

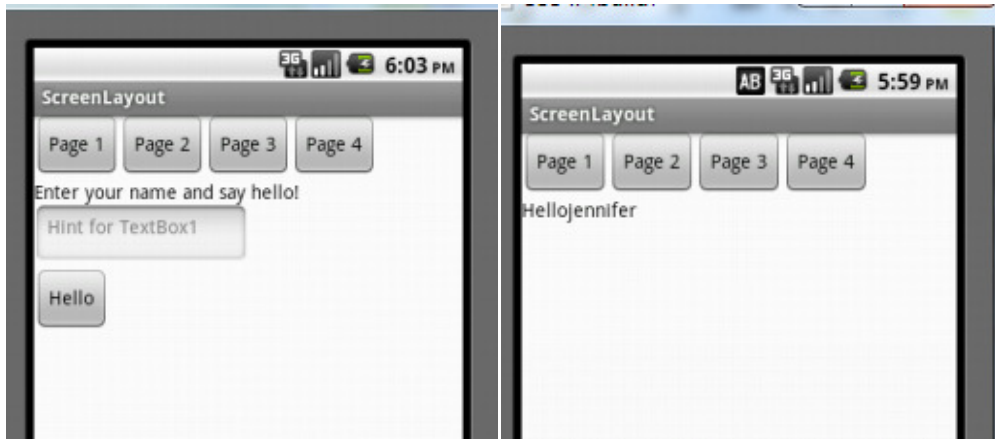
App Inventor Tutorial 7 – Understanding Screen Layouts

This is an app which will demonstrate the use of the screen layout components. These components will overcome any limitations caused by the 'add screen' capability in App Inventor.

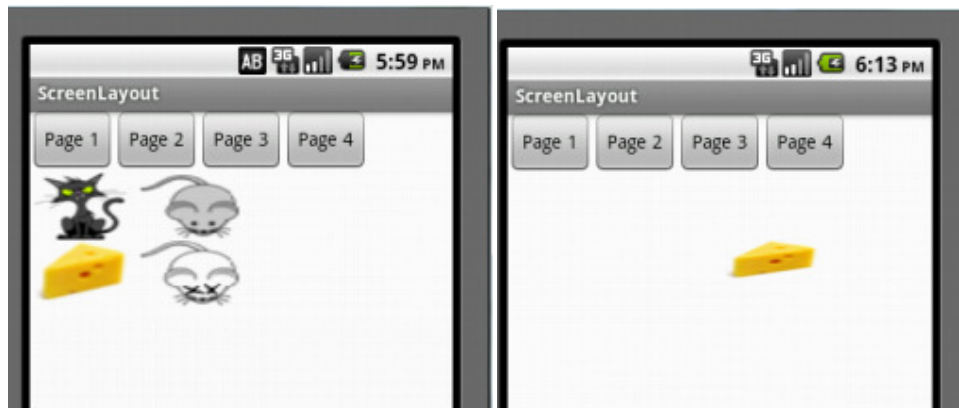
Currently – in the App Inventor stand alone package the use of multiple screens cannot be deployed on a device. It suggests that the hardware cannot support multiple screens in development mode. This is likely to be updated in future releases of App Inventor. One way to overcome this limitation at this stage is to use the layout objects, i.e. the Screen Arrangement components.

What you will develop is an app which can simulate multiple screens by showing and hiding the screen layout components. ☺

'Page 1' will contain the *HelloYou* app (Tutorial 1) with the output displaying on 'page 2'



'Page 3' will contain a 2 x 2 table arrangement with 4 buttons containing an image. 'Page 4' will load with the image from the pressed button moving around another table arrangement (2 x 3).



Step 1: Open App Inventor

Once you have the App Inventor running type into your browser <http://localhost:8888>

Like the previous tutorials, create a new project.

Give the project the name – ScreenLayouts and click OK.

