CONTROL

INTERACTION DESIGN

MULTITOUCH

& OSC (OPEN SOUND CONTROL)*

*MIDI on steroids :-)

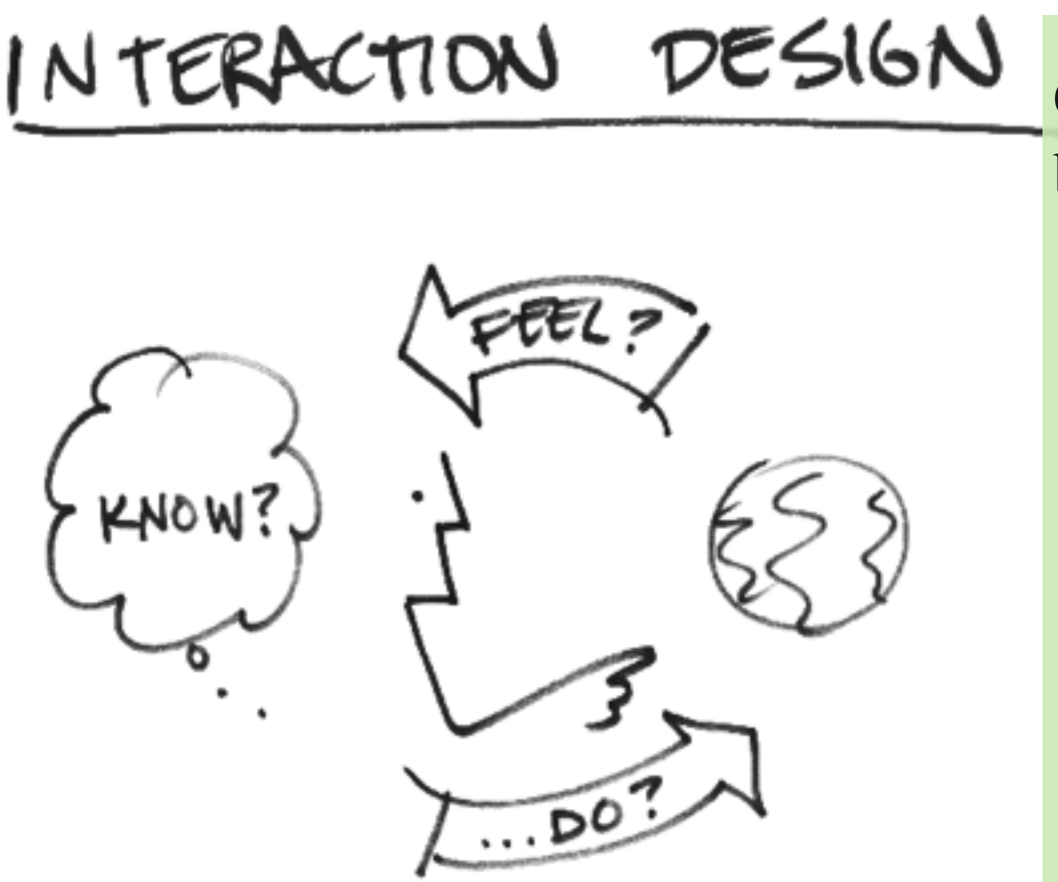
BD.Bridges@ulster.ac.uk
brian.randomtwist.com
www.brianbridges.net

PART 0: EASY MULTITOUCH

Mira 'It Just Works'!
Cycling 74: iPad only :-(

Just log on to same Wifi network as machine running Max

PART 1: INTERACTION DESIGN CONCEPTS AND IDEAS



One of the important choices for "How do you do"; is between **HANDLES and BUTTONS.**

- Handles are better for continuous control (e.g. trombone)
- Buttons are better for discrete control (e.g. piano keyboard)
- Handles leave you in control (e.g. opening a car door).
- Buttons are more likely to trigger something automatic (e.g. opening an elevator door).
- Q: in Max, what the counterpart object to a handle?
- A: a slider!

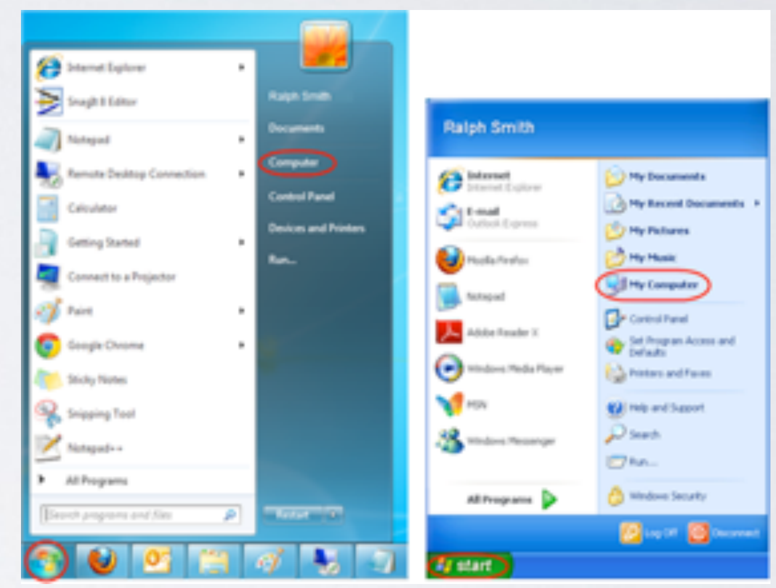
Sketches: Bill Verplank, guest lecture/sketching at CCRMA, Stanford, 2001

INTERACTION: SOFTWARE MEETS HARDWARE MEETS YOU!

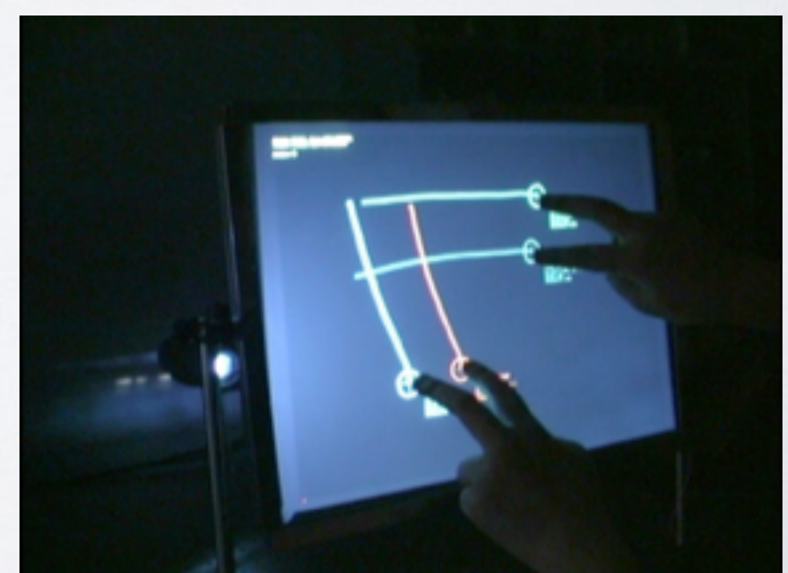
ADDING OPTIONS IN SOFTWARE

SEE (WESSEL AND WRIGHT, 2002)

- (a) artefact/control area multiplication



- (b) artefact/control area magnification



INTERFACING

- We need to interface between human actions (gestures which are repeatable/controllable with relative degrees of accuracy) and machine actions (output modalities: position in video, selection of sound-file etc.)

- => We need to choose technologies to best match human capabilities (gestures) to interface devices and output modalities

- Buxton (2011, chapter 1): 'Appropriate gestures can simplify syntaxes'

APPROACHING COMPUTERS

- Buxton (2011, chapter 1): "'**Hands-on**" computing is a myth'... "**finger on**" is typically more accurate
- Physical contact with **transducers** results in input to computer (movement translated to electrical signal, further translated to digital signal)
- **Variety of interaction modes** (continuous control from mice, tablets, touchscreens, discrete control from keyboards/switches), but feedback is rare
- But some form of feedback is often important in facilitating interactive gestures (informs user of *boundaries*) etc.
- **Taxonomy:** beyond form factor (joystick, mouse, trackpad etc.) to input form (force required to operate, force required for operation within various **ranges, structure/ dimensionality of input gestures** etc.=>2D, 3D etc.)

