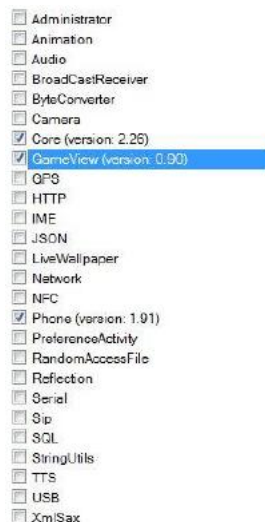


## Mise en situation : Mise en œuvre d'un cadran à aiguille

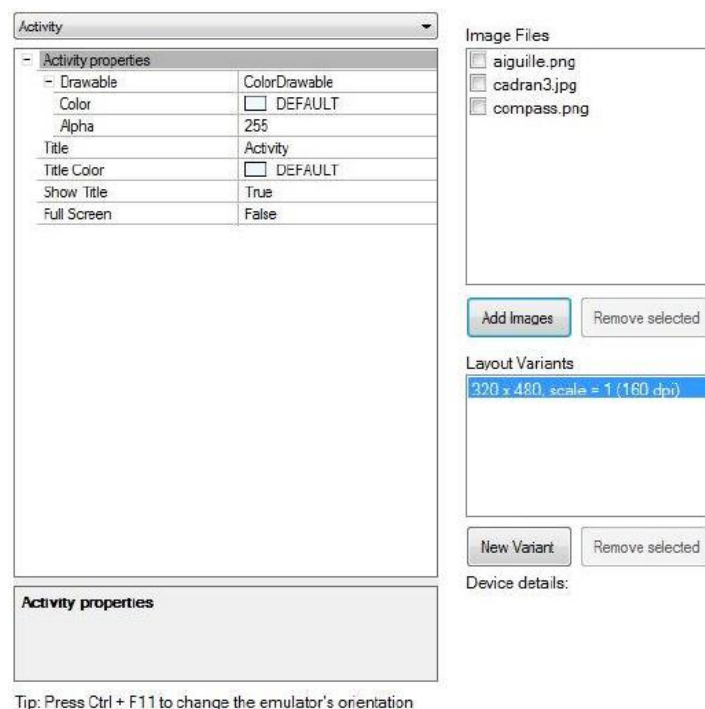
Nous allons maintenant concevoir un programme qui affichera un cadran à aiguille. Deux boutons permettront de faire tourner l'aiguille dans un sens ou dans l'autre. On va donc devoir travailler autour de l'animation d'objet graphique. Pour cela nous allons mettre en œuvre une autre bibliothèque : la **GameView**.

## Programme

- 1- Créez un nouveau projet B4A que vous sauvegarderez dans un dossier **programme4** et que vous nommerez « **Programme n4** ».
- 2- Pour mettre en œuvre notre programme on va utiliser une librairie additionnelle : **GameView**



- 3- Lancez le Designer.
- 4- Dans la partie droite ajoutez les fichiers images suivants : **aiguille.png** ; **cadran3.jpg** ; **compass.png** :



- 5- Dans votre feuille placez un objet **ImageView** ou vous voulez, avec la taille que vous désirez. Appelez cet objet **cadran** et associez-le avec le fichier **compass.png**.
- 6- Placez deux boutons que vous nommerez **gauche** ; **droite** ; et qui contiendront les textes correspondant.
- 7- Générez les membres (Click sur les deux boutons et initialisation du cadran), sauvegardez votre feuille sous le nom **layout1**, puis quittez le Designer.
- 8- Nous avons donc positionné notre cadran. L'aiguille se déplaçant dedans est un autre objet que nous allons définir plus tard dans le programme. Néanmoins, pour que cette aiguille puisse se placer devant le cadran, il faut que notre objet **cadran** soit placé en arrière-plan de l'image. Modifiez votre programme pour charger votre feuille et placer le cadran en fond d'image.

```
18 Sub Activity_Create(FirstTime As Boolean)
19     'Do not forget to load the layout file created with the visual designer.
20     Activity.LoadLayout("Layout1")
21     cadran.SendToBack
```

