



# **MultiMultiTouchTouch**

## **Making your own Space Palette**

**Tim Thompson**

**<http://timthompson.com>**

**[me@timthompson.com](mailto:me@timthompson.com)**

# MultiMultiTouchTouch (MMTT)

- C++ program, supports Kinect, Kinect 2, and Senz3D
- Uses depth image only
- Blob detection using OpenCV
- Trainable interactively on new frames, holes of any shape
- Trainable without a frame, using coordinates or colored image
- Browser interface to control it over HTTP, with JSON API
- Output is TUIO (a standard multitouch format) over OSC (a standard UDP protocol)
- Windows-only, open source: <http://multimultitouch.com>

# Quick Start

- <http://multimultitouch.com>
- Download and install
- Look in:
  - All Programs->Nosuch Media->MultiMultiTouchTouch
- Run: Install EVERYTHING
- Plug in Senz3D camera
- Run: MMTT
- Run: Example\_1

# Modifying Example\_1

- Run Processing
- Inside Processing, use “File Open” to open example\_1.pde in Documents\Nosuch Media\MultiMultiTouchTouch\example\_1
- Click on “Sprite” tab
- Change:  
    `rect(x0,y0,w,h)`  
    To:  
    `ellipse(x0,y0,w,h)`
- Play/Run it

# TUIO/OSC format

- Created by Reactable project – <http://tuio.org>
- Represents cursor information, with different “profiles”
- MMTT uses “2.5D Interactive Surface” profile, with messages:
  - `/tuio/25Dblob alive sid0 sid1 ... sidN`
  - `/tuio/25Dblob set s x y z a w h f X Y Z A m r`
  - `/tuio/25Dblob fseq frameid`
- Variables in “set” message which convey cursor information:
  - s = session ID
  - x, y, z = position in 3D space
  - w, h = width, height,

# Monitoring the OSC Output

- Run: Monitor OSC on port 3333
- Only one program can read from port 3333 at a time
- Format: `/tuio/25Dblb set s x y z a w h f X Y Z A m r`
- Output example:

```
'/tuio/25Dblb', ',sifffffffffffff', 'set', 1000,  
0.546875, 0.17500001192092896, 0.2936505377292633,  
0.0, 51.0, 80.0, 0.05312500149011612,  
0.0, 0.0, 0.0, 0.0, 0.0, 0.0
```

# Modifying the configuration

- Run: Open Config Directory
- Edit: `mmtt.json`

```
"camera": "senz3d",          # Other values: kinect, kinect2
"patch": "quadrants_senz3d", # a file in config/mmtt directory
"tuio.25d.clientlist" : "127.0.0.1:3333",    # OSC client(s)
```

- Format is JSON, be careful of comma placement  
(e.g. there's no comma after the last value in the file)

# Browser Interface to MMTT

- Make sure MMTT is running
- Run: Open Browser Interface
- Click: Calibration page
- Adjust the “Detection Plane”:
  - Depth for Top of Detection Plane
  - Depth for Bottom of Detection Plane
- Adjust values
  - Maximum Blob Size
  - Minimum Blob Size
- Toggle switch
  - Show Region Rects



# Using a physical frame

- Run: MMTT
- Run: Open Browser Interface
- Click: Advanced Calibration page
- Adjust: Depth for Top and Bottom of Detection Plane
- Click: New Registration Start
- Type: new patch name in text field
- Click: Save Patch

The resulting patch file is in:

Documents\Nosuch Media\MultiMultiTouchTouch\config\mmtt

- Edit: patch file (e.g. to change “first\_sid” values)

# Modifying Example\_1 – more details

- `cursorDownEvent()` in `example_1.pde`
  - Gets called when a cursor is moved or dragged
  - Sends MIDI NoteOn
  - Creates a graphical “sprite” and sets it in motion
- `cursorUpEvent()`
  - Sends MIDI NoteOff
- `pitchof(Cursor c)` and `velocityOf(Cursor c)`
  - Computes the MIDI pitch and velocity values for a cursor position
- `initializeAreas()`
  - Each Area is assigned a range of TUIO Session IDs, along with the color (for the graphical sprite) and MIDI channel.

# What else does MMTT have?

- Shared-memory interface
  - Faster than OSC
  - Can replace OSC, or just augment it
  - Transmits blob outlines, not just center

# More information

- Run: `README`
- Email: `me@timthompson.com`



# **MultiMultiTouchTouch**

**Making your own Space Palette**

**Tim Thompson**

**<http://timthompson.com>**

**[me@timthompson.com](mailto:me@timthompson.com)**