UPDD API - Examples

Revision 1.5 – 31st Dec 2014
www.touch-base.com\documentation\API

Contact

This document links to a number of useful examples and will be expanded as and when more are made available.

It also highlights other API examples created internally or by our customers.

Downloadable examples

Drawing application, multi-monitor, multi-stylus	Example GUI drawing program with source code has been created to demonstrate the UPDD API interface.
Console mode program	Example non GUI console program with source code has been created to demonstrate the UPDD API interface.
C# VS2005	C# example program with source code
USB Macro Utility	Sending USB requests to a UPDD supported USB device
QT touch demos (all – 75mb)	QT development environment example
QtTouchPaintDemo (25mb)	**QtTouchPaintDemo**
QtTUIOPaintDemo (25mb)	**QtTUIOPaintDemo**
QtUPDDPaintDemo (25mb)	**QtUPDDPaintDemo**

Demonstrates how to build a Qt application that allows the user to paint in a window using both the mouse and touches from a touch-pad or touch-screen, using Qt's touch events. These events receive touch input from OS calls, such as WM_touch in Windows and NSTouch in Mac OS X. In this application a simple widget is implemented called "QPaintWidget", whose behavior is when the user clicks in the window or generates a touch, a trail of color will follow, with a label identifying which mouse button is pressed or the index of a touch. The touches are received as QTouchEvents in the widget. More up-to-date modules using QT 5.x are available for Linux and Mac.

QtTUIOPaintDemo

Demonstrates how to build a Qt application that allows the user to paint in a window using both the mouse and touches from a TUIO server. The "QPaintWidget" is used again in this demo, except this time it has outward facing functions for receiving touches from any source. This demo shows how to set up a TUIO client using the demonstration TUIO client implementation available on the TUIO website, and then pass the received touches to the QPaintWidget. A 64 bit Linux version is available here (to be used in conjuction with our 64 TUIO server). App only is here: Mac OS X

QtUPDDPaintDemo

Demonstrates how to build a Qt application that allows the user to paint in a window using both the mouse and touches received via the UPDD API. The same implementation of "QPaintWidget" is used as before in QtTUIOPaintDemo, except this time touches are received from a callback registered with the UPDD and then passed on to the widget for drawing.

Under Windows there are 32 bit and 64 bit versions. The 32bit version runs with UPDD 4.1.10 (any) and 5.0.x (32 bit). The 64 bit version runs with UPDD 5.0.x (64bit).

A version with touch logging is available here (the mac version needs UPDD 5.1.x):

Mac OS X Linux Windows 32 Windows 64

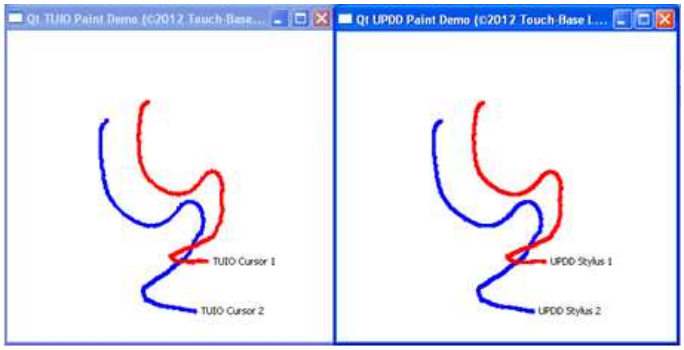
Gesture calculation

The latest demo programs make use of an in-house written gesture analyzer to calculate the gesture being performed on the touchscreen and list this in the demo screen. To prevent the analysis function being used outside our demo programs it only functions for 5 minutes at which point is stops working. Both QtUPDDPaintDemo and QtTUIOPaintDemo use the TBGestureAnalyzer static library.

