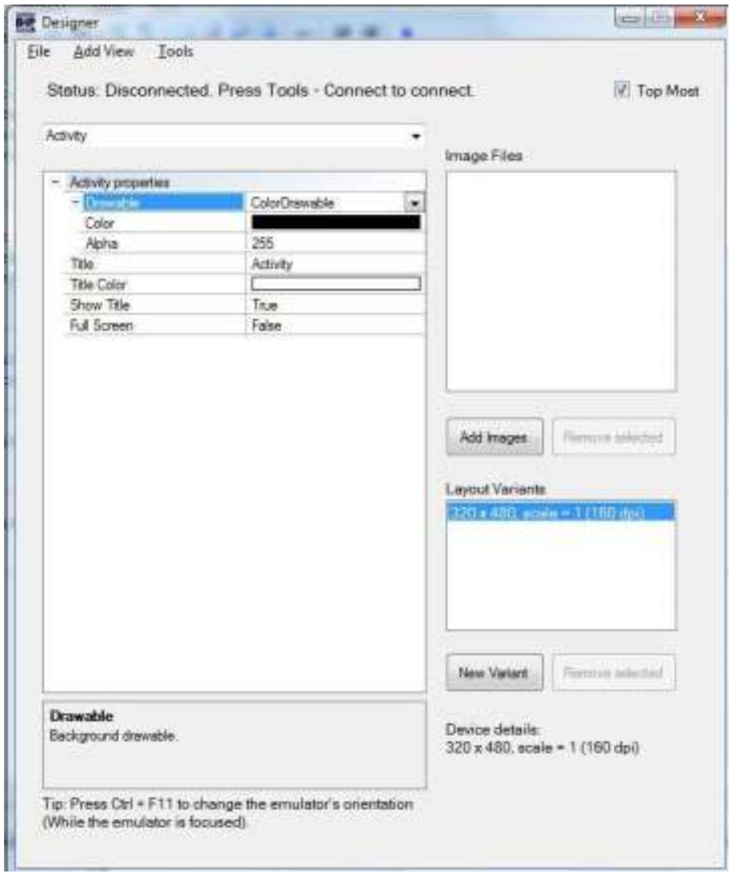
Guess my number - Visual designer & Events

In this tutorial we will use the designer to create the layout. The layout includes an EditText (TextBox) and a Button.
The user needs to guess a random number. The user enters the number in the EditText view (control) and presses on the button to submit the guess.
A "toast" message will appear, indicating to the user whether the number is larger or smaller than the chosen number.

- Create a new project and save it.
- Open the designer by choosing the Designer menu.

The designer is made of two main components. The "control panel" which contains all the available properties and options, and is part of the IDE:



Status: Disconnected. Press Tools - Connect to connect.

[✓] Top Most

Activity

Activity properties	
Drawable	ColorDrawable
Color	■
Alpha	255
Title	Activity
Title Color	□
Show Title	True
Full Screen	False

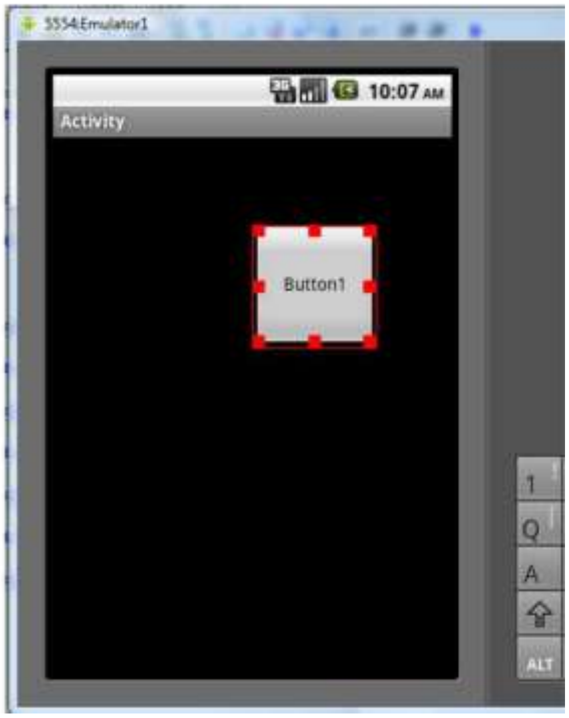
Drawable
Background drawable.

Tip: Press Ctrl + F11 to change the emulator's orientation
(While the emulator is focused).

