Formation SIN

Utilisation du logiciel Basic4Android

Formateur : Pierre Aguillon

académie d'Orléans-Tours

PARTIE 1 : PROGRAMME 1

Objectifs :

- **Prendre en main le logiciel ;**
- **Travailler sur les propriétés des objets de façon statique et dynamique ;**
- **Découvrir les notions d'appartenances entre les objets ;**
- **Découvrir l'objet TIMER ;**
- **Implanter un programme dans une tablette ;**

Premier Programme

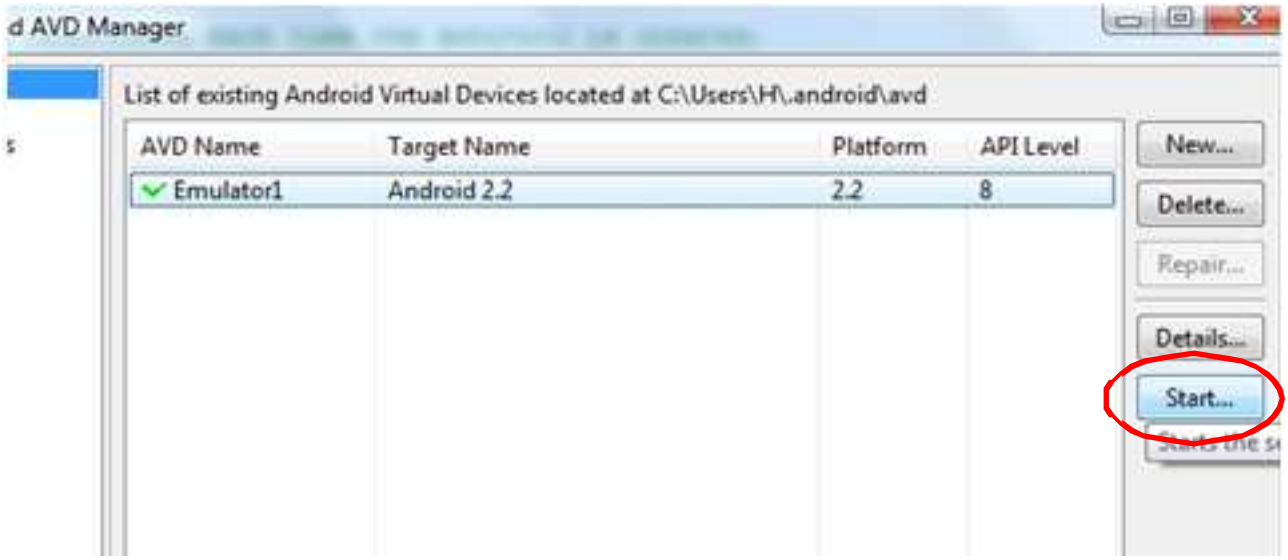
Lancez B4A. Vous devriez obtenir l'écran suivant :

La première chose à faire est d'enregistrer notre programme, pour se faire, cliquez sur le menu ***File-Save,*** je vous recommande de créer un dossier pour chaque programme.

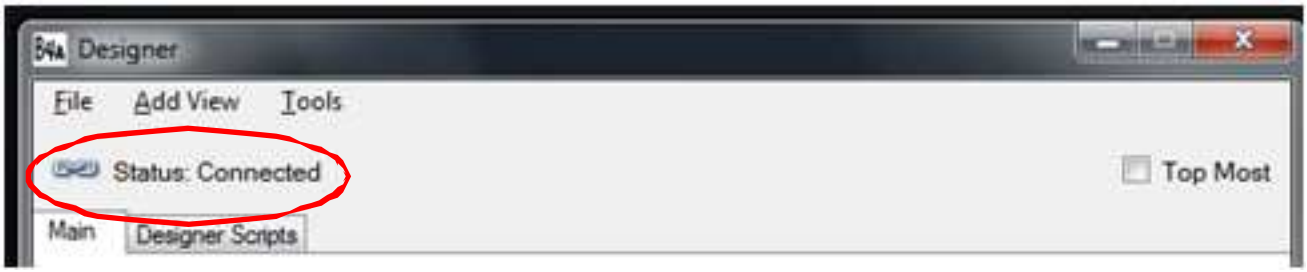
Créer donc un dossier nommé [programme1] puis donner un nom à votre projet par exemple "programme n1"

Avant de concevoir votre programme assurez-vous que votre émulateur est lancé (pour rappel ***Tools-run AVD manager***, sélectionnez l'émulateur et cliquer sur le bouton ***start***.)

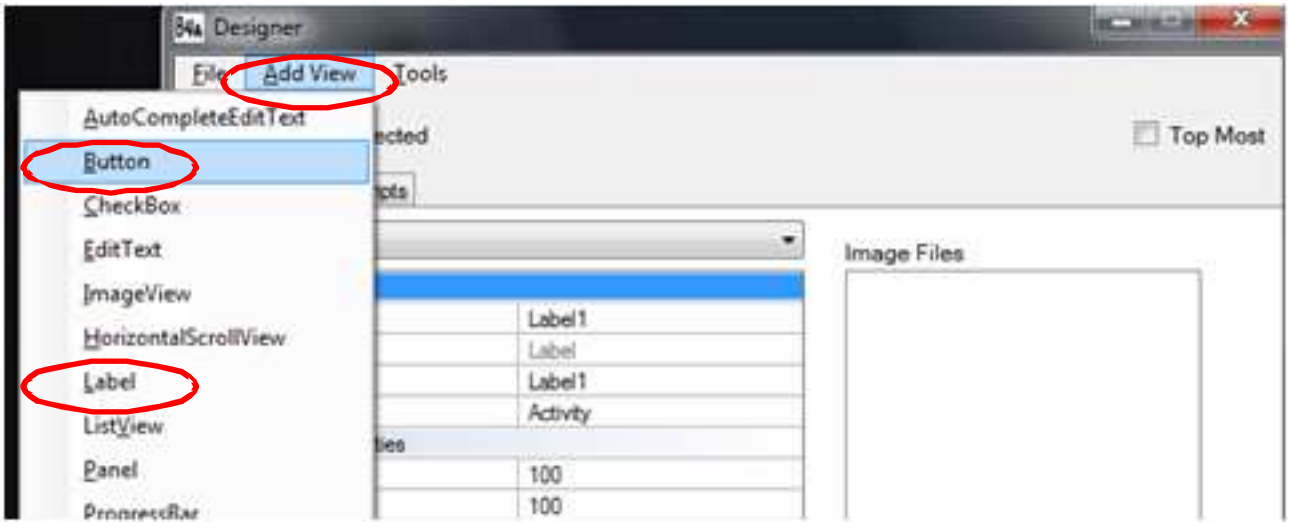
Si aucune tablette virtuelle n'a été définie utilisez le bouton **New**, puis voir la notice d'installation du logiciel sur le site de B4A. Sinon choisissez-en une dans la liste



