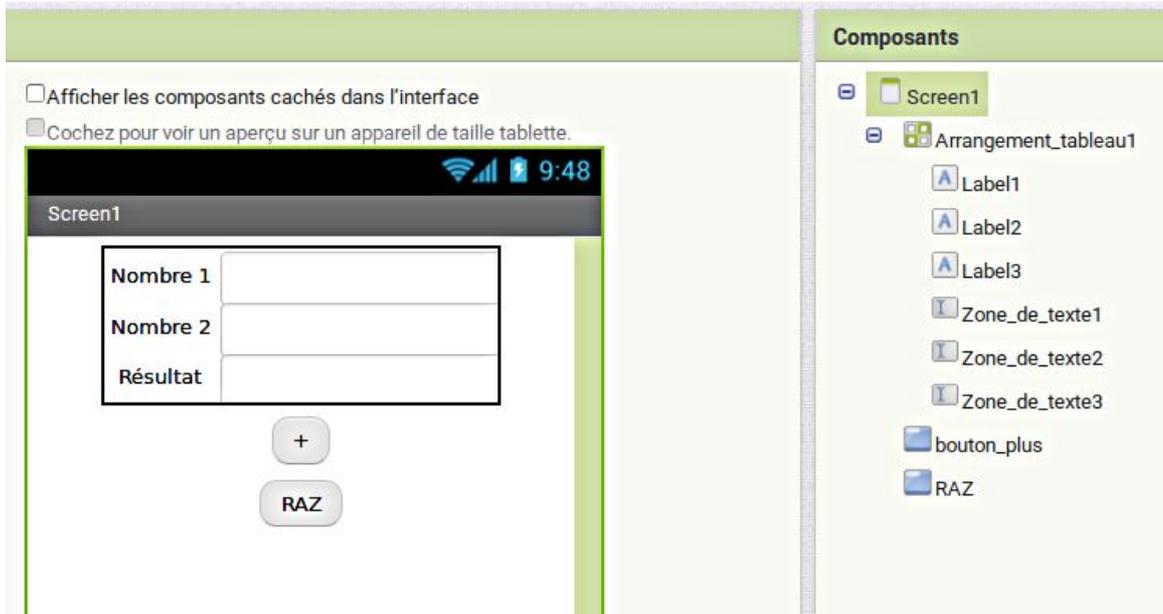
3. *Avant de faire la programmation des blocs, écrire l'algorithme du programme précédent (cela doit décrire de manière intelligible ce que réalise le programme).*
4. *Faites le programme et testez-le (copies d'écran à fournir). Quand tout fonctionne faites valider par le professeur.*

IV. Utilisation de fonctions mathématiques

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

Description de l'interface :

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



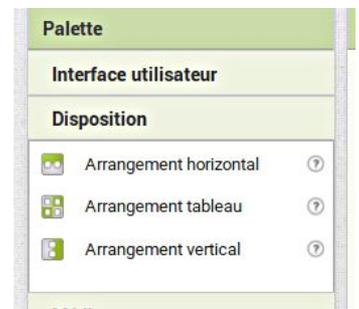
Pour placer nos différents éléments correctement les uns par rapport aux autres nous allons les mettre dans un tableau :

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")