Step 2: Build the Interface of Screen Layouts

The ScreenLayouts app needs many components. These include:

- A horizontal screen arrangement
 - 4 buttons
- 4 vertical screen arrangement
- HelloYou – (2 labels, a text field and a button)
- 2 table arrangements – (4 buttons and 6 canvas objects)

Let's start by naming our application.

In the properties panel change the Title field of the screen to ScreenLayout.

The first component we need is a Horizontal arrangement component. Drag this to our screen and change its properties as follows:

Width – Fill Parent

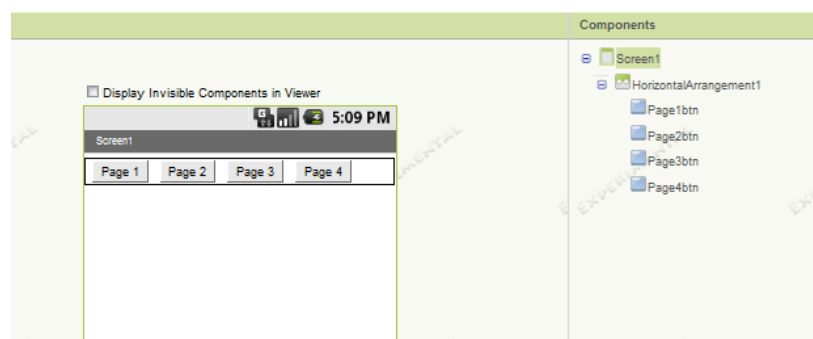
This horizontal component allows us to place items side by side on the screen horizontally. We will fill this component with 4 buttons. Drag over each button and change its properties as follows:

1st Button: Rename –Page1btn Text – Page 1

2nd Button: Rename – Page2btn Text – Page 2

3rd Button: Rename – Page3btn Text – Page 3

4th Button: Rename – Page4btn Text – Page 4



As we are going to simulate 4 different pages in this app, we will use the vertical arrangement component. This component allows us to arrange objects on the screen in a list horizontally – basically we can think of the vertical arrangement as a mini screen.

Add in 4 vertical arrangement components and change their properties as follows:

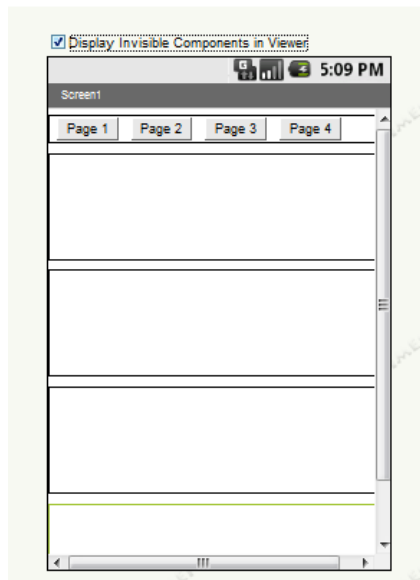
1st Vertical Arrangement: Rename – SLPage1 Width – Fill Parent

2nd Vertical Arrangement: Rename – SLPage2 Width- Fill Parent Visible – untick

3rd Vertical Arrangement: Rename – SLPage3 Width – Fill Parent Visible – untick

4th Vertical Arrangement: Rename – SLPage4 Width – Fill Parent Visible – untick

(We cannot see these components as we made them invisible – we can tick the ‘Display Invisible Components in Viewer’ check box to see our components)



The first page will contain the features from the ‘HelloYou’ app. Drag into ‘SLPage1’ component the following components:

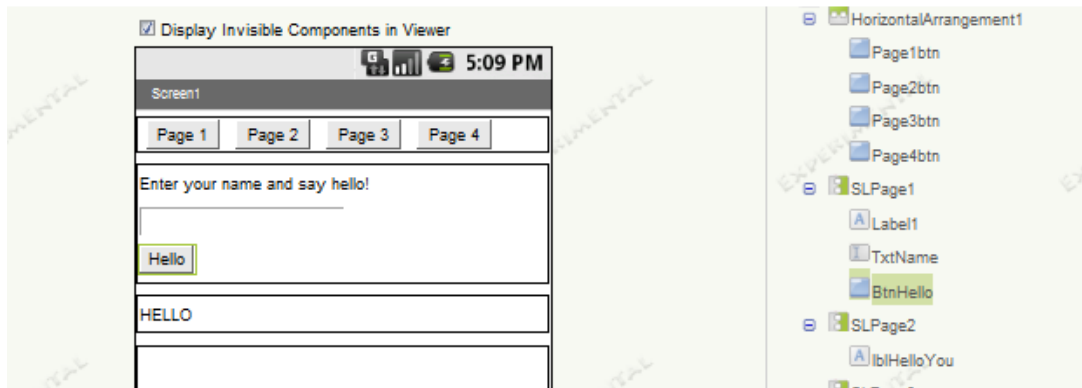
A Label: Text- Enter your name and say Hello!

A TextBox: Rename – TxtName

A Button: Rename – BtnHello Text - Hello

The final label from the ‘HelloYou’ app will be displayed on Page 2 therefore drag a label component into ‘SLPage2’.

A Label: Rename – lblHelloYou Text: HELLO



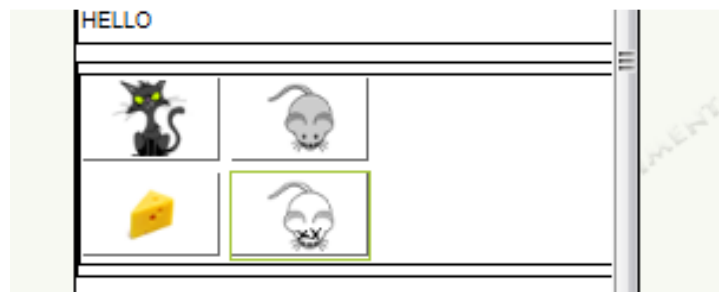
Page 3 will hold 4 images which are buttons. When a user clicks on a button the image will be displayed in Page 4. To display the 4 buttons, we can use a 2 x 2 table arrangement. Into 'SLPage3' drag over a Table Arrangement component.

This component has default values of 2 rows and 2 columns which results in 4 different cells.

Cell 1	Cell 2
Cell 3	Cell 4

In each cell place a button component and change its properties as follows:

Cell 1: Button	Rename – BtnCat Width- 80 pixels	Image – Cat.png Height – 50 pixels
Cell 2: Button	Rename – BtnXmouse Width- 80 pixels	Image – deadmouse.png Height – 50 pixels
Cell 3: Button	Rename – BtnCheese Width- 80 pixels	Image – cheese.png Height – 50 pixels
Cell 4: Button	Rename – BtnMouse Width- 80 pixels	Image – mouse.png Height – 50 pixels



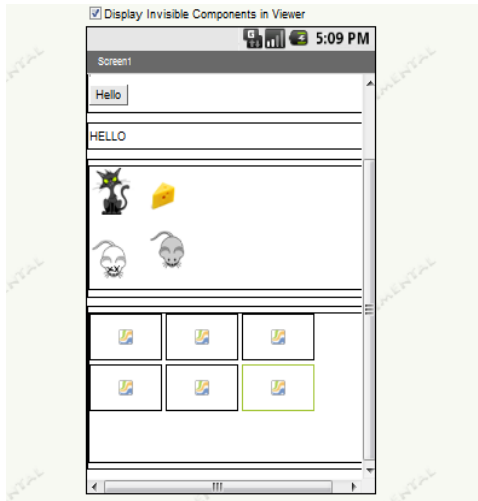
In 'SLPage4' we are going to display one of these 4 images. This image will move between cells of a table arrangement. In 'SLPage4' insert a Table arrangement component and change its properties as follows:

Rename- TableLayout Row - 2 Column - 3
 Width - 80 pixels Height - 50 pixels

This results in a table like this:

cell1	cell2	cell3
cell4	cell5	cell6

In each cell insert a canvas object - we will later change each canvas background based on the button the user selects in 'SLPage3'.