Lancez le Designer (menu ***Designer***) et connectez-le avec votre AVD en cliquant sur l'icône (ou menu ***Tools-Connect to Device*****)**

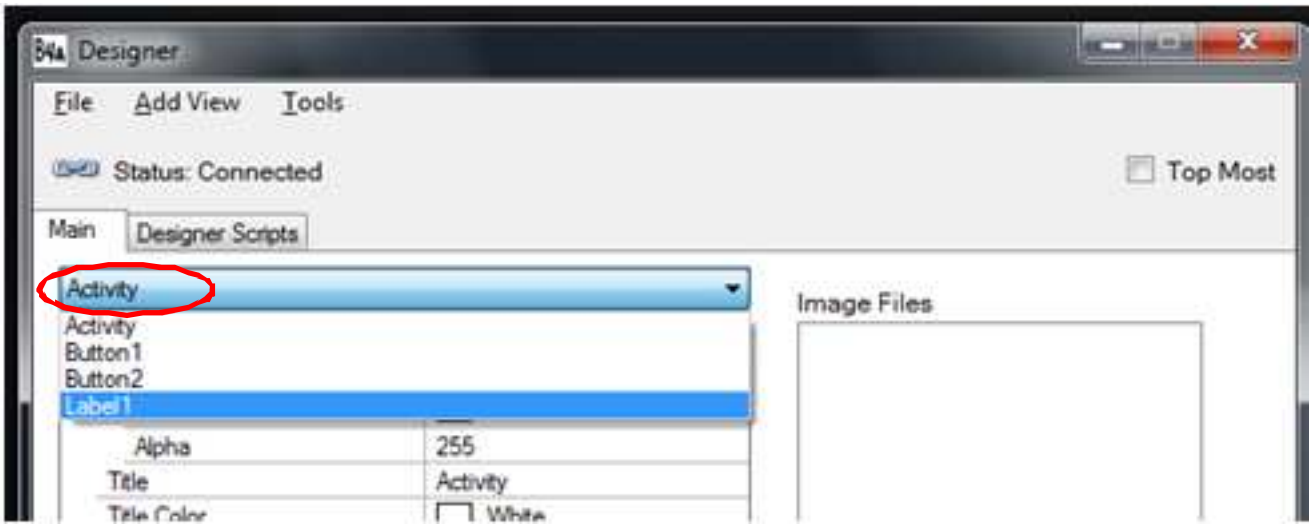


Nous allons placer trois objets sur notre feuille : deux boutons (**Button**) et une zone de texte (**Label**) : menu ***AddView – Button*** et ***Addview-Label***



Organisez les éléments pour qu'ils prennent place comme sur l'image suivante :

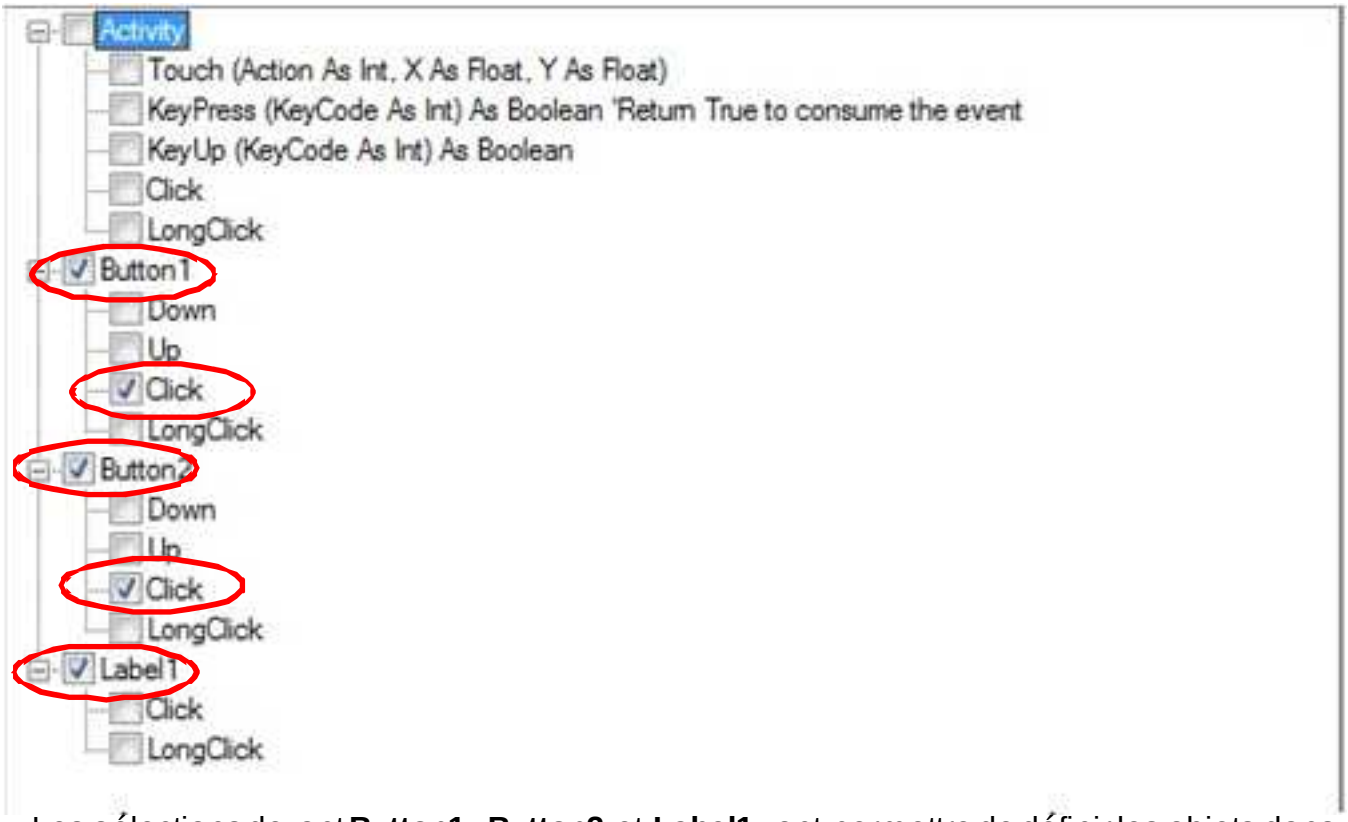
Votre projet propose donc maintenant quatre objets accessibles par le menu-déroulant



