USER EXPERIENCE AND INTERACTIVE DESIGN FOR DEVELOPERS

THE JOY OF UX

David PLATT

The Joy of UX

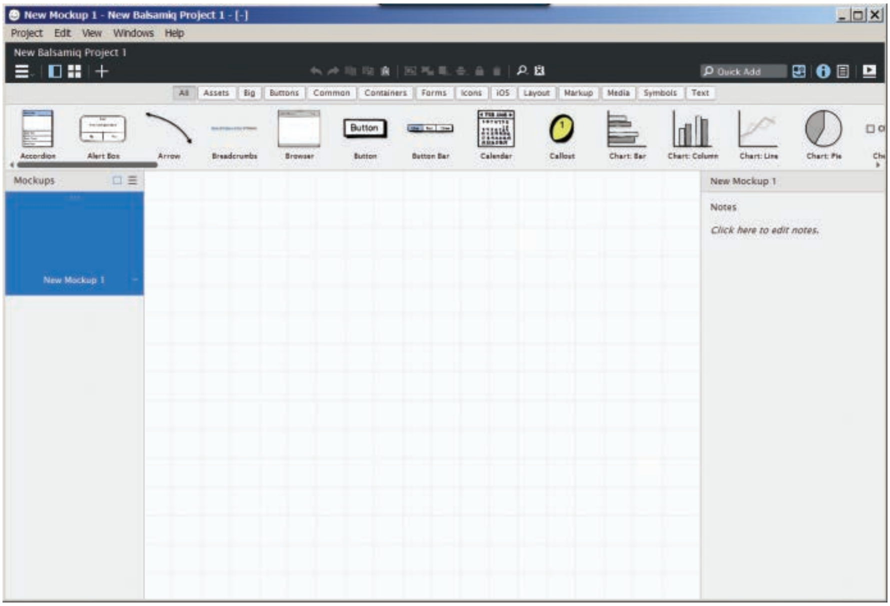
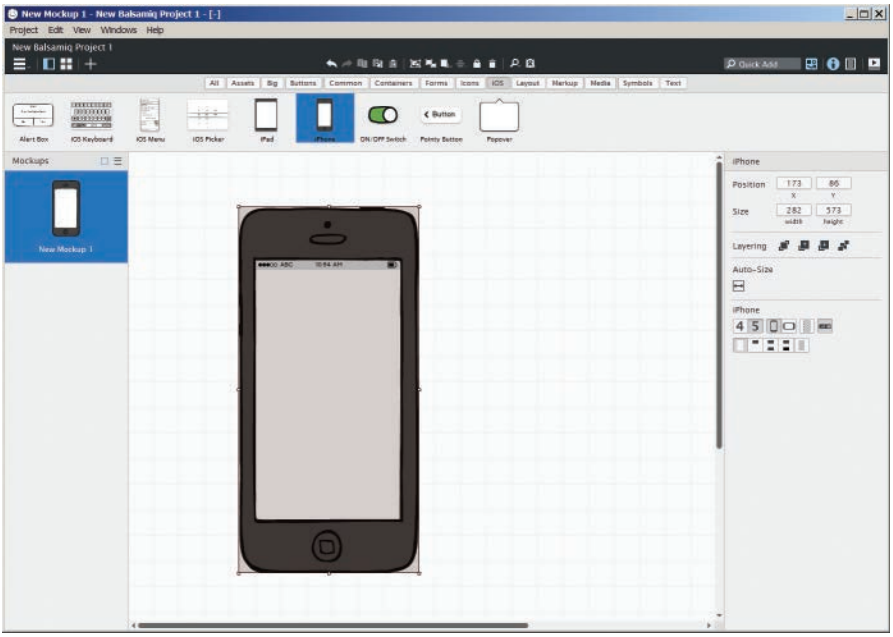

New Mockup 1 - New Balsamiq Project 1 - [-]
Project Edit View Windows Help
New Balsamiq Project 1
Quick Add
All Assets Big Buttons Common Containers Forms Icons iOS Layout Markup Media Symbols Text
Accordion
Alert Box
Arrow
Breadcrumbs
Browser
Button
Button Bar
Calendar
Callout
Chart: Bar
Chart: Column
Chart: Line
Chart: Pie
Mockups
New Mockup 1
New Mockup 1
Notes
Click here to edit notes.