- Cell1: Canvas Rename-Cell1
- Cell2: Canvas Rename-Cell2
- Cell3: Canvas Rename-Cell3
- Cell4: Canvas Rename-Cell4
- Cell5: Canvas Rename-Cell5
- Cell6: Canvas Rename-Cell6

In order to move the image around these cells we can add a timer event. Drag over a Clock component to the screen. We can change the clocks interval property to 3000 which is 3 seconds (This means we can access the fired event of the timer object every 3 seconds) and untick its enabled property.

Your interface is now complete!

Step 3: Add the functionality to the Interface

Once the interface is ready we can set the functionality of the components. This app can be broken into 3 main parts as follows:

1. The Hello You app: demonstrating how to use multiple layouts imitating different pages. ('page 1' and 'page 2')
2. To show how to use the table layouts we can select an image and it will pass and move to the different cells on the 'next page'. ('page 3' and 'page 4')
3. To check for control over accessing 'different pages'

The functionality of the program (step 3) is added by using the Blocks Editor - Open the Blocks Editor to start adding code blocks.

This process may take a few minutes so please be patient ☺

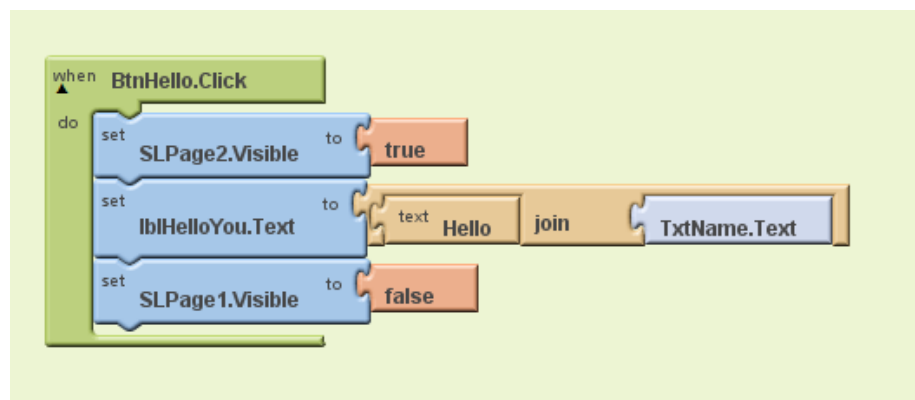
Once the Blocks editor has loaded we want to add the code blocks for:

1. **The Hello You app: demonstrating how to use multiple layouts imitating different pages.**

The Hello You app was documented earlier in Tutorial 1: Copy the following Code Blocks for the functionality of the 'HelloYou App'.



In addition to this we want the 'Page 2' layout to become visible and the current page 'Page1' layout to become invisible. We can therefore complete this block by adding the following blocks displayed below.



2. To show how to use the table layouts we can select an image and it will pass and move to the different cells on the 'next page'.

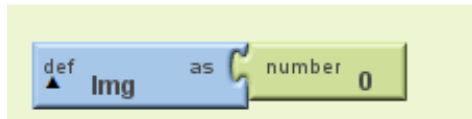
In 'page 3' we have 4 buttons; BtnCat, BtnMouse, BtnCheese and BtnXmouse.

For each of these buttons we want the same thing to happen.

1. We want to make 'page 4' visible – Set SLPage4.visible to true
2. We want to make the current page 'page 3' invisible – SetSLPage3.visible to false
3. We also want to start our Timer – SetClock1.TimerEnabled to true

As there is 4 buttons, we want to know which button was called to display the correct image. To do this we can use a global variable.