320 x 480, scale = 1 (160 dpi)

Device details:
320 x 480, scale = 1 (160 dpi)

and the "visual" component which runs on a device or emulator:

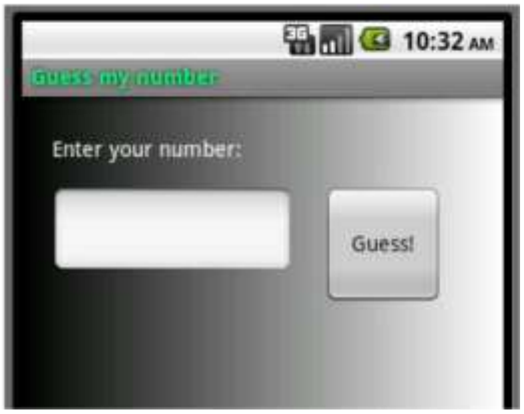


The visual component, as it names suggests, displays the layout. It also allows you to move and resize the views (controls).
Changing the layout in the visual component will also change the values stored in the control panel.

Note that all the data is stored in the control panel component. Therefore nothing bad will happen if the emulator crashes or is turned off.
You can connect to it again and the layout will appear.

The first step is to connect to the device. Press Tools - Connect.
This step takes several seconds. Note that the IDE will stay connected until the IDE closes. Closing the designer will not disconnect the connection.

Use the Add View menu to add a Button, an EditText and a Label.
Change the views Text property and position them similar to this:



Change the Activity Drawable property to GradientDrawable to achieve the gradient effect.

Tip: When working with a small monitor you may find it convenient to check the "Top Most" option (in the upper right corner). It will cause the control panel to stay on top and not be hidden by the emulator.

Save your layout, name it Layout1.

An **important concept** about layouts is that there is a complete separation between your code and the layouts.
The layout is saved as a file, with ".bal" extension. Each project can have any number of such files and unless you explicitly load a layout file, it will not have any effect on your application.

Once you have saved a layout, it is automatically added to the "File manager". You can see it under the "Files" tab in the IDE right pane.

We want to catch the button's click event.
Each view has an EventName value. It is a property in the Designer, and a parameter passed to the Initialize method when adding views programmatically.
In order to catch an event you should write a Sub with the following structure (it is simpler than it sounds):
Sub <EventName>_<Event> (event parameters).

In the designer, the EventName property is set by default to the view's name.
If we want to catch the Click event of a button with EventName value of Button1 we should write the following sub signature:
Sub Button1_Click

So here is the complete code:

Code:

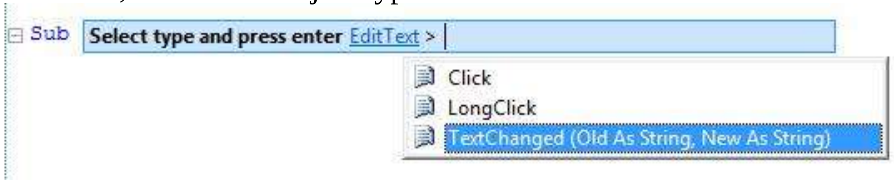
```
Sub Process_Globals

End Sub

Sub Globals
    Dim MyNumber As Int
    Dim EditText1 As EditText 'will hold a reference to the view added
by the designer
End Sub

Sub Activity_Create(FirstTime As Boolean)
    Activity.LoadLayout("Layout1") 'Load the layout file. MyNumber =
    Rnd(1, 100) 'Choose a random number between 1 to 99
End Sub

Sub Button1_Click
    If EditText1.Text > MyNumber Then
        ToastMessageShow("My number is smaller.", False)
    Else If EditText1.Text < MyNumber Then
        ToastMessageShow("My number is larger.", False)
    Else
        ToastMessageShow("Well done.", True)
    End If
    EditText1.SelectAll
End Sub
```

Some notes:

- Every activity module comes with an Activity object that you can use to access the activity.
- Activity.LoadLayout loads the layout file.
- There are other views that can load layout files. Like Panel and TabHost. For TabHost each tab page can be created by loading a layout file.
- In order to access a view that was added with the designer we need to declare it under Sub Globals.
- ToastMessageShow shows a short message that disappears after a short period. Using a toast message in this case is not optimal as it may be unnoticed when the soft keyboard is open.

Tip for writing events subs:

In the IDE, write Sub and press space. You should see a small tooltip saying "press tab to insert event declaration".
Press tab, choose the object type and choose the event.


