App Development

This section is for application developers adding Wacom tablet support to their Android application.

Android API Reference

Draw Simple sample code

Motion Dump sample code

Android API for Wacom Digitizers

Wacom Technology Corp.
<`developer@wacom.com`>
version 1.3, 2013

From integrated EMR sensors to USB- and Bluetooth-connected tablets, Wacom hardware is used with an ever-increasing number of Android devices. While Android was limited to only reporting basic stylus data in the past, there has been significant work in recent releases to improve the richness of the pen experience. This document provides an overview of the updates that have been made, and is intended to serve as a guide for app developers looking to make full use of the data made available to them.

Compatibility

The Android API was originally designed with touch input in mind. With recent releases, Google has worked to expand the API to be more accomodating of other input devices such as styli. Targeting apps for these more-recent releases can provide access to nearly the full compliment of available pen data (e. g. eraser, side-switches, tilt, etc.).

The following is a list of important milestones in the development of Android's input API. Available features should be weighed against the required degree of device support.

API Level 1 (Android 1.0)

Android API level 1 defines the base Android API. Input events are provided to apps through instances of the `android.view.MotionEvent` class. Android makes no distinction between the type of input device: touchscreen, stylus, and mouse input all produce `MotionEvent` objects that can be processed by an app.

Only basic tool information is available at this API level. High-resolution coordinates and pressure can be obtained, but little else. Multitouch is not yet supported, nor are interesting stylus attributes like button state and eraser distinction.

API Level 5 (Android 2.0: Éclair)

Android 2.0 introduced support for basic multitouch. Functions have been updated to accept a pointer index argument, which is used to access event data for contacts other than the first. It should be noted that Éclair only supports up to three simultaneous contacts, which may be fewer than the maximum supported by hardware.

While multitouch allows for tracking multiple contacts from a single device, tracking contacts between multiple devices (e.g. touchscreen and EMR) continues to work as it did in Android 1.0. Android may stop sending events from the touchscreen, and start sending new ones from the stylus instead. The switch can be detected by a change in device ID, though it is not possible until Gingerbread to determine the type of each device.

API Level 8 (Android 2.2: Froyo)

Froyo removes the three contact limit imposed by Éclair, allowing apps to track as many contacts as the hardware allows.

API Level 9 (Android 2.3: Gingerbread)

Gingerbread expands on the number of axes it can track for each `MotionEvent`, and introduces the `android.view.InputDevice` class to provide information about the devices themselves.

One of the new axes is "orientation", which measures the azimuthal angle of a stylus or finger. This is one-half of the tilt data measured by Wacom's Intuos line of tablets. The other "altitude" angle is not available except to those apps targeting Ice Cream Sandwich or newer.

The `InputDevice` class can be used to obtain data about the input devices responsible for each `MotionEvent`. Device names, axis ranges, and even (limited) type information is available. This class sees improvement in later API levels (e.g. the addition of `SOURCE_STYLUS` in Ice Cream Sandwich).

API Level 12 (Android 3.1: Honeycomb MR1)

Honeycomb was Google’s first release of Android to be targeted for tablet use. In version 3.1, Android gained the ability for apps to track a stylus even as it hovers above the screen. Apps can use this information to draw a cursor under the pen, display tooltips, or otherwise inform the user of what will occur should they tap the screen.

API Level 14 (Android 4.0: Ice Cream Sandwich)

Android 4.0 expands on the tablet support introduced in Honeycomb and provides access to a substantial fraction of the features provided by Wacom tablets.

The hover API was reworked somewhat in Ice Cream Sandwich. While Honeycomb would use `onGenericMotionEvent` to notify apps of a hovering stylus, ICS introduces `onHoverEvent` to take over this duty. Additionally, two new hover actions are introduced: `ACTION_HOVER_ENTER` and `ACTION_HOVER_EXIT`. These allow apps the ability to dynamically show and hide cursors as the pen enters and leaves proximity of the sensor.

Additional tool axes are available. In particular, the second half of the tilt data reported by Intuos tablets (altitude angle) can be read and used to determine the precise orientation of the pen in space. Distance data is also provided, giving apps a rough idea of how far away the stylus is from the screen surface.

