Prior Work in Touch Prototyping

Valkyrie Savage, UC Berkeley BiD

26 October 2011

Designers have more or less settled on a process for creating the look and feel of new physical artifact. For most, this involves either the traditional foam carving for low-fidelity prototypes or the more recent 3D printing for more exact ones. However, it is important that designers be able to communicate how the object should *work* at an early stage in order to get feedback from customers and future users. It should be possible to add all kinds of input sensitivities to a prototype object, but particularly touch.

There have been a number of people coming from many angles to think at the problem of properly interactive prototypes. Beginning with the Phidgets [7] physical interface system in 2001, we have seen an onslaught of people working to make buttons [5] [2] [9] [3] [9] [8] a part of the prototyping process in stages ranging from the cardboard, low-fidelity stage [9] to a finished product stage [2]. There has also been some work to include accelerometers [4] and other sensing devices as a part of the design process via the process of programming by demonstration. Many of the systems emphasize the designer, although a few [2] [3] are still geared more towards programmers. The interfaces for connecting the parts of these systems range from addition of VB code to a program [7] to a GUI toolkit for a designer to inject keyboard and mouse events into the UI of an existing and unistrumented computer program [9]. This second approach is valuable for broadening the reach of the BOXES platform.

However, accelerometers and buttons aren't enough any more. With the ubiquity of the iPhone and similar systems, touch is becoming an increasingly important and attractive method of input for new devices. A few groups have begun to look into touch interactions for arbitrary objects through different methods. The use of time domain reflectometers, previously employed to find breaks in long underground cables, has been explored as a method of finding a touch point along a one-dimensional conductive wire [11]. This approach involves the use of a large, expensive, and finicky time-domain reflectometer, which is not a reasonable acquisition for most

designers to justify. This approach also does not permit the sensing of 2D touch points, except when one for instance folds a wire into a Hilbert curve on a plane. Additionally, there is no interface for a designer to prototype interactions: she can only sense the touch. Other systems, like the Kinect [10] allow for touch on arbitrary objects, but they, too, lack an interface for designers.

Much of the touch-sensing world is still off-limits to non-programmers. MacLean, et al., [6] back in 1990 discussed the differences between workers, tinkerers, programmers, and handymen as potential audiences for new technologies. Workers are those who wish to complete a goal using a technology and who have no interest in its low-level workings. Tinkerers are mainly goal-oriented, but they dabble in the core pieces of systems as the situation dictates. Programmers are deep in the system, working to grok its functionality and possibilities. Designers are not, in general, programmers: more systems need to be created which focus on workers.

My proposed system will draw inspiration from all these papers, allowing a designer to augment a prototype with touch input that can be easily mapped to existing program interactions. It will step beyond what has thus far been worked on to include the ability to print designer circuitry for touch sensing, easily prototype touch interactions to computer-based software, and allow for sensing touch in two dimensions rather than one.

How can we use small, cheaply-available, and hobbyist-friendly tools to help designers prototype touch interactions with arbitrary objects? It is obvious that Arduinos in concert with the MPR121 can be used to sense digital change in capacitance on copper wires, and it should be extensible to both analog signal (i.e. pressure) and more dimensions with some thought and engineering. All that is required is creativity and a lot of copper tape.

By adding the ability to easily map touch interactions to computer interactions, designers will have an easier time of determining whether the ergonomics of their products are sensible. Additionally, adding the ability to create attractive-looking touch input sensors for arbitrary objects will allow for creation of higher-fidelity prototypes earlier in a designers development cycle.

I will also re-implement the piece of the BOXES framework that is used to interact directly with computer-based programs to extend the interaction prototyping. It will consist of a glass-window overlay on the designers computer screen which will be able to inject keyboard and mouse events into an un-instrumented computer program, e.g. iTunes, and which contains an interface for mapping touch interactions on the device to interactions with the computer screen. However, it will be extended to allow for "capturing"

of events, rather than hand-assignment. This capturing will be used for both the input action and the output action.

Further, I want to explore the ability to "print" touch sensors in interesting shapes for arbitrary surfaces: there is a large market these days for "designer" circuitry, particularly for Lilypad Arduinos as they are used primarily for clothing, where style matters. I want to develop a system for designers to draw aesthetically pleasing touch sensor designs which can then be printed into copper foil using the BiD lab vinyl cutter. The copper foil has an adhesive backing, so it can be moved around at the designer's will.

I plan to experiment first on classmates, and then on people who typically prototype three-dimensional interactive objects. As there are a number of mechanical engineers in BiD, I plan to recruit from them and their friends. The study that I would do would likely involve one guided task and one free-form task where they could perhaps work with a prototype that they were currently working on. The most important results of the study, I imagine, would come from Liekert-scale satisfaction questions and free-form responses from the participants; timing data would not really be useful for any of the study beyond that the participants were able to complete the tasks within a reasonable amount of time.

Since designers are the designated population of the study, it seems reasonable that I perform my evaluation on them. I believe that observational studies will determine whether the tools are successful or usable for their stated purpose (prototyping touch interactions), as its findings will indicate whether users were successful and how well they felt they were able to use the tools.

Probably the biggest risk to consider in this research direction is whether someone will have beaten me to it! There seem to be many people these days working on touch interaction: notably the UIST paper on time-domain reflectography. It will be sad if I finish up and cannot publish anything new because someone else finished a month or two before me.

Another worry of mine is that it may take some significant time to tune the Arduino system to accept two-dimensional and pressure-sensitive input. I have already experienced some difficulties in getting two-dimensional sensing to work, caused I think by the sensitivity thresholds for the device.

If there is time, I'd certainly like to investigate a system similar to DisplayObjects [1], except perhaps one that combines picoprojectors with the Kinect rather than using the arcane marking system used in the paper. If it were possible to mock up screens and touch sensitivity concurrently on an arbitrary object, that would be a valuable contribution, indeed.

References

[1] Akaoka et al. Displayobjects: Prototyping functional physical interfaces on 3d styrofoam, paper, or cardboard models, 2010.

[2] Ballagas et al. istuff: A physical user interface toolkit for ubiquitous computing environments, 2003.

[3] Gellerson et al. Physical prototyping with smart-its, 2004.

[4] Hartmann et al. Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition, 2007.

[5] Lee et al. The calder toolkit: Wired and wireless components for rapidly prototyping interactive devices, 2004.

[6] MacLean et al. User-tailorable systems: Pressing the issues with buttons, 1990.

[7] Saul Greenberg and Chester Fitchett. Phidgets: Easy development of physical interfaces through physical widgets, 2001.

[8] Scott Hudson. Forming interactivity: A tool for rapid prototyping of physical interactivity, 2002.

[9] Scott Hudson and Jennifer Mankoff. Rapid construction of functioning physical interfaces from cardboard, thumbtacks, tin foil, and masking tape, 2006.

[10] Andrew D. Wilson. Using a depth camera as a touch sensor, 2010.

[11] Raphael Wimmer and Patrick Baudisch. Modular and deformable touch-sensitive surfaces based on time domain reflectometry, 2011.