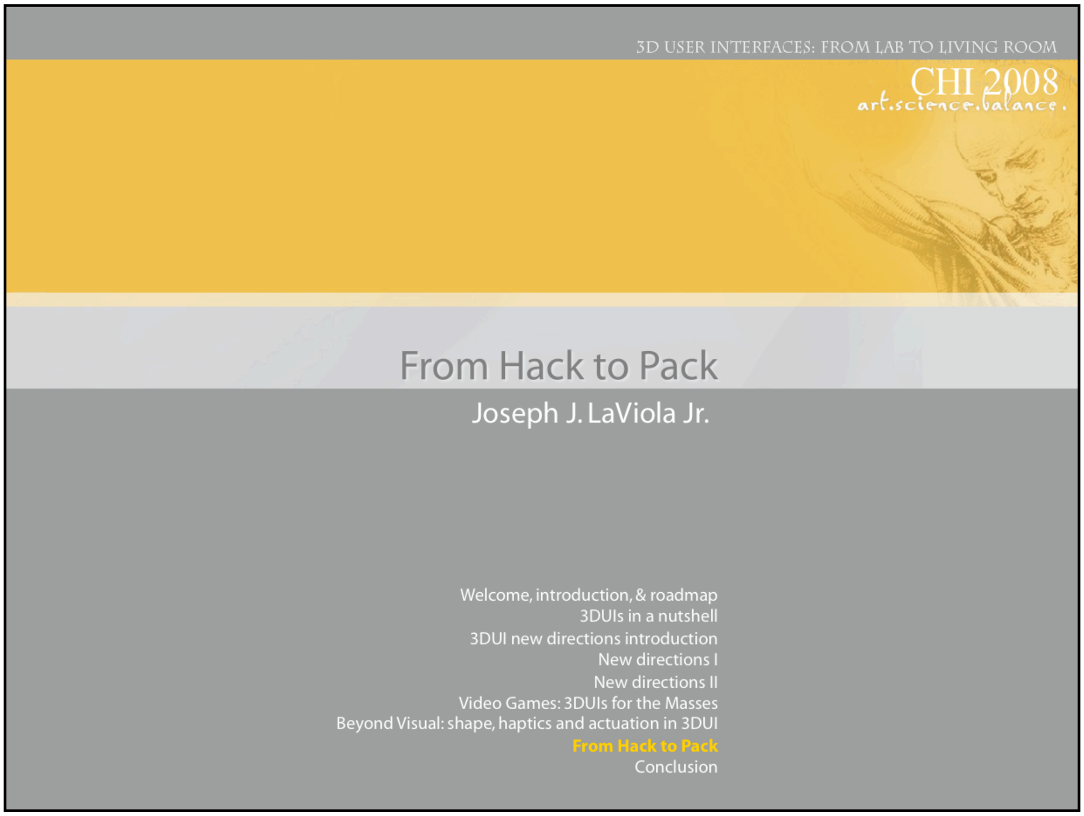
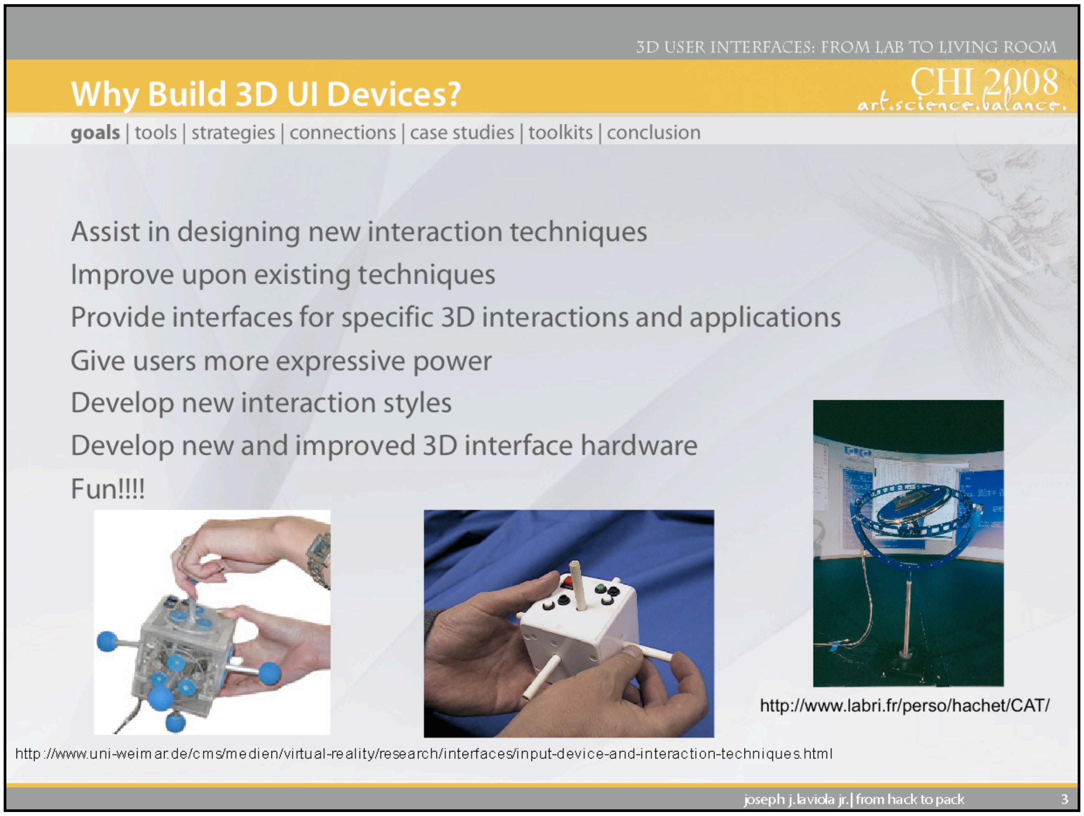


From Hack to Pack

Joseph J. LaViola Jr.
Assistant Professor
University of Central Florida
School of EECS
4000 Central Florida Blvd.
Orlando, FL 32816

