The boundaries between 'the digital' and our everyday physical world are dissolving as we develop more physical ways of interacting with computing. This forum presents some of the topics discussed in the colorful multidisciplinary field of tangible and embodied interaction.

Eva Hornecker, Editor

Research with a Hacker Ethos: What DIY Means for Tangible Interaction Research

Amanda Williams
Wyld Collective Ltd. | meta@wyldco.com

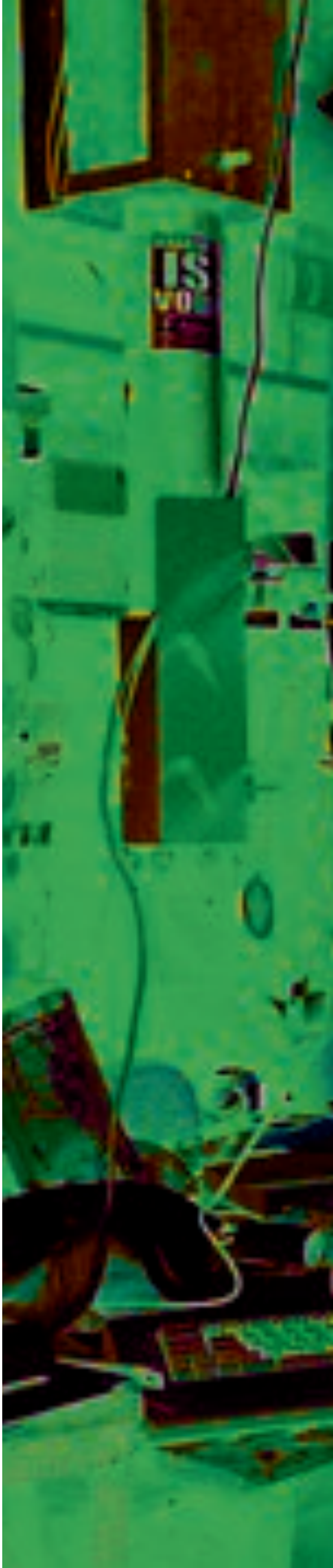
Alicia Gibb
NYCResistor | pipix@nycresistor.com

David Weekly
Hacker Dojo | dweekly@gmail.com

Let's be honest. Much of what we do as researchers in tangible interaction design is *hacking*. We reappropriate game controllers; we break open toys and cannibalize their parts; we bend and flex and stuff electronics into objects they were never meant to go into. We usually mention this hackery, briefly, in the implementation sections of our research papers and then move on to discuss their interactional affordances, which is where our innovation really lies, right?

