



Handling User Interaction Events

To make your new widget interactive, it will need to respond to user events like key presses, screen touches, and button clicks. Android exposes several virtual event handlers, listed below, that let you react to user input:

- `onKeyDown` Called when any device key is pressed; includes the D-pad, keyboard, hang-up, call, back, and camera buttons
- `onKeyUp` Called when a user releases a pressed key
- `onTrackballEvent` Called when the device's trackball is moved
- `onTouchEvent` Called when the touch screen is pressed or released, or it detects movement

The following code snippet shows a skeleton class that overrides each of the user interaction handlers in a View:

```
@Override
public boolean onKeyDown(int keyCode, KeyEvent keyEvent) {
// Return true if the event was handled.
return true;
}
@Override
public boolean onKeyUp(int keyCode, KeyEvent keyEvent) {
// Return true if the event was handled.
return true;
}
@Override
public boolean onTrackballEvent(MotionEvent event) {
// Get the type of action this event represents
int actionPerformed = event.getAction();
// Return true if the event was handled.
return true;
}
@Override
public boolean onTouchEvent(MotionEvent event) {
// Get the type of action this event represents
int actionPerformed = event.getAction();
// Return true if the event was handled.
return true;
}
```

Further details on using each of these event handlers, including greater detail on the parameters received by each method, are available in Chapter 11.