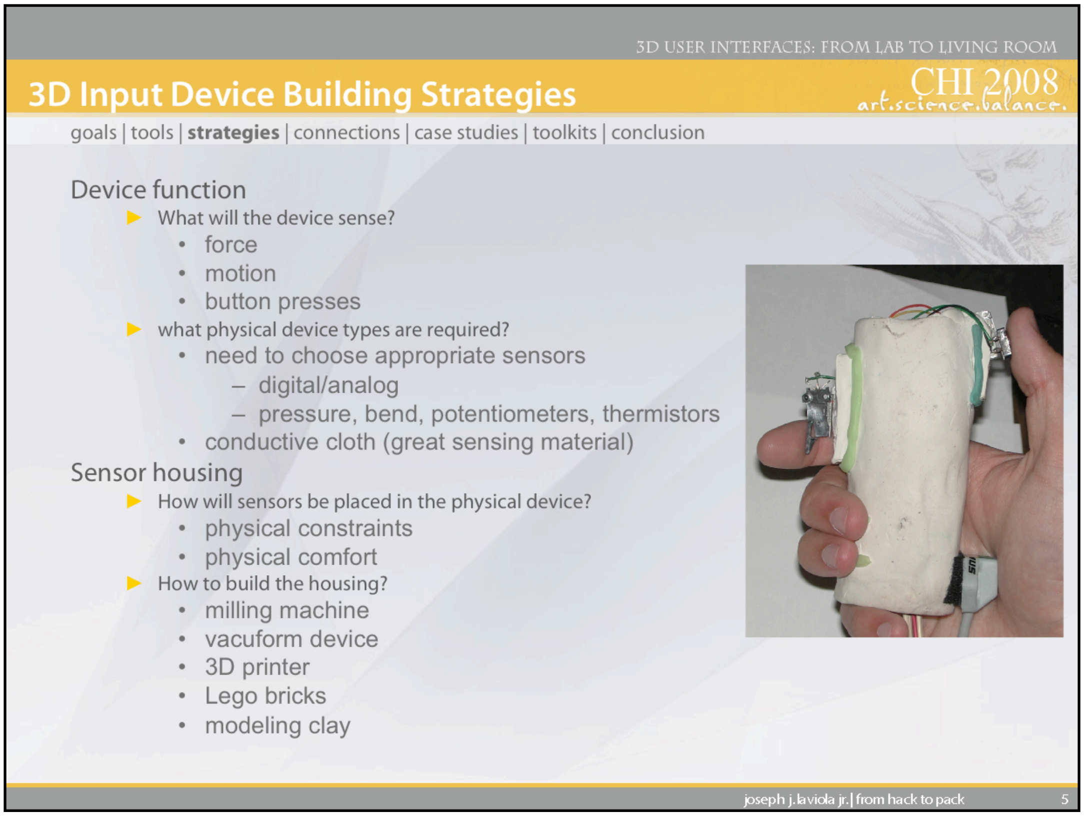
Email: jjl@cs.ucf.edu
http://www.eecs.ucf.edu/~jjl/

In this lecture, we will be talking about constructing custom built input device hardware. We will discuss why it is important to build custom-made 3D input devices. We will discuss the tools that you need to construct these devices and also some strategies for building the physical hardware and connecting them to a computer. We will examine several custom built devices through three case studies. Finally, we will discuss two input device prototyping toolkits that make it very easy to build custom input devices.



Why do would we ever want to build our own 3D input devices. Besides it being a fun and rewarding experience, there are several reasons to do so in the context of 3D user interface design. First, building new and interesting input devices can help user interface developers and researchers design novel interaction techniques and improve on existing techniques. There are many cases where a given interaction technique was designed using a particular input device and a new device would make significant improvements to the technique in user performance, ease of use, and ergonomics. Second, there are many instances in 3D user interfaces and 3D applications where that require specific forms of interaction and a device well suited to these forms may not be available. Thus, building a custom input device would greatly improve usability for that particular technique or application. Third, as a general rule, we always want to find new and innovative ways to interact with computers, especially in 3D, and custom built devices are way to explore that space.

Building custom made input devices requires a variety of different components including microcontrollers, device controls such as buttons, sliders, switches, and various sensors. Also needed are cables to connect the device to the computer, power supplies, and some type of housing for the electronics and the device itself. A soldering iron is also a common tool to fuse connections and a circuit breadboard for prototyping electronics.



There are several strategies and guidelines to follow when building 3D input devices. Having a good plan of attack and answers to the following questions will make it much easier to build a device.

What will the device sense?

It is import to know what the device is supposed to do. Will it just have several buttons or will it sense motion, force, temperature, etc…

What physical device types are required?

Once you have an idea for what the device is doing to do, you need to determine how it is going to do it. For example, will the device need to convey digital information or analog or both. What kind of sensors will be required for the device to observe its surroundings? Will it need bend sensors, accelerometers, potentiometers, etc… A great sensing material is conductive cloth due to its flexibility and low cost. We will see examples of how conductive cloth is used in building custom devices later on in the lecture.

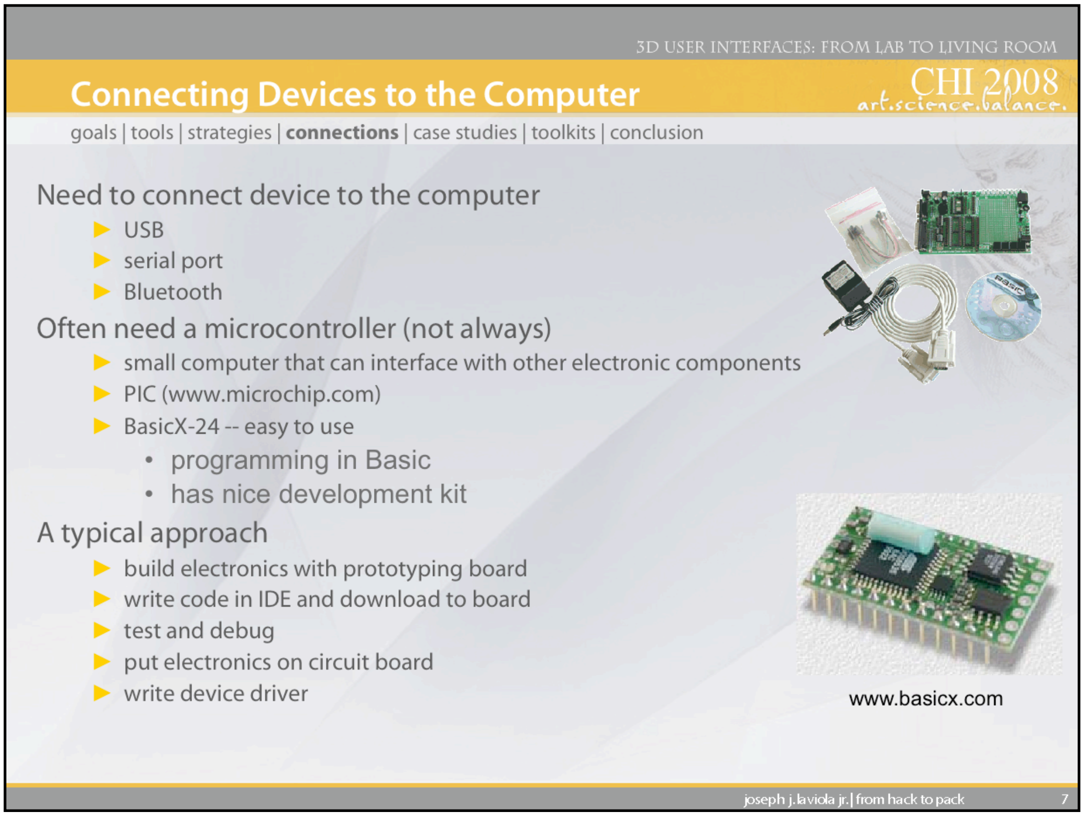
How will the sensors and buttons be placed in the physical device?