Figure 3.4 Empty Balsamiq project, initial mockup.



Position 173 X 86 Y

Size 282 width 573 height

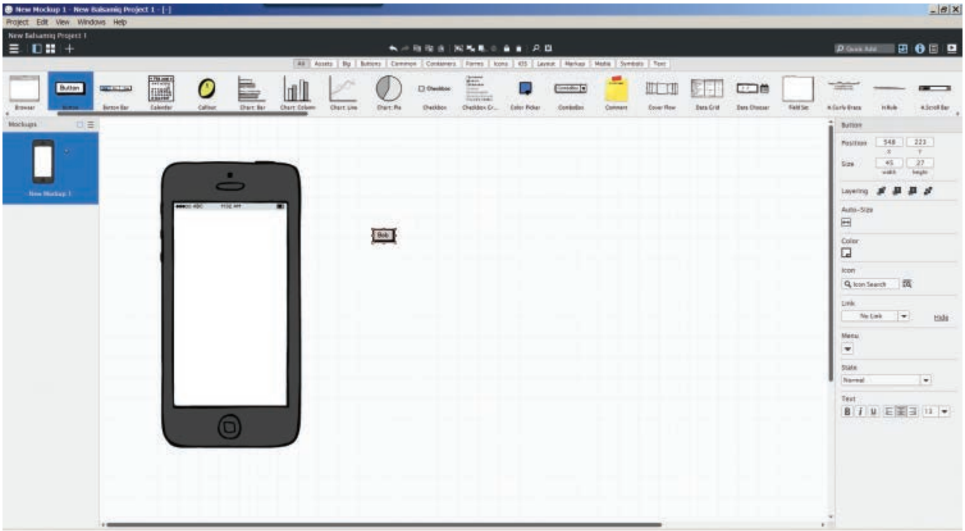
Figure 3.5 Balsamiq project with the iPhone outline.

Phones are almost always used in portrait orientation while talking, so we'll leave that setting intact as well. Again, remembering the design funnel we saw earlier in this chapter, we don't want to spend time on these details now.

The location and size of the phone are shown in PC screen pixels. The actual resolution of an iPhone 6 display is 1136 x 640 pixels, about double what Balsamiq is showing. The pixels on phones, particularly iPhones, are more densely packed than on most PC monitors. That lack of fidelity is precisely not the point here. We're not going to spend time matching pixel for pixel. We are looking to do a relatively large number of layouts, quickly and cheaply, so we can decide which ones we want to spend time developing further. We can zoom in (Ctrl +) for a closer look, or out (Ctrl -) for a wider one.

How should we start? What would the simplest design be? Well, this app is about frequently calling a small number of people, the same ones all the time. So let's plan from the very beginning to optimize this case. How about putting some large buttons on the app's home screen for quick access? Maybe with contact names in them?

Let's select the Button control from the UI library toolbar, drag it, and drop it on the screen. When we do this, the data entry editor for the button opens, so we can type in the text that we want it to display. In this case, it's "Bob," the employee we want the button to call. The right column now shows the properties of the new text area. The default is centered, which is what you expect from a button most of the time. The button's size is automatically set to the size of the text string (Figure 3.6).



Position 548 223
x y
Size 45 27
width height
Link: No Link
State: Normal
Text: 13

Figure 3.6 Balsamiq project with the iPhone outline and one button.