Nous n'allons pas pour le moment nous occuper des propriétés de chaque objet, mais plutôt voir comment ils s'intègrent dans notre projet.

On désire que si l'utilisateur clique sur **Button1** le texte de **Label1** soit « Bonjour » ; s'il clique sur **Button2** le texte soit « Au revoir ». Rien d'extraordinaire mais cela va nous permettre de changer les propriétés d'un objet suite à un évènement sur un autre objet.

On va maintenant automatiquement générer les lignes de commande correspondant aux évènements que l'on désire traiter. Pour cela utilisez le menu ***Tools-Generate Members***. Ouvrez toutes les arborescences et cochez les cases comme sur l'image suivante :

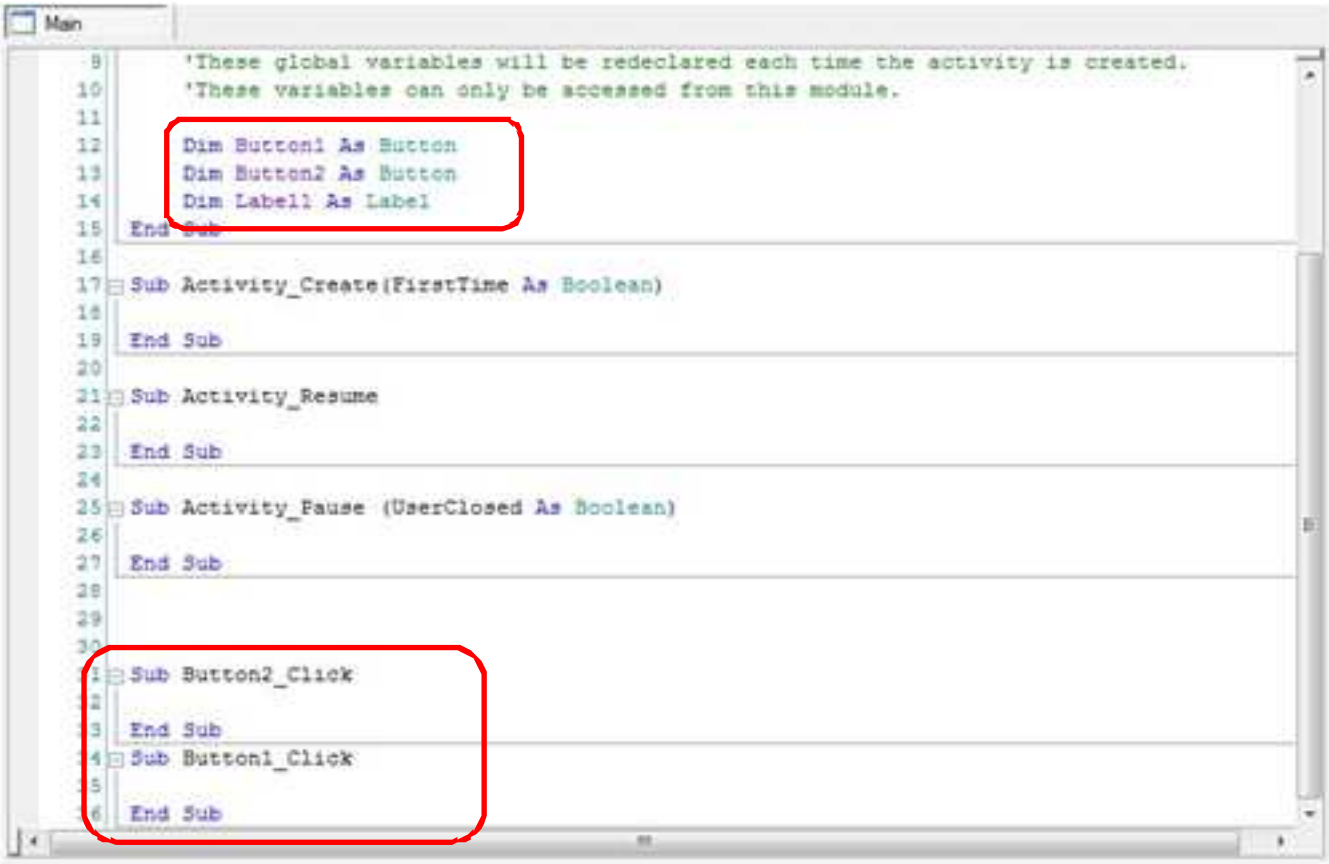


Les sélections devant **Button1**, **Button2** et **Label1** vont permettre de définir les objets dans notre programme. Nous n'en avons pas forcément besoin ; dans notre cas seul **Label1** aurait pu être coché. La sélection des **Click** sous **Button1** et **Button2** va automatiquement créer les fonctions associées à cet événement dans notre programme.