The device is going to require some type of housing and it is important to ensure that any controls the user actively must invoke are placed in or on the physical housing so that the user is comfortable and there is no undue physical strain.

How to build the physical housing of the device?

There are many different approaches to building the device housing. A milling machine or vacuform device would probably do the best job (along with some 3D modeling software). Alas, not everyone has access to these machines. Less expensive alternative include modeling clay and Lego bricks.

Another important consideration when building a input device is ergonomics. Good ergonomic design is crucial when constructing input devices. The device should be lightweight so as to avoid fatigue and undue strain. It should also be simple to use and make it easy to reach all of the buttons and controls. The picture in the slide shows an input device called the CyberGrasp. This device provides force feedback on the users fingers. Unfortunately, the device has rather poor ergonomic design. However, this is more a function of the state of the art in haptic technology.

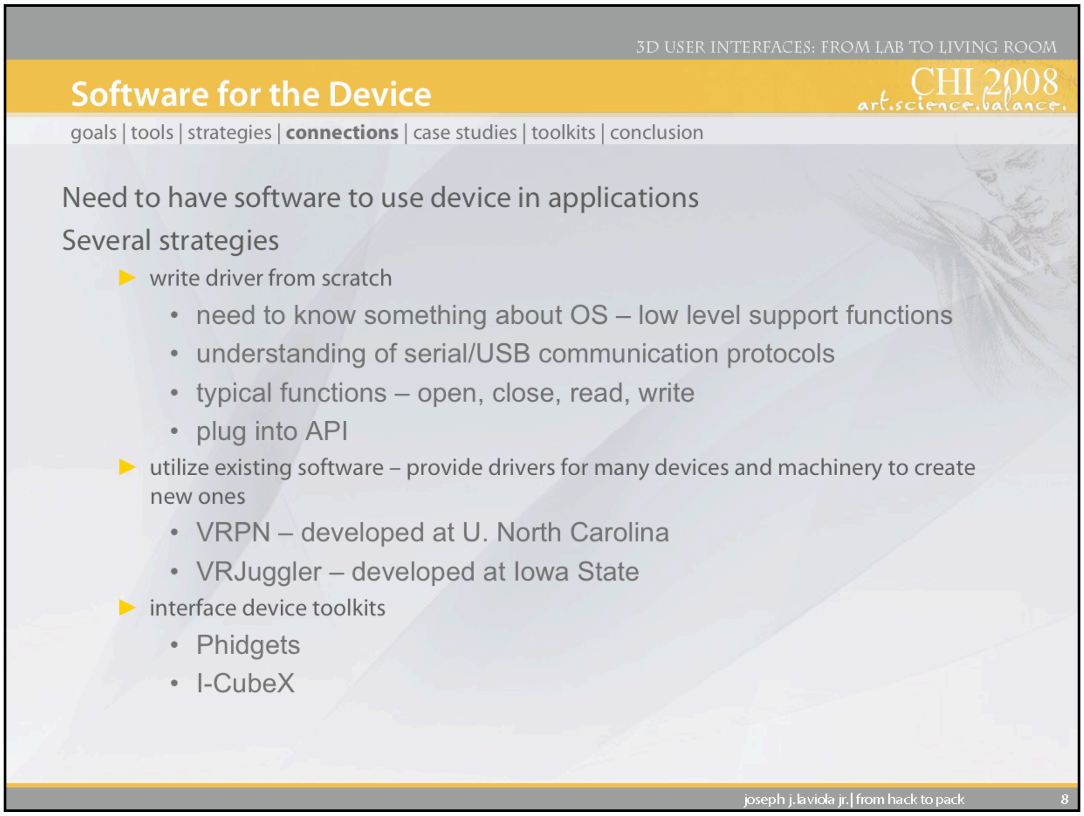


When building input devices, we must have as way to connect them to the computer. There are several different approaches to doing so including USB, serial port, and Bluetooth. Very often a microcontroller is needed as a mechanism for communicating to the computer from the physical device. A microcontrollers is simply a small computer that can interface with various electronic components. There is a wide variety of microcontrollers on the market today varying in size, functionality, and power consumption. Two of the most common are PIC and BasicX microcontrollers. They both have development kits and are programmed with either Basic or C.

Typically, building a device consists of building the electronics on a prototyping board, writing the necessary logic, and downloaded it to the board for testing and debugging. Once testing is done, the electronics could be put on a circuit board. A device driver then would need to be written so the computer could communicate with the device.

A great resource for getting started with microcontrollers is

Forman and Lawson, Building Physical Interfaces: Making Computer Graphics Interactive, Course #30, SIGGRAPH 2003.



Once the device is built and along with the necessary electronics, software is needed to interface to it so that people can use the device in their application. One approach is to write the device driver from scratch. Typically one has to know something about the operating system as well as understanding the appropriate communication protocols (serial/USB/Bluetooth). Functions that are often required are open, close, read, and write. The device driver can then be incorporated into an API so developers and researchers can use it.