We now need to figure out the size and arrangement of buttons within the phone. Remember, we're talking about fast access to a small, tight group. We don't want to spend time scanning a long list. Carpenters work with their hands, so they tend to have larger fingers with less fine control. So how about something large, say, two columns, with maybe three or four buttons in each? Let's drag Bob's button into the iPhone layout and size it to take up about half the width. We'll adjust its height to where it looks and feels good to us. The text string will probably get small, so we'll use the properties window to adjust its font. Twenty-four point looks OK for now (Figure 3.7).

Now that we have one button in our iPhone layout, we want to add more buttons. To make them the same size as the first one, we'll just click on that button to select it, type Ctrl-C to copy, then Ctrl-V to paste. We repeat Ctrl-V two more times, giving us a total of four buttons. We drag the new ones to form a square arrangement in the mockup. Then we double-click each button to edit its text string. We can change "Bob" to "Pete," "Jimmy," and "Everyone." Oops, the legend "Everyone" doesn't quite fit this button size in 24-point font. What to do? Change its font size maybe? Or add a \n character into the text which splits it onto two lines? Change its name to "All"? We'll try one of those; if we don't like it, we'll try another. But above all, we won't spend much time fiddling with it. Now we have what's shown in Figure 3.8. The touch zones are large, and the button names are too. The buttons are easy to see and easy to tap, hard to screw up. Simple, as a walkie-talkie should be.

We now have one mockup to show to some potential users. Let's try another type of layout to see which they like better. Suppose we wanted to explore the possibility of using photos on our buttons instead of just the contact's name. That's a common feature on many cell phones

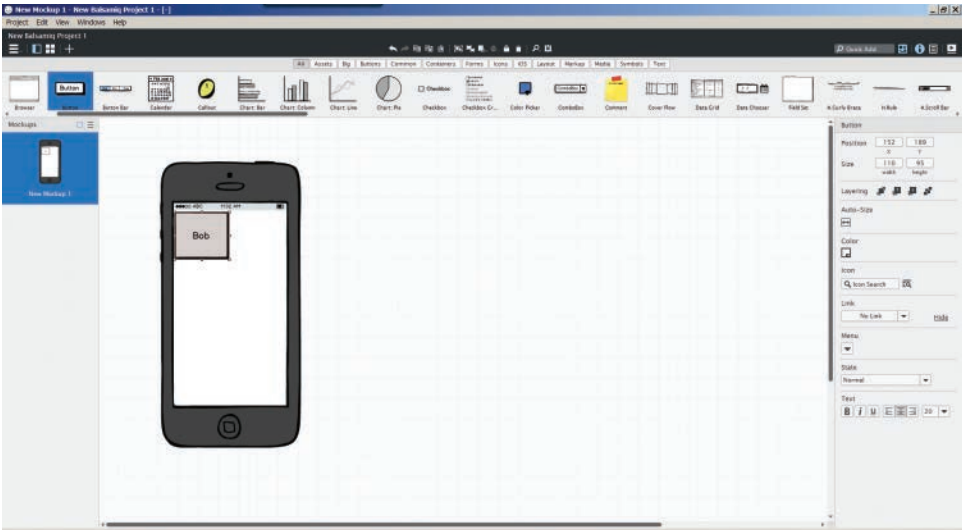

Button
Position 152 189
Size 110 95
Link: No Link
State: Normal
Text: 20

Figure 3.7 Balsamiq project with the iPhone outline and one button in the layout.

