

# PyMT



“An open source Python library for developing multitouch applications.”

Safe Hammad  
Python Northwest  
21<sup>st</sup> October 2010

# The Multitouch (R)evolution

- The ability to simultaneously register 3+ distinct positions of input touches on a touch screen.
- Lends itself to usable multiuser applications / applications which don't follow the usual mouse based desktop paradigm.
- “Gesture Enhanced” e.g.:
  - One finger drag
  - Two finger pinch to zoom and rotate
  - Three finger scroll
- Set to become more widely used because of its intuitiveness. Note that gestures are not standardised between between platforms and applications. Yet?

# The Multitouch (R)evolution (cont.)

- Think iPhone / iPad / Helm of USS Enterprise!



<http://en.wikipedia.org/wiki/File:Multitouch-table.jpg>  
Creative Commons Attribution-ShareAlike 3.0 Unported

# The PyMT Library

- Linux/OSX/Windows.
- GNU LGPL License.
- Python 2.6.x and higher. Python 3.x not yet supported.
- Uses NumPy, PyGame, PyOpenGL.
- Optional acceleration with Cython based C extensions making direct OpenGL calls.
- Actively developed.
- Google search for PyMT: **Second** highest hit. See for yourself which is the highest! (A good lesson in choosing a suitable name for your project :)

# DEMO 1

Bundled kitchen sink example.

# Hardware Support

Good hardware support. If your hardware works with your OS, it will work with PyMT:

- Linux ( $\geq 2.6.34$ ): native support for multitouch events from the kernel.
- OSX: all Apple multitouch devices supported including builtin trackpads, magic mouse, magic trackpad.
- Windows 7: native support for WM\_Touch and WM\_Pen events.

# Hardware Options

- Touchscreen tablets / laptops.
- Trackpads.
- TUIO.
- Wiimote.
- Mouse emulation for those who can't beg borrow or steal any of the above.



<http://en.wikipedia.org/wiki/File:Wiimote.png>  
Creative Commons Attribution-ShareAlike 3.0 Unported

# PyMT Code – "Hello, World"

```
from pymt import *

class CircleDrawer(MTWidget):
    """Draw a circle at touch positions."""
    def draw(self):
        set_color(1, 0, 0)
        for touch in getCurrentTouches():
            drawCircle(touch.pos, 50)

runTouchApp(CircleDrawer())
```



# DEMO 2

Hello, World.

# PyMT Widgets

- The widget is the basic PyMT graphical element handling state, updates, drawing and input.
- A selection of basic widgets e.g. buttons, sliders etc.
- A selection of more complex / compound widgets that lend themselves to multitouch use.
- Can be styled and customised using CSS.
- Widgets respond to events e.g.:
  - `on_touch_up()`
  - `on_touch_down()`
  - `on_draw()`

# DEMO 3

Widget gallery.

# PyMT Utilities

- Drawing framework based on OpenGL provides easy drawing of basic shapes, polygons, textures. OpenGL available whenever needed for more advanced use.
- Image and SVG handling.
- Audio and video handling.
- Animation framework. “Kinetic” object response to touch is an important part of the user experience.

# DEMO 4

Animation framework.

# Roadmap

- TUIO v2. Timing information, finger pressure, additional geometry information.
- OpenGL 3.
- Network / proxy widgets – serialise and share widgets over a network.

# Thanks!

<http://pymt.eu>



Safe Hammad  
<http://safehammad.com>