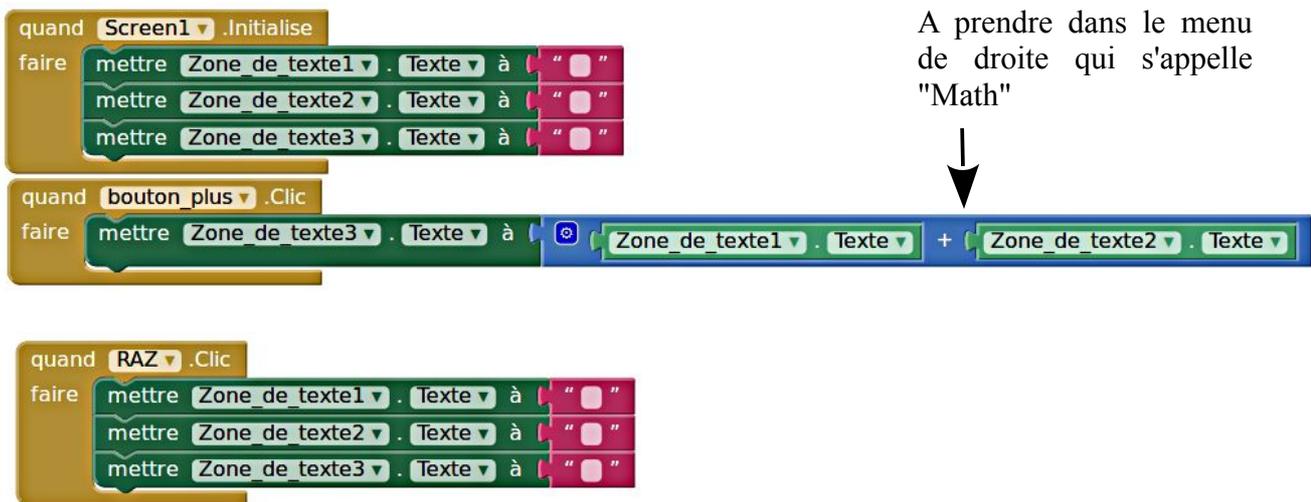
Pour les zones de texte il faudra préciser "nombres uniquement" et pour la 3ème zone de texte ne pas l'activer (en effet elle ne doit pas pouvoir être modifiée manuellement).



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 :



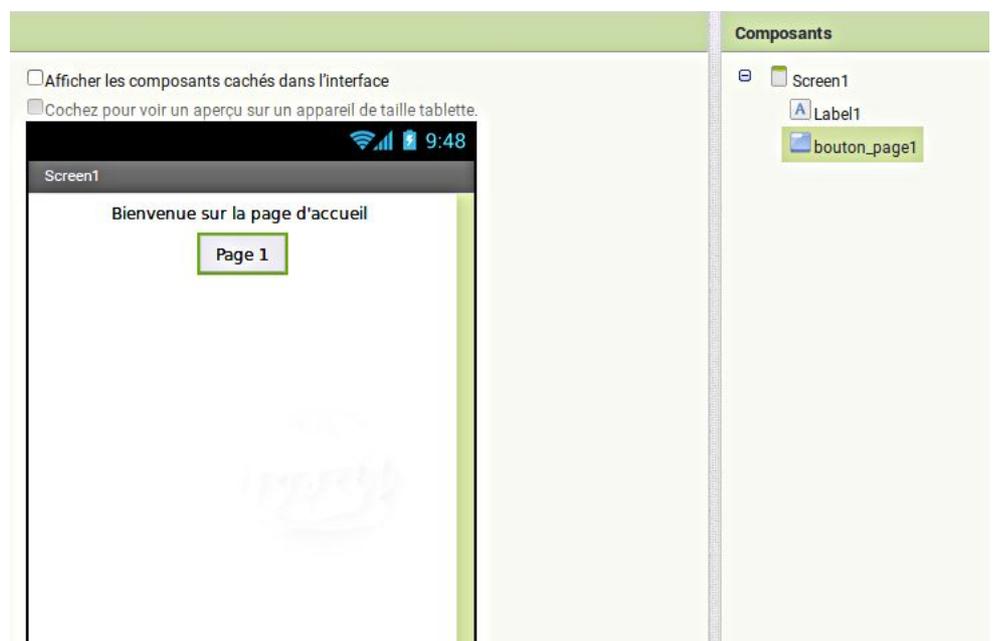
1. *Avant de faire le paramétrage de l'interface et la programmation des blocs, écrire l'algorithme du programme précédant (cela doit décrire de manière intelligible ce que réalise le programme).*
2. *Faites le programme et testez-le (copies d'écran à fournir). Quand tout fonctionne faites valider par le professeur.*