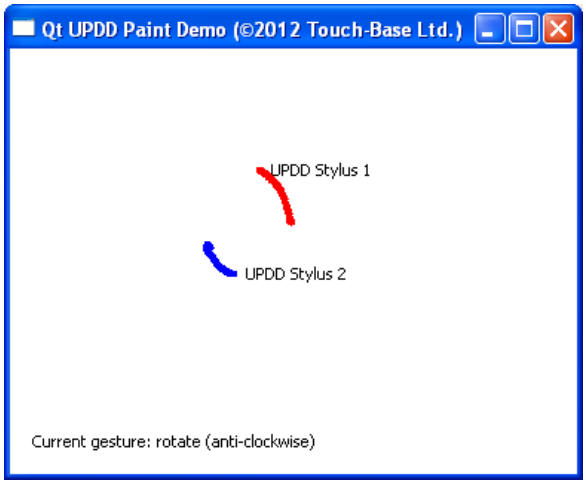
QtTouchPaintDemo uses Qt's own gesture recognition for gestures. However, it seems that this doesn't work consistently between platforms. In Windows 7 (not XP as Qt uses the Windows 7 multitouch API), it will only detect "pinch", "pan", and "tap" gestures in our touch screen tests. In OS X, it can detect all gestures that Qt supports, but different ones will be detected depending on if used with an Apple magic trackpad or a touch screen with UPDDGestures running. Reading online, other developers are seeing the same behavior, so we consider this an issue with Qt. We haven't been able to get it to work at all in Linux so far when testing under Ubuntu 11.10).

Examples

In this example, run under Windows XP, the UPDD-TUIO-Bridge software is running to offer a TUIO server. Both the UPDD and TUIO applications have been invoked and placed side-by-side. A dual touch screen has been touched. TUIO app is drawing via TUIO events and UPDD is drawing via the UPDD API Callback function.



In this example, with built in gesture analysis, the rotate anti clockwise movement of the dual touch is calculated and listed:



Notes

Executables are available for Mac, Linux and Windows

Windows and Mac apps statically linked with required libraries, Linux dynamically linked. UPDDPaintDemo should be run in UPDD folder so as to access correct versions of tbapi.dll and ACE_UPDD_n.n.n.DLL.

Windows may require you to install the Microsoft Visual C++ Redistributable Package. Applications will report that a required .DLL is missing if the run-time is not installed.

UPDD Demonstration program	UPDD Demo – Source code of our UPDD demonstration program

Other examples

Real-Time annotation	We wanted to write a desktop annotation utility (Annotate) that could be used at the same time as the system mouse. Therefore we directly accessed the pointer device co-ordinate information via the UPDD API rather than be driven by mouse emulation. The application places a transparent drawing layer over the entire desktop such that when the UPDD controlled device is in use, such as a touch screen or electronic whiteboard, then real-time desktop annotation can be performed, leaving the system mouse to be used as normal
Controller communication	A space saving system was utilized inside emergency vehicles. The touch screen controller performed many system functions and processed much more than just the touch co-ordinates. UPDD processed the touch co-ordinates as expected but also passed through commands, in both directions, between the controller and the system control program. The system control program, using the API, was able to receive and send non-touch data via the touch controller.
UPDD Toolbar handling	UPDD allows toolbars to be defined on the pointer device. A toolbar is an area of the pointer device that is calibrated separately from, or on top of, the normal 'mouse emulation' calibrated area. When the stylus enters a toolbar area notice is given to any external applications using the UPDD callback API that this has occurred. This is a very powerful feature that allows external functions to be triggered by the pointer device. UPDD toolbars are often used in touch surround type applications.
UPDD functionality and settings	Utilities are supplied to change driver and controller settings and also to perform such actions as calibration. It may be required that your own application be responsible for updating UPDD settings or performing calibration. In this instance the application will need to utilise the UPDD API calls. One such application is used on an electronic Whiteboard. One element of this application is to switch between two different calibrations, one calibration used when the whiteboard is being used

for drawing purposes (calibrated in the corners) and the other calibration is used when the desktop is projected onto the whiteboard (calibration to the limits of the projected area). The API is used to switch between the two calibrations.

Contact

For further information or technical assistance please email the technical support team at technical@touch-base.com.