HISTORY AND TAXONOMY OF USER INTERFACES

- **From the beginning—switches:** keyboard, button, switch/toggle
- **1960s on—2D:** joystick, mouse, trackball, trackpad, touchscreen, tablet
- **3D and beyond**: joystick with rotating shaft, Kinect, accellerometer, gyroscope, Wii (increasingly sophisticated 3D movement, sometimes comprising '6 degrees of freedom' across X, Y, Z axes with rotation about these axes (roll, yaw, pitch)

Added dimensionality: add force/ acceleration controls (e.g. MIDI keyboard—switch plus scale)

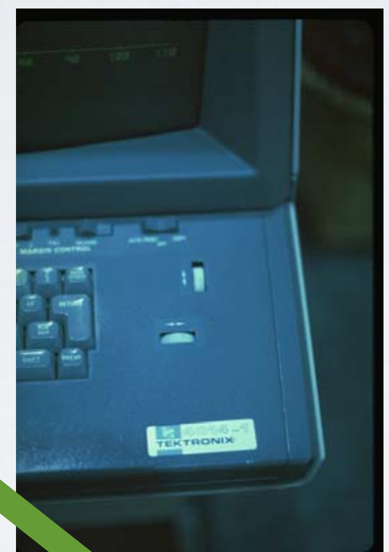
Added dimensionality, along with feedback for user, facilitates more ecological ('natural') user interaction (which may therefore efficiently tie in with our everyday (inter)action strategies

HISTORY AND TAXONOMY OF USER INTERFACES

- Mouse (Englebart and English, 1963)
- 2D control + switch (buttons/modifiers)=> therefore potentially 4D or more (non–simultaneously)

APPROPRIATE INPUT MODALITY

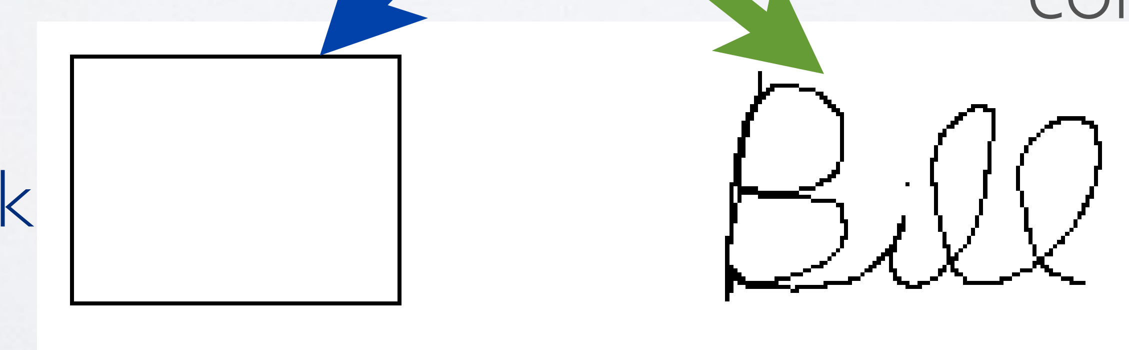
From Buxton (2011): chapter 1



Etch–a–sketch and similar: two one-dimensional controls (x and y movement)

Alternative drawing toy: integrated x/y control (joystick)

Difficult task

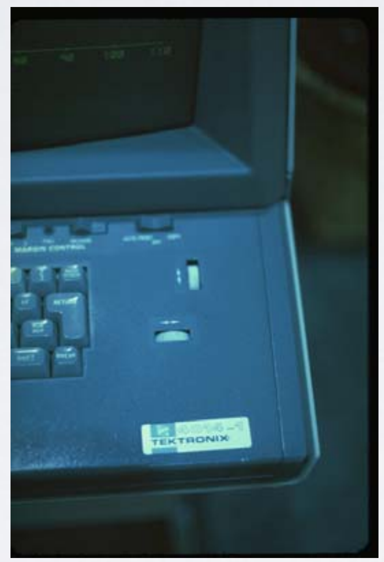


(a) Geometric Figure (b) Cursive Script

Difficult task

APPROPRIATE INPUT MODALITY

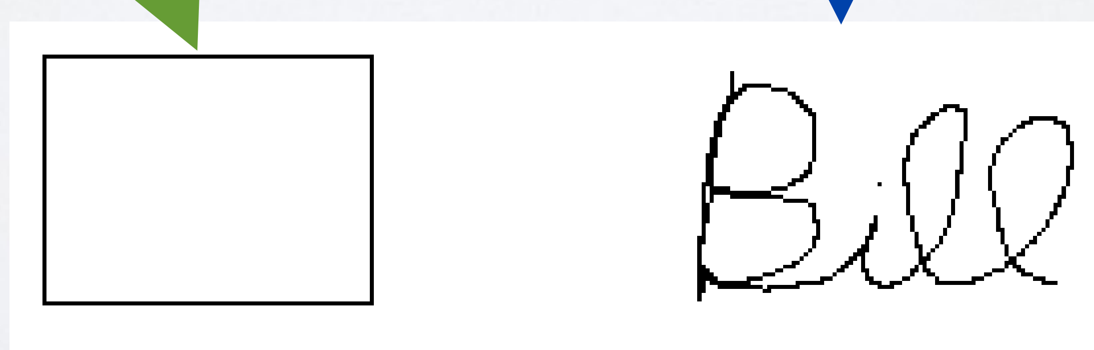
From Buxton (2011): chapter 1



Etch–a–sketch and similar: two one-dimensional controls (x and y movement)

Alternative drawing toy: integrated x/y control (joystick)

Easy task



(a) Geometric Figure (b) Cursive Script

Easy task

APPROPRIATE INPUT MODALITY

From Buxton (2011): chapter 1

Impose mapping to constrain x, y control to separate acts (make like two separate x, y controls)

(a) Geometric Figure (b) Cursive Script

Task (a) becomes easier

HISTORY AND TAXONOMY OF USER INTERFACES

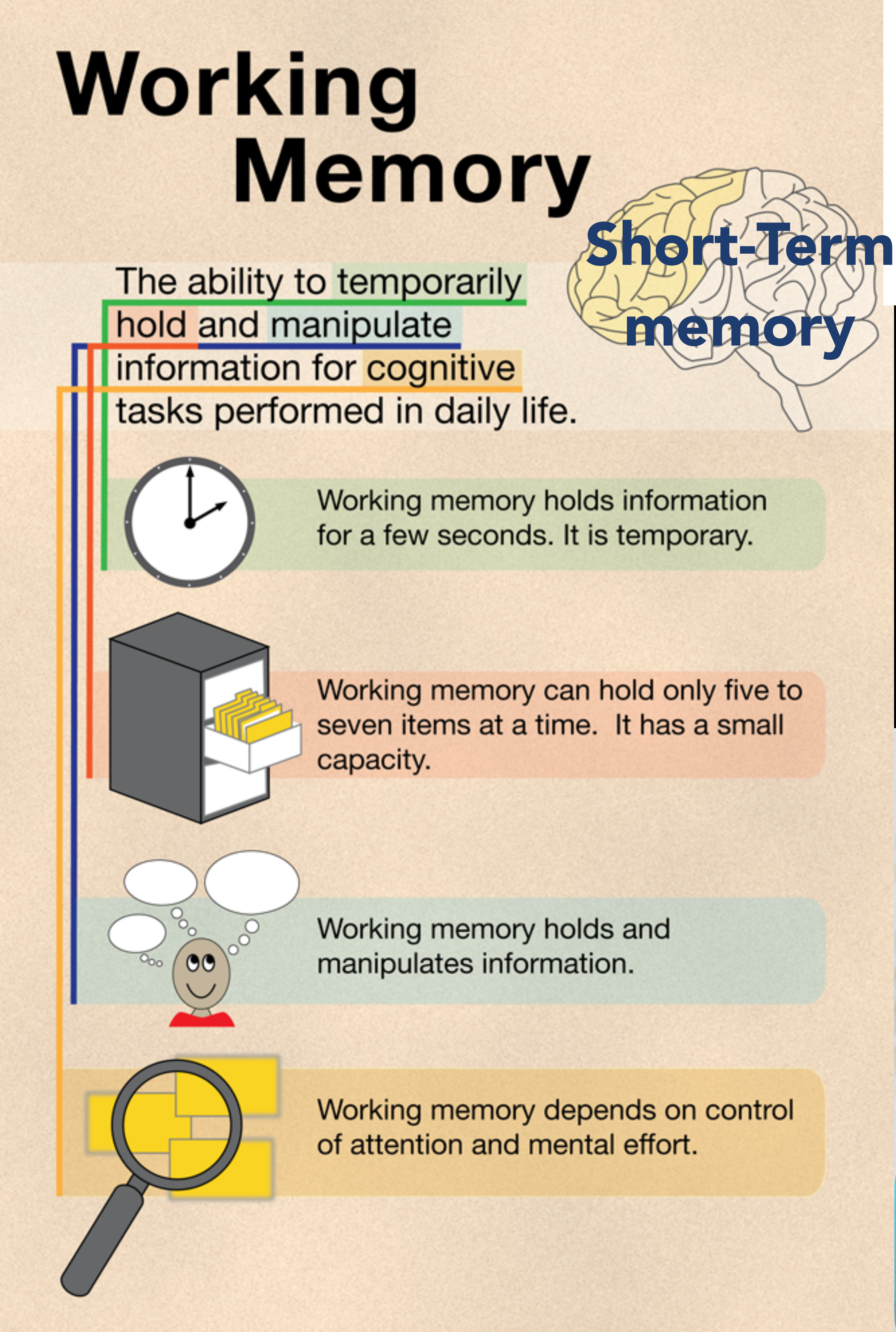
- Multi-touch
- Multiple parallel 2D outputs!
- Allows for clear user feedback at location of interaction
- Allows for registering of more complex gestures (relative changes of 2D positions for points tracked)=> beyond discrete dimensionality

