

Doomsday Engine - Feature #1229

Input plugins: generate events from connected controllers

2003-07-09 03:04 - skyjake

Status: Rejected	Start date: 2003-07-09
Priority: High	% Done: 0%
Assignee:	
Category: Redesign	
Target version:	
Description	
Much like the audio plugins, input devices should be abstracted behind a plugin mechanism. An input plugin would feed input events into the engine's input subsystem from any controllers specific to / configured in the plugin.	
One input plugin should be able to generate any kind of input events, for instance mouse, joystick, and keyboard events.	
See related old proposal: Input drivers	
Related issues:	
Related to Feature #1541: Joystick axis settings (sensitivity, max speed)	Progressed 2011-06-28
Related to Bug #778: Gamepad usability issues	Progressed 2007-06-07
Related to Feature #1636: Support for Oculus Rift	Closed 2013-10-23
Related to Feature #1741: Upgrade to SDL 2	Closed 2014-03-27
Related to Feature #1758: Separate left/right modifier keys (Shift, Alt, etc.)	New 2014-04-18
Related to Feature #1759: Plugins 2.0	Rejected 2014-04-18
Related to Feature #1886: Use SDL 2 for window management, display modes, col...	In Progress 2014-10-20

History

#1 - 2003-07-10 03:16 - skyjake

(originally posted by anonymous SF.net user)

Logged In: NO

Proper support for analog. Instead of holding down run, have the stick determine your speed.

#2 - 2003-07-16 05:31 - skyjake

(originally posted by anonymous SF.net user)

Logged In: NO

Ability to drag and drop controllers to be used by players 1-4 when splitscreen arrives.

#3 - 2013-10-20 21:06 - skyjake

- Tags set to Input, Plugin

- Category set to Redesign

- Priority changed from Normal to High

There is a related proposal: [Input drivers](#)

#4 - 2013-10-20 21:07 - skyjake

- Tags changed from Input, Plugin to Input, Plugin, UI

#5 - 2013-10-22 11:32 - skyjake

- Description updated

#6 - 2014-04-18 07:54 - skyjake

- Subject changed from Controller plugins to Controller plugins (input drivers)

#7 - 2014-10-20 15:15 - skyjake

- Target version set to 42

#8 - 2014-10-20 15:17 - skyjake

- Related to Feature #1886: Use SDL 2 for window management, display modes, color correction, and keyboard/mouse/gamepad input added

#9 - 2014-10-20 15:20 - skyjake

- Subject changed from Controller plugins (input drivers) to Input plugins: generate events from connected controllers

#10 - 2014-10-20 15:22 - skyjake

- Description updated

#11 - 2014-10-20 15:22 - skyjake

- Description updated

#12 - 2015-04-22 05:23 - danij

- Assignee set to danij

#13 - 2015-05-03 17:03 - skyjake

- Target version changed from 42 to 2.0 – Home UI & Packages

#14 - 2016-03-27 08:15 - skyjake

- Target version changed from 2.0 – Home UI & Packages to Input and game controllers

#15 - 2017-07-11 19:13 - skyjake

- Status changed from New to Rejected

- Assignee deleted (danij)

- Target version deleted (Input and game controllers)

If SDL 2 is used for input (see [#1886](#)), having input plugins doesn't really provide much additional value.

#16 - 2017-09-07 16:29 - danij

There are other motivations for at least a virtualized abstraction. For example, while for the most-part we will be communicating with APIs that will all largely follow the same device / control / input abstractions there are countless uncommon controllers used for all kinds of purposes for various accessibility reasons.

Originally input customization features have been limited to console scripts, for the purposes of automating the generation of event sequences. Nowadays things are handled a bit differently, with bindable modifiers et al. Also, on game side we have a special case mechanism for handling cheat codes using a high priority event responder.

Architecturally it would be nice to deal with both issues. Perhaps we could leverage Doomsday Script here?