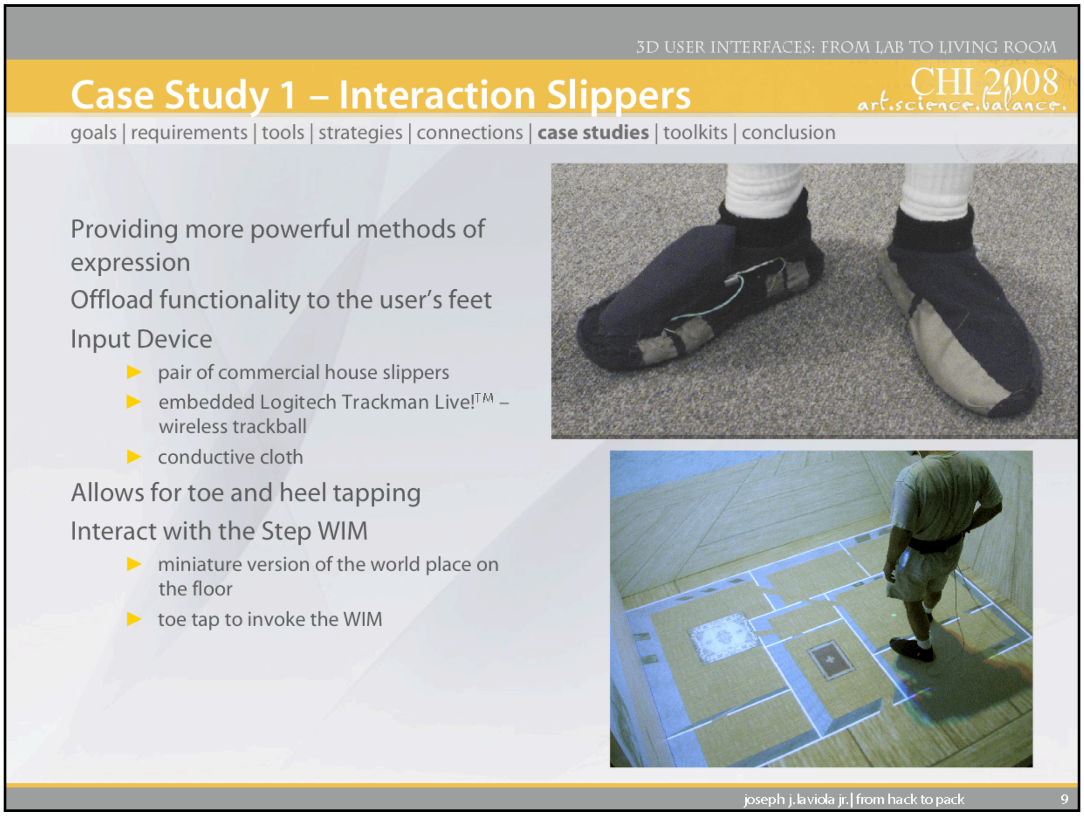
A better approach is to use existing software to assist in getting the device ready for use in applications. There are several software frameworks that make it easier to create device drivers. For example, VRPN, developed at the University of North Carolina, Chapel Hill, provides a framework for connecting devices to applications. It currently supports many devices and provides infrastructure to create new drivers. Another example is VR Juggler, developed at Iowa State.

Probably the best approach is to make use of interface device toolkits. We will look at two examples of them later in the lecture.

References:

http://www.cs.unc.edu/Research/vrpn/

http://www.vrjuggler.org/



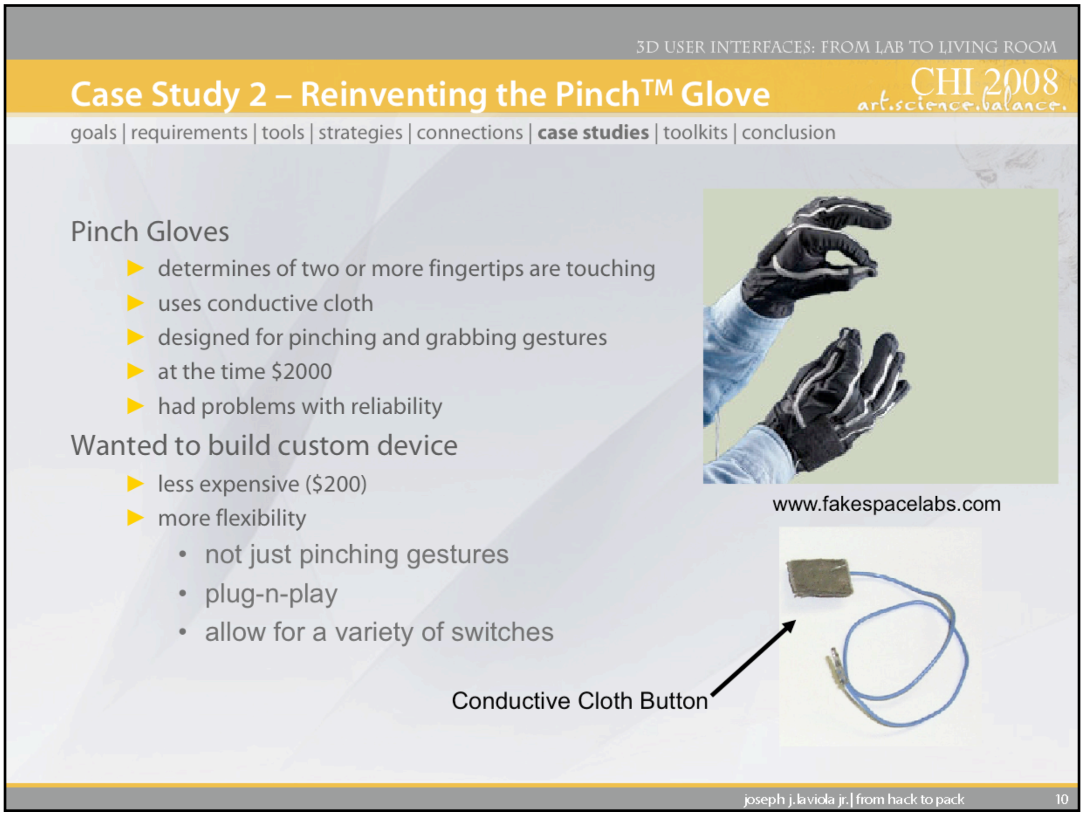
In our first case study, we look at the Interaction Slippers, a device for interaction in CAVEs, specifically the Step WIM.

The Step WIM is a interaction widget for quickly navigating through a virtual environment. It is a miniature version of the world placed underneath the user's feet and acts as an augmented roadmap. The user can either walk around the Step WIM to get a better understanding of the virtual world or navigate to a specific place by simply walking to a desired location in the WIM and invoking a scaling command, causing the Step WIM to animate, scaling up around the user's feet, thereby seamlessly transporting the user to the specified location.

In order to invoke, navigate and dismiss the Step WIM, users wear a pair of slippers (slippers with an imbedded wireless mouse) which gives them the ability to perform toe and heel tapping. To invoke the Step WIM, users simply tap their toes together. The device uses conductive cloth so the buttons fit easily onto the slippers. A Logitech Trackman Live is imbedded into a pouch on top of the right slipper. The beauty of this design is that no special device driver is needed. Developers can simply use mouse button events.

Reference:

LaViola, J., Zeleznik, R., Acevedo, D., and Keefe, D. "Hands-Free Multi-Scale Navigation in Virtual Environments", *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, 9-15, March 2001.



In our second case study, we will look at how trying to reverse engineer a commercially available input device provided for a variety of new input devices and interaction techniques. The Pinch Glove is an input device that knows when two or more fingertips are touching. It was designed to detect simple pinching and grasping postures. At the time of its popularity (the late 1990s), a pair of Pinch gloves cost $2000. They often had problems with reliability as the conductive cloth used in the device would fade, losing its conductivity. Thus, we decided to build our own Pinch Glove. However, we also wanted more flexibility beyond simple pinching postures. Thus, we designed a system where one could easily plug-n-play with different interface components. Our approach cost only $200 and it lets us to create several different devices and interaction techniques.

