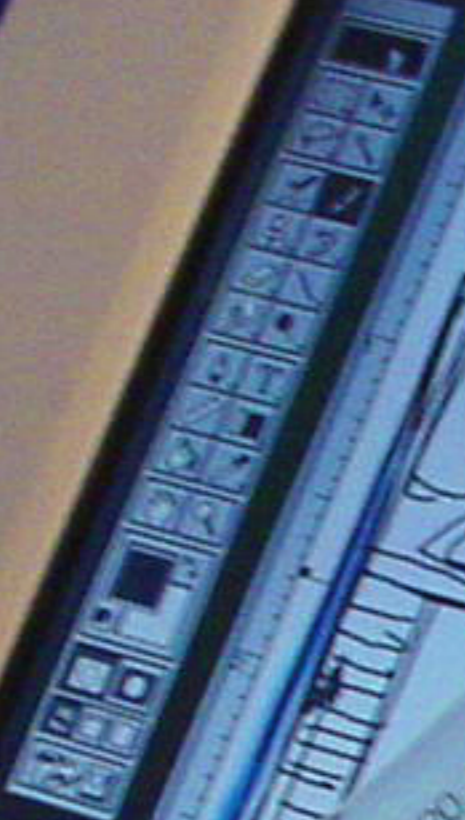
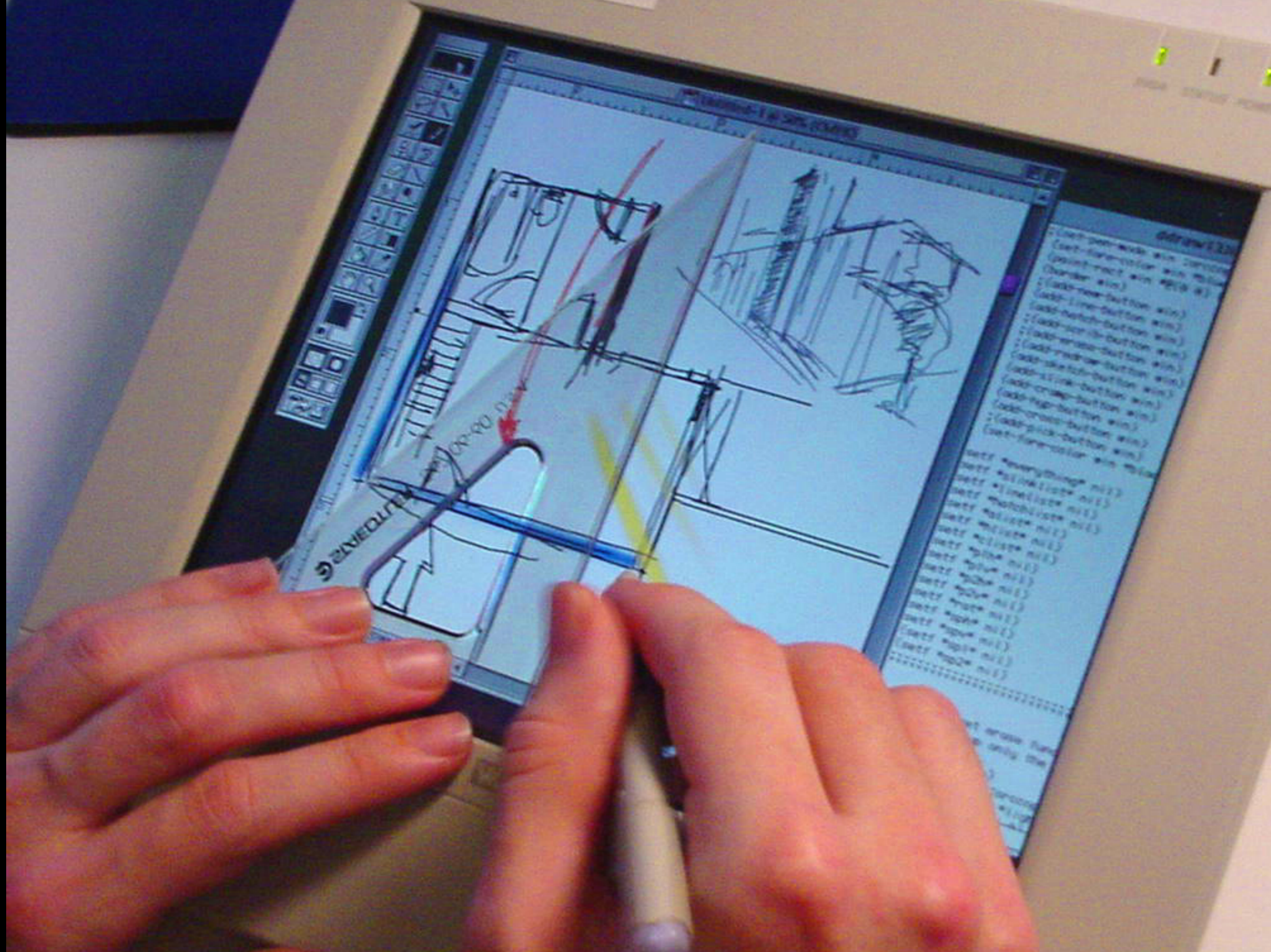


designers need
high level languages
that support rapid
prototyping

Mark D Gross (CMU)
Ellen Yi-Luen Do (Georgia Tech)