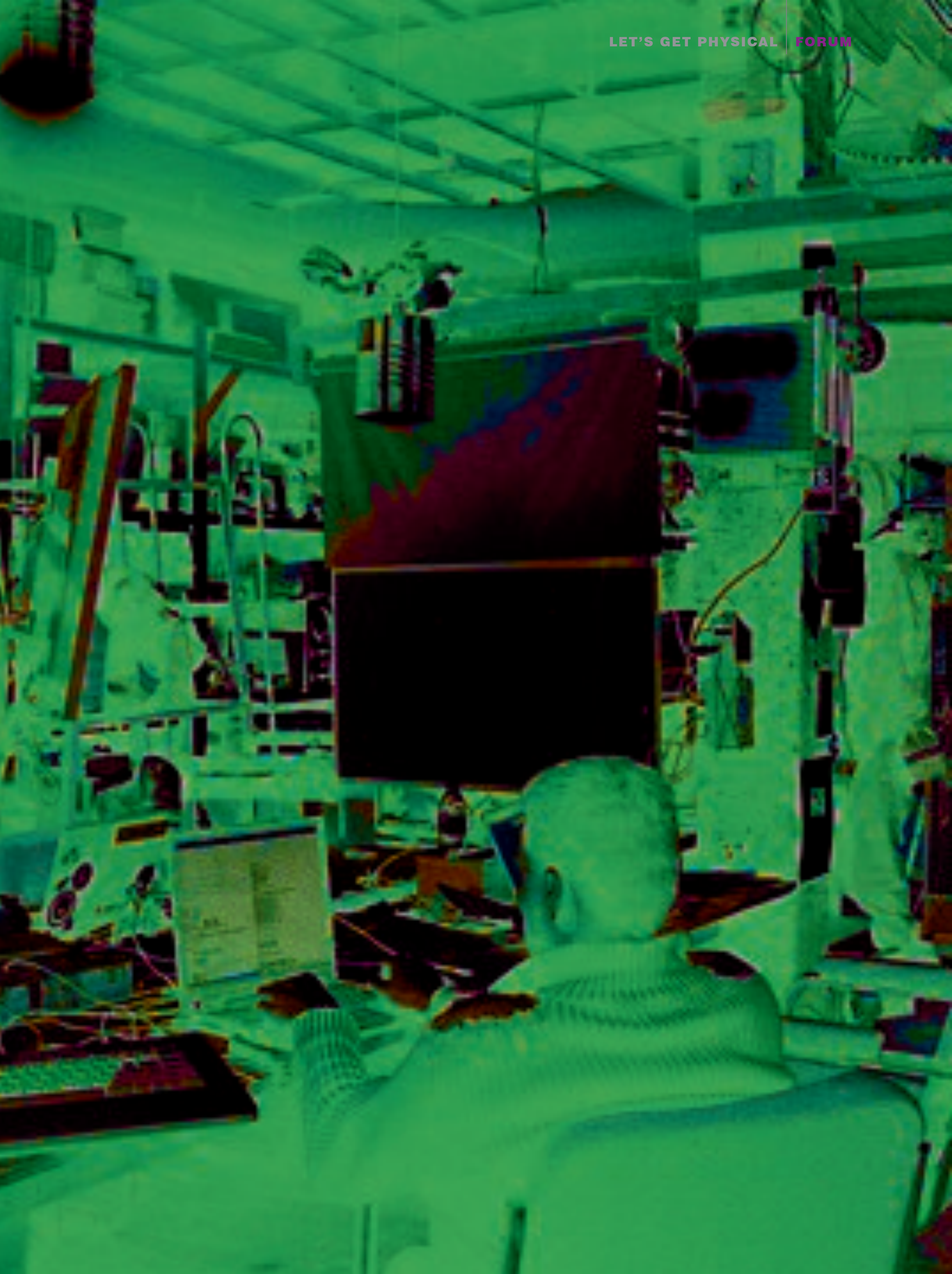
In this article, we will discuss the ways in which hacking and hackers [1] make our world go round. Widespread use of open source hardware tools like Arduino, and knowledge-sharing tools like Instructables, has made tangible interaction prototyping vastly easier over the course of just a few years.

A concrete example may help illustrate how the game has changed. One of the authors of this article designed a tangible prototype called Nimio in late 2005. It consisted of a collection of networked toys that sensed sound and motion, communicated with one another, and engaged users via colored LEDs [2]. No one else in our department had experience with tangibles, so we scoured *everywhere* for information, including sites for cosplayers and sex-toy manufacturers. Mostly we learned by trial and error, by casting molds over many weeks and earning a few chemical burns. For the electronics we used MICA motes, a technology developed at UC Berkeley that runs TinyOS, along with some sensor boards. While these motes were a significant achievement for wireless sensor networks, they were not intended for our purposes: distributed display and interaction, not just distributed sensing. We pried the



▸ Foulab, Montreal, Canada

Photograph by Amanda Williams



NIMIO IN 2005		NIMIO TODAY	
1 MICA2 mote	$120	1 Arduino Fio	$25
		1 Xbee	$25
1 sensor board	$60	1 ADXL335 triple axis accelerometer with breakout board	$25
		1 electret microphone with breakout board	$8
9 LEDs, 3 transistors, resistors, and some wire	$10	9 LEDs, resistors, and some wire (no transistors needed)	$8
TOTAL:	**$190**		**$91**

▸ Table 1: Cost comparison, Nimio, 2005 and today.

surface-mount LEDs off the boards and soldered ultra-bright LEDs and transistors to the bare solder pads. Though the interaction was simple, our physical implementation took several months and required us to familiarize ourselves with a large code base, irreversibly alter expensive hardware—and did we mention the chemical burns? Yet this was still cheaper and faster than building our own wireless, sensing, communicative hardware from scratch. We are, after all, interaction designers, not electrical engineers.

If we were doing the project today, we could build the electronics and software in just a few days, at less than half of the cost (see Table 1).

An Arduino-based Nimio would be not only cheaper but also much easier to implement (and this doesn't account for the physical fabrication, which would also be easier today—see our sidebar of design resources). Designed to accommodate a variety of uses and users, it would allow us to skip both the daunting code base and the X-Acto-knife circuit surgery.

The availability of such cheap and easy tools is a game changer. Not only does it allow us to prototype our ideas more quickly, cheaply, and easily (getting down to the interactions that we care about), but it also enables new formats for learning and communicating. It's hard to imagine, for example, TEI's popular studio format [3] thriving as it does without the available, affordable, and usable hardware platforms that have become popular in the past few years.