Ice Cream Sandwich also provides button state information, including the state of the side-switches and the ability to distinguish between the stylus and eraser ends of a pen.

API Level 16 (Android 4.1: Jelly Bean)

Android 4.1 adds the ability for apps to obtain a stable device descriptor. This can be used as a key to store and retrieve device-specific settings.

Functionality

Event Listeners

The `View` class provides methods that allow you to register event listeners that will be notified when a `MotionEvent` occurs. While all versions of Android support touch event listeners, support for hover event listeners was introduced with Honeycomb.

PUBLIC STATIC INTERFACE VIEW.ONTOUCHLISTENER PUBLIC VOID VIEW.SETONTOUCHLISTENER (VIEW.ONTOUCHLISTENER L)Defines a callback that will be invoked when a touch event is sent to this View. Touch events are generated for active interactions: e.g. touching the touchscreen, or dragging the stylus/mouse.PUBLIC STATIC INTERFACE VIEW.ONGENERICMOTIONLISTENER PUBLIC VOID VIEW. ONGENERICMOTIONLISTENER (VIEW.ONONGENERICMOTIONLISTENER L)Defines a callback that will be invoked when a generic (non-touch) motion event is sent to this view. Hover events are no longer sent to generic motion listeners as of Ice Cream Sandwich.PUBLIC STATIC INTERFACE VIEW.ONHOVERLISTENER PUBLIC VOID VIEW.ONHOVERLISTENER (VIEW.ONHOVERLISTENER L)Defines a callback that will be invoked when a hover event is sent to this view.

Event Actions

The `MotionEvent` class defines constants which describe the kind of action that is occurring. As with the event listeners, hover actions are only available in Honeycomb and newer.

PUBLIC FINAL INT MOTIONEVENT.GETACTION () PUBLIC FINAL INT MOTIONEVENT.GETACTIONMASKED ()Return the kind of action being performed. The first of these three methods returns the action, combined with the pointer index which performed the action. The second method returns just he action for easier comparison with the constants below.PUBLIC STATIC FINAL INT ACTION_DOWN"A pressed gesture has started, the motion contains the initial starting location."PUBLIC STATIC FINAL INT ACTION_DOWN"A change has happened during a press gesture (between ACTION_DOWN and ACTION_UP. The motion contains the most recent point, as well as any intermediate points since the last down or move event."PUBLIC STATIC FINAL INT ACTION_UP"A pressed gesture has finished, the motion contains the final release location as well as any intermediate points since the last down or move event."PUBLIC STATIC FINAL INT ACTION_HOVER_MOVE"A change happened, but the pointer is not down (unlike ACTION_MOVE). The motion contains the most recent point, as well as any intermediate points since the last hover event."PUBLIC STATIC FINAL INT ACTION_HOVER_ENTER"The pointer is not down but has entered the boundaries of a window or view. This action is always delivered to the window or view under the pointer."PUBLIC STATIC FINAL INT ACTION_HOVER_EXIT"The pointer is not down but has exited the boundaries of a window or view.

Event Pointers

Android can track an arbitrary number of "pointers" (contacts/tools) from a single device, each with its own set of axis values and tool type information. Indicies

PUBLIC FINAL INT GETPOINTERCOUNT ()"The number of pointers of data contained in this event. Always >= 1."PUBLIC FINAL INT GETACTIONINDEX ()If an "interesting" action occurs (e.g. ACTION_POINTER_DOWN or ACTION_POINTER_UP), then this method will return the index of the pointer to pay attention to. For more mundane events (e.g. ACTION_MOVE) there is no need to call this method.PUBLIC FINAL INT GETPOINTERID (INT POINTERINDEX)Return the pointer identifier associated with a particular pointer data index. Unlike the pointer index, the pointer identifier is unique to the pointer throughout its life, and is safe to use for tracking purposes.