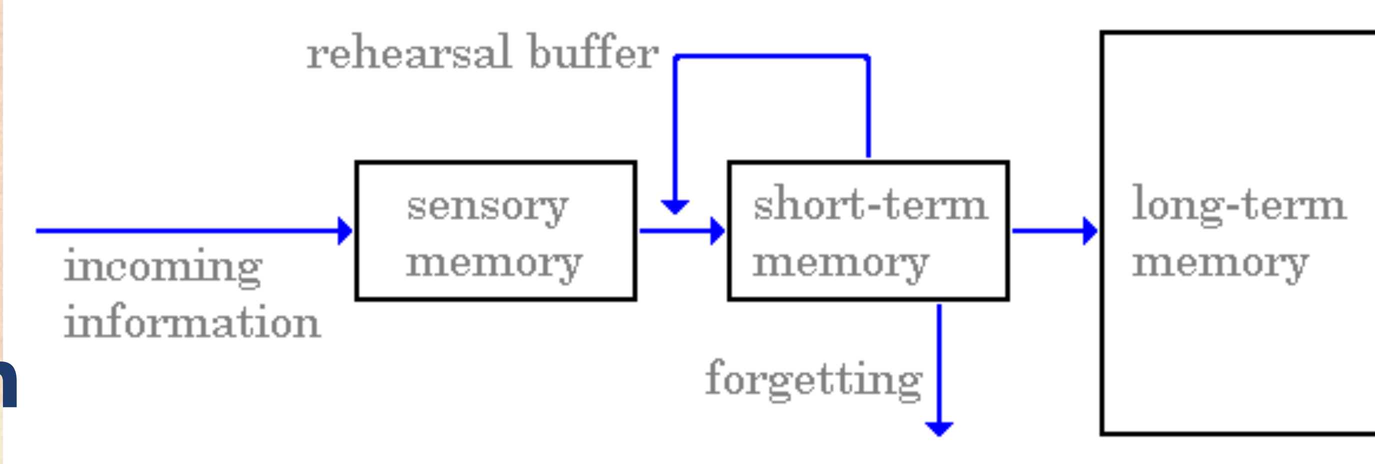
=> 'pinch to zoom' (unless you're scared of Apple's patent lawyers)

CONTROLS AND MAPPING

- e.g. **Switch: 2 states** (on/off or A/B for routing)
- **Simple control:** solution—use in combination, or use in cases where output is binary, too! Or employ more complex/continuous control (dials/sliders/mouse/joystick/trackpad etc.)
- Bill Verplank's *Interaction Design Sketchbook* (included with slides) notes 3 key elements in interaction design: **interface** (physical structure), **mapping strategy** (how the data from the interface is 'massaged' and routed) and **output** (the result)
- Throughout the history of interaction design, these elements have been accorded various degrees of importance at various stages of development=>**progression from interest/obsession with the basic affordances** (interface/mapping during 'honeymoon period') **to focus on new output forms** as the technology and usage conventions become more established

Short-Term memory





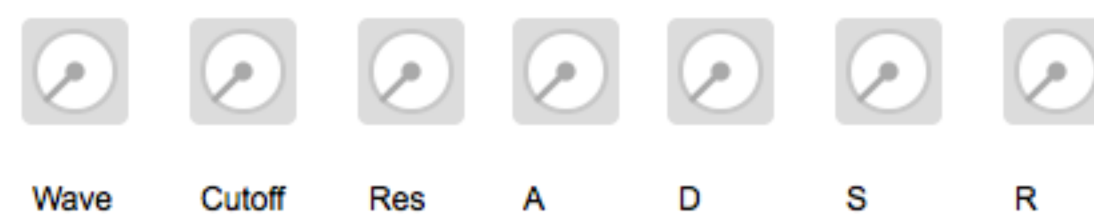
LONG LIST BUT ONLY 7+/-2???

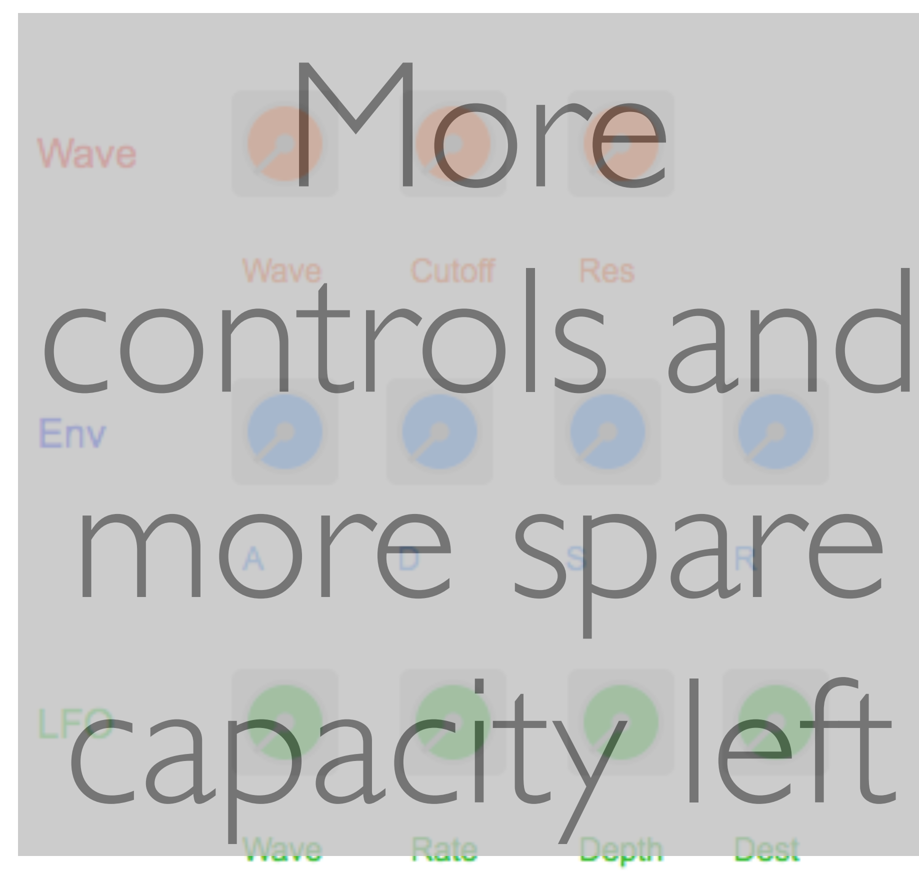
GROUP LONGER LISTS INTO C.5-7 SUB–GROUPS OF ITEMS (E.G. TELEPHONE CODES, 028 71...)