Reference:

LaViola, J., Keefe, D., Acevedo, D., and Zeleznik, R. “Case Studies in Building Custom Input Devices for Virtual Environment Interaction”, Proceedings of the *IEEE VR 2004 Workshop on Beyond Wand and Glove-Based Interaction*, 67-71, March 2004.

Building the Custom Device

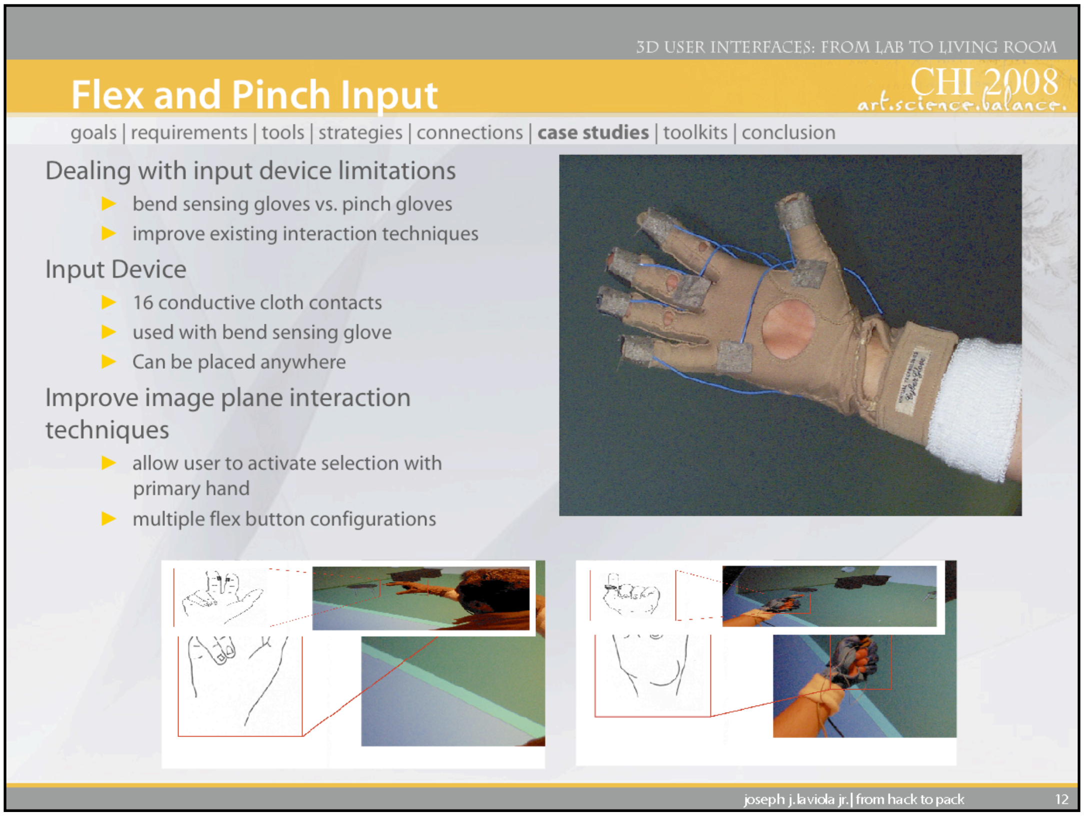
- Uses conductive cloth (other buttons as well)
- 16 connections provides many different combinations

PART	USAGE
PIC16C63	8 bit microcontroller with built in UART primary interface chip
16x 20K ohm resistors	pull up resistors
16x 2K ohm resistors	protection resistors
16x 1000pF capacitors	protection capacitors
LT1081 RS232 driver/ receiver	converts 5 volt PIC output to RS232 levels

Electronics Pseudocode

```
initializeMemory()
for each pin
    do set a voltage on pin;
        for each (otherpin > pin)
            do check for voltage on otherpin;
                if (pin status changed)
                 increment keycode timer;
                if (timer expired)
                 set keycode status;
                 transmit status change;
```

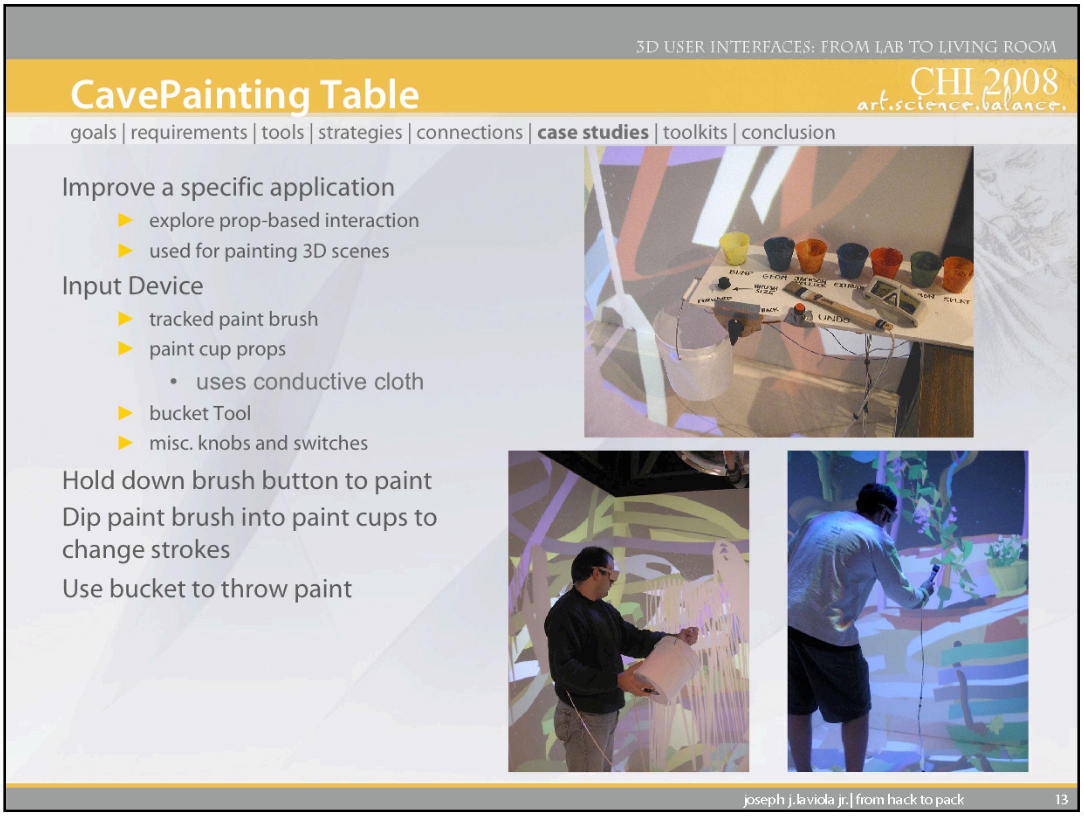

The most important part of building our custom device was ensuring that the electronics allowed for generality and easy plug-n-play with different device components. The parts and electronic pseudo code to drive the device are shown in the slide.