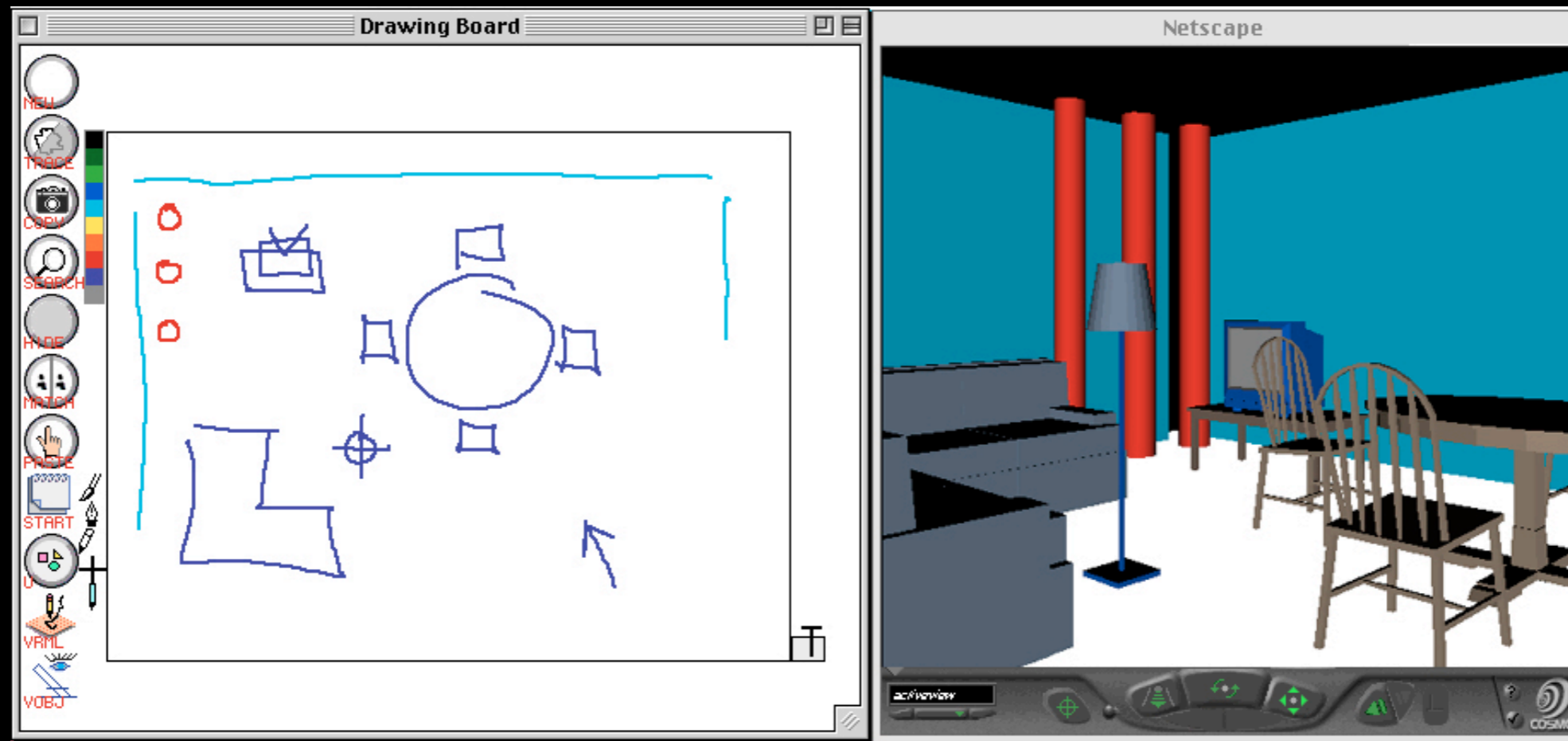




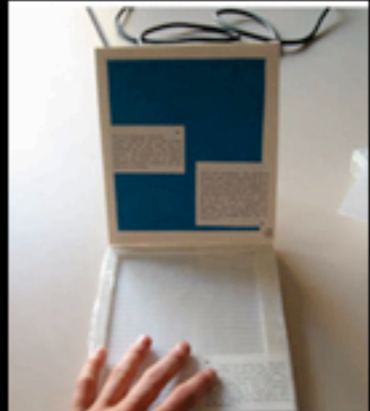
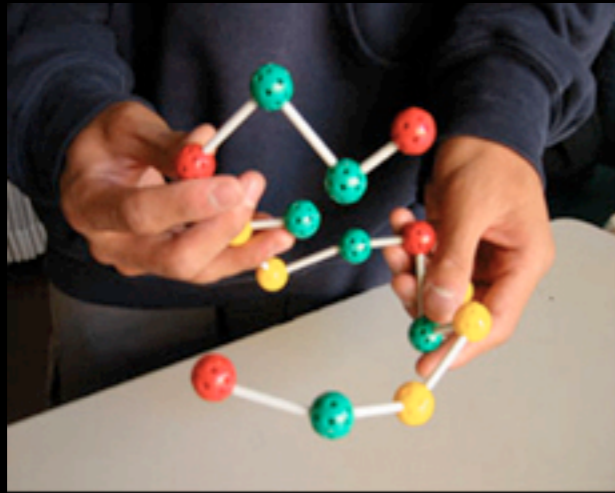
ELIANT