HUMAN MEMORY CAPACITIES AND UI ELEMENTS

- Bearing in mind the need for alignment with a **limited capacity of short–term memory,** colour codes in user interfaces should use a small number (5–9) of clearly distinct colours

- Similarly, large numbers of unrelated interface elements—without clear structural grouping—should be avoided: 7+/-2 also describes this limit, even if labelling of controls is present...however, this may be overcome through **grouping related controls** together

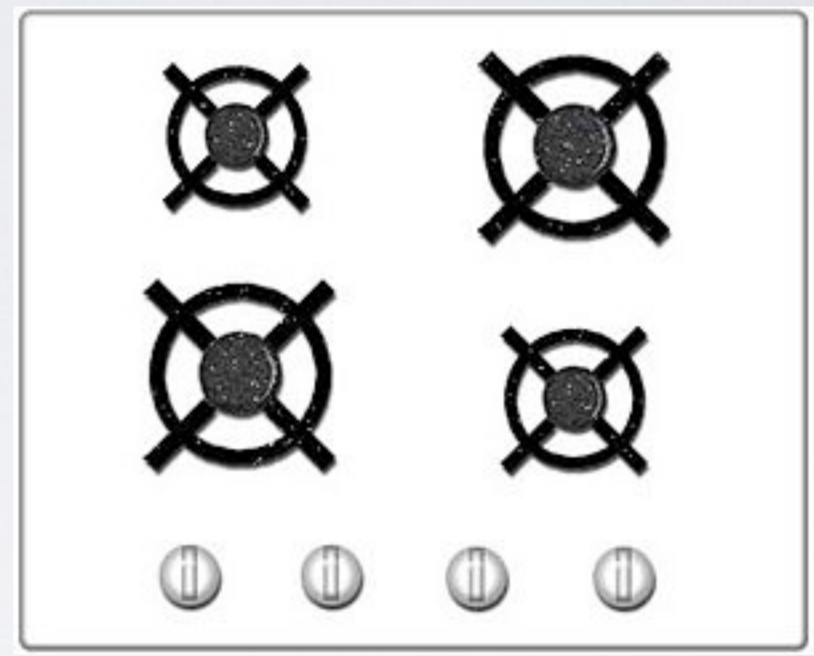


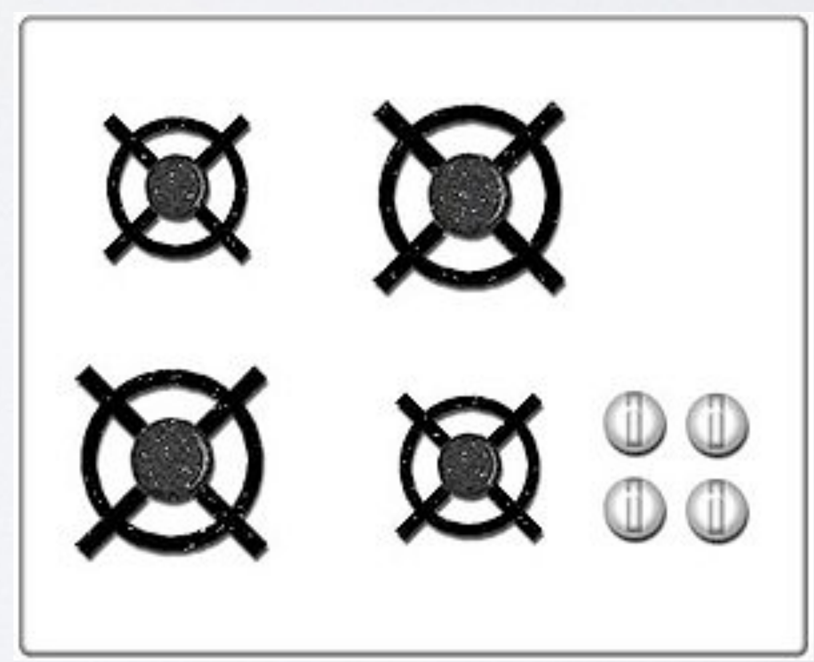


MAPPING

- Mapping strategy will clearly affect usability/learnability mentioned earlier
- Simple strategies may produce clear results, but may suffer from lack of complexity/developmental potential
- Complex strategies may increase cognitive load/decrease learnability
- **Natural mappings versus arbitrary mapping** (natural mapping of controls will reflect the organisational structure of the outputs)

See: Norman (1988, p.75)





MAPPING

• One–one mapping: single input matches single control output	Straightforward/predictable
• One–many mapping: single input is scaled/translated/mapped to multiple control outputs	Simple control input, but potentially more complex/less predictable result (requires learning of relationship between input and output)
• Many–many mapping: multiple inputs mapped to similar number of multiple outputs	Straightforward/predictable in terms of control associations, but multiple controls places ‘strain’ on STM

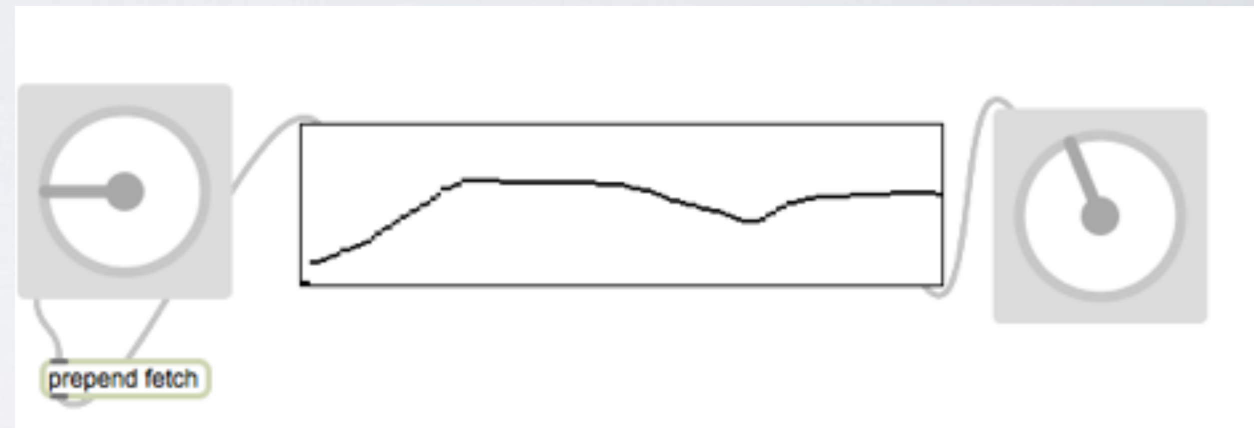
MAPPING

- One–one mapping: single input matches single control output
- One–many mapping: single input is scaled/translated/ mapped to multiple control outputs
- Many–many mapping: multiple inputs mapped to similar number of multiple outputs

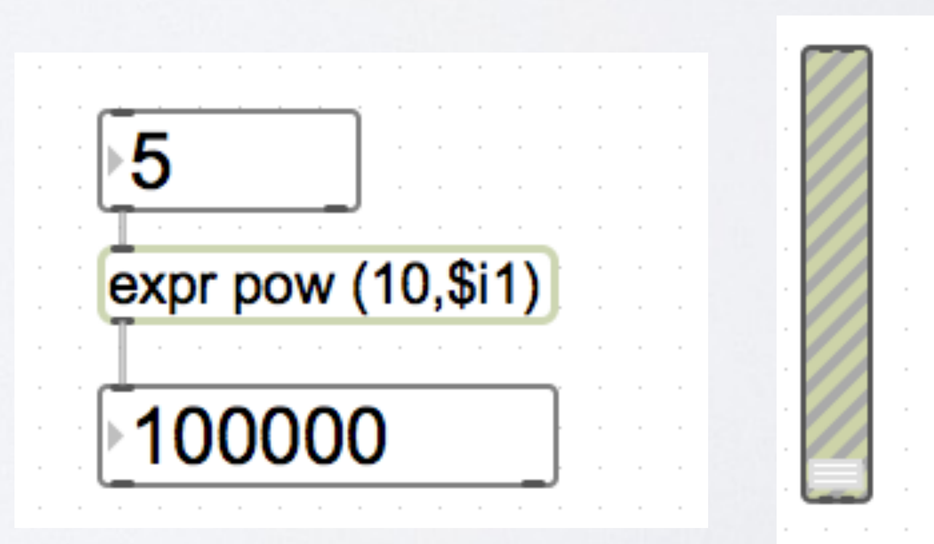
May use Max tables to map values in non–linear fashion

MAPPING AND SCALING

- As we've found in our Max experience, mapping values from a source (controller) to an output frequently involves scaling
- Scaling may be linear (using scale object) or non–linear (use a lookup table—e.g. send values to multislider, read off multislider values)...non–linear may be especially useful in one–to–many mappings (you can also use tables, colls etc.: coll will be useful for smaller and more precise ranges of input and output values)
- For some attributes, you may want to scale via a predictable but non-linear mapping, such as a power/exponential law (which we have seen can relate a small range of input values to a wide range of output values, producing a rapid scaling for very small changes)