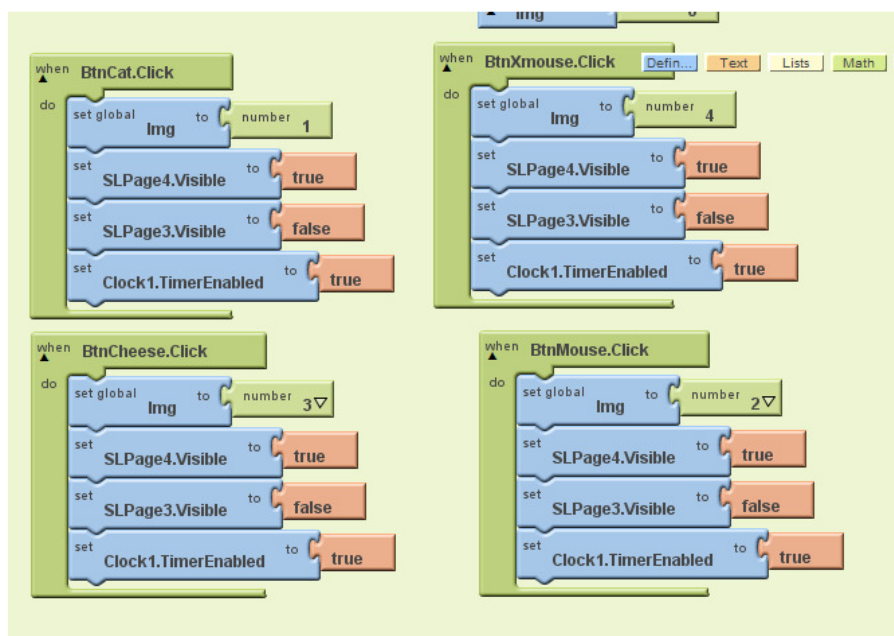
Drag a global variable definition to the canvas and name it 'Img' Initialise this with a number piece = 0.



For each button we will change the value of this variable.

- BtnCat → Img = 1 BtnMouse → Img = 2
BtnCheese → Img = 3 BtnXmouse → Img = 4

Complete the code blocks for the described steps above for each button click event. You should end up with the code blocks as outlined below.



From the steps above we can see the order of the cat (X) movements as follows:

Previous Cell	Current Cell	Next Cell
4	2	3
2	3	6
3	6	1
6	1	5
1	5	4
5	4	2

In order to add this functionality we must take each step above and logically work out an algorithm.

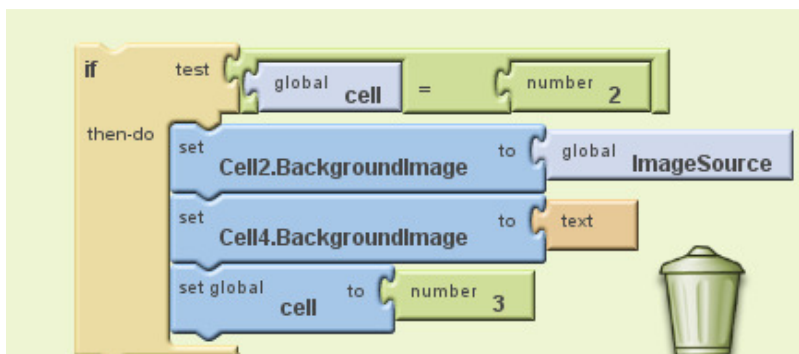
For each cell location we must make that cell background the image and set the previous cell background to null. We also must set the new cell variable to the next cell location.

We can say: For each of the 6 cells:

IF Cell = Current Cell THEN set previous cellBackgroundImage to Null AND set current cellBbackground Image to the Imagesource variable AND set the cell variable to the next cell value.

- Think about this logic for a moment!