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

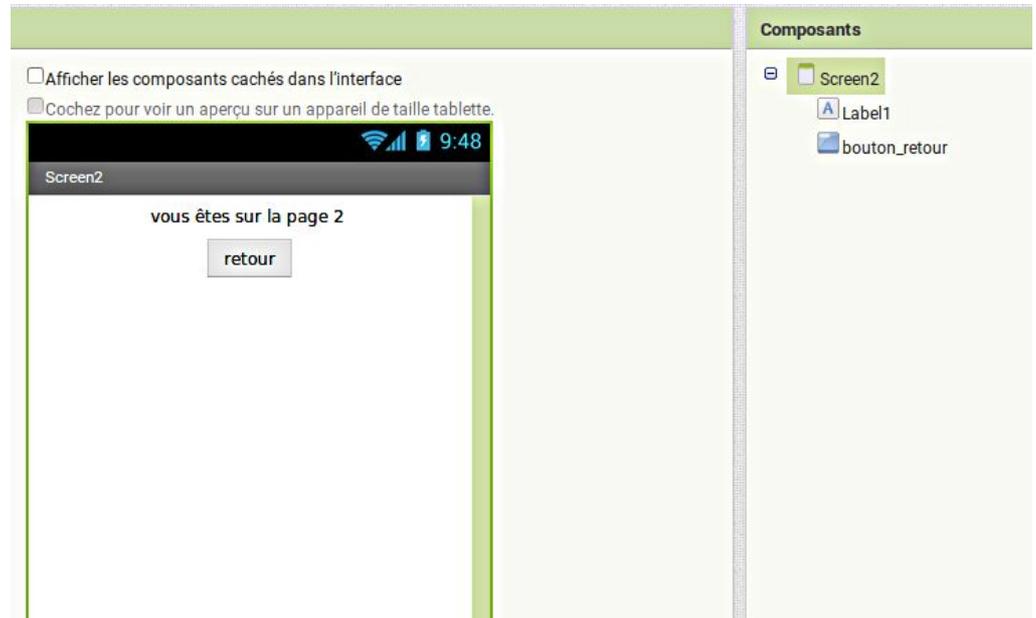
Créez une nouvelle application et paramétrez le "Screen1" de la façon suivante :

- un texte
- 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
- 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.

