

UserInputTouch

General

This plug-in channel extends the Quest3D user interface to support multitouch activities in Win 7. This allows following many individual independent touches or touch movements. The complementary plug-in UserInputGesture detects more kinds of single gestures.

In brief, UserInputTouch is providing independent move gestures for a set of (different) fingers, and UserInputGesture is for detecting a set of different gestures.

Children

1. TouchEvent: OUT X,Y position in local window space and Flag (vector), as growing list ...

Description

UserInputTouch tracks up to 20 (later versions 32) independent touch events.

A call to this channel provides a list of all active touches. The return value indicated the number of active touches in progress. The individual touch events are mapped to the connected children vectors. A typical application would read the return value and then process the affected children. Each child vector delivers the position in local window space and a flag. For many purposes it's sufficient to test the flag for not active (0), and active (>0). For more details, see appendix A.

A negative return value of -1 indicates that no appropriate touch device was found, -2 tells that trial version limits are reached. In summary, this value channel returns with:

-2	limitations reached (trial versions only)
-1	touch device not available
0	idle
>0	number of touch events received

Tips

- Use only one UserInputTouch or UserInputGesture channel. They cannot operate simultaneously.
- An active touch sequence requires movement or just a touch.
- Touches are not limited to 2D. In order to manipulate an object in a 3D scene, it has to be identified first with a pick operation. Any further processing is similar to well established ways of mapping screen to object/world space.
- Make yourself familiar with the touch and feel of interactions by checking other Win7 gesture enabled applications first. Slow down in the beginning and don't forget to lift up the fingers.
- Look into the Debug window for detailed reports.
- Known problems: Quest3D editor may crash while saving or publishing a cgr, if a Preview Window is open. Remedy: close all Preview beforehand.
- Detection of touch events works only in published projects, not in the editor.

Legal note:

Permission granted for evaluation and educational purposes only (trial version), commercial use requires an explicit license agreement.

Volume, runtime duration, and expiration restrictions are enforced with trial licenses. Details can be found by reading the output of the Debug Window, and on my web page. Be prepared to cope with this situation in your application.

Contact: quest3d.godbersen.eu

Supported Quest3D Versions

- 4.3.2
- 5.0 (x64), 32 on request only

Tutorial:

See the annotated [Multi-Touch-Demo.cgr](#), and [Touch3dpicking.cgr](#)

A screenshot of the Debug window:

- Indicates all applicable restrictions
- Informs about the actual **device properties**

```
Previous project cleared, opened new proje
>>>> Trial license restrictions apply
- Edit access allowed through: 7 2011
- Volume restriction : 100
- Edittime restriction (min.): 10
- Runtime restriction (min.): 5
>>>> Read touch device properties:
NID_INTEGRATED_TOUCH
NID_MULTI_INPUT
NID_READY
>>>> Touch device found, flags = 193
```

Appendix A

Flags correspond to TOUCHINPUT data structure. <http://msdn.microsoft.com/de-de/library/windows/desktop/dd317334%28v=vs.85%29.aspx>

Flag	Value Hex / decimal	Description
TOUCHEVENTF_MOVE	0x0001 1	Movement has occurred. Cannot be combined with TOUCHEVENTF_DOWN.
TOUCHEVENTF_DOWN	0x0002 2	The corresponding touch point was established through a new contact. Cannot be combined with TOUCHEVENTF_MOVE or TOUCHEVENTF_UP.
TOUCHEVENTF_UP	0x0004 4	A touch point was removed.
TOUCHEVENTF_INRANGE	0x0008 8	A touch point is in range. This flag is used to enable touch hover support on compatible hardware. Applications that do not want support for hover can ignore this flag.
TOUCHEVENTF_PRIMARY	0x0010 16	Indicates that this TOUCHINPUT structure corresponds to a primary contact point. See the following text for more information on primary touch points.
TOUCHEVENTF_NOCOALESCE	0x0020 32	When received using GetTouchInputInfo, this input was not coalesced.
TOUCHEVENTF_PALM	0x0080 128	The touch event came from the user's palm.

Example: (all values in decimal)

Value received	Flag list:
57	1 TOUCHEVENTF_MOVE 16 TOUCHEVENTF_PRIMARY 32 TOUCHEVENTF_NOCOALESCE
41	1 TOUCHEVENTF_MOVE 8 TOUCHEVENTF_INRANGE 32 TOUCHEVENTF_NOCOALESCE

More advice:

- Extract an individual flag:
 - Use ExpressionValue, e.g. $(A\&2)>0$ to find a DOWN flag set
 - Save result immediately, as flag settings are updated at full framerate
- Follow each finger track:
 - Keep track of DOWN flags,
 - The upcoming upgrade will provide an additional Growing Child Value TouchID (value)*