But these aren't just tools; they are systems created and supported by communities of smart and dedicated individuals. They are inexpensive and easy to use because of these communities and the design philosophy they espouse. Projects like Arduino, Instructables, Makerbot, and others arise out of the DIY (do it yourself) movement. We won't be able to address all aspects of DIY here, but we will focus on two important trends. First, we unpack how the tenets of open source hardware foster a proliferation of creative, inexpensive, and well-supported tools. Second, we explore the ways in which this has democratized innovation. Hackerspaces, easily accessible organizations that are beginning to rival many university and corporate labs in inventiveness, are growing quickly. These movements feed into one another in a virtuous cycle, with huge benefits for all of us.

Open Source Hardware

While the open source hardware movement is still young, it is growing rapidly. Publications like *Make* magazine, and events such as Maker Faire and the Open Hardware Summit, are helping to expand the overlapping communities around DIY, hackerspaces, and open source hardware.

Building open source hardware means giving others the ability to remanufacture, redistribute, and redesign the original work by sharing all the documentation necessary to completely rebuild the hardware. A definition of open source hardware [4], created by the community and publicly signed, also encourages the use of readily available components and open infrastructures, and the inclusion of accessible, redistributable documentation to maximize the potential for reappropriation. Further details specify that open source hardware must allow for the creation, sale, and distribution of derivatives, a stipulation that has allowed small OSH-based startups to flourish.

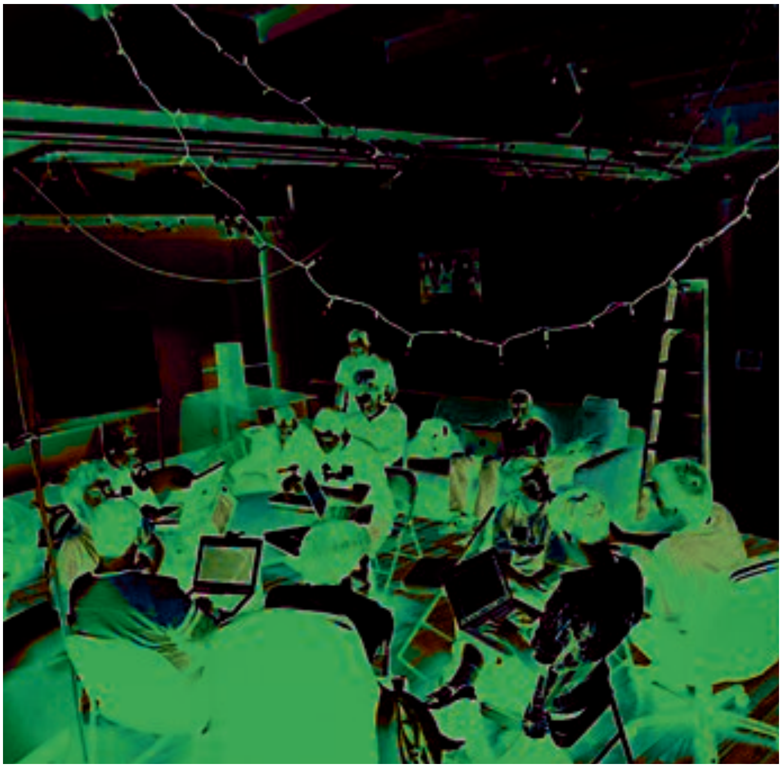
CERN used this definition to write an open source hardware license [5] under which it has released several projects. It believed there would be numerous advantages to using open source hardware in scientific research [6]. Here, we describe some of those advantages: First, the participation of a community helps projects last longer and grow autonomously; second, opening hardware designs to scrutiny makes them more robust and better supported; third, remanufacture allows for more improvement and variation on the original design; and last, open hardware can generate commercial value for projects.

The reasons for open source projects' longevity are straightforward: If organizational support for the project disappears, the design files can still get picked up and produced by others. At the 2011 Open Hardware Summit, Massimo Banzi from the Arduino team remarked that they knew Interaction Design Institute Ivrea (IDII) was closing, and they wanted to keep working on the Arduino platform. They open-sourced the hardware, allowing Arduino to live on after Ivrea closed [7].