Cliquez sur ***Generate Members*** puis ***Close***.

Fermez le Designer et sauvegardez votre feuille sous le nom « feuille1 »

Regardez le résultat dans votre programme



Maintenant la chose est simple à comprendre : si l'utilisateur clique sur **Button1** alors ce sont les lignes de code placées dans la fonction **Sub Button1_click** qui seront exécutées, et réciproquement pour **Button2**.

On va donc modifier ces lignes pour que la propriété **Text** de **Label1** soit modifiée suite au clique sur un des boutons.

Pour cela complétez les lignes de codes suivantes :

Il ne nous reste plus qu'a définir ce que doit faire notre programme au démarrage du logiciel (un peu notre Void Main(void)…).Cela se passe sous la fonction :
Sub Activity_Create (FirstTime as Boolean).

La seule chose à effectuer est de charger notre feuille pour permettre à l'utilisateur d'avoir accès aux deux boutons. C'est ici que l'objet **Activity** intervient. Saisissez la ligne de commande suivante :

```
16
17 Sub Activity_Create(FirstTime As Boolean)
18     Activity.LoadLayout("feuille1")
19 End Sub
20
```

La simulation du programme se lance soit en utilisant l'icône avec la flèche **Run** en bleu, soit par le menu ***Project – Compile and Run***.

Si c'est la première fois que vous exécutez le programme, deux fenêtres vont s'ouvrir. La première vous permet de définir le nom package à créer. Il doit respecter la syntaxe xxx.xxxx. Utilisez par exemple b4a.programme1.

Le nom du package est utilisé comme identificateur unique de l'application. Ce n'est pas le nom du fichier que vous allez générer pour installer l'application sur votre tablette, mais un identifiant de l'application une fois installée.

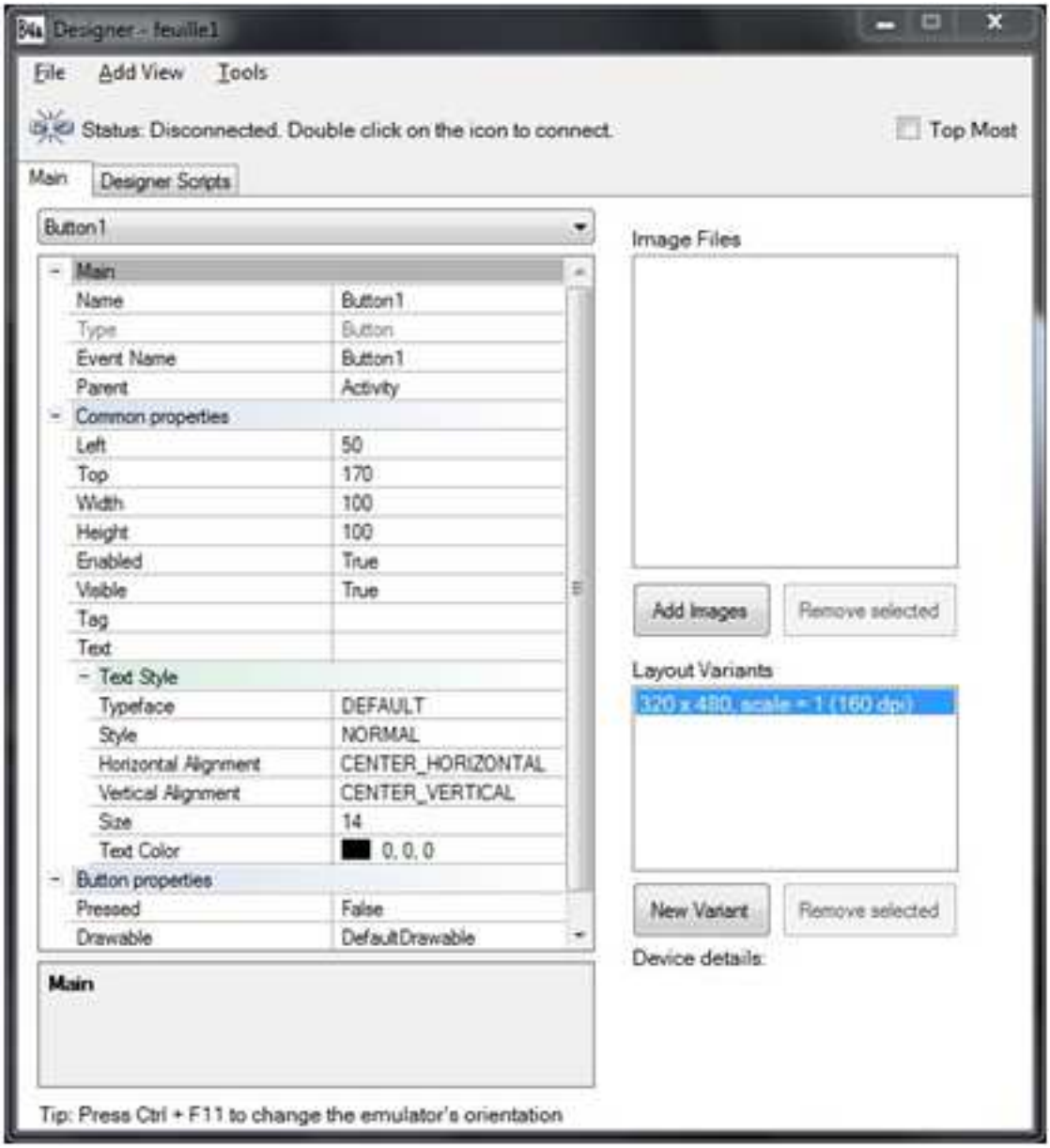
La deuxième est le nom du label apparaissant dans Android. Tapez par exemple « Programme n°1 ». Le programme se compile alors puis est transféré dans votre AVD.