- 9- Il faut maintenant créer un objet **aiguille** de type **BitmapData**

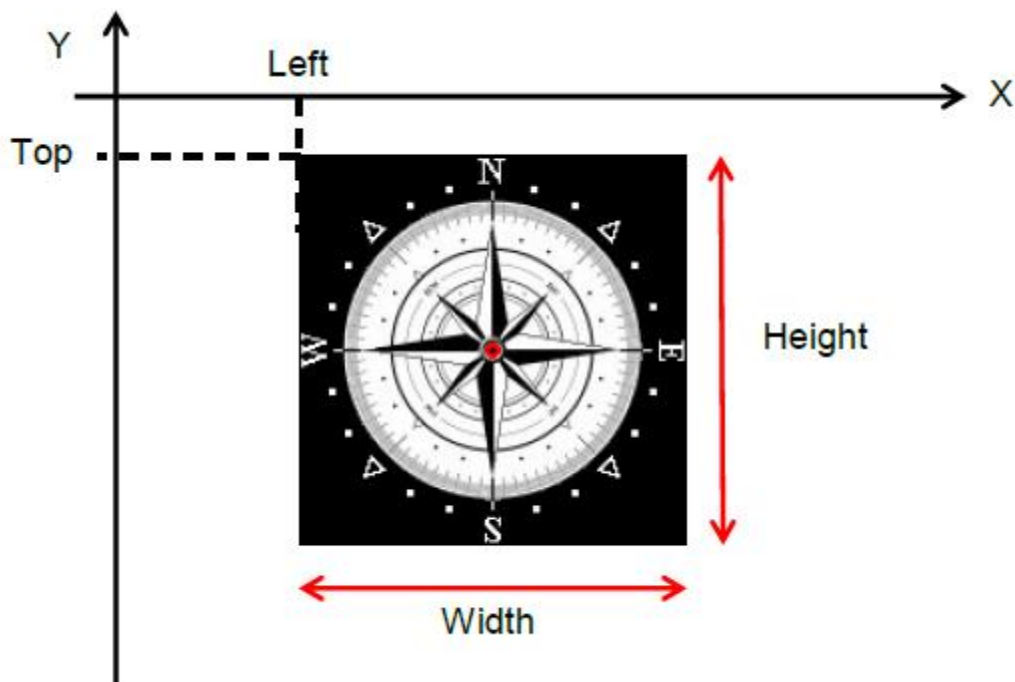
```
7 Sub Globals
8     'These global variables will be redeclared each time the activity is created.
9     'These variables can only be accessed from this module.
10    Dim cadran As ImageView
11    Dim droite As Button
12    Dim gauche As Button
13    Dim aiguille As BitmapData
```

- 10- Une fois créé, on lui associe un fichier image, dans notre cas le fichier **aiguille.png**. Ce fichier ayant été précédemment chargé dans le Designer, B4A l'a copié dans le répertoire de votre programme. Ainsi pour le charger on utilise le mot clef **DirAssets**.

```
18 Sub Activity_Create(FirstTime As Boolean)
19     'Do not forget to load the layout file created with the visual designer.
20     Activity.LoadLayout("Layout1")
21     cadran.SendToBack
22     aiguille.Bitmap = LoadBitmap(File.DirAssets, "aiguille.png")
```

Pour donner l'impression que l'aiguille tourne dans le cadran, il faut que nos deux images soient l'une sur l'autre, centrées en leurs milieux. Hélas, il n'existe pas de fonction permettant de retourner les coordonnées X et Y d'une image. Il va donc falloir les calculer.

Une image est dessinée de la façon suivante :



Le point en haut à gauche est le point d'ancrage et est défini par des coordonnées X (propriété **Left**) et Y (propriété **Top**). Puis on détermine la largeur de l'image (propriété **Width**) et sa hauteur (propriété **Height**). Aussi on peut calculer facilement les coordonnées X et Y du centre de l'image.