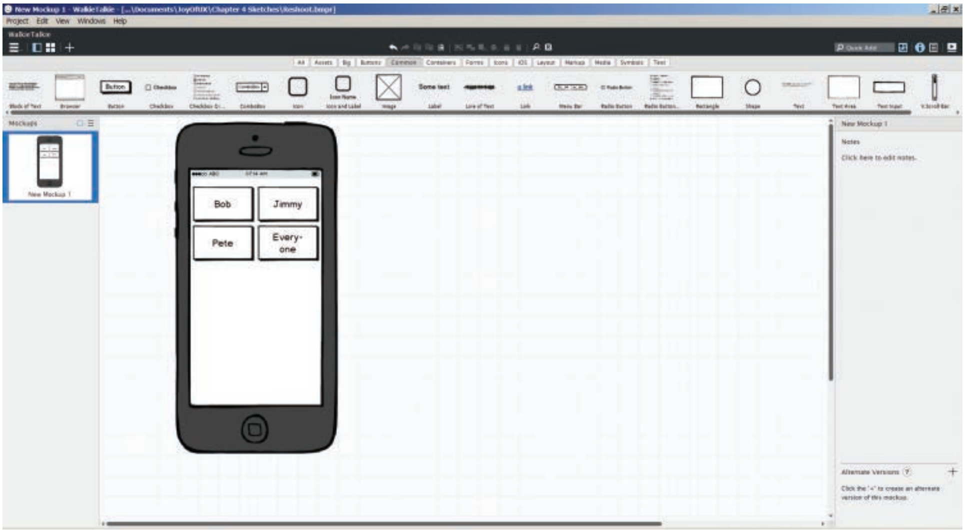

New Mockup 1 - WalkieTalkie - [...\Documents\JoyOfUX\Chapter 4 Sketches\Reshoot.bmpr]
Project Edit View Windows Help
WalkieTalkie
Quick Add
All Assets Big Buttons Common Containers Forms Icons iOS Layout Markup Media Symbols Text
Block of Text
Browser
Button
Checkbox
Checkbox Gr...
Combobox
Icon
Icon and Label
Image
Some text
Label
Line of Text
Link
Menu Bar
Radio Button
Radio Button...
Rectangle
Shape
Text
Text Area
Text Input
V.Scroll Bar
Mockups
New Mockup 1
Bob
Jimmy
Pete
Every-
one
New Mockup 1
Notes
Click here to edit notes.
Alternate Versions
Click the '+' to create an alternate version of this mockup.

Figure 3.8 Balsamiq project with the iPhone outline and multiple recipient buttons.

today. On the other hand, the reason these customers are demanding this app is a return to the simple old days. Which would they prefer, pictures or just text? I don't know. And I won't believe you if you say that you know. In fact, I don't think I'd believe the users if they told me without seeing it first. Fortunately, we can mock this up quite easily and show it to them.

If we right-click on our mockup in the Mockups list on the left side, we'll see a context menu with the choices of Rename and Duplicate. We'll click Duplicate, and it will make a copy of the existing mockup (Figure 3.9). You tend to get a lot of these cloned mockups because it's so easy to do. We might as well give it a name right now, at creation time. So a descriptive name is a good idea. However, not too much of the name shows in the list, so very long ones get clipped. Try to make mockup names distinctive in the first couple of dozen characters. We'll call this new one "Picture Buttons." (While we're at it, we might as well change the name of the first mockup, perhaps to "Text-Only Buttons," as shown.)

Now we start putting in buttons that could contain pictures. How do we get these? Balsamiq doesn't supply an image button on the control list. We could try pasting a picture onto a regular button, but Balsamiq doesn't allow us to change the vertical position of the text. We could drag a rectangle symbol into each button location, and a label, and a picture, four times or more. But that would get complicated, and we'd have to keep repositioning and resizing all of the sub-items, especially as we develop the idea further.

Fortunately, we can simulate an image button by making a *group*. This is a logical combination of several Balsamiq control symbols that we manipulate as a unit. Again, it's easier to show you one than to describe it. We select a rectangle control and drag it onto the design surface. We