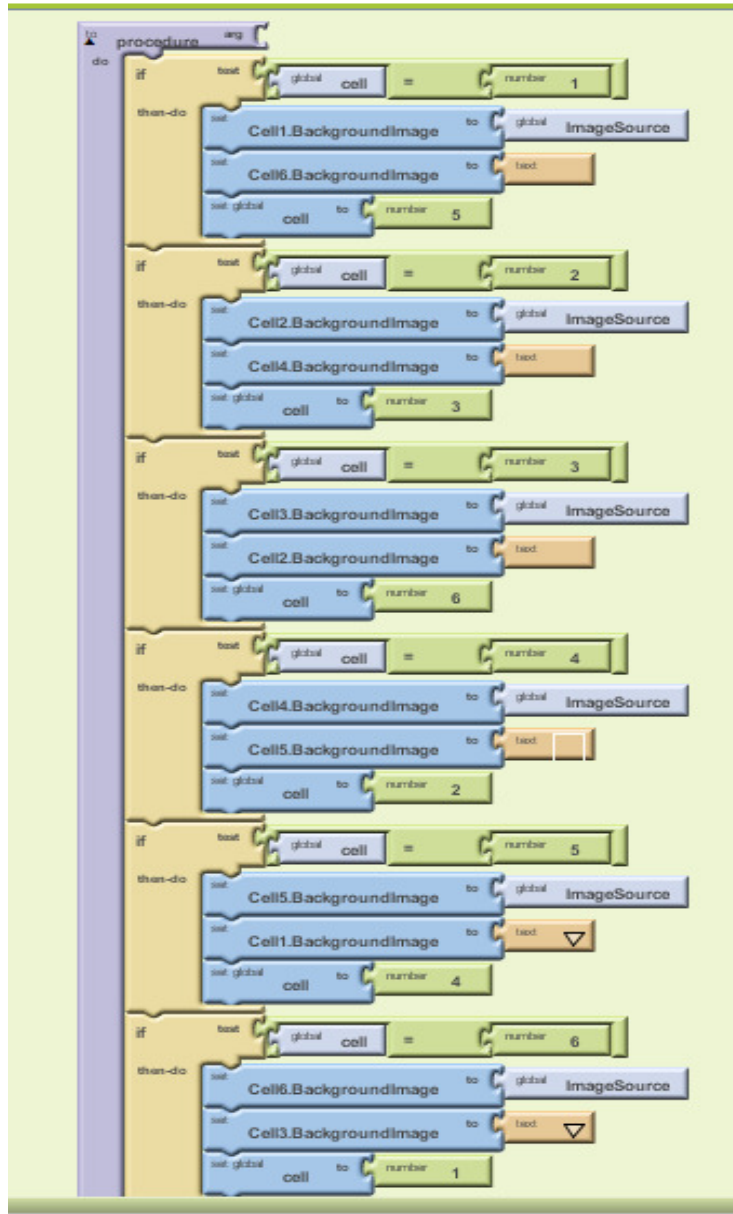
In the code blocks this can be built as follows:



To set a text object as Null – drag a text block and leave it empty.

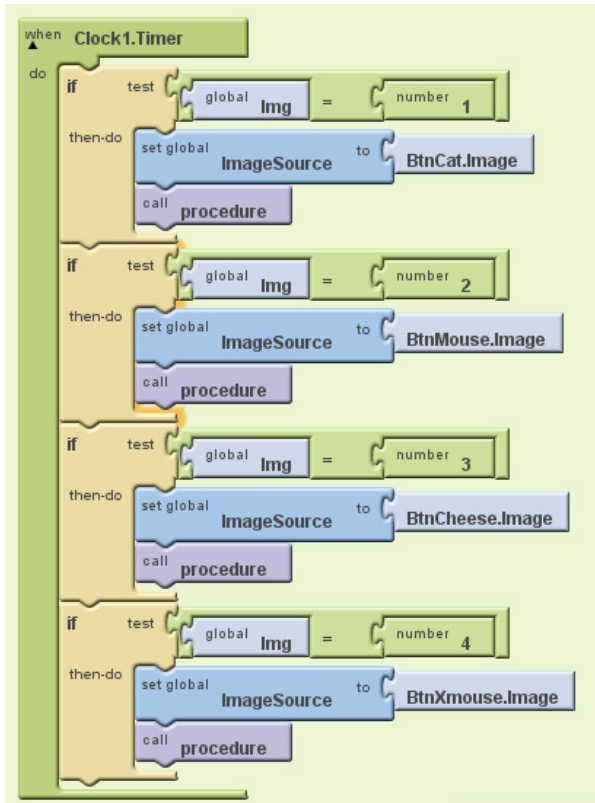
Repeat this IF/Then-do block for each cell in the horizontal arrangement, following the table outlined about.