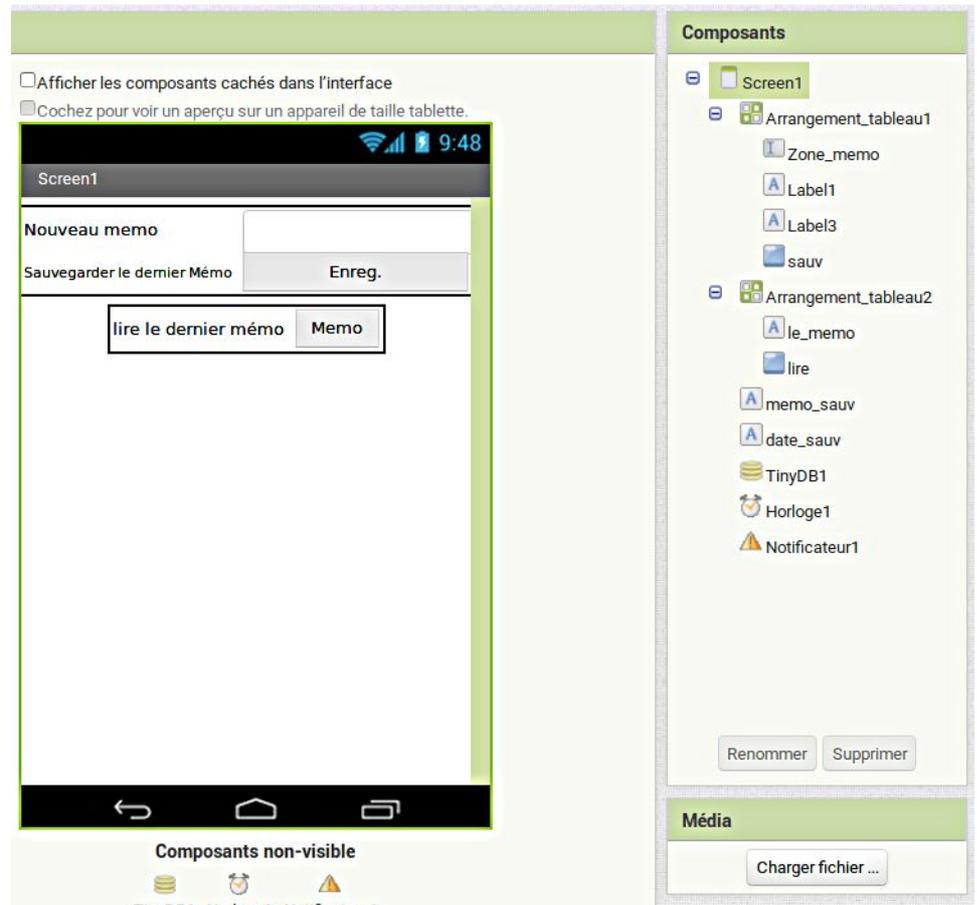
VI. Sauvegarde de données

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

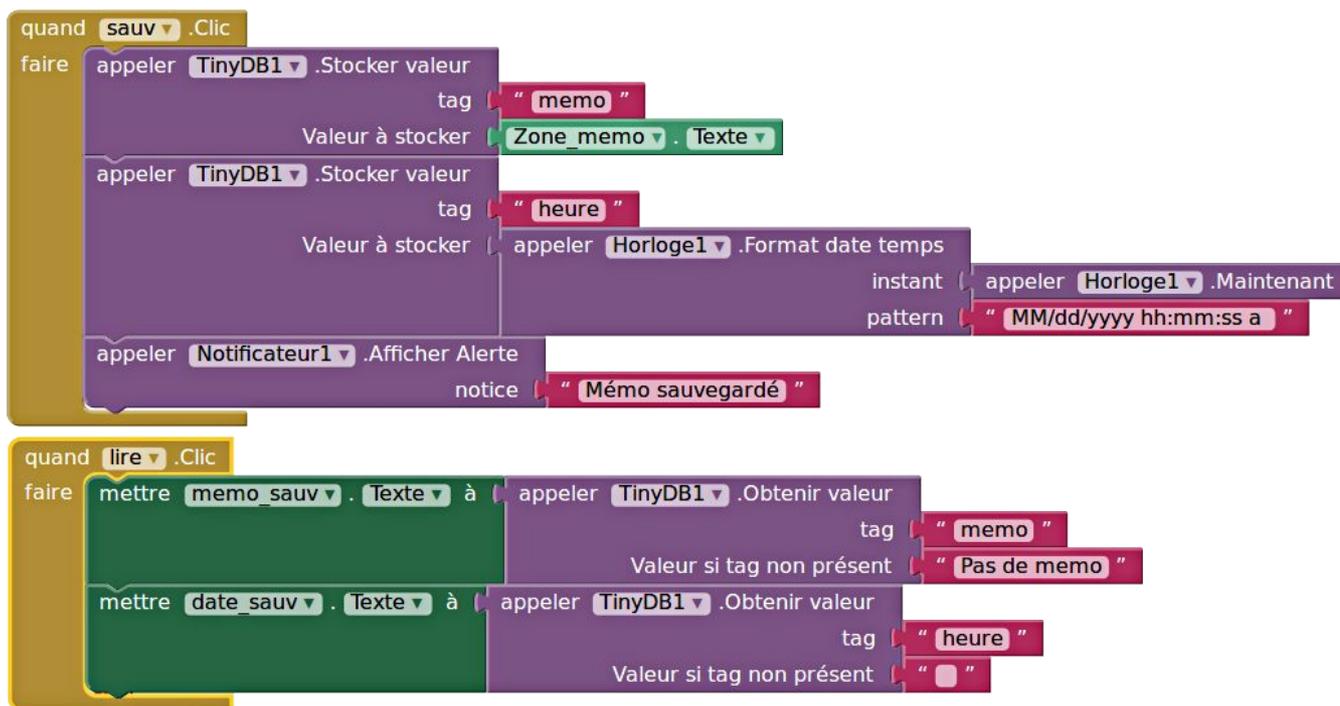


Le fonctionnement :