Non–linear scaling



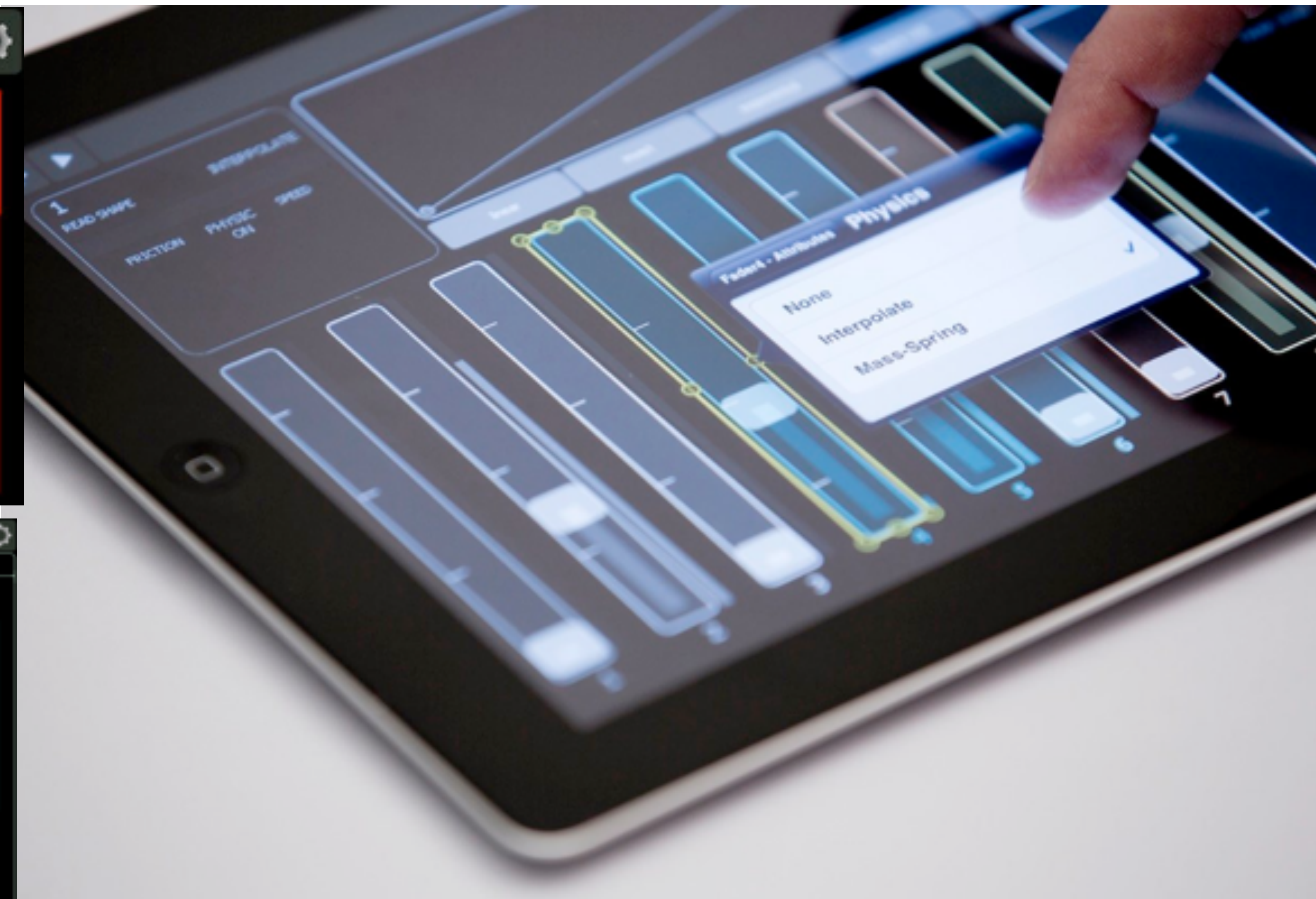
Some Max objects handle this for you!

INTERACTION DESIGN: CONCLUSION

- Think about the task
- Select an input modality and note dimensionality
- Think about structure of the task, then think about grouping of controls and mapping strategy which is similar in structure
- Experiment and refine, ask friends who are unfamiliar with Max to try your patch out!

PART II:
FROM MIDI TO OSC AND MULTITOUCH APPLICATIONS

EASY MULTITOUCH II: LEMUR



Cross-platform (Android/iOS), outputs MIDI and OSC

PRECURSOR: MIDI RECAP

- Musical Instrument Digital Interface is a communications protocol which has enjoyed great longevity in the performance and studio fields
- Introduced in 1983 (at the January 1983 NAMM music industry trade show), it works on the basis of a serial message communication format using discrete messages to transmit data about the specification of events such as notes on synthesisers, control of recording devices and lighting cues
- Its longevity has been due to the advantages posed by continuity in this field (e.g. the ability to use older/vintage gear in integration with more modern equipment), coupled with a degree of commercial inertia with regard to the development of an alternative
- In addition, the MIDI model has shaped expectations of its users: its structure permeates the design of modern synthesisers and commerical digital audio workstations (DAWs)

MIDI: LIMITATIONS

- Primarily discrete model of communications: specify note as 'triggered event' of certain length—note too expressive as a model of musical performance/interaction; standardises expectation of musical note structure based on piano keyboard (12TET note pitch divisions, limited variety of articulation models)

- Low-bandwidth/low resolution: advantage in the early years of MIDI (hardware could not cope with high-bandwidth data), but now a problem as more is expected of control data (e.g. 7 bit/0-127 applied to a wider range of values provides limited resolution); MIDI's resolution is limited in both magnitude values of controllers (128 values by default) and timing (e.g. compare timing demands of Max scheduler and MSP signal network)=> result: granularity in control (e.g. audible stepping/'zipper noise')

- Naming convention: MIDI control change (CC) messages are organised by numbers from 0-127: limited number of controllers and a somewhat abstracted 'naming' convention (would a language-based 'symbolic' name be clearer?)

OPEN SOUND CONTROL (OSC)

- Commercial providers have yet to agree on a successor to MIDI, although Yamaha pioneered its 'somewhat open' MLAN (Music Local Area Network—uses FireWire IEE1394)—patented but available under royalty-free license (appears to be dormant since 2008)
- University research has been more successful/influential: the University of California at Berkeley's CNMAT (Centre for New Music and Audio Technologies) developed Open Sound Control (Wright and Freed, 1997)
- Open Sound Control has learned from some of MIDI's mistakes: it is more easily extensible and makes fewer assumptions about the structure of what it may be controlling; as a result, it has become extremely popular in electronic arts communities

OSC: KEY FEATURES

- Open/extensible URL-style symbolic naming (easier to read 'at a glance' due to symbolic names) featuring use of '/' to define 'path' (clearer communications structure)

- Uses standard network hardware (MIDI started to do this later in life)

- High-resolution numeric data for magnitude and temporal resolution

- Some similarities to MIDI: controller name + value as 'bundled' message, with optional time tag

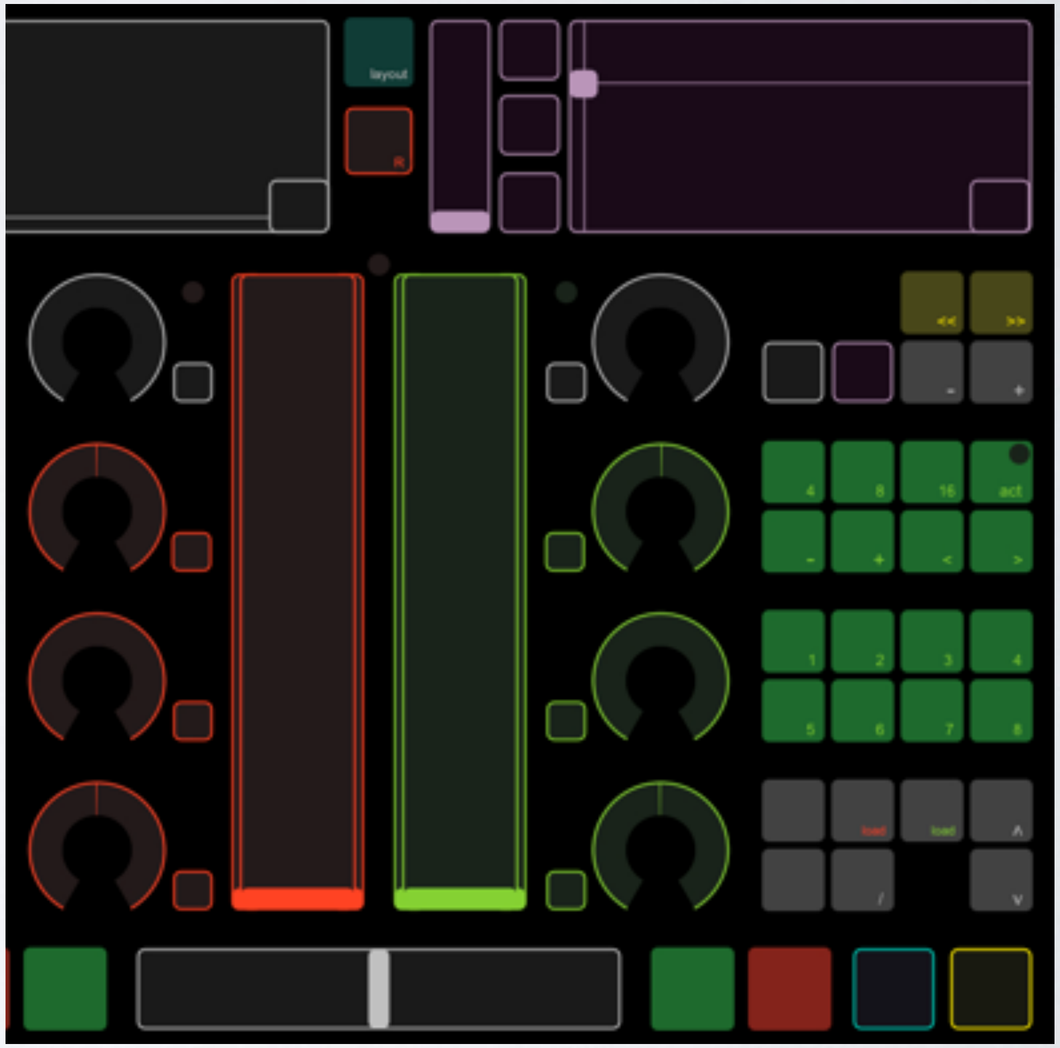
- Data packets are 32 bits in size (as opposed to MIDI: 7 bit)4,294,967,295 values (in memory address terms, equivalent to 4 GB of different memory addresses)