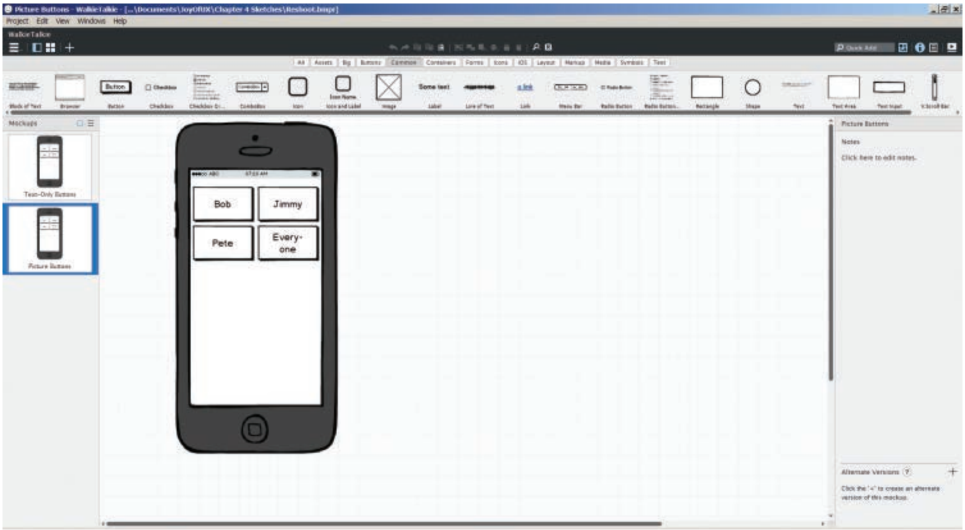

Picture Buttons - WalkieTalkie - [...\Documents\JoyOfUX\Chapter 4 Sketches\Reshoot.bmpr]
Project Edit View Windows Help
WalkieTalkie
All Assets Big Buttons Common Containers Forms Icons iOS Layout Markup Media Symbols Text
Button
Checkbox
Some text
a link
Block of Text Browser Button Checkbox Checkbox Gr... Combobox Icon Icon and Label Image Label Line of Text Link Menu Bar Radio Button Radio Button... Rectangle Shape Text Text Area Text Input V.Scroll Bar
Quick Add
Mockups
Text-Only Buttons
Picture Buttons
Bob
Jimmy
Pete
Every-
one
Picture Buttons
Notes
Click here to edit notes.
Alternate Versions
Click the '+' to create an alternate version of this mockup.

Figure 3.9 Balsamiq project with a cloned mockup to use for picture buttons.

adjust its size to be the same as that of the button controls we used previously. Now we're ready for the picture. We could use an image control, which can display a picture of any type. But it's easier to use the control called Webcam, which already contains a caricature of a person's face. We drag a webcam control into the rectangle and adjust its size. We still need a name, so we drag a label control into the box. We adjust its position underneath the webcam control and set its text to "Jimmy."

Now that we have these together, we want to tie them together so that we can manipulate them as a unit—drag and drop, position, size, and so on. That's quite easy. We select all three of these items—the rectangle, the label, and the webcam picture—in the standard Windows way, by holding down the Control key and clicking each item. When we have them all selected, we click the Group button on the toolbar, as shown in Figure 3.10.

Now, all three of these controls are bound together in a group. We can click just once to select the group, move it around, and position it wherever we want. The group sizes as a unit, more or less. We can copy the group as a unit and paste several groups into the new mockup, as we did before with the buttons—Ctrl-C, then Ctrl-V three times. We drag each group where we want it, more or less where the buttons are in the other mockup.

But wait. What about changing the text? We surely don't want four buttons to say "Jimmy," do we? Of course not (unless we *really* like Jimmy). We can edit the internal items of a group without ungrouping them. If we double-click on the group, we enter a mode in which we can change the contents of the group. If we then double-click on Label with the group, it opens up the text editor, so we can change the label's contents (Figure 3.11).