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 :

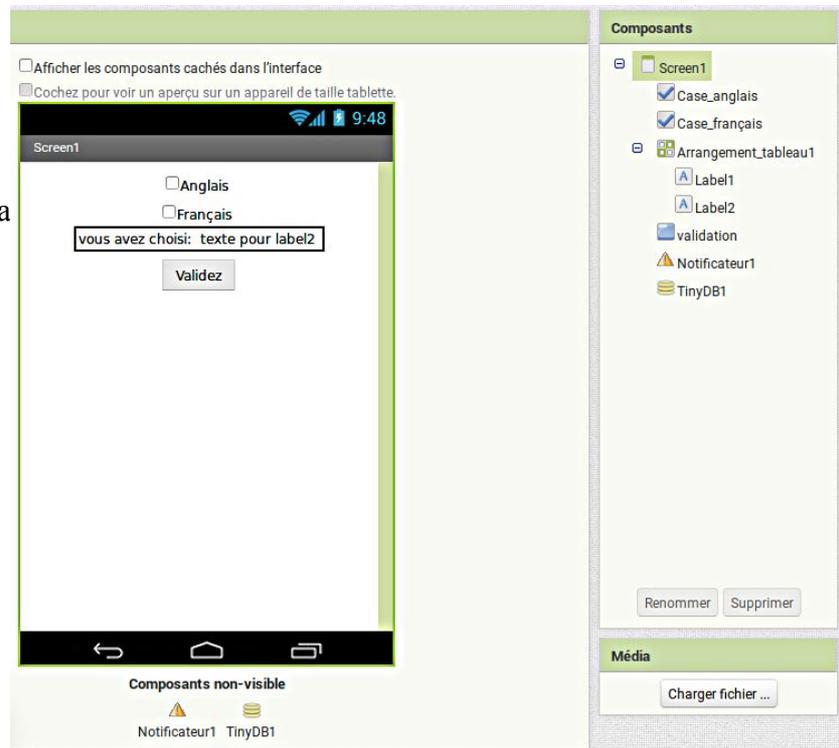
Nous allons maintenant créer une nouvelle application dont le but est de mémoriser le choix d'une langue et, suivant ce choix, d'afficher des pages différentes.



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.

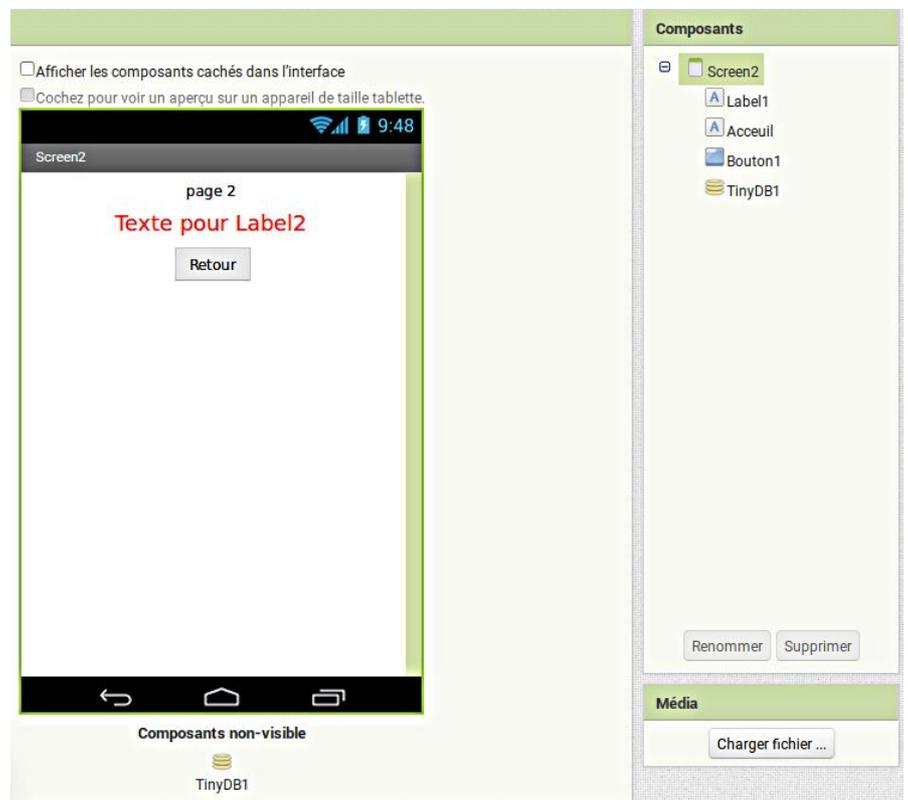
L'interface homme/machine pour la page 1 (Screen1) sera la suivante :