Sub
Select type and press enter EditText >
Click
LongClick
TextChanged (Old As String, New As String)

Now all you need to do is to write the required event name and press enter.

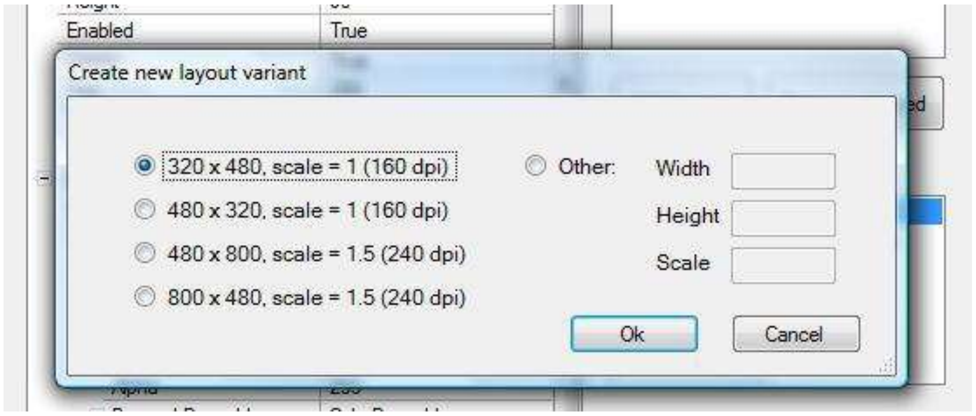
Supporting multiple screen resolutions and orientations

Each layout file can include a number of layout variants. Each layout variant holds a different set of values for the position and size of the views.
If for example, you change the text of any view it will be changed in all variants automatically. However if you change the position of a view it will only affect the current variant.
Note that scaling is handled automatically if required. Which means that if we run our program on a high resolution device, the layout will be automatically scaled. Still you may choose to create different variants for different scales.

When you load a layout file the variant that best matches the current device will be loaded.

Usually you will need two variants:
- 320 x 480, scale = 1 (160 dpi). This is the default scale in portrait mode.
- 480 x 320, scale = 1 (160 dpi). Default scale in landscape mode.

Ok, so open the designer again. Load Layout1 file if it is not opened.
Choose "New Variant" and choose 480 x 320 (second option).

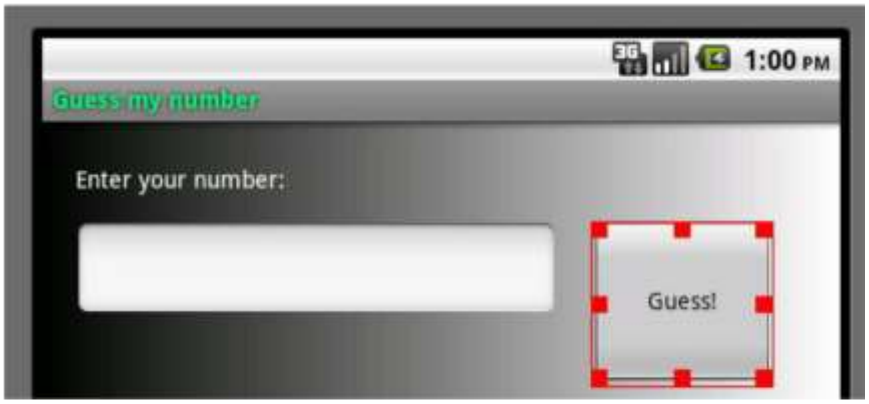


Create new layout variant

[✓ 320 x 480, scale = 1 (160 dpi)]
480 x 320, scale = 1 (160 dpi)
480 x 800, scale = 1.5 (240 dpi)
800 x 480, scale = 1.5 (240 dpi)
Other: Width Height Scale

Change the emulator orientation by clicking on the emulator and then press on Ctrl + F11.
Note that the device layout details appear under the list of variants.

Change the layout to be similar to this:



You can change the current selected variant and see how it affects the visual layout.
Save the layout and run the program.
Change the emulator orientation and see how the layout changes accordingly.

Android destroys the old activity and creates a new activity each time the orientation changes. Therefore Activity.LoadLayout will be called again each time. Unfortunately the number will also be randomly chosen again each time. This can be easily fixed... But not in this tutorial.

The project is attached.
Last tip for this tutorial:
- The IDE includes an "Export as zip" option under Files menu. This method creates a zip file with all the required files.