

Touch Feedback

Use color and illumination to respond to touches, reinforce the resulting behaviors of gestures, and indicate what actions are enabled and disabled.

Whenever a user touches an actionable area in your app, provide a visual response. This lets the user know which object was touched and that your app is "listening".



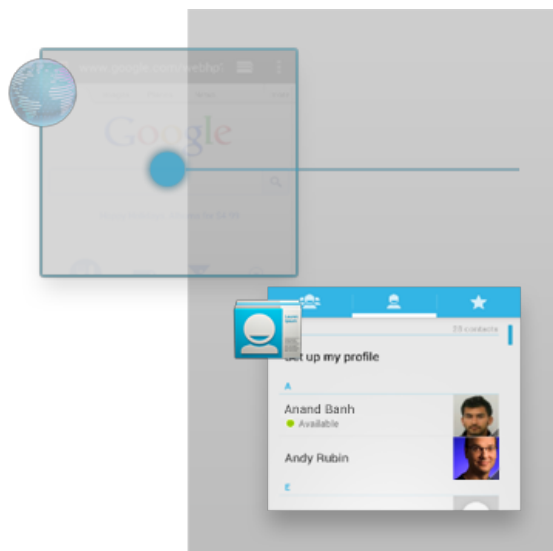
States

Normal	REMAINS STATIC
Pressed	ILLUMINATES WITH COLOR
Focused	DRAWS 50% OF THE PRESSED VALUE (OPTIONAL 2DP BORDER FILL AT PRESSED VALUE)
Disabled	DRAWS 30% OF THE NORMAL STATE
Disabled & focused	DRAWS 30% OF THE FOCUSED STATE

Most of Android's UI elements have touch-feedback built in, including states that indicate whether touching the element will have any effect.

Communication

When your objects react to more complex gestures, help users understand what the outcome of the operation will be. For example, in Recents, when you start swiping a thumbnail left or right, it starts to dim. This helps the user understand that swiping will cause the item to be removed.





Boundaries

When users try to scroll past the upper or lower limit of a scrollable area, communicate the boundary with a visual cue. For example, if a user attempts to scroll past the first home screen panel, the screen content tilts to the right to indicate that further navigation in this direction is not possible. Many of Android's scrollable UI widgets (e.g. lists or grid lists) already have support for boundary feedback built in. If you are building custom, keep boundary feedback in mind and provide it from within your app.