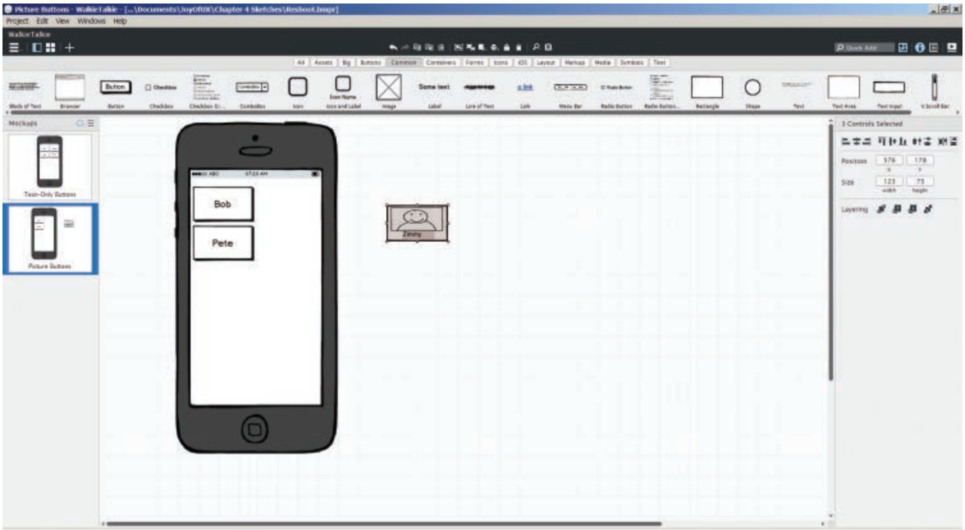

Picture Buttons - WalkieTalkie - [...\Documents\JoyOfUX\Chapter 4 Sketches\Reshoot.bmpr]
Project Edit View Windows Help
WalkieTalkie
All Assets Big Buttons Common Containers Forms Icons iOS Layout Markup Media Symbols Text
Button
Checkbox
Some text
a link
Mockups
Text-Only Buttons
Picture Buttons
Bob
Pete
Jimmy
3 Controls Selected
Position 576 176
Size 123 73
Layering

Figure 3.10 Balsamiq project with a custom button being grouped.

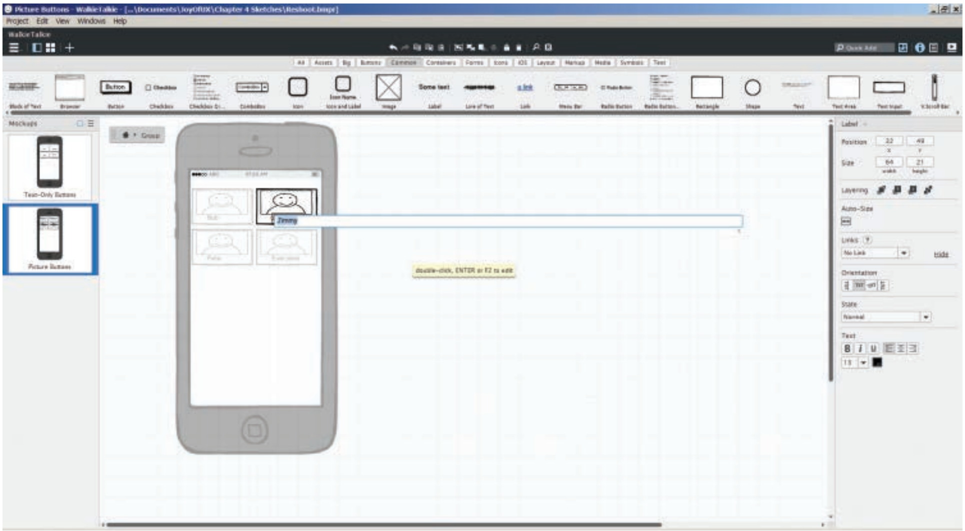


Figure 3.11 Balsamiq project, changing the label text inside a custom button group.

That's all we'll look at with layouts in Balsamiq. There is a whole tutorial on the Balsamiq Web site, exercising each part of the editor product. However, don't spend too much time on tutorials—just jump in, and consult the documentation only as you get stuck.