Types

PUBLIC STATIC FINAL INT MOTIONEVENT.GETTOOLTYPE (INT POINTERINDEX)"Gets the tool type of a pointer for the given pointer index. The tool type indicates the type of tool used to make contact, such as a finger or stylus, if known."PUBLIC STATIC FINAL INT MOTIONEVENT. TOOL_TYPE_STYLUS"The tool is a stylus."PUBLIC STATIC FINAL INT MOTIONEVENT.TOOL_TYPE_ERASER"The tool is an eraser or a stylus being used in an inverted posture."PUBLIC STATIC FINAL INT MOTIONEVENT.TOOL_TYPE_FINGER"The tool is a finger."PUBLIC STATIC FINAL INT MOTIONEVENT.TOOL_TYPE_UNKNOWN"This constant is used when the tool type is not known or is not relevant."

Event Axes

Android stores both the most-recent axis values reported by a tool, as well as a list of intermediate values that occurred since the last `MotionEvent` was delivered. The basic coordinate and pressure axes are supported by all Android versions, but other axes may depend on more recent releases.

There are generally several methods to obtain the data for any given axis. A bare method will provide the most-recent data for the first pointer index, while other variants may allow you to specify the index or age of data to be retrieved.
History

To minimize the processing overhead that would be incurred by immediately forwarding every input event from the kernel, Android buffers events for a short period of time before sending a single batched `MotionEvent`. For applications that demand smooth input, the intermediate input device states are available though various `getHistoricalFoo(int pos)` functions.

PUBLIC FINAL INT GETHISTORYSIZE ()"Returns the number of historical points in this event. These are movements that have occurred between this event and the previous event."PUBLIC FINAL LONG GETHISTORICALEVENTTIME (INT POS)"Returns the time that a historical movement occurred between this event and the previous event, in the uptimeMillis() time base."

Location

Android provides the location of events in screen units, with sub-pixel precision. If for some reason you find it necessary to convert a coordinate to device units, the functions `getXPrecision()` and `getYPrecision()` are also provided.

PUBLIC FINAL FLOAT MOTIONEVENT.GETX () PUBLIC FINAL FLOAT MOTIONEVENT.GETX (INT POINTERINDEX)PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALX (INT POS) PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALX (INT POS, INT POINTERINDEX) PUBLIC STATIC FINAL INT MOTIONEVENT.AXIS_X"X axis of a motion event."PUBLIC FINAL FLOAT MOTIONEVENT.GETY () PUBLIC FINAL FLOAT MOTIONEVENT.GETY (INT POINTERINDEX)PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALY (INT POS) PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALY (INT POS, INT POINTERINDEX)PUBLIC STATIC FINAL INT MOTIONEVENT.AXIS_Y"Y axis of a motion event."PUBLIC FINAL FLOAT GETRAWX ()"Returns the original raw X coordinate of this event. For touch events on the screen, this is the original location of the event on the screen, before it had been adjusted for the containing window and views."PUBLIC FINAL FLOAT GETRAWY ()"Returns the original raw Y coordinate of this event. For touch events on the screen, this is the original location of the event on the screen, before it had been adjusted for the containing window and views."

Force / Size

Devices can measure and report force through several different attributes. A stylus will pen pressure, while touchscreens may report contact size or major /minor axis lengths.