OTHER ADVANTAGES

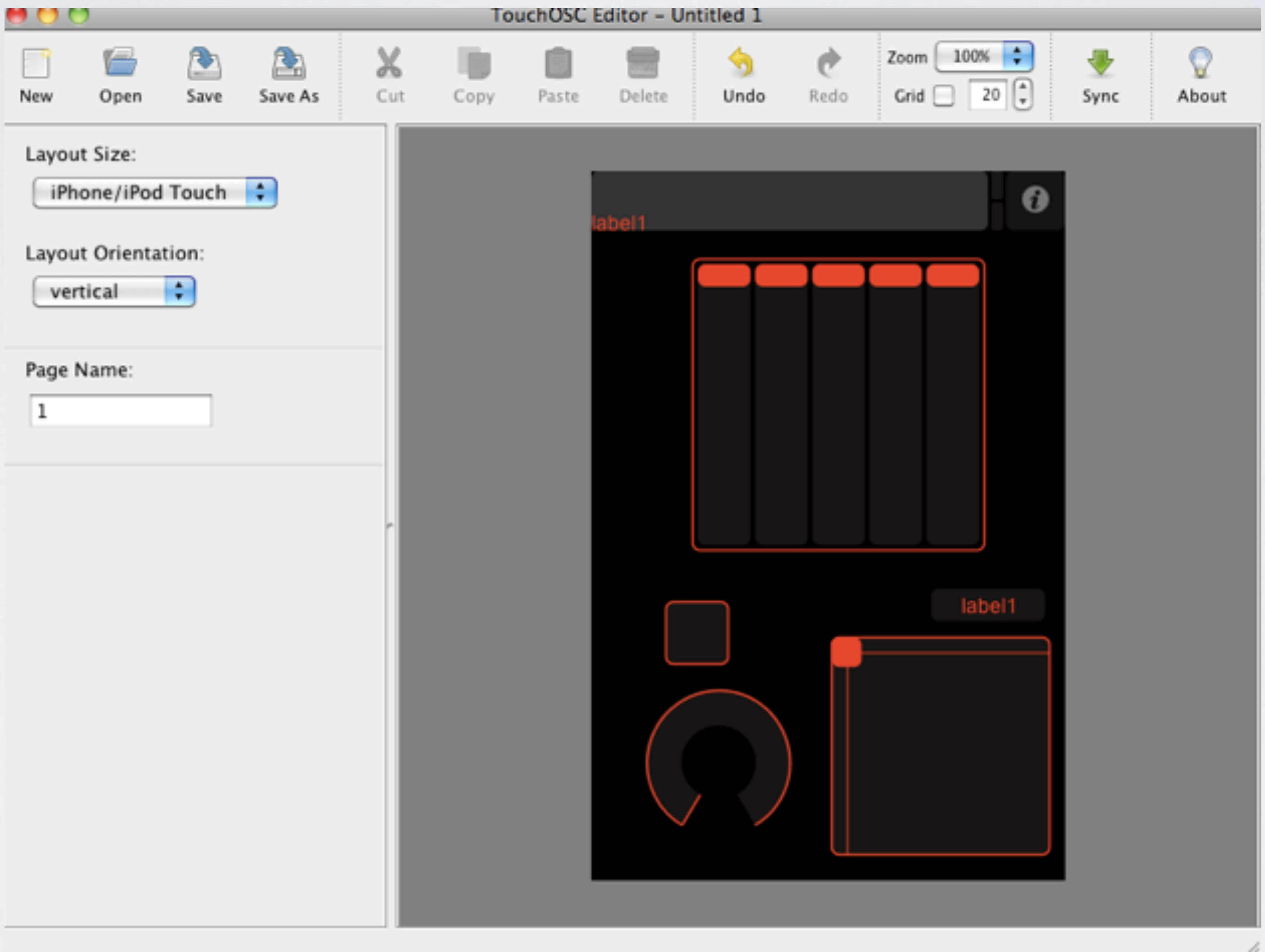
- Vibrant, distributed community (not just at CNMAT); grew out of NIME (New Interfaces for Musical Expression) conference
- Lack of commercial 'lock-in' of features to particular hardware/ software platforms (available as library for all major operating systems, including all of the usual 'desktop' suspects, plus iOS, Android, Java)
- As physical computing becomes more prevalent, expect OSC to be around for a long time to come as a communications protocol of choice

OSC UI/CONTROL SURFACE APPLICATIONS

- A number of smartphone apps transmit Open Sound Control messages over networks
- These can be used to control computer-based applications (such as Max)
- TouchOSC is available as a paid download for iOS and as donationware for Android
- It takes advantage of Open Sound Control to define extensible user interfaces and transmit high-resolution data from the phone's multitouch control surface