These 6 IF/Then-do blocks can be added under a 'procedure' component as this will be called every 3 seconds.



As this is now a procedure, we can call this for our clock timer event. Drag onto the screen 'Clock1.Timer' – Any code we put in here will be called every 3 seconds (as defined in our interval property of the clock component)

When a button on Page 3 is clicked, we set a variable 'Img' to a value. Every 3 seconds we need to check which value this is and ensure the 'ImageSource' variable is set to the correct image. Looking at where we set this variable we can identify the code needed for each of the 4 buttons.



IF 'img' = 1 THEN set ImageSource to BtnCat.Image AND call our 'procedure'

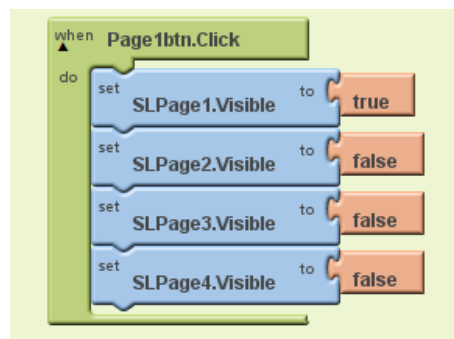
IF 'img' = 2 THEN set ImageSource to BtnMouse.Image AND call our 'procedure'

IF 'img' = 3 THEN set ImageSource to BtnCheese.Image AND call our 'procedure'

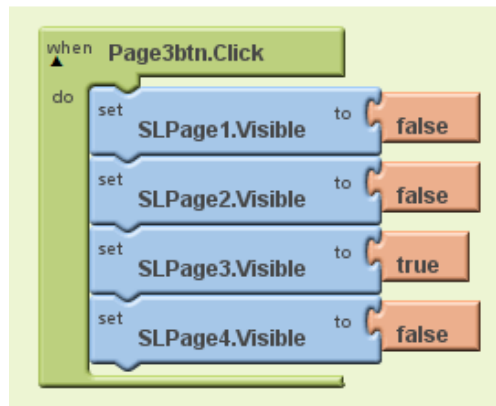
IF 'img' = 4 THEN set ImageSource to BtnXmouse.Image AND call our 'procedure'

3. To check for control over accessing 'different pages'

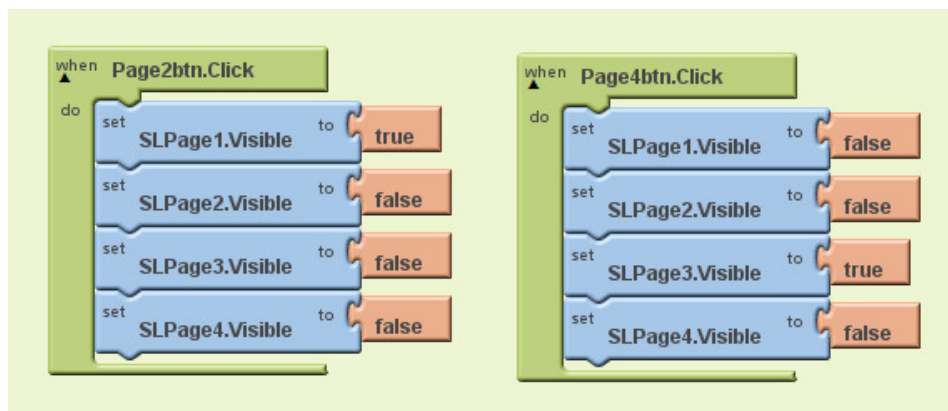
Along the top of our interface, we added 4 buttons – This simply shows the basic method to show/hide a page. We can add the code block for 'Page1btn' to simply show the Page 1 - 'SLPage1' and hide the others – 'SLPage2', 'SLPage3', 'SLPage4' as follows:



We can also do this for Page 3 – 'SLPage3' :



When we think of our page 2 'SLPage2' and page 4 'SLPage4' we do not directly want to access them. We only want to access page 2 after we access page 1, as we simulate the passing of information between the 'pages'. Similarly for page 4. For this reason we ensure 'Page2btn' to open 'SLPage 1' and 'Page4btn' to open 'SLPage3'



Step 4: Try it out - Connect to the device and test the program

Create a new emulator by clicking on the 'New emulator' button at the top of our blocks editor.

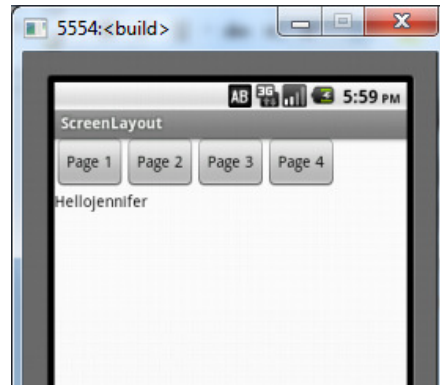
You must wait for the emulator to load fully before you test your app.

Once the emulator is ready - connect and send your app. Select the device from the device list. This process may take a few minutes. Please be Patient.



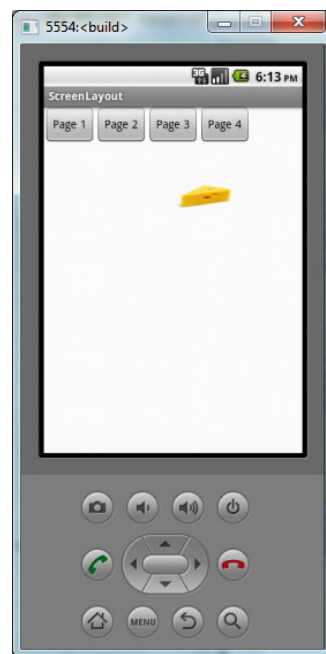
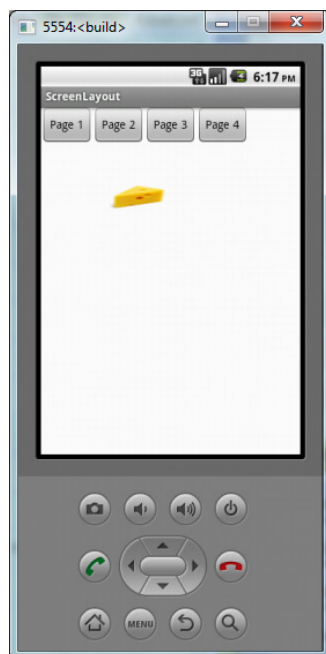
Your initial screen will load like the HelloYou App – with the 4 additional buttons.

If we enter our name in the text box – and click on Hello – the page will ‘disappear’ and ‘page 2’ will load with Hello (yourname)



Click on Page 2 button – youll notice you will return to the page 1.

Click on Page 3 button – click on any image button – this button will then load on next page and every 3 seconds change its cell location ☺



EndNote: This app has demonstrated the 3 different uses of the Screen Arrangement components – Using the table arrangement allows for component alignment like a grid. The horizontal arrangement allows components to sit side by side on the screen whilst the vertical arrangement allows the simulation of different pages. ☺ What else can you do with these arrangements . ~ Have Fun ~