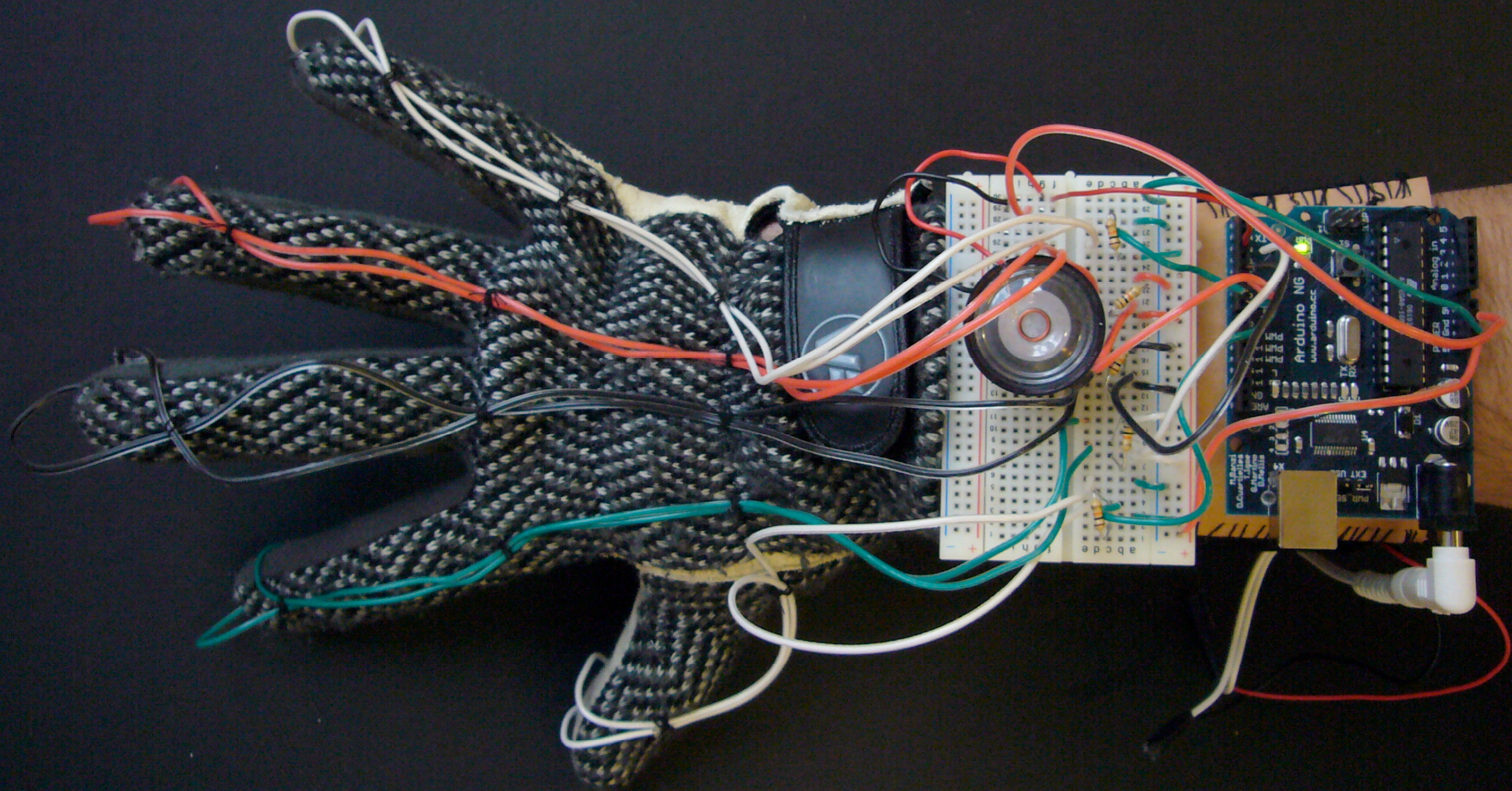
00-00-00

```
draw123  
[set-pen-mode win console  
(set-fill-color win white)  
(paint-rect win white)  
(clear win)  
(add-new-button win)  
(add-line-button win)  
(add-rect-button win)  
(add-erase-button win)  
(add-sketch-button win)  
(add-undo-button win)  
(add-redo-button win)  
(add-pick-button win)  
(set-fill-color win white)  
  
setf *everything* nil)  
setf *line-list* nil)  
setf *rect-list* nil)  
setf *erase-list* nil)  
setf *sketch-list* nil)  
setf *undo-list* nil)  
setf *redo-list* nil)  
setf *pick-list* nil)  
setf *pen* nil)  
setf *fill* nil)  
setf *brush* nil)  
setf *text* nil)  
(setf *pen* nil)  
(setf *fill* nil)  
(setf *brush* nil)  
.....  
  
set mouse fun  
only the  
  
spread  
w/lig
```