PUBLIC FINAL FLOAT MOTIONEVENT.GETPRESSURE () PUBLIC FINAL FLOAT MOTIONEVENT.GETPRESSURE (INT POINTERINDEX)PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALPRESSURE (INT POS)PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALPRESSURE (INT POS, INT POINTERINDEX) PUBLIC STATIC FINAL INT MOTIONEVENT.AXIS_PRESSUREObtain the pressure the contact is exerting on the screen. This value is normalized from the range 0.0 - 1.0, but may exceed these bounds depending on the input driver.PUBLIC FINAL FLOAT MOTIONEVENT. GETSIZE () PUBLIC FINAL FLOAT MOTIONEVENT.GETSIZE (INT POINTERINDEX)PUBLIC FINAL FLOAT MOTIONEVENT. GETHISTORICALSIZE (INT POS) PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALSIZE (INT POS, INT POINTERINDEX)PUBLIC STATIC FINAL INT MOTIONEVENT.AXIS_SIZEObtain the size of the contact, relative to the maximum size that can be sensed.PUBLIC FINAL FLOAT MOTIONEVENT.GETTOUCHMAJOR () PUBLIC FINAL FLOAT MOTIONEVENT.GETTOUCHMAJOR (INT POINTERINDEX)PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALTOUCHMAJOR (INT POS)PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALTOUCHMAJOR (INT POS, INT POINTERINDEX) PUBLIC STATIC FINAL INT MOTIONEVENT.AXIS_TOUCH_MAJOR"Returns the length of the major axis of an ellipse that describes the touch area at the point of contact."PUBLIC FINAL FLOAT MOTIONEVENT.GETTOUCHMINOR () PUBLIC FINAL FLOAT MOTIONEVENT. GETTOUCHMINOR (INT POINTERINDEX)PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALTOUCHMINOR (INT POS)PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALTOUCHMINOR (INT POS, INT POINTERINDEX) PUBLIC STATIC FINAL INT MOTIONEVENT. AXIS_TOUCH_MINOR"Returns the length of the minor axis of an ellipse that describes the touch area at the point of contact."PUBLIC FINAL FLOAT MOTIONEVENT.GETTOOLMAJOR () PUBLIC FINAL FLOAT MOTIONEVENT.GETTOOLMAJOR (INT POINTERINDEX)PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALTOOLMAJOR (INT POS)PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALTOOLMAJOR (INT POS, INT POINTERINDEX) PUBLIC STATIC FINAL INT MOTIONEVENT.AXIS_TOOL_MAJOR"Returns the length of the major axis of an ellipse that describes the size of the approaching tool. The tool area represents the estimated size of the finger or pen that is touching the device independent of its actual touch area at the point of contact."PUBLIC FINAL FLOAT MOTIONEVENT.GETTOOLMINOR () PUBLIC FINAL FLOAT MOTIONEVENT.GETTOOLMINOR (INT POINTERINDEX)PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALTOOLMINOR (INT POS)PUBLIC FINAL FLOAT MOTIONEVENT. GETHISTORICALTOOLMINOR (INT POS, INT POINTERINDEX) PUBLIC STATIC FINAL INT MOTIONEVENT.AXIS_TOOL_MINOR"Returns the length of the minor axis of an ellipse that describes the size of the approaching tool. The tool area represents the estimated size of the finger or pen that is touching the device independent of its actual touch area at the point of contact."

Orientation

Recent versions of Android are capable of reporting three other tool axes if supported by the hardware. These provide additional information about the 3D orientation of the tool in space.

PUBLIC FINAL FLOAT MOTIONEVENT.GETORIENTATION () PUBLIC FINAL FLOAT MOTIONEVENT.GETORIENTATION (INT POINTERINDEX) PUBLIC FINAL FLOAT MOTIONEVENT.GETHISTORICALORIENTATION (INT POS)PUBLIC FINAL FLOAT MOTIONEVENT. GETHISTORICALORIENTATION (INT POS, INT POINTERINDEX) PUBLIC STATIC FINAL INT MOTIONEVENT.AXIS_ORIENTATIONAzimuth angle of contact, relative to the top of the screen.PUBLIC STATIC FINAL INT MOTIONEVENT.AXIS_TILTAltitude angle of contact, relative to the screen’s perpendicular.PUBLIC STATIC FINAL INT MOTIONEVENT.AXIS_DISTANCEDistance between the tool and the screen.

Event Buttons

If a motion event comes from a stylus or mouse, there may be button state information available. Unlike a tool axis, button state is not available per-pointer, and no historical data is provided.

PUBLIC FINAL INT MOTIONEVENT.GETBUTTONSTATE ()"Gets the state of all buttons that are pressed such as a mouse or stylus button." Historic button state information is not available.PUBLIC STATIC FINAL INT MOTIONEVENT.BUTTON_PRIMARY"Primary button (left mouse button). This button constant is not set in response to simple touches with a finger or stylus tip. The user must actually push a button."PUBLIC STATIC FINAL INT MOTIONEVENT.BUTTON_SECONDARY"Secondary button (right mouse button, stylus first button)."PUBLIC STATIC FINAL INT MOTIONEVENT.BUTTON_TERTIARY"Tertiary button (middle mouse button, stylus second button)."