The Flex and Pinch input system is a custom built device which takes the functionality of the Pinch Glove system and combines it with the bend sensing technology of a data glove. The pinch buttons are made from conductive cloth and can be placed anywhere on the bend sensing glove.

Reference:

LaViola, J. and Zeleznik, R. “Flex and Pinch: A Case Study of Whole Hand Input Design for Virtual Environment Interaction”, *Proceedings of the Second IASTED International Conference on Computer Graphics and Imaging*, 221-225, October 1999.



The Painting Table is another example of a custom built device that spawned from our work in reverse engineering the Pinch glove. This input device was used in the CavePainting application, a system for painting 3D scenes in a virtual environment. The device uses a set of conductive cloth contacts as well as traditional buttons and digital sliders. Users can dip the paint brush prop into the colored cups to change brush strokes. The bucket is used to throw paint around the virtual canvas.

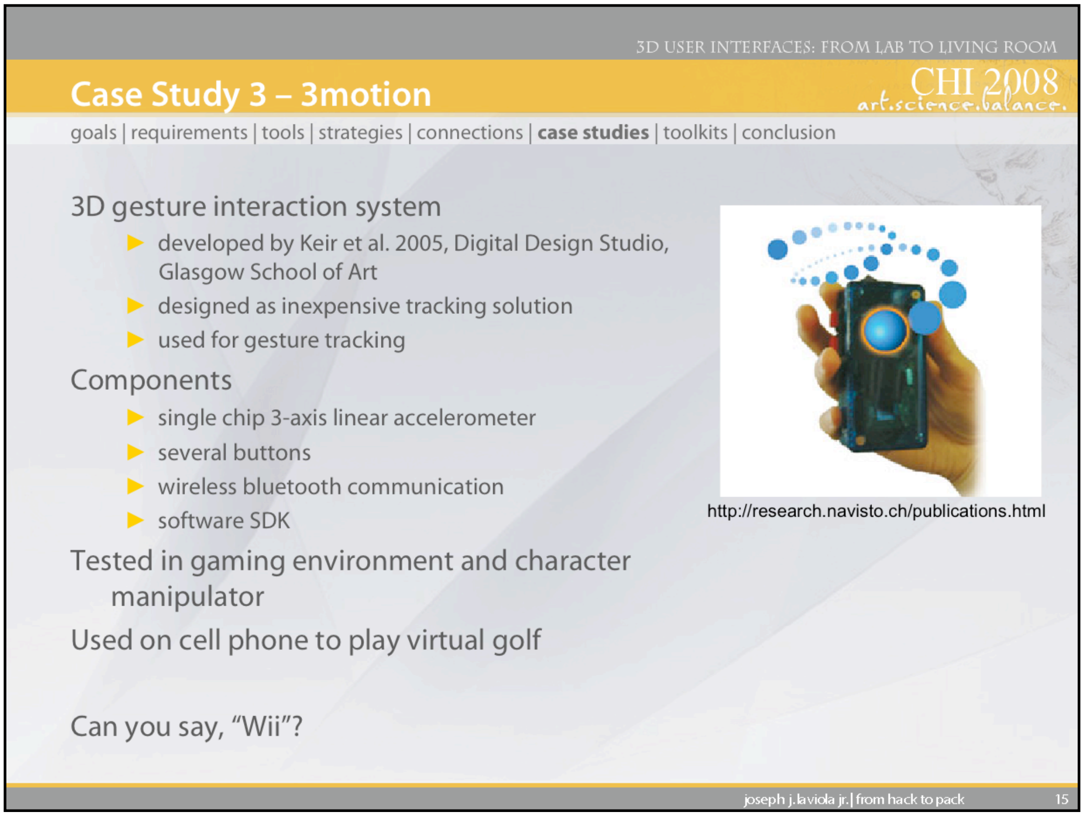
References:

Keefe, D., Acevedo, D., Moscovich, T., Laidlaw, D., and LaViola, J. “CavePainting: A Fully Immersive 3D Artistic Medium and Interactive Experience”, Proceedings of the 2001 Symposium on Interactive 3D Graphics, 85-93, 2001.

The FingerSleeve, is the last custom built device that was built as part of the reverse engineering Pinch glove work. It can be worn on the index finger of either the left or right hand. The frame is made out of an elastic fabric and a small piece of flexible plastic that can be found at any arts and crafts store. The fabric is sewn into a sleeve with a varying diameter that fits snuggly for most users. The plastic is sewn onto the front of the sleeve to provide a solid mount for pop through buttons -- buttons which have two clearly distinguished activation states corresponding to light and firm finger pressure. The buttons are glued into place a few millimeters apart on top of the plastic. Finally, a 6 DOF tracker is secured to the back of the sleeve using Velcro. This device allowed for the creation of three interaction techniques, ZoomBack, LaserGrab, and Snapshot.

References:

Zeleznik, R., LaViola, J., Acevedo, D., and Keefe, D. “Pop Through Button Devices for VE Navigation and Interaction”, *Proceedings of IEEE Virtual Reality 2002*, 127-134, March 2002.