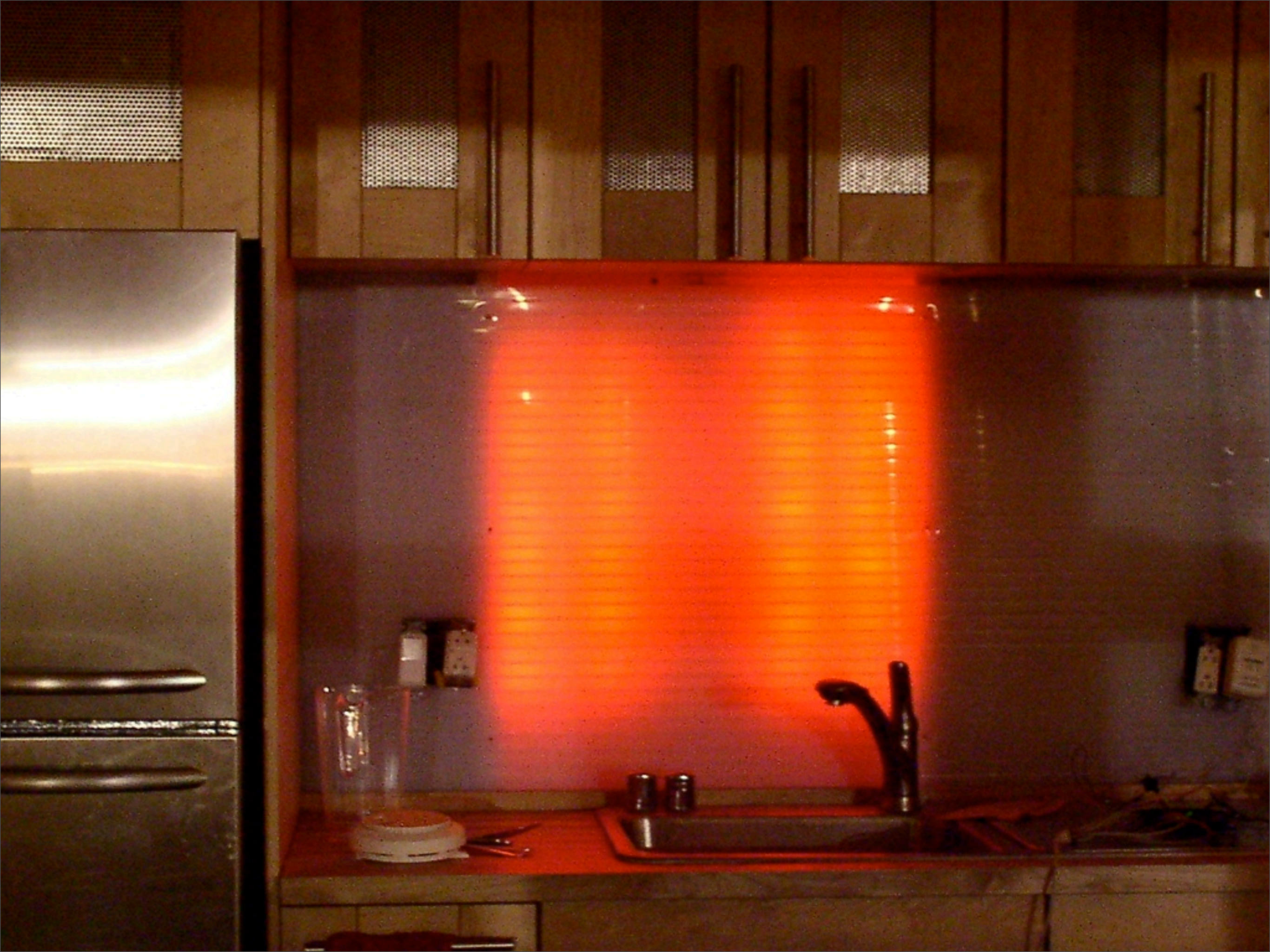


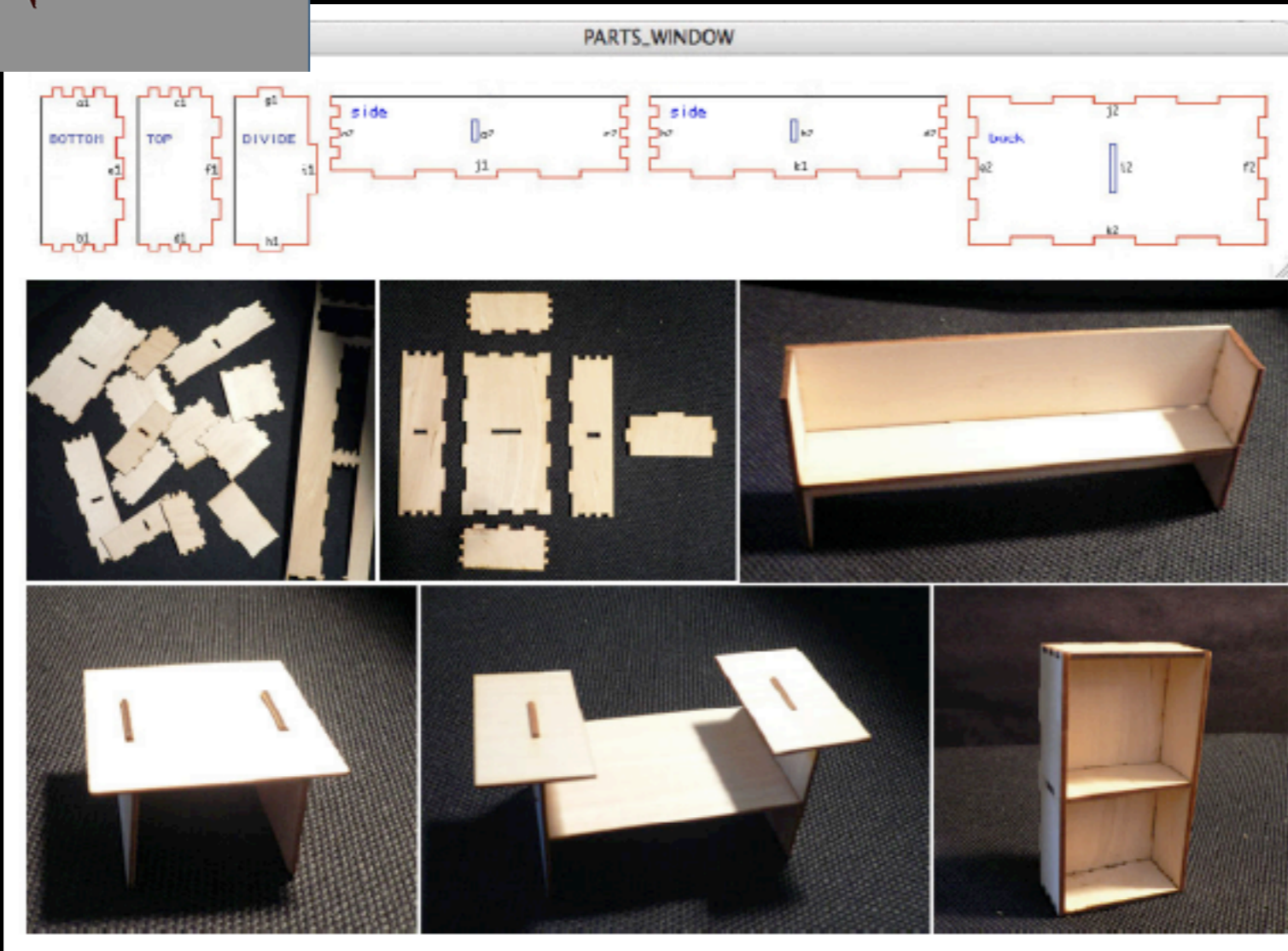
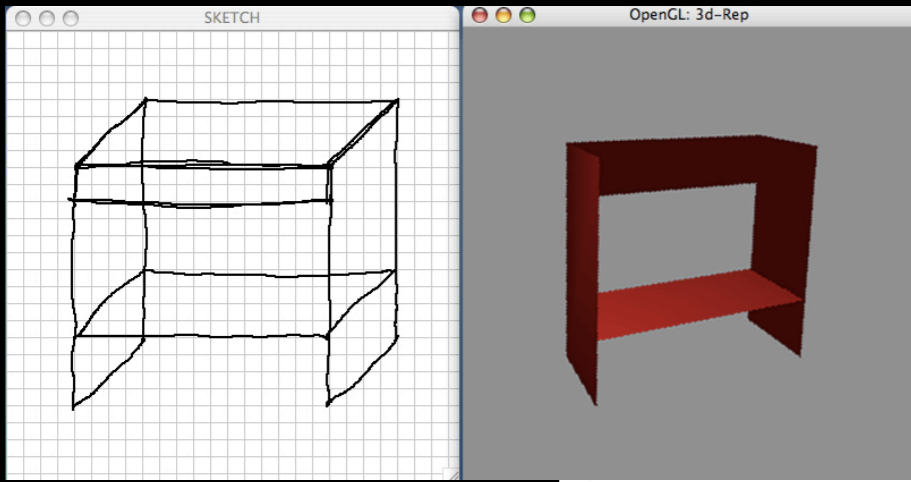
designers sketch (“thinking with a pencil”)
so CAD tools should support sketching
AND understand the drawing