Testez-le.

Modification n°1

On va maintenant réaliser deux modifications sur les propriétés de nos objets : une sur le Designer et une dans notre programme ; afin que sur les boutons on puisse lire « Arrivé » et « Départ » et que le texte du **Label** apparaissent en rouge pour le message « Bonjour » et en bleu pour « Au revoir ».

Ouvrez le Designer et dans la liste déroulante choisissez **Button1** pour faire apparaître ses propriétés :



C'est la propriété **Text** qui nous intéresse ici. Saisissez « Arrivé » et constatez le résultat dans vote AVD. Si cela vous plait changez aussi d'autres propriétés du texte (taille, couleur ….).

Réalisez la même opération sur le **Button2**.

Fermez le Designer en sauvegardant les modifications (il n'est pas nécessaire dans ce cas d'utiliser la commande ***Generate Members*** du Designer afin de modifier votre programme).

On peut dire pour clarifier les choses que vous venez de modifier de façon statique les propriétés de vos objets.

Il faut maintenant modifier la couleur du texte du **Label1**. Cette modification, contrairement aux précédentes doit s'effectuer pendant l'exécution de votre programme, et doit prendre deux valeurs différentes.

Modifiez votre programme de la façon suivante :

```
Sub Button2_Click
    Label1.Text="Au revoir"
    Label1.TextColor=Colors.Blue
End Sub
Sub Button1_Click
    Label1.Text="Bonjour"
    Label1.TextColor=Colors.Red
End Sub
```

Lancez la simulation.

On peut dire pour clarifier les choses que vous venez de modifier de façon dynamique les propriétés de votre objet.

Notre programme est opérationnel, mais on peut regretter que nos objets portent des noms aussi peut évocateur de leur fonction. Effectivement, si mon programme est bien plus conséquent, comment pourrais-je me rappeler que le **Button1** est le bouton « Arrivé » ?

Pour cela B4A permet de nommer les objets.

Relancez le Designer et modifiez la propriété **Name** de **Buton1** en saisissant « BP_arrive ». Constatez que la propriété **Event Name** vient aussi de changer. Cette propriété correspond au nom donné par B4A dans votre programme aux fonctions associées à l'objet (**Sub BP_arrive_Click** par exemple). Vous pouvez très bien imaginer que votre objet porte un nom mais que ses fonctions associées en portent un autre. Néanmoins je ne vous le conseille pas.

Modifiez les autres objets pour que **Button2** devienne « BP_Depart » et **Label1** devienne « Message ».

Ayant changé les noms des objets, il faut à nouveau demander au Designer de lier vos « nouveaux » objets avec votre programme : commande ***Generate Members***

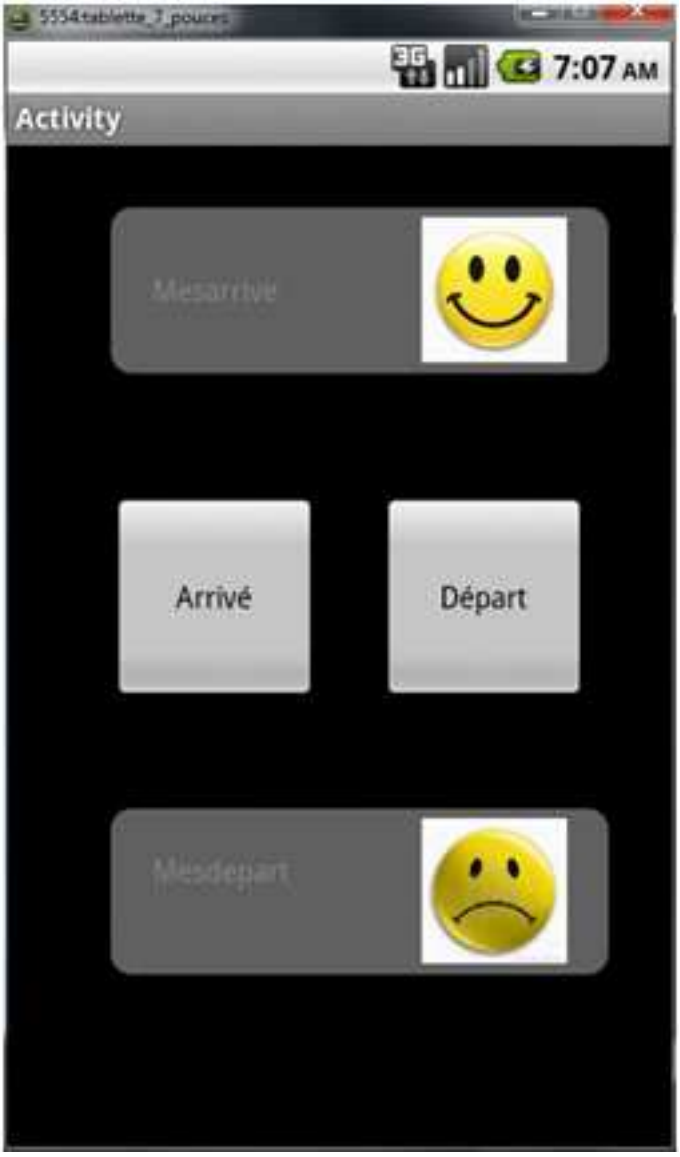
Fermez le Designer et constatez le résultat sur votre programme. B4A a bien généré les lignes liées aux modifications dans le Designer mais il n'a pas substitué nos anciens noms par nos nouveaux. Il faut donc le faire à la main, même si les outils du menu ***Edit*** rendent les choses assez simples.