11-Créez deux variables **centerX** et **centerY** de type **Int**

```
12 Dim aiguille As BitmapData
13 Dim centerX, centerY As Int
14 End Sub
15
```

12-Procédez au calcul du centre de votre cadran

```
18 Sub Activity_Create(FirstTime As Boolean)
19     'Do not forget to load the layout file created with the visual designer.
20     Activity.LoadLayout("Layout1")
21     cadran.SendToBack
22     aiguille.Bitmap = LoadBitmap(File.DirAssets, "aiguille.png")
23     centerX=cadran.Left+(cadran.Width/2)
24     centerY=cadran.Top+(cadran.Height/2)
```

13-On va maintenant positionner l'objet aiguille (pour le moment il s'agit pour B4A d'une image) dans un rectangle virtuel dont on définira la position en X, en Y, sa largeur et sa hauteur en fonction du centre de notre cadran, afin que ce dernier et notre rectangle soient centrés par les milieux. L'image de l'aiguille occupera la globalité du rectangle.

Notre aiguille fait donc 10dpi de large ((centerX+5)-(centerX-5)) et 100dpi de haut.

```
18 Sub Activity_Create(FirstTime As Boolean)
19     'Do not forget to load the layout file created with the visual designer. For
20     Activity.LoadLayout("Layout1")
21     cadran.SendToBack
22     aiguille.Bitmap = LoadBitmap(File.DirAssets, "aiguille.png")
23     centerX=cadran.Left+(cadran.Width/2)
24     centerY=cadran.Top+(cadran.Height/2)
25     aiguille.DestRect.Initialize(centerX-5, centerY-50, centerX+5, centerY + 50)
```

Nous allons maintenant créer un objet de type **GameView**. Cet objet est une sorte de calque dynamique sur lequel on peut placer d'autres objets et qui vient se positionner par-dessus notre layout dessiné dans le Designer.

14-Créez un nouvel objet de type **GameView** que vous nommerez **gv**.

```
7 Sub Globals
8     'These global variables will be redeclared each time the activity is created.
9     'These variables can only be accessed from this module.
10    Dim cadran As ImageView
11    Dim droite As Button
12    Dim gauche As Button
13    Dim aiguille As BitmapData
14    Dim centerX, centerY As Int
15    Dim gv As GameView
```

15-Une fois l'objet créé il faut l'initialiser, lui associer des objets et le positionner sur notre écran (ici il occupera la globalité de la tablette).

```
18 Sub Activity_Create(FirstTime As Boolean)
19     'Do not forget to load the layout file created with the visual designer. For exan
20     Activity.LoadLayout("Layout1")
21     cadran.SendToBack
22     aiguille.Bitmap = LoadBitmap(File.DirAssets, "aiguille.png")
23     centerX=cadran.Left+(cadran.Width/2)
24     centerY=cadran.Top+(cadran.Height/2)
25     aiguille.DestRect.Initialize(centerX-5, centerY-50, centerX+5, centerY + 50)
26     gv.Initialize("gv")
27     gv.BitmapData.Add(aiguille)
28     Activity.AddView(gv, 0, 0, 100%x, 100%y)
```

Pour que l'aiguille tourne dans un sens ou dans l'autre on va utiliser la propriété Rotate de notre objet aiguille. En fait c'est le rectangle lui-même associé à l'objet aiguille que l'on tourne. Une fois la rotation réalisée, on réinitialise l'objet GameView qui se « redessine » avec notre rectangle, donc notre aiguille dans sa nouvelle position.

16-Complétez les lignes suivantes :

```
40
41 Sub gauche_click
42     aiguille.Rotate = (aiguille.Rotate -2) Mod 360 'rotate the aiguille 2 degrees
43     gv.Invalidate
44 End Sub
45 Sub droite_click
46     aiguille.Rotate = (aiguille.Rotate +2) Mod 360 'rotate the aiguille 2 degrees
47     gv.Invalidate
48 End Sub
```

17-Sauvegardez votre travail et testez-le. A chaque appui sur les boutons l'aiguille doit tourner d'un degré dans un sens ou dans l'autre.

### Quelques modifications

18-Modifiez votre programme pour que votre aiguille soit plus longue et testez-le.

19-Changer l'image du cadran par le fichier cadran3.jpg puis testez votre programme.

20-Vous devriez constater que l'aiguille est mal placée avec ce nouveau cadran. Modifiez votre programme pour remédier au problème précédent. Testez-le.

21-Empêchez votre aiguille de dépasser les limites de votre cadran.