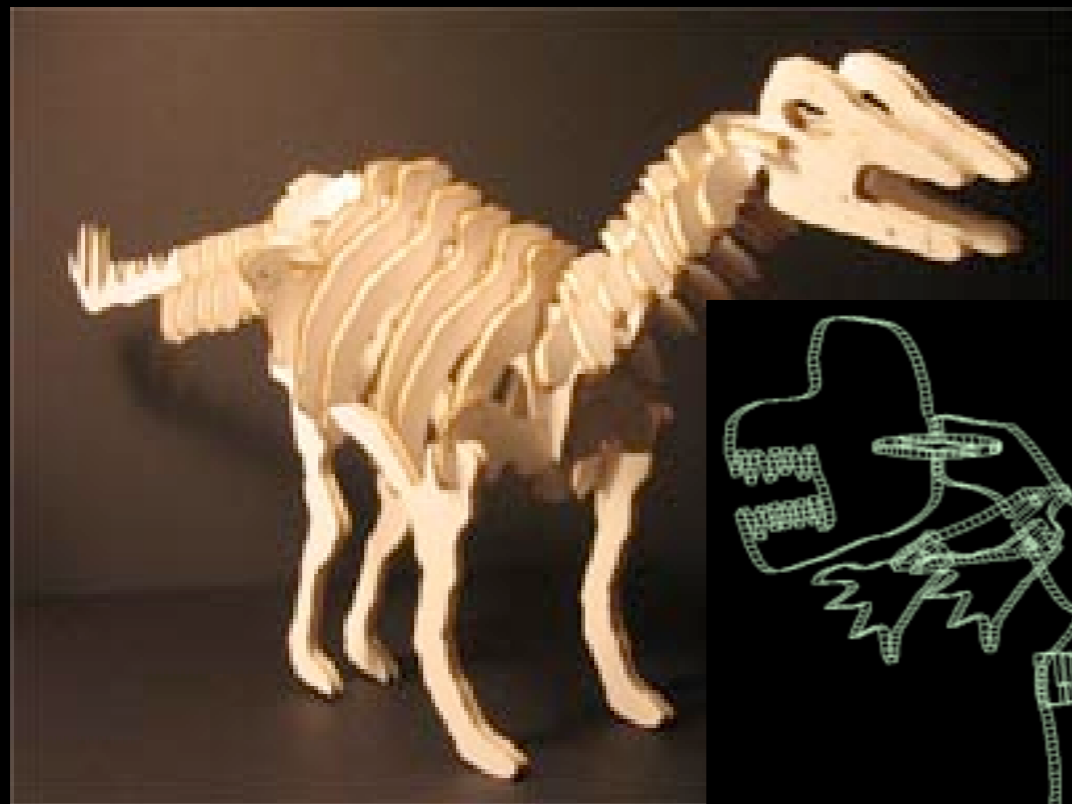
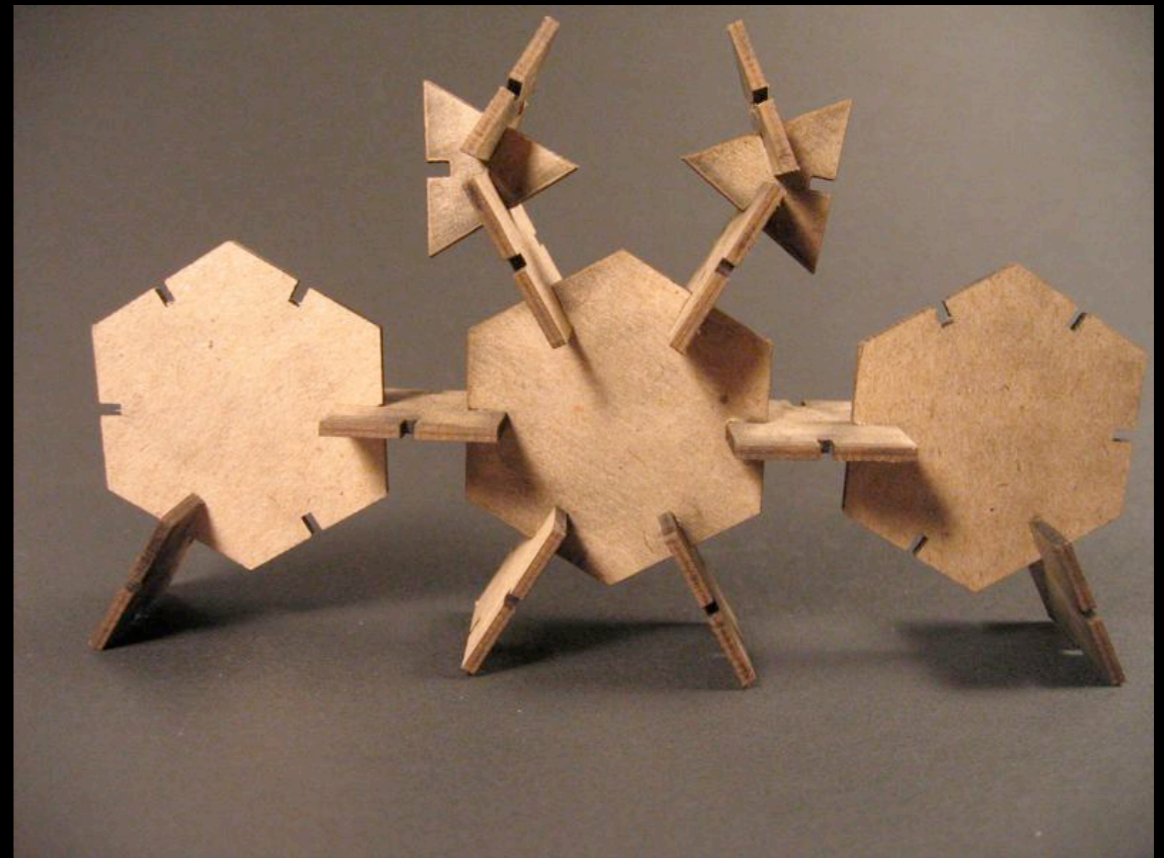
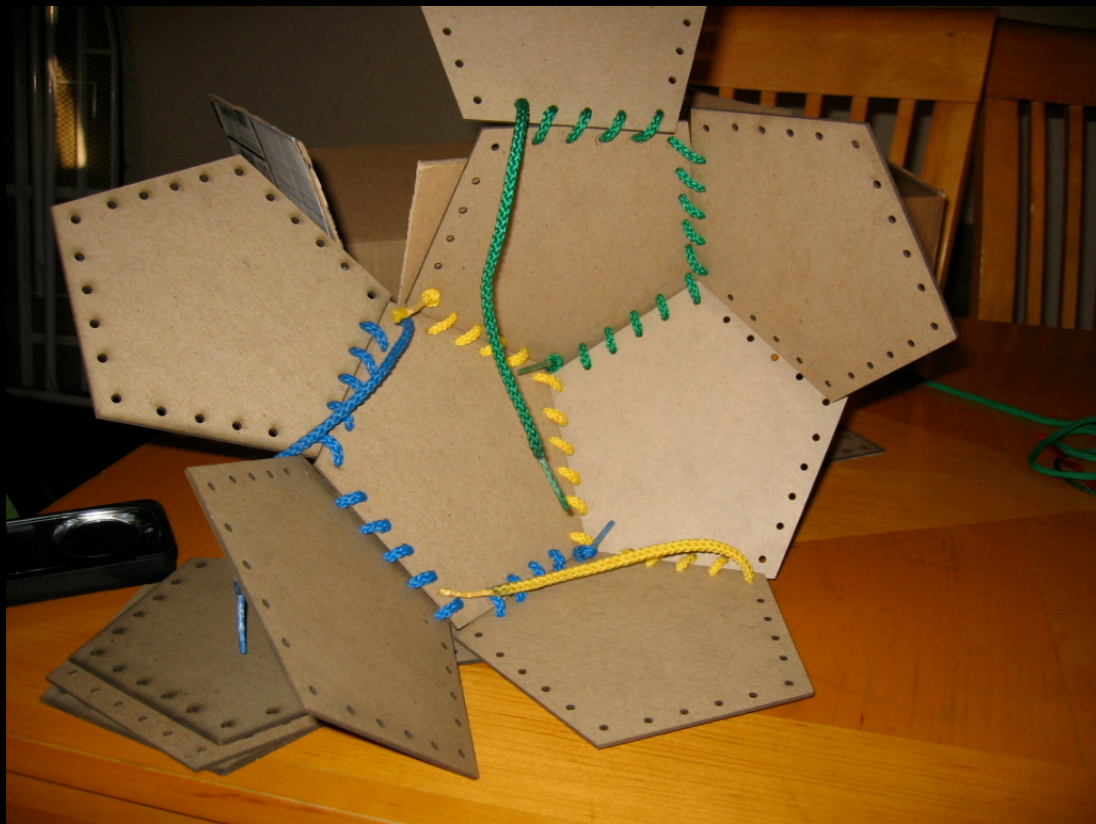