Modifiez votre programme pour le rendre à nouveau opérationnel en nommant les deux boutons BParrive et BPdepart.

Modification n°2

On va maintenant encore modifier notre programme pour voir l'intérêt que peut avoir la notion d'appartenance dans le Designer.

Supposons que l'on veuille agrémenter notre projet d'une image accompagnant le « Bonjour » (un smiley souriant) et d'une associée au « Au revoir » (un smiley triste). Par ailleurs les deux textes et images devront apparaître à deux endroits différents de notre écran (en haut pour l'arrivée, en bas pour le départ).



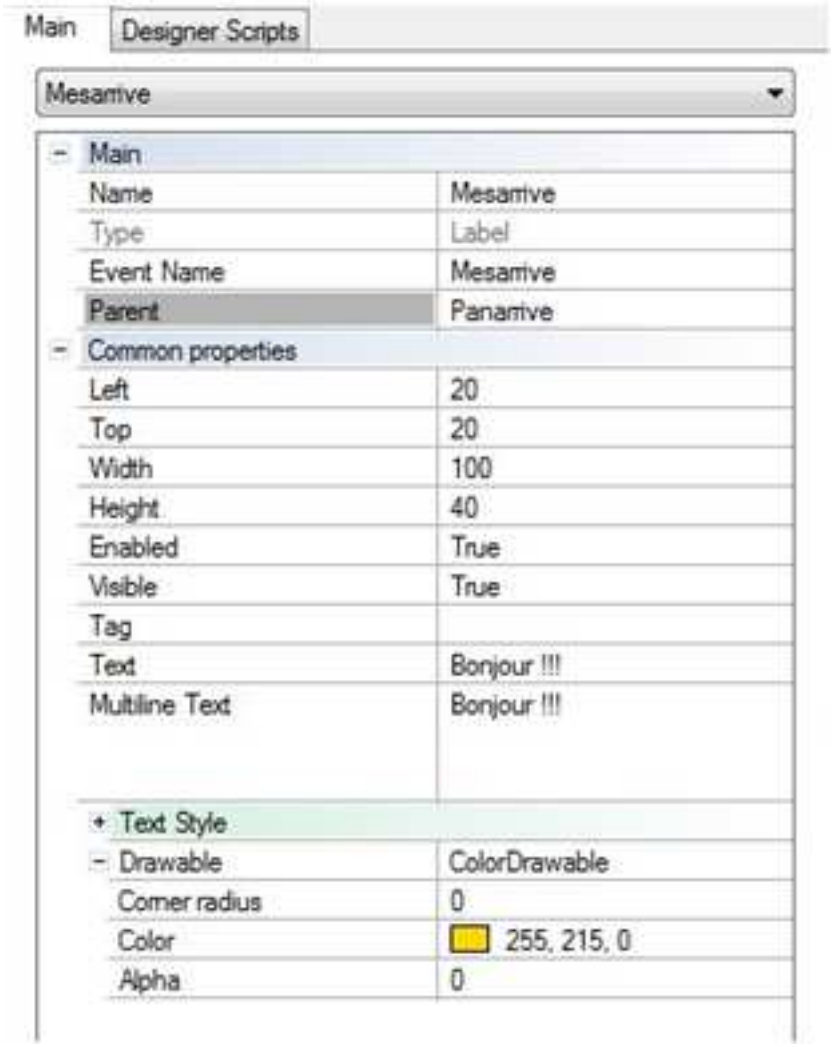
Ouvrez le designer et supprimez l'objet **Label** (il faut cliquer dessus pour le sélectionner dans le AVD et utiliser le menu ***Tools- Remove Selected Views***).

On va placer deux panneaux (**panel**), un en haut l'autre en bas. Nommez les respectivement Panarrive et Pandepart. Définissez leur la même taille en largeur (**Width**) et hauteur (**Height**). Choisissez une couleur qui se détache du fond de l'écran.

Sélectionnez **False** à deux propriétés :

- **Visible** : cela aura pour effet de ne pas rendre visible le panneau au lancement du programme ;
- **Enabled** : cela interdira à l'utilisateur d'utiliser le panneau.