- 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 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 "
```

```

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
  
```

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) , 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.*

VIII. En bonus

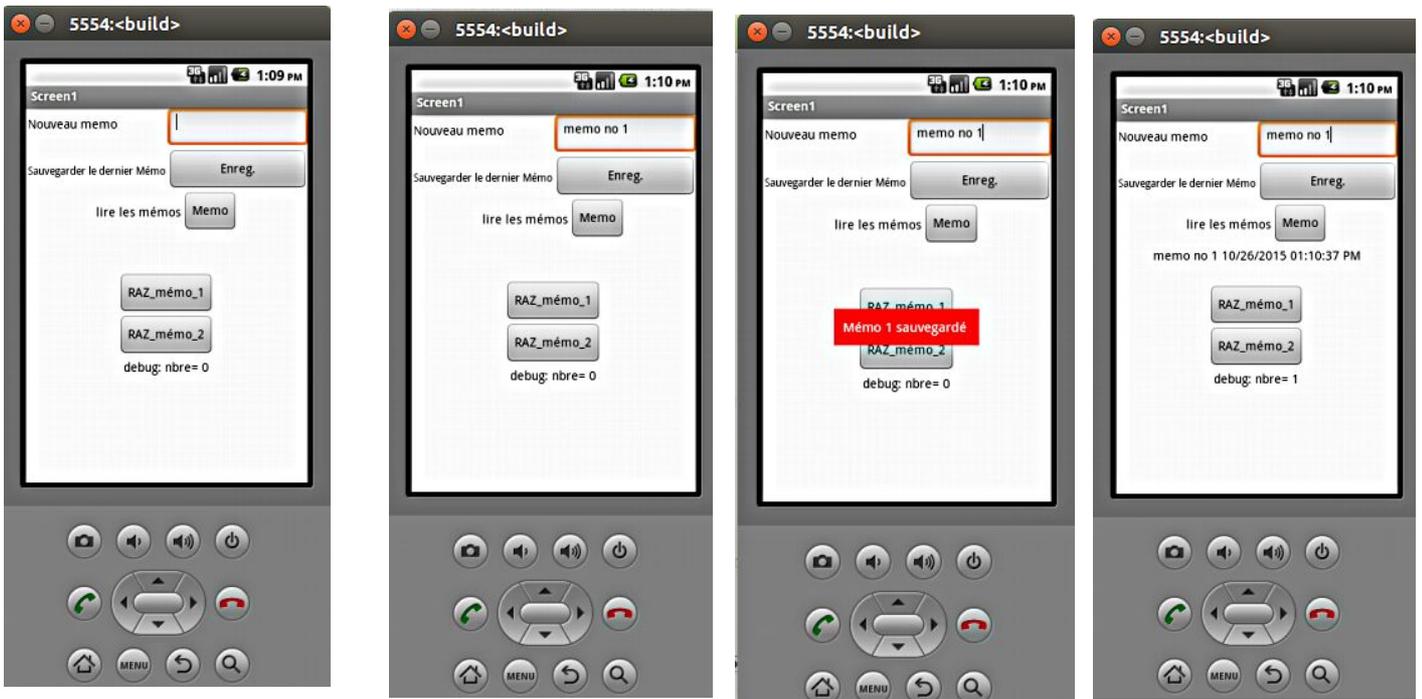
Vous allez reprendre l'application de la question VI (sauvegarde de données)

Vous allez maintenant rajouter :

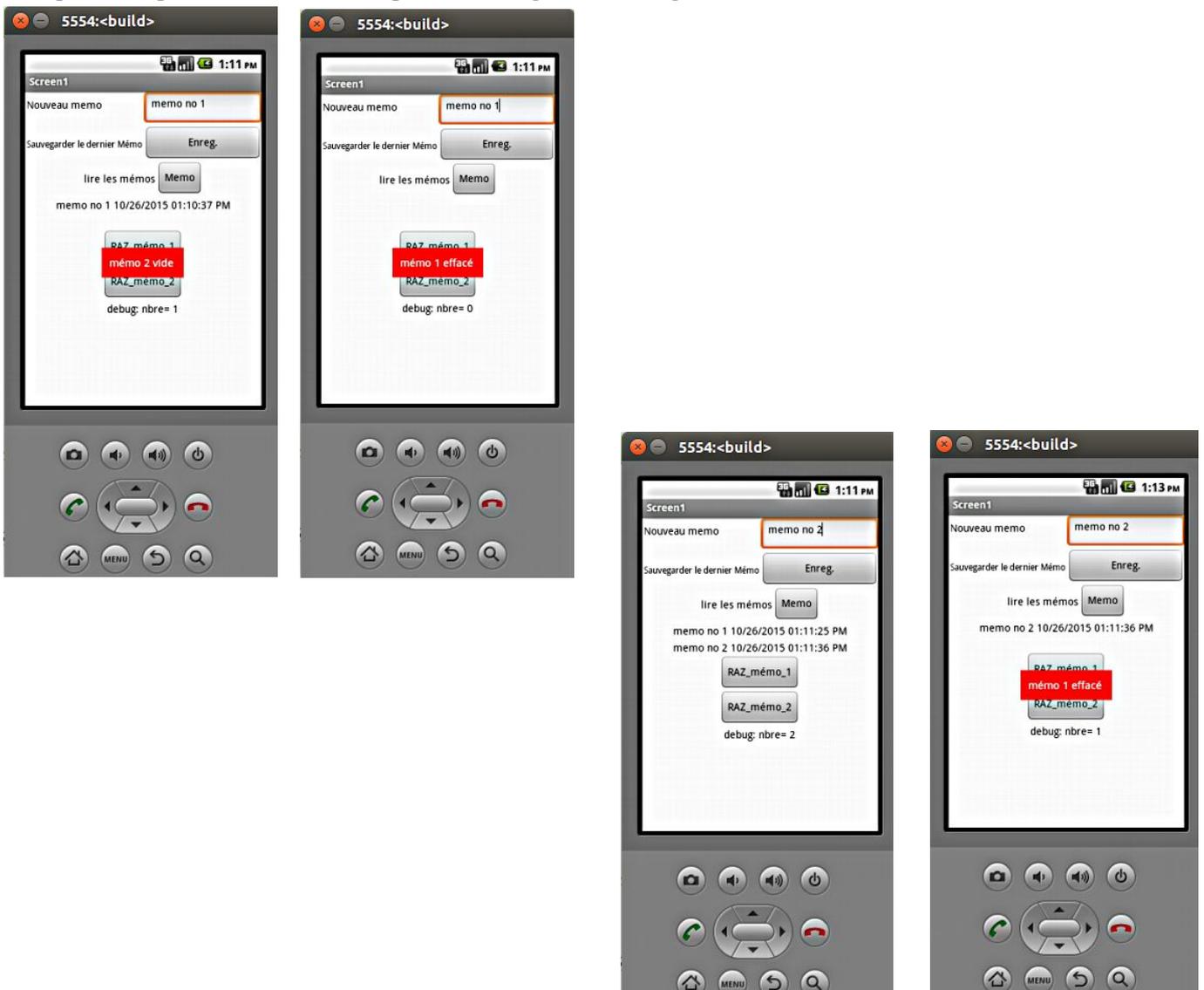
- un deuxième "mémo" mémorisable dans la base de données (il aura comme tags "memo2" et "heure2")
- 2 boutons pour effacer le mémo 1 ou le mémo 2