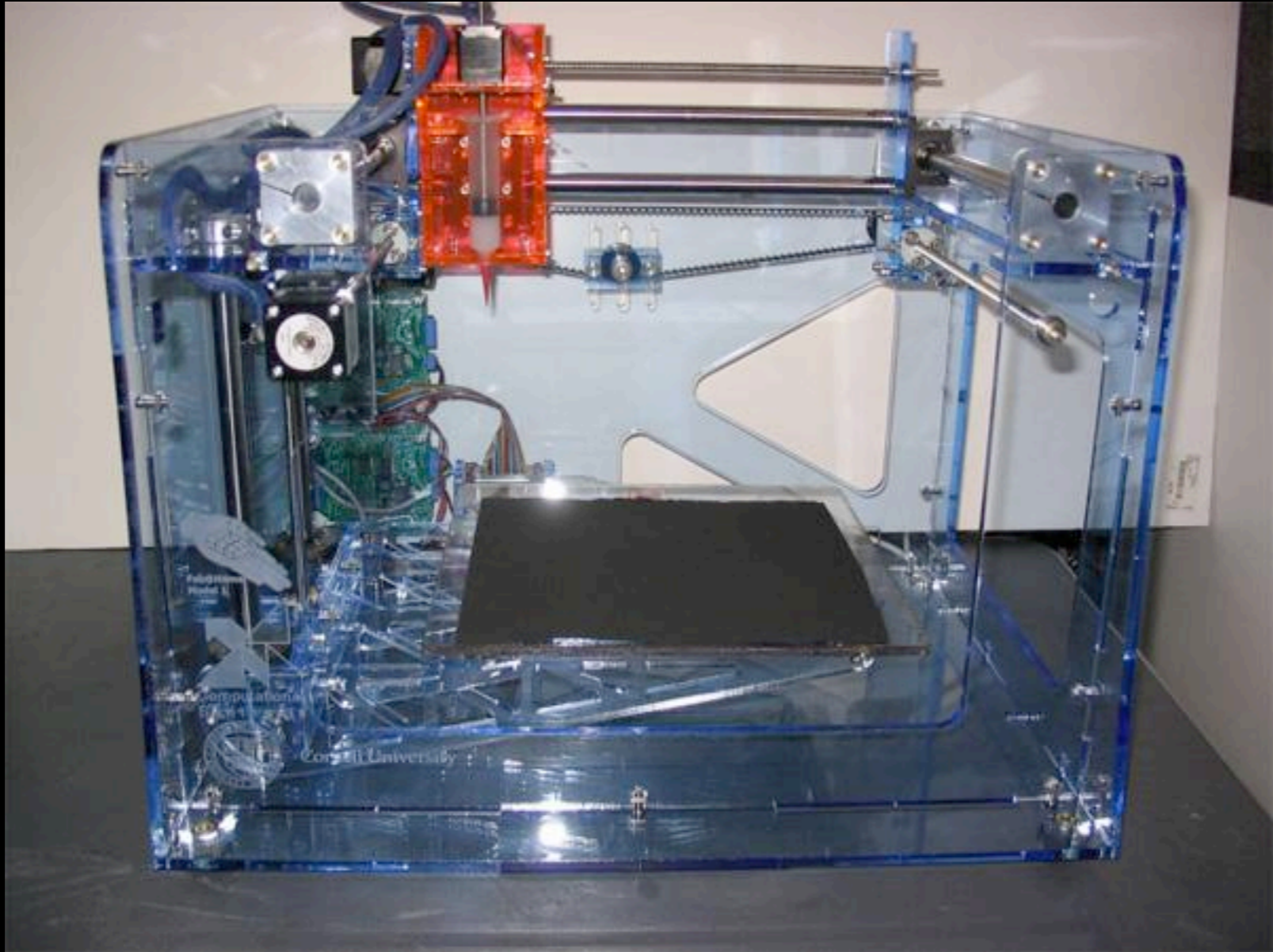


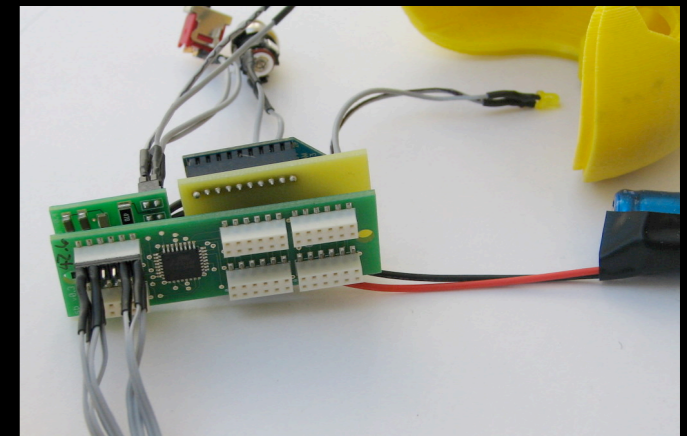
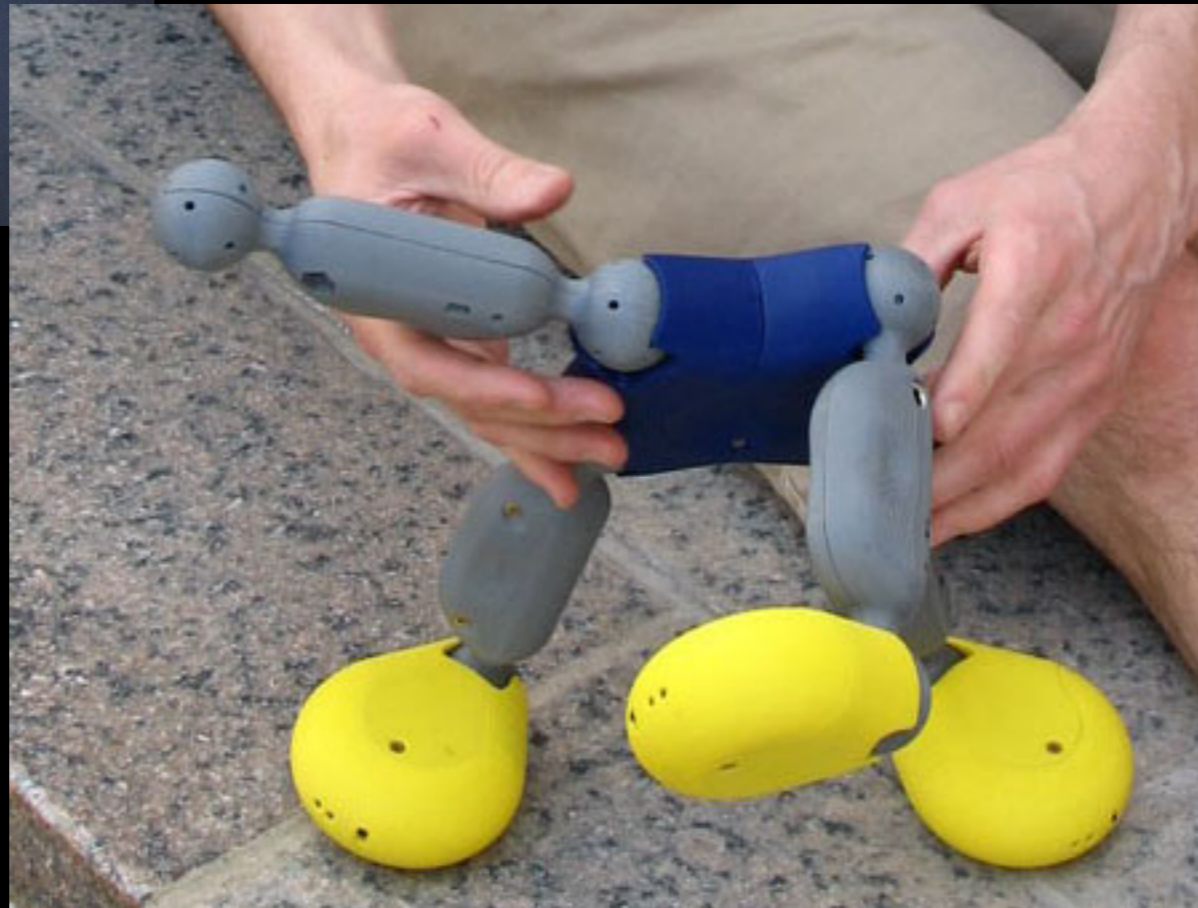
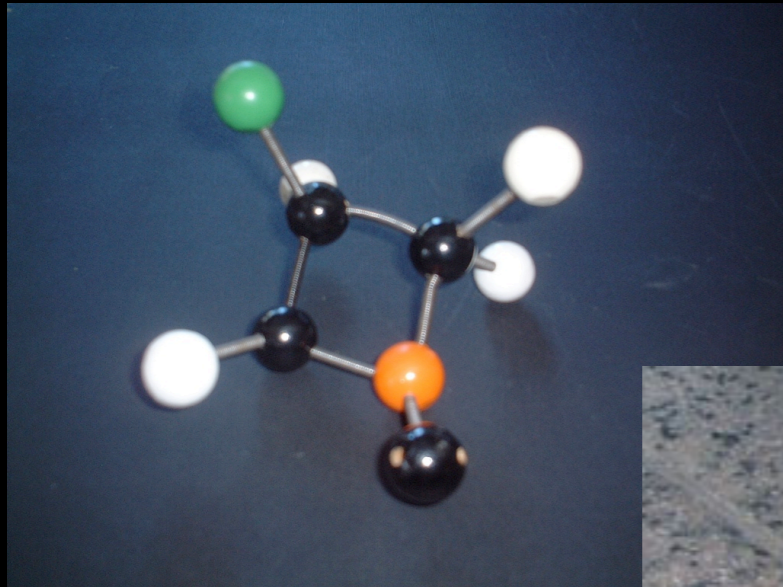


furniture factory (yeonjoo oh)



flatCAD (gabe johnson)





posey (michael weller)
hub-and-strut geometry construction kit
that senses connections and angles



roBlocks (eric schweikardt)

sensors: light, sounds, touch, knob, ...

actuators: tread, twisty, display, ...

operators: average, min, threshold, ...

the robot is the program

many different tools
don't communicate well
work at a low (hardware-specific) level

arduino, wiring, processing, MaxMSP, IC, PBASIC, ...

Eagle, ExpressPCB, ...

SolidWorks, Maya, Rhino, ...

leads to getting lost in uninteresting detail

assembler
high level languages
higher level languages
very high level languages
domain oriented languages

different languages are good for different things.
some are better than others for prototyping.
this is not entirely a matter of taste.

compilers take high-level description
of desired behavior and generate
implementation in a specific
technology

C compiler
silicon compiler
mechanical compiler?
mechatronic compiler?

sketchy languages:

very quick and low-cost exploring
no/low overhead

challenges:

how to describe what you want?
integrated mechatronic and software designs
designing for 10 , 10^2 , 10^3
supporting debugging