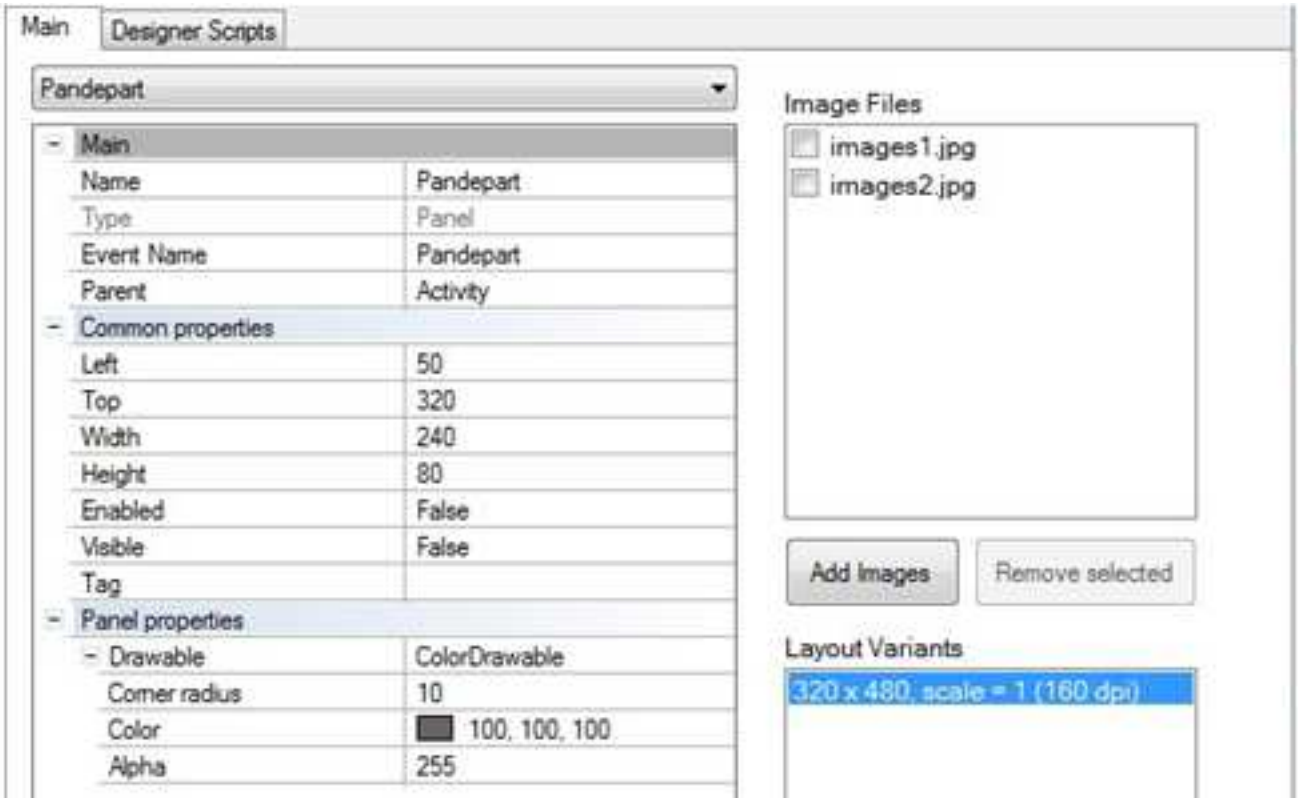
Placez maintenant un **label** que vous nommerez Mesarrive. Dans la propriété **Text** saisissez le texte à afficher « Bonjour !!!! ». Modifiez sa couleur pour qu'il s'écrive en jaune. Par défaut votre label appartient à l'objet **Activity**. On va changer sa parenté pour qu'il soit rattaché au panneau arrivée (Panarrive).



Une fois apparenté au panneau, votre label ne peut se déplacer qu'à l'intérieur de ce dernier. Par ailleurs il hérite des propriétés du panneau : si le panneau n'est pas visible, le label non plus.

Réalisez la même opération pour un deuxième **label** « Mesdepart » parenté avec le panneau Pandepart.

Reste maintenant à placer les images. Il faut d'abord charger les fichiers images pour qu'ils puissent s'insérer dans votre projet. Pour cela utilisez la commande ***Add Images*** et chargez les fichiers images1.jpg et images2.jpg.

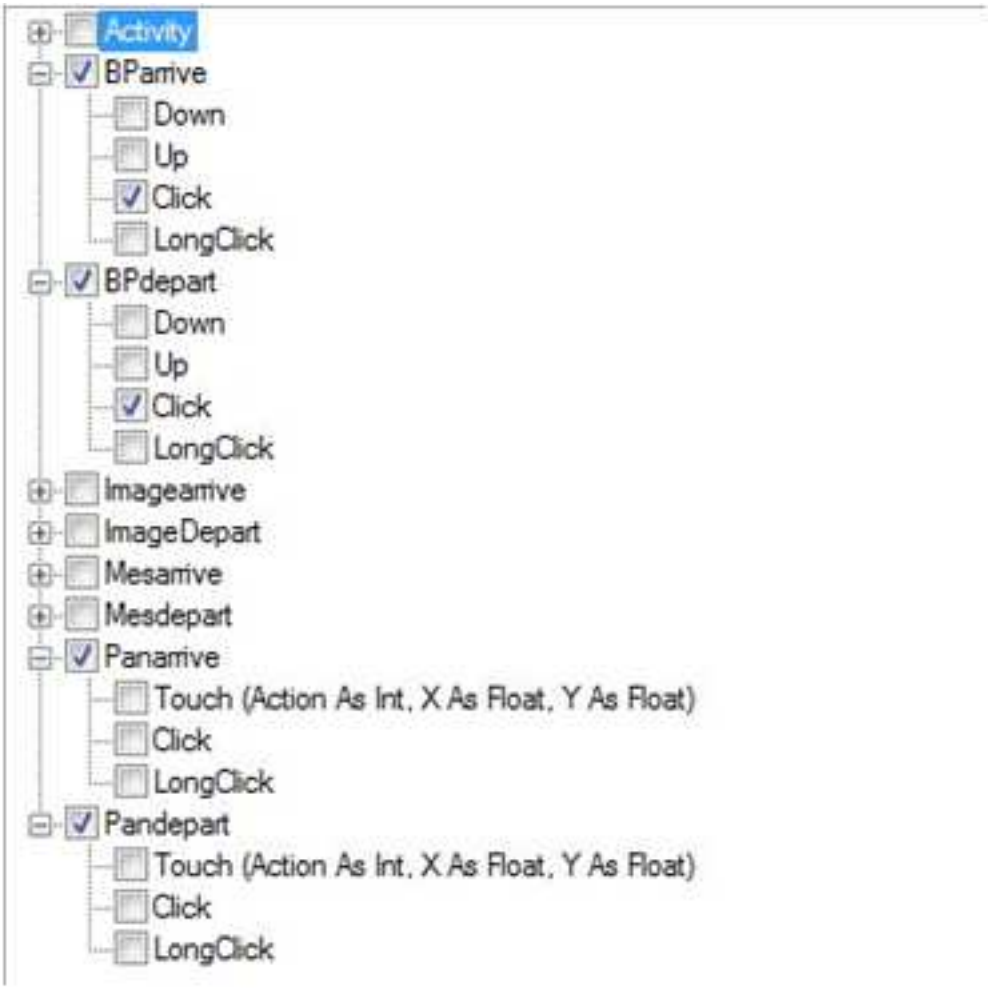


Placez maintenant un objet **Imageview** ; renommez le Imagearrive ; choisissez comme parent Panarrive ; choisissez comme fichier image (**Image file**) le fichier images1.jpg ; modifiez la taille de l'image et sa position pour qu'il se place dans le panneau.