- un label pour afficher le 2ème mémo

Le fonctionnement :

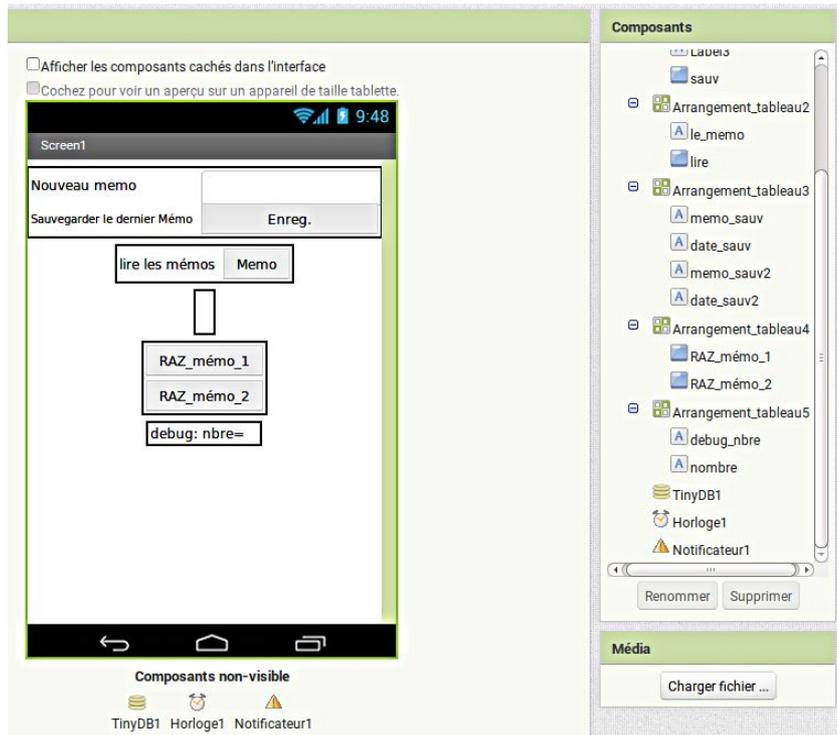
- au démarrage de l'application il faut aller chercher dans la base de données le nombre de mémo
- si on rentre un texte et que l'on appuie sur "Enreg", et qu'il y a encore de la place on enregistre le nouveau mémo. S'il n'y a plus de place on affiche un message bref (notificateur).



- Si l'on veut effacer un mémo, il faut choisir celui que l'on efface et si l'on efface le mémo 1, le mémo 2 prend sa place. Si l'on se trompe un message nous le signale :



L'interface homme/machine doit ressembler à cela :



Vous remarquerez la présence d'une variable debug. Celle-ci est le nombre de mémo mémorisés et a servi à tester le programme. Elle sera enlevée une fois l'application testée et validée (vous n'êtes pas obligé de la mettre si vous n'en ressentez pas le besoin).

Astuces ou aide :

- on peut avoir besoin d'une variable globale "nombre_memo" que l'on sauvegardera dans la base de données avec le tag "nombre"
- il faut penser à tous les cas possibles et pour cela réaliser avant toute chose un algorithme (sous la forme que vous voulez) pour bien avoir la structure du programme.

Faites le programme et testez-le. Quand tout fonctionne faites valider par le professeur.