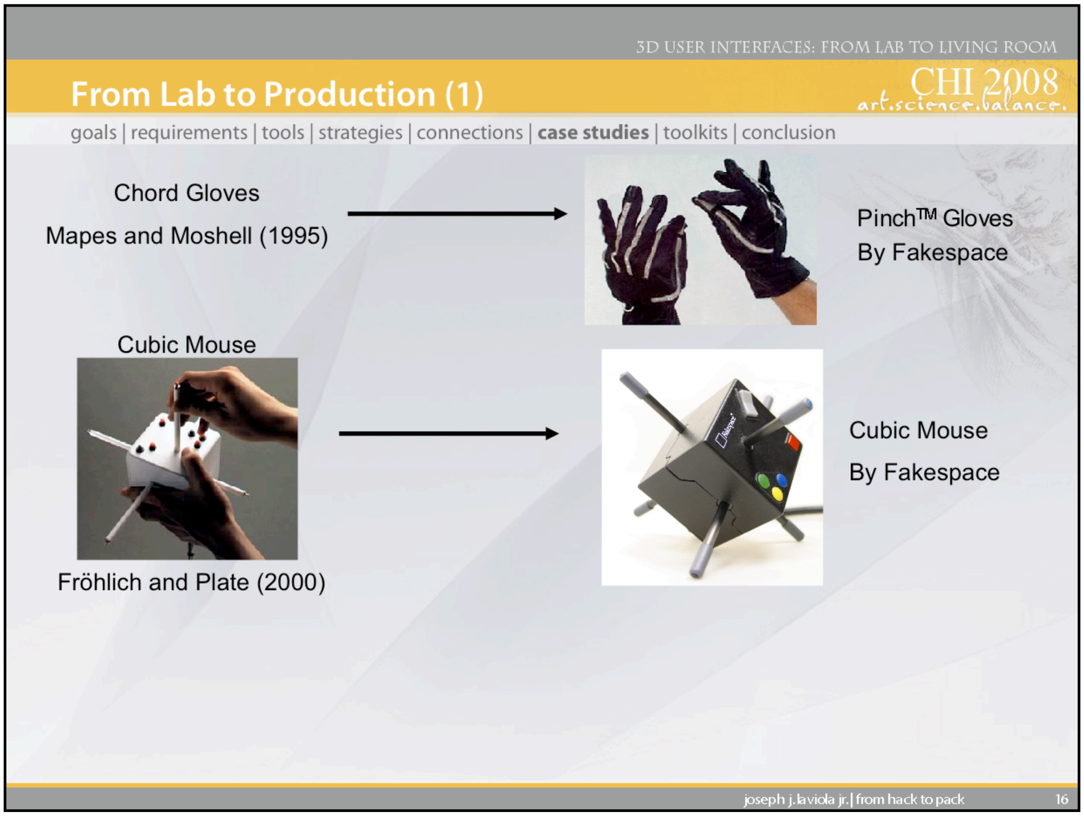


In this last case study, we examine a custom built 3D gesture interaction system. The device was designed to be an inexpensive non-referenced tracking solution used in gaming environments. It consisted of a single chip 3-axis linear accelerometer with several buttons and wireless Bluetooth communication. One of the main goals of the project was to minimize the cost of the device so it could be mass manufactured inexpensively. The device is small and lightweight and was tested in several gaming environments.

Of note, this device looks very similar to the Nintendo Wii controller. There was an attempt to commercialize the device shortly before the Wii came out.

References:

Keir, P., Payne, J., Elgoyhen, J., Horner, M., Naef, M., and Anderson, P. Gesture-recognition with Non-referenced Tracking. In *Proceedings of the 3D User interfaces (3DUI'06),* IEEE Computer Society, 151-158, 2006.

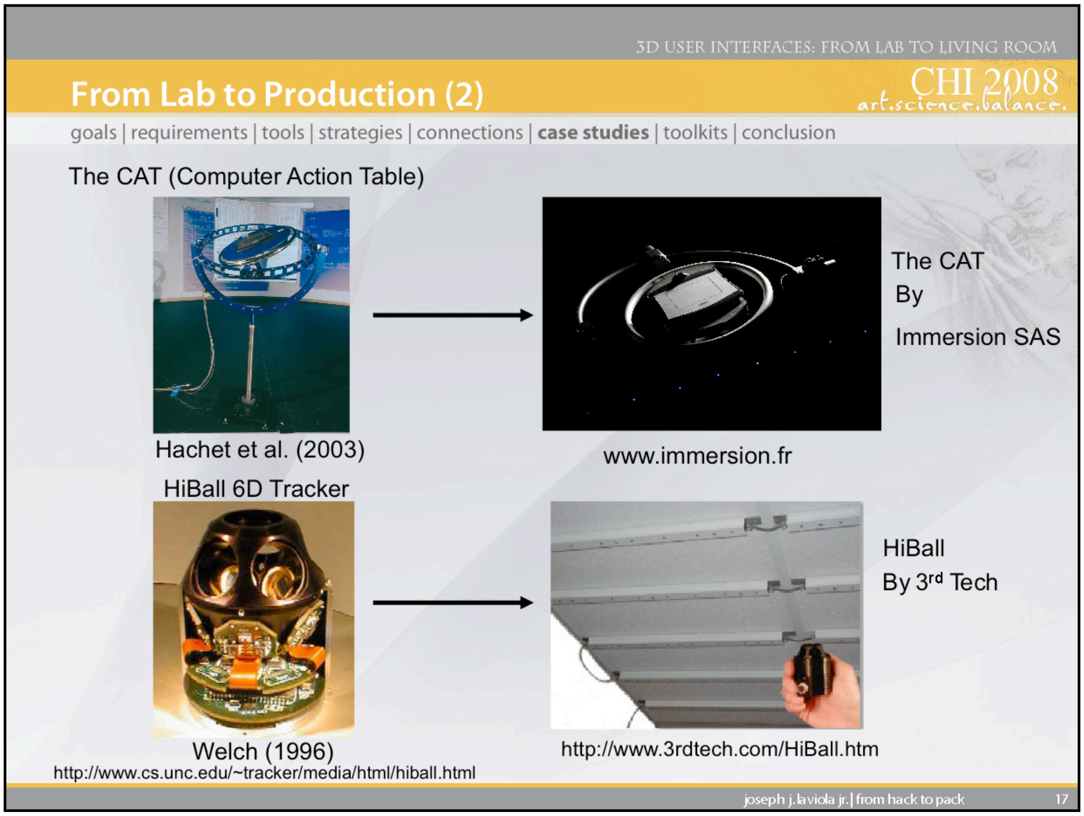


There have been a number of cases where custom built input devices from research laboratories have been successfully commercialized. The next two slides show some examples.

References:

Mapes, Daniel P. and Moshell, J. M. A Two-Handed Interface for Object Manipulation in Virtual Environments, *Presence: Teleoperators and Virtual Environments*. Vol. 4(4):403-416. Fall 1995.