Device Information

Android provides apps with metadata about the devices and tools that are used to create motion events. This metadata includes contact IDs, tool types, and descriptions of the source itself. Data specific to a particular contact is typically hosted in the `MotionEvent` itself, while information about the underlying device is available through the `InputDevice` class.

Devices

Android tracks every input device that is attached the the system and assigns each an ID number that can be used both to distinguish motion events that come from different devices and to obtain information about the responsible device.

PUBLIC FINAL INT MOTIONEVENT.GETDEVICEID ()"Gets the id for the device that this event came from. An id of zero indicates that the event didn’t come from a physical device and maps to the default keymap. The other numbers are arbitrary and you shouldn’t depend on the values."PUBLIC STATIC INPUTDEVICE INPUTDEVICE.GETDEVICE (INT ID)Returns the InputDevice associated with the specified id.PUBLIC STATIC INT[] INPUTDEVICE.GETDEVICEIDS ()"Gets the ids of all input devices in the system."PUBLIC STRING INPUTDEVICE.GETNAME ()"Gets the name of this input device." PUBLIC STRING INPUTDEVICE.GETDESCRIPTOR ()"Gets the input device descriptor which is a stable identifier for an input device." This descriptor can be used to safely store settings associated with an input device across application launches.

Event Sources

Each motion event may come tagged with information about the "source" of the event. This provides additional device type information that can be used to e.g. separate touchscreen events from stylus events.

PUBLIC FINAL INT MOTIONEVENT.GETSOURCE ()"Gets the source of the event."PUBLIC INT INPUTDEVICE.GETSOURCES ()"Gets the input sources supported by this input device as a combined bitfield." Note that a single input device may have multiple sources, so a simple equality test with the constants below is insufficient.PUBLIC STATIC FINAL INT INPUTDEVICE.SOURCE_TOUCHSCREEN"The input source is a touch screen pointing device."PUBLIC STATIC FINAL INT INPUTDEVICE.SOURCE_STYLUS"The input device is a stylus pointing device."PUBLIC STATIC FINAL INT INPUTDEVICE.SOURCE_UNKNOWN"The input source is unknown."

Misc. Properties

Google provides a limited set of additional generic axes that can be used to relay other information to apps. These axes may contain arbitrary data, so it is important to positively identify the source device before attempting to interpret their values. For Wacom hardware, this can be done by verifying the device name begins with "Wacom".

PUBLIC STATIC FINAL INT MOTIONEVENT.AXIS_GENERIC_1Serial number of the stylus in use. The returned value is the float value corresponding to the bit pattern of the original 32-bit integer. Before making use of this serial number, it is important to recover the original value by passing it through Float.floatToRawIntBits.PUBLIC STATIC FINAL INT MOTIONEVENT.AXIS_GENERIC_2Tool type of the stylus in use. The returned value is the float value corresponding to the bit pattern of the original 32-bit integer. Before making use of this identifier, it is important to recover the original value by passing it through Float.floatToRawIntBits.The values may be interpreted as follows (see the Linux kernel driver’s wacom_wac.c file for an exhaustive list):
Tool Stylus ID Eraser IDGrip Pen(value & 0xfffff) == 0x60802(value & 0xfffff) == 0x6080a(value & 0xfffff) == 0x00802(value & 0xfffff) == 0x0080aArt Pen (value & 0xfffff) == 0x00804(value & 0xfffff) == 0x0080cClassic Pen(value & 0xfffff) == 0x40802(value & 0xfffff) == 0x4080aAirbrush Pen(value & 0xfffff) == 0x00902(value & 0xfffff) == 0x0090aInking Pen(value & 0xfffff) == 0x20802N/APUBLIC STATIC FINAL INT MOTIONEVENT.AXIS_GENERIC_3This value measures the rotational angle of the Art Pen about its long axis. As with the orientation property, the value is measured from -PI to +PI radians. A measurement of zero indicates the side-switches are facing to the left.PUBLIC STATIC FINAL INT MOTIONEVENT.AXIS_GENERIC_4This value measures the current position of the fingerwheel for the Airbrush Pen. As with the pressure property, the value is measured from 0.0 to 1.0.

Version 1.3
Last updated 2013-03-28 09:20:09 PDT