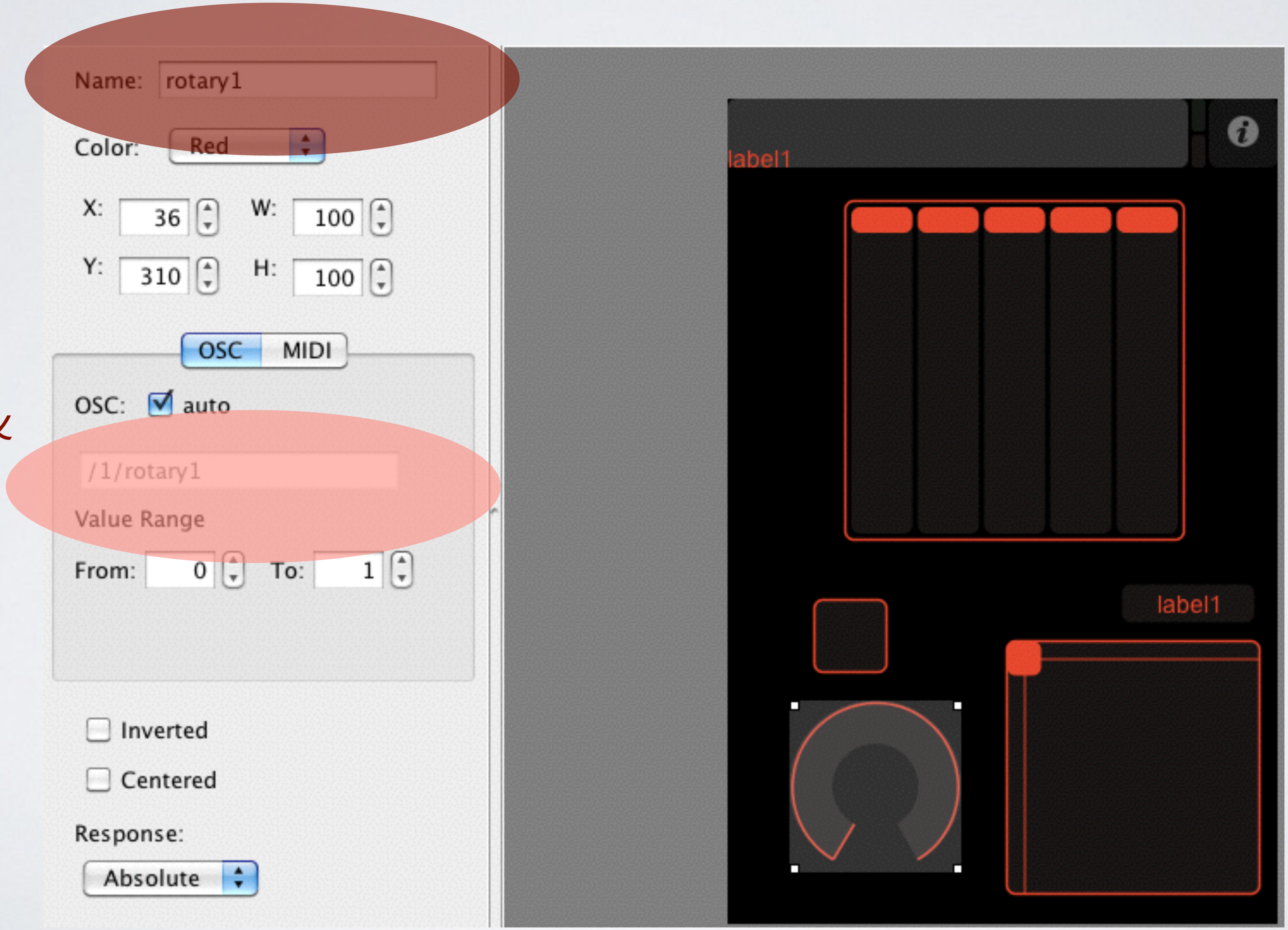
TOUCHOSC: CROSS-PLATFORM JAVA-BASED EDITING APPLICATION FOR CUSTOM UI LAYOUTS



OSC OBJECT NAMING CONVENTION

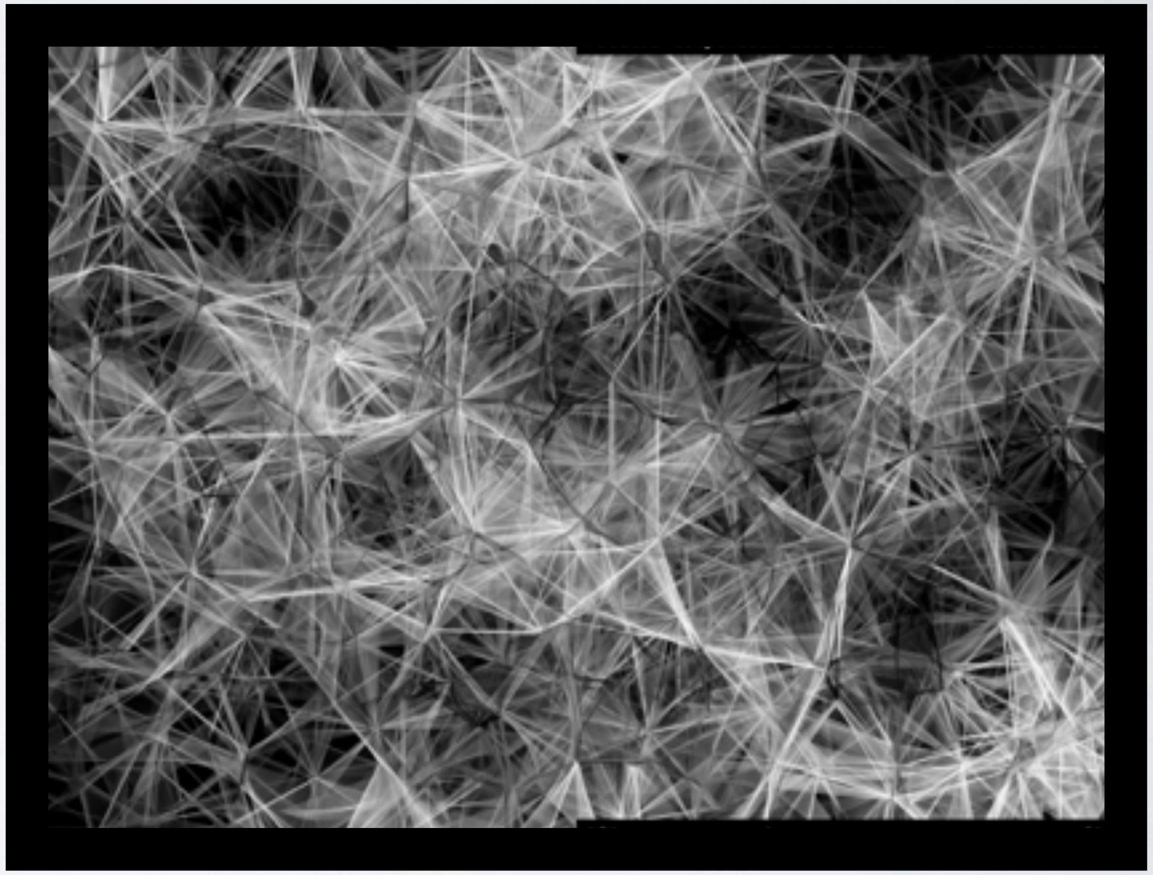
Name

Name& path



RECEIVING OSC

- You can receive OSC data in Max and some of its digital arts software brethren (Pd, Supercollider, Csound and Processing), along with the possiblitiy of using it with hardware such as the Arduino via Maxuino; you can also use it with MIDI-based applications via a commercial intermediary/translator such as OSCulator...but wait, you could also make a Max patch to do this translation into MIDI!



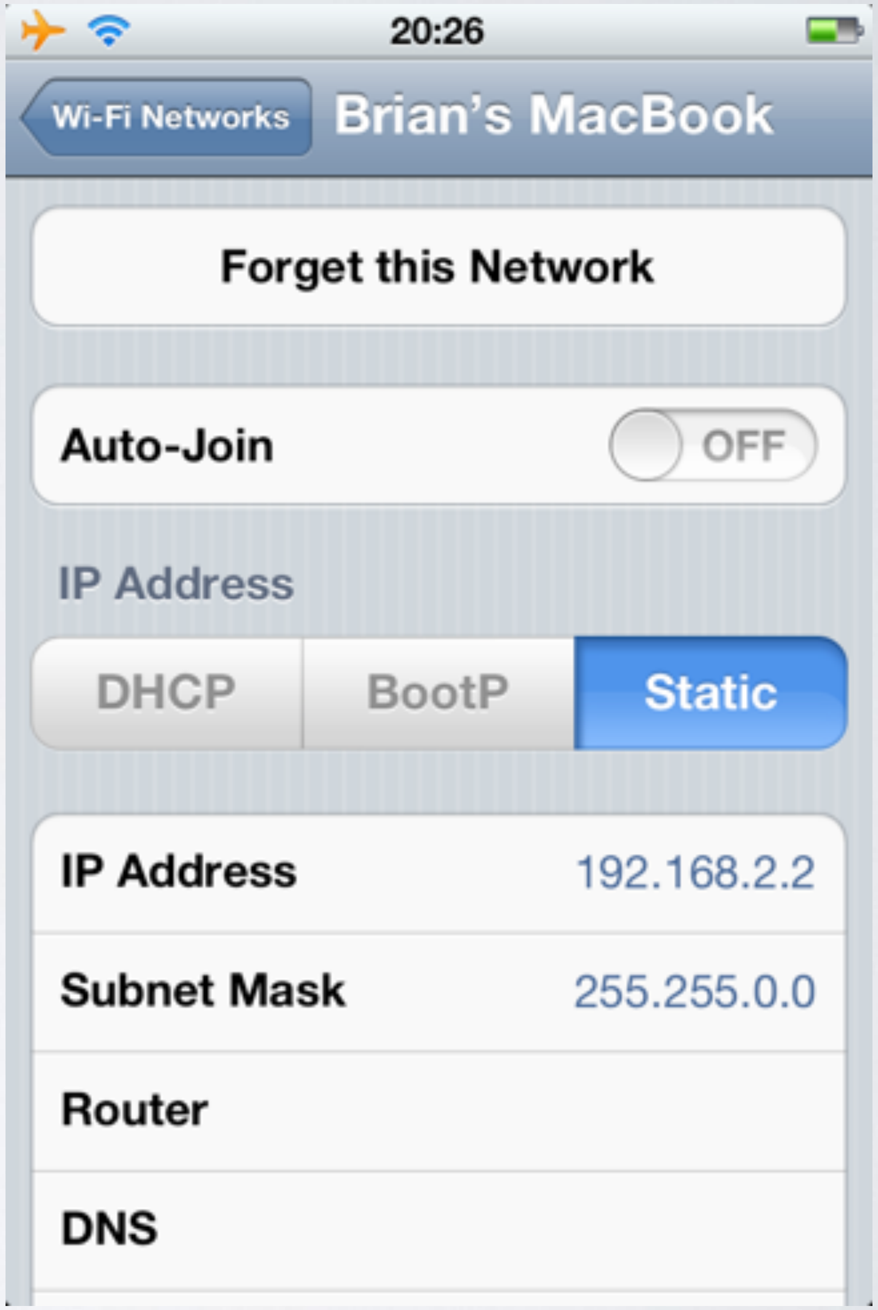
- We'll use it in Max to apply some basic control messages and to 'discover' the format of OSC messages, but it can be used to transfer much higher-bandwidth data