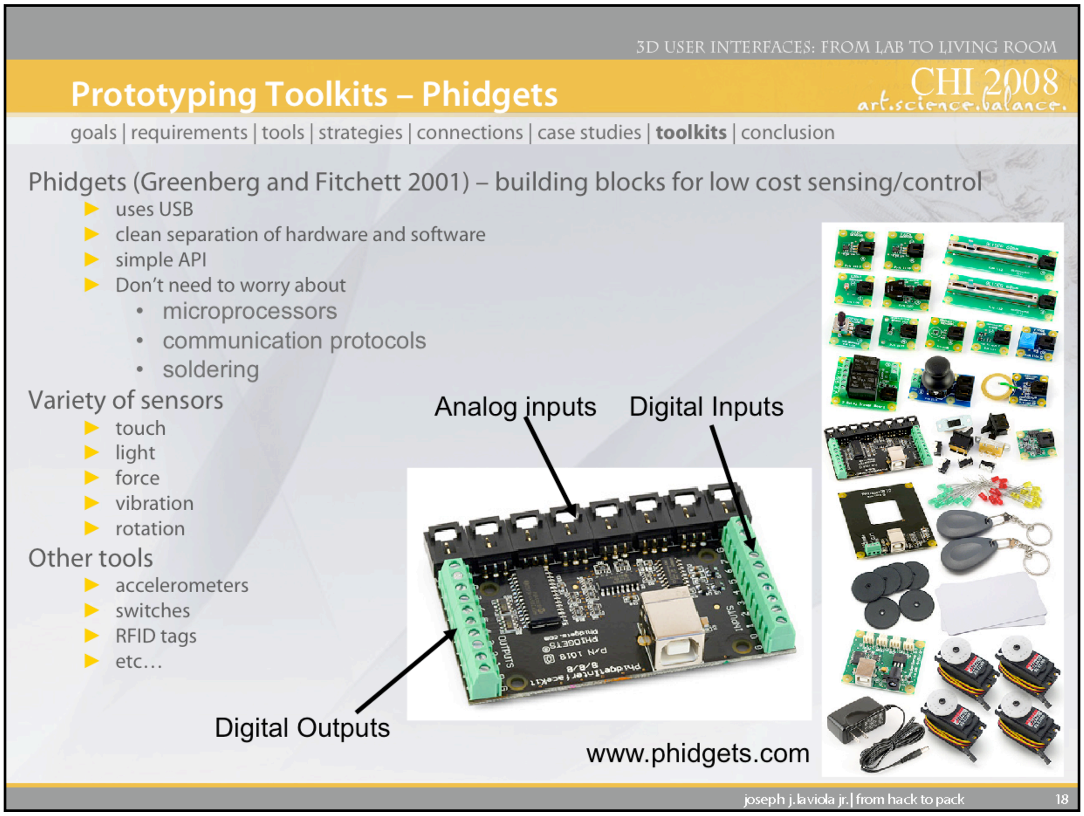
Fröhlich Bernd and John Plate. The Cubic Mouse: A New Device for Three-Dimensional Input. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2000), 526-531, 2000.



References:

Hachet, M., Guitton, P., and Reuter, P. The CAT for Efficient 2D and 3D Interaction As An Alternative to Mouse Adaptations. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (VRST '03), 225-112, 2003.

Welch, Greg, Gary Bishop, Leandra Vicci, Stephen Brumback, Kurtis Keller, and D'nardo Colucci High-Performance Wide-Area Optical Tracking: The HiBall Tracking System, *Presence: Teleoperators and Virtual Environments* 10(1): 1-21, 2001.

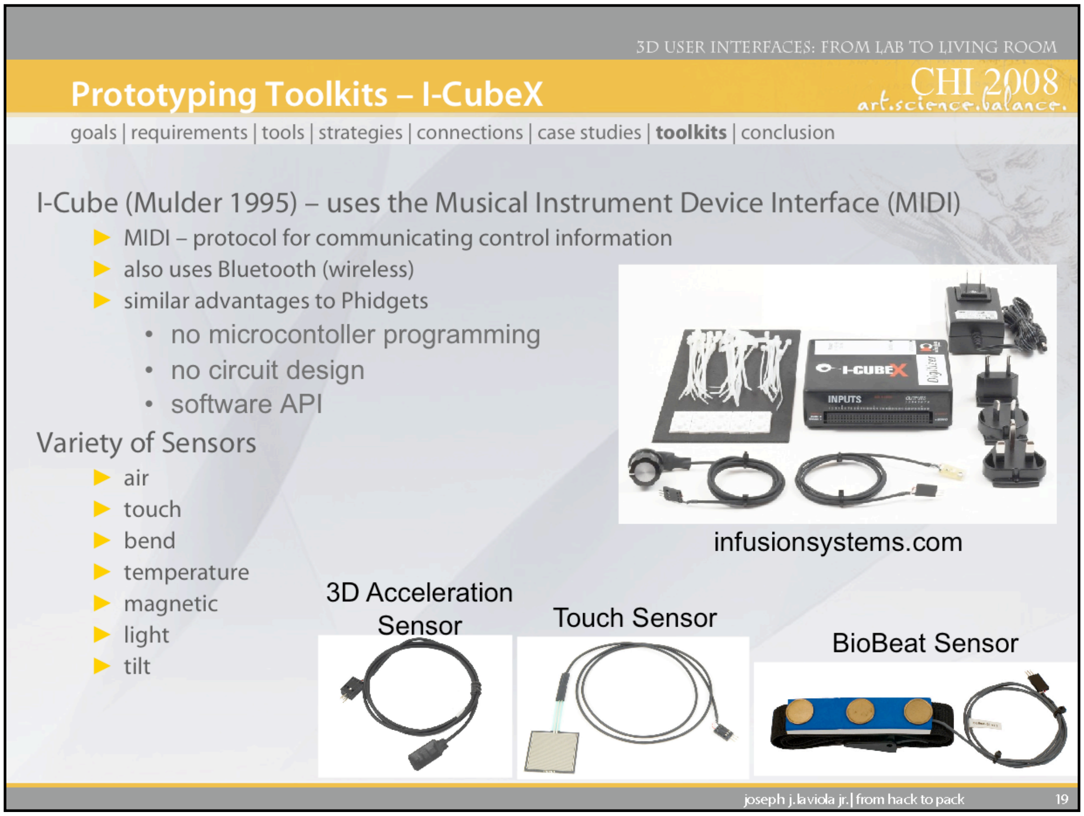


We have seen that there are many things to think about when designing and building a custom made input device. To make it easier to develop new input devices, there are a number of prototyping toolkits that mask unwanted details in the development process. Phidgets are one example of a set of building blocks for developing input devices. Phidgets provide a variety of sensors and other tools to make it easy to create input devices without having to worry about microcontrollers, communication protocols, device drivers, and soldering.

References:

www.phidgets.com

Greenberg, S. and Fitchett, C. Phidgets: Easy Development of Physical Interfaces through Physical Widgets. *Proceedings of the UIST 2001 14th Annual ACM Symposium on User Interface Software and Technology,* 209-218, 2001.



Another toolkit for prototyping and building new input devices is the I-CubeX system. I-CubeX makes use of the musical instrument device interface (MIDI). It also utilizes Bluetooth for wireless communication. The I-CubeX toolkit has similar advantages to Phidgets and a plethora of sensors comparable to Phidgets as well.

References:

infusionsystems.com

Mulder, Axel. The I-Cube System: Moving Toward Sensor Technology for Artists. *Proceedings of the 6th International Symposium on Electronic Art,* Montreal, QC, Canada, 1995.

Building novel 3D input devices has many advantages. These custom built devices can assist in the development of new interaction techniques, help to improve on existing techniques, and provide interfaces for specific 3D interactions and applications. In addition, these new devices can give users more expressive power and researchers with assistance in coming up with new interaction styles. In the past, one had to have electrical engineering knowledge to build new devices. However, today's input device toolkits (e.g., Phidgets and I-CubeX), provide UI developers and researchers with all the necessary components in both hardware and software to quickly get starting in building new devices.