
pyzmo documentation

Release 0.2.0

Georgi Valkov

January 15, 2014

1 Usage (Python)	3
2 Usage (Hy)	5
3 Installing	7
4 Similar Projects	9
5 License	11

Pyzmo is a small hotkey library for Python **2.7+** based on [python-evdev](#). It can call arbitrary Python functions in response to key presses, key chords (simultaneously pressed keys) and key sequences (keys pressed in succession).

Usage (Python)

```
from pyzmo import *
from evdev.ecodes import *

# triggered when KEY_PLAYPAUSE is pressed
@key(KEY_PLAYPAUSE)
def playpause(events):
    pass

# triggered when KEY_B is pressed, held or released
@key(KEY_B, states=['down', 'hold', 'up'])
def back(events):
    pass

# triggered when either KEY_C or KEY_D are pressed
@key(KEY_C, KEY_D)
def c_or_d(events):
    pass

# triggered when KEY_LEFTCTRL, KEY_LEFTALT and KEY_DELETE are
# pressed at the same time
@chord(KEY_LEFTCTRL, KEY_LEFTALT, KEY_DELETE)
def ctrlaltdel(events):
    pass

# triggered when KEY_A, KEY_B and KEY_C are pressed one after the other
@keyseq(KEY_A, KEY_B, KEY_C)
def abc(events):
    pass

# specifying multiple sequences for one callback (syntax applies for
# @chord and @event as well)
@keyseq( (KEY_1, KEY_2, KEY_3), (KEY_Z, KEY_X, KEY_C) )
def zxc_or_123(events):
    pass

# each handler receives the list of input events, because of
# which it was triggered
@chord(e.KEY_LEFTMETA, e.KEY_A)
def term(events):
    for event in events:
        print(event) # instance of evdev.events.InputEvent
        #event at 1352244701.749908, code 125, type 01, val 01
        #event at 1352244701.861897, code 30, type 01, val 01
```

```
## Note 1:
# If we define two triggers:
# - @chord(KEY_LEFTCTRL, KEY_V)
# - @key(KEY_V)
#
# Pressing 'ctrl-v' will run both their callbacks. To stop
# processing any further triggers after a match is made, use:
@chord(KEY_LEFTCTRL, KEY_V, quick=True)
def copy(events):
    pass

## Note 2:
# Pyzmo can actually match arbitrary input events. The following
# will be triggered on scroll-wheel movement:
@event(EV_REL, REL_WHEEL, -1)
def vertical_scroll(events):
    pass

# start main loop
poll('/dev/input/event1', '/dev/input/event2')

# .. or if you wish to get exclusive access to a input device
from evdev import InputDevice
dev = InputDevice('/dev/input/eventX')
dev.grab()
poll(dev)
```

You can also avoid polluting the global namespace with:

```
from pyzmo import EventHandler
from evdev import ecodes as e

app = EventHandler('name')

@app.key(e.KEY_F)
def f(events): pass

@app.poll(...)
```

Usage (Hy)

Pyzmo also has a DSL that can be used from **Hy**, which is a dialect of lisp that is embedded in Python.

```
(require pyzmo.dsl)
(import pyzmo [pyzmo [poll util]])

# triggered when KEY_PLAYPAUSE is pressed
(key PLAYPAUSE (print events))

# triggered when KEY_B is pressed, held or released
(key B [:down :hold :up] (print events))

# triggered when either KEY_C or KEY_D are pressed
(key [C D] (print events))

# triggered when KEY_LEFTCTRL, KEY_LEFTALT and KEY_DELETE are
# pressed at the same time
(chord [LEFTCTRL LEFTALT DELETE] (print events))

# triggered when KEY_A, KEY_B and KEY_C are pressed one after the other
(seq [A B C] (print events))

# start main loop
(poll "/dev/input/eventX" "/dev/input/eventY")
```

Installing

The latest stable version of pyzmo is available on pypi, while the development version can be installed from github:

```
$ pip install pyzmo # latest stable version
$ pip install git+git://github.com:gvalkov/pyzmo.git # latest development version
```

Alternatively, you can install it manually like any other python package:

```
$ git clone git@github.com:gvalkov/pyzmo.git
$ cd pyzmo
$ git reset --hard HEAD $versiontag
$ python setup.py install
```

Similar Projects

- [triggerhappy](#)
- [actkbd](#)
- [inputexec](#)

License

Pyzmo is released under the terms of the [Revised BSD License](#).