SETUP — DEVICES

- Create 'ad-hoc' WiFi network on Mac (Airport/WiFi menu, Create Network)— more reliable than connecting over a remote network, which may be carrying lots of other data; check and note the IP address of the computer on this network

- Connect Android Smartphone/iOS device to this network (instructions will follow iOS method but Android operations are similar)

- You may wish to set a static IP address for your device which you always use in your patch (e.g. 192.168.2.2)



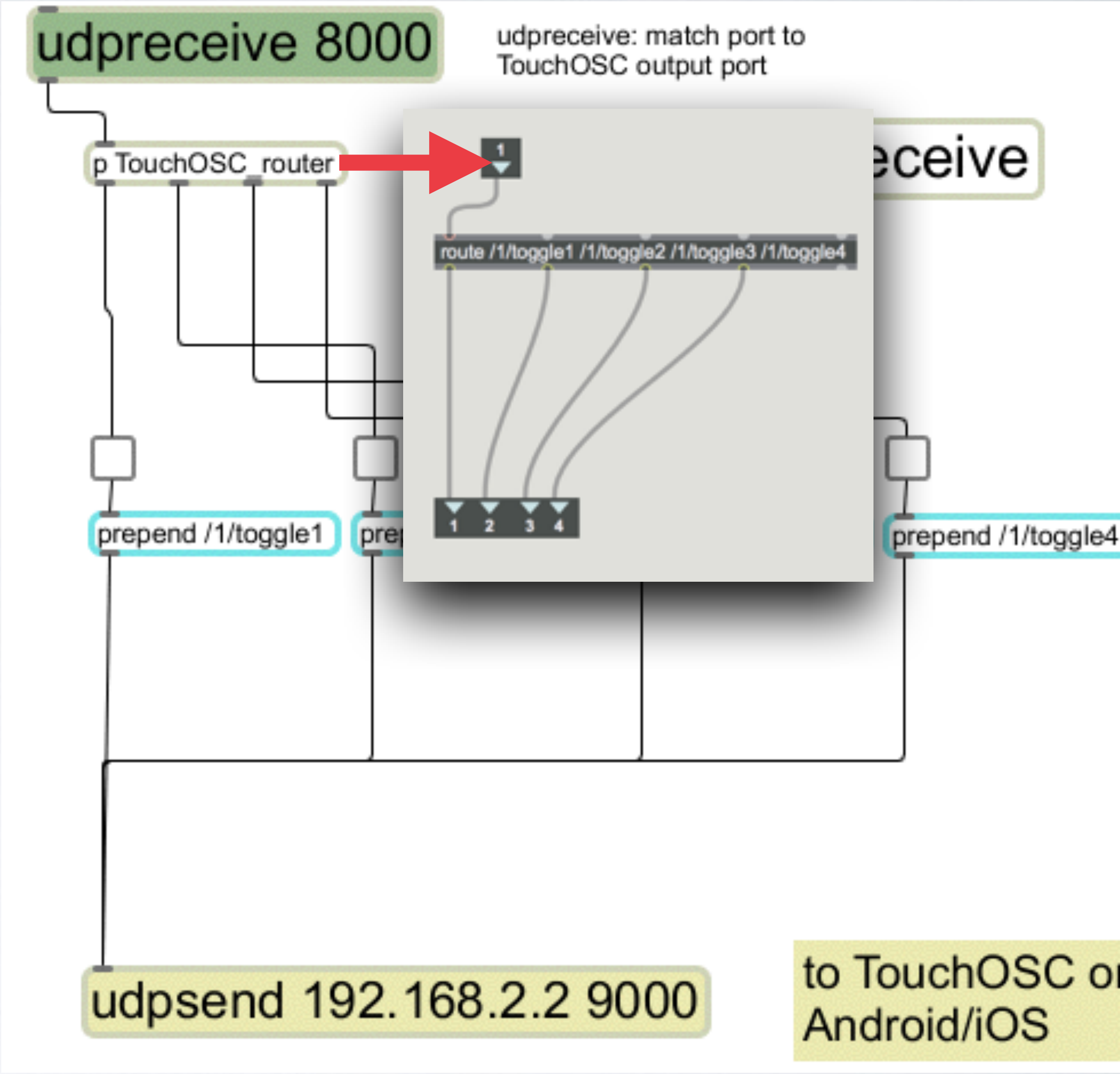
Setup: TouchOSC

SETUP (2) — TOUCHOSC AND MAX

- Click on 'Network' tab and enter your computer's IP address (noted from earlier)— in this case, I've set up a manual address under the TCP/IP tab (192.168.1.73), but you could simply note the address which you have been automatically assigned (DCHP)

- Note incoming and outgoing ports: these enable you to organise your sending and receiving of messages and are essential information if you want to set up your Max patch correctly (I'm using 8000 for outgoing—to Max—and 9000 for incoming—to iOS)

- You can process OSC messages via a suitably configured **route** object (you can also use the OSC externals from CNMAT—original home of OSC—but these have to be installed manually)



NAMING/VALUE CONVENTION

- A little like MIDI: name followed by value—invariably in floating point(e.g. /fader1 1.0)

- Object names can be something clear like 'fader' (a symbolic name) instead of *unclear* like MIDI CC 10 (all MIDI inputs being 'named' as CCs with numbers)

print	/1/fader1 0.462025
print	/1/fader1 0.459916
print	/1/fader1 0.457806
print	/1/fader1 0.455696
print	/1/fader1 0.453587
print	/1/fader1 0.451477
print	/1/fader1 0.453587
print	/1/fader1 0.455696
print	/1/fader1 0.453587
print	/1/fader1 0.451477
print	/1/fader1 0.449367

FURTHER READING (INTERACTION)

- Buxton, W. 2007. *Sketching User Experiences.* Amsterdam and Boston: Elsevier. Available at University of Ulster e-library: (click link from catalogue whilst logged in to Portal)
- Buxton, W. 2011. *Human Input to Computer Systems: Theories, Techniques and Technology.* Unpublished ebook, available at: http://www.billbuxton.com/inputManuscript.html
- http://www.istartedsomething.com/20100322/bill-buxton-on-making-user-interfaces-natural/
- Heim, S. 2008. *The Resonant Interface: HCI Foundations for Interaction Design.* Boston: Pearson
- Miller, G. 1956. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review*, vol. 63, pp. 81-97. Available at: http://www.musanim.com/miller1956/
- Moggridge, B. 2007. *Designing Interactions.* Cambridge, Mass.: MIT
- McCloskey, B., Bridges, B., & Lyons, F. 2015. Accessibility and dimensionality: enhanced real time creative independence for digital musicians with quadriplegic cerebral palsy. In: *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME),* Louisiana State University. Available at: http://www.nime.org/proceedings/2015/nime2015_250.pdf
- Norman, D. 1988. *The Design of Everyday Things*. Cambridge, Mass.: MIT

REFERENCES/FURTHER READING ON OSC

- www.opensoundcontrol.org
- Wessel, D. and Wright, M. 2002. Problems and Prospects for Intimate Musical Control of Computers. in *Computer Music Journal*, 26(3):11-22.
- Wright, M. 2005. Open Sound Control: an enabling technology for musical networking. in *Organised Sound* 10(3): 193–200
- Wright et al. 2001. Managing Complexity with Explicit Mapping of Gestures to Sound Control with OSC. in *Proceedings of the 2001 International Computer Music Conference.*
- Freed, A. and Shmeder, S. 2009 Features and Future of Open Sound Control version 1.1 for NIME. in *Proceedings of the 2009 New Interfaces for Musical Expression conference*.

SOFTWARE AND COMMENTARY

- http://hexler.net/docs/touchosc
- http://createdigitalmusic.com/2011/02/a-few-good-touchosc-layouts-from-waldorf-to-traktor-to-ableton-and-a-brief-rant/
- http://createdigitalmusic.com/2010/11/jazzmutant-lemur-controller-is-dead-long-live-multitouch/
- http://createdigitalmusic.com/2011/04/expanding-touch-and-midi-mobile-ios-control-gets-more-mature-in-new-and-updated-apps-round-up/
- http://chrisjeffs.com/wiregui/
- http://www.nr74.org/c74.html
- http://pinktwins.com/fantastick/