Réalisez la même chose pour le départ avec un objet **ImageView** que vous nommerez Imagedepart dans le panneau Pandepart.

Votre conception est terminez, lancez la commande ***Tools-Generate Members*** pour ne gérer que les choses suivantes :

- La définition des deux panneaux ;
- La création des deux fonctions associées au Click sur les deux boutons



Fermez le Designer et analysez votre programme. Il devrait ressembler à celui-là :

```
7
8  Sub Globals
9      'These global variables will be redeclared each time the activity is created.
10     'These variables can only be accessed from this module.
11     Dim Panarrive As Panel
12     Dim Pandepart As Panel
13     Dim BParrive As Button
14     Dim BPdepart As Button
15 End Sub
16
17 Sub Activity_Create(FirstTime As Boolean)
18     Activity.LoadLayout("feuille1")
19 End Sub
20
21 Sub Activity_Resume
22
23 End Sub
24
25 Sub Activity_Pause (UserClosed As Boolean)
26
27 End Sub
28
29 Sub BPdepart_Click
30
31 End Sub
32 Sub BParrive_Click
33
34 End Sub
```

Modifiez les lignes des deux fonctions de la manière suivante :

```
29
30  Sub BPdepart_Click
31      Pandepart.Visible=True
32      Panarrive.Visible=False
33  End Sub
34  Sub BParrive_Click
35      Pandepart.Visible=False
36      Panarrive.Visible=True
37  End Sub
```

Compilez votre programme puis exécutez-le.

On voit que le fait d'avoir associé les objets ImageDepart et Mesdepart au panneau Pandepart nous a ainsi permis de ne pas avoir eu à changer leurs propriétés. Ils ont hérité de celles du panneau.

Modification n°3

On va encore améliorer le programme. Pour le moment le panneau Panarrive reste visible tant que l'utilisateur ne clique pas sur le bouton Départ (réciproquement pour l'arrivé). On souhaite qu'en cas de non activité pendant un certains temps (disons 5 secondes) les deux panneaux disparaissent.

La première solution consisterait à mettre en place un délai de 5s après l'apparition du panneau, puis de l'effacer. Une boucle comptant (type For…Next) permet de réaliser cette temporisation. Mais ce n'est pas du tout la bonne solution :

- Comment être certains de la précision de la durée ;
- Pendant la boucle votre programme est bloqué.

Pour remédier à notre problème B4A propose un objet **Timer**. Celui-ci n'est pas disponible dans le Designer puisqu'il n'a pas d'interaction possible avec l'utilisateur. Il va donc falloir travailler directement avec le programme en VB.

Cet objet possède trois paramètres :

Initialize : initialisation du timer en définissant le nom de l'événement associé à l'objet et l'intervalle en ms où le timer déclenche un événement.

Syntaxe : nom_du_timer.Intialize(« nom_de_l'événement » as String, Intervalle as Long)
Exemple : Timer1.initialize(« Timer1 »,1000)

Interval : pour redéfinir l'intervalle en ms où le timer déclenche un événement.
Syntaxe : nom_du_timer.Interval=Intervalle as Long
Exemple : Timer1.Interval=1000

Enabled : valide ou non les fonctionnalités du timer
Syntaxe : Timer1.Enabled=True ou False

Un seul événement est associé : **Tick** lorsque le timer atteint son intervalle de temps
Sub Timer1_Tick

La première étape consiste donc à déclarer l'objet **Timer**. Modifiez le programme :

```
'Activity module
Sub Process_Globals
    'These global variables will be declared once when the application starts.
    'These variables can be accessed from all modules.
    Dim Montimer As Timer
End Sub
```

Au lancement de votre programme il faut initialiser le **Timer** :

```
Sub Activity_Create(FirstTime As Boolean)
    Activity.LoadLayout("feuille1")
    Montimer.Initialize("Effacement",5000)

End Sub
```

Ensuite, il faut créer la fonction associée à l'objet :

```
Sub Effacement_Tick
    Pandepart.Visible=False
    Panarrive.Visible=False
End Sub
```

Et enfin, il faut relancer pour 5 secondes le **Timer** après l'appui sur un des deux boutons poussoirs :

```
Sub BPdepart_Click
    Montimer.Enabled=False
    Pandepart.Visible=True
    Panarrive.Visible=False
    Montimer.Enabled=True
    End Sub
Sub BParrive_Click
    Montimer.Enabled=False
    Pandepart.Visible=False
    Panarrive.Visible=True
    Montimer.Enabled=True
```

Testez votre programme.

Installation de l'application dans la tablette

Une fois votre programme opérationnel, il ne vous reste plus qu'à l'installer dans votre tablette. B4A a compilé un fichier package dont le nom est result.apk (result comme résultat de votre travail et apk comme Android Package). C'est ce fichier qu'il va falloir copier dans votre tablette. Il se trouve dans le répertoire Objects de votre dossier de travail (normalement …\programme1\objects\result.apk).

Connectez au port USB de votre ordinateur la tablette, et regardez l'écran de cette dernière. Suivant le type de tablette la réaction peut être différente. Quoi qu'il en soit vous devez activer (ou monter) la mémoire de stockage USB, afin que votre ordinateur détecte la tablette et vous propose un disque de sauvegarde (souvent LOST.DIR). Placez-y le fichier result.apk.

Il faut maintenant désactiver (ou démonter) la mémoire de stockage USB de votre tablette. Vous pouvez alors retirer le câble USB. Ne reste plus qu'à utiliser le gestionnaire de fichier de votre tablette pour retrouver le fichier apk et l'installer.