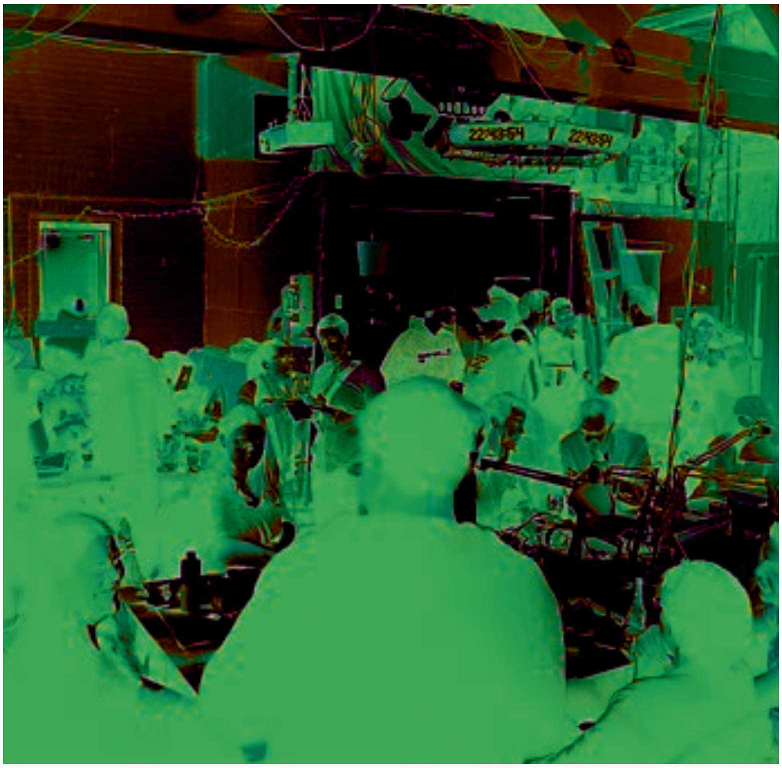
Opening up hardware, by making its design easier to scrutinize, generates feedback that helps make that hardware design more robust and allows for features to be added as the need for them becomes evident. Widespread review by an active and growing open source hardware community also makes it fairly easy for new users to find help and advice. (This is also true of open source software, and is similar in principle to academia's peer-review system; work has more legitimacy when a community of people who know the field can review and contribute to it.)

Open hardware also embodies a hacker ethos of improving, altering, and personalizing the items you own. It is usually competitively priced because the design files are available for anyone to rebuild and resell the product. This lowers barriers to entry for small businesses and labs, but at the same time, competition requires companies to continuously increase the quality and usability of their products. As these features attract an ever-growing community of active users, the benefits bootstrap themselves, creating a collective effort to add features to the product, improve it, or create specialized versions of it (such as the Lilypad Arduino for e-textiles). The license states that those alterations must also stay open, allowing the original designer to incorporate changes from the community and perpetuating the project's openness through its derivatives in a viral manner.

Last, and perhaps counterintui-



▸ Hacker Dojo, Mountain View, CA



▸ NYCResistor, Brooklyn, NY

Photograph on bottom by Trammell Hudson

Links and Resources

Tired of reading? Rather be building? Here are some of our favorite resources: free tools, inexpensive services, interesting events, knowledge-sharing, and more.

Open source tools:
arduino.cc
processing.org
openframeworks.cc
fritzing.org
openpicus.com

Big projects from small businesses:
reprap.org
labs.nortd.com/lasersaur/
makerbot.com
solidoodle.com
adafruit.com
littlebits.cc
buglabs.net

Small-scale and open fabrication:
ponoko.com
shapeways.com
thingiverse.com
upverter.com
eagleup.wordpress

Organizations and events:
dorkbot.org
makezine.com
makerfaire.com
openhardwaresummit.org
hackerspaces.org

More information:
instructables.com
archive.org/details/hackerspaces-the-beginning

tively, open source hardware can be an effective business model. In fact, the open source hardware community is largely made up of companies and start-ups; most speakers at the Open Hardware Summit in 2010 and 2011 represented companies creating open source hardware. Because big hardware projects involve electronic parts that require capital to purchase, they might have high start-up costs and benefit from economies of scale; even a small business is better equipped to take advantage of those conditions than an individual would be. A few key projects, notably the Arduino and LilyPad Arduino, were developed in universities and are now successful products. Others, such as Texas Instruments's Beagleboard and Sun Microsystems's SunSPOTS, have the backing of large corporations. But, notably, many small open source companies—such as SparkFun, MakerBot, BugLabs, Adafruit, and littleBits—are challenging the traditional business model of making money from trade secrets and closed technologies [8].

The open source hardware movement began as a way for people to share documentation for how hardware is built, a practice encouraged by the cooperative ethos of the larger DIY community. Sharing designs in an open manner makes the hardware more accessible to a broader audience. In turn, active and engaged members of this audience help make the hardware less expensive in a competitive marketplace, easier and faster to build, and more malleable to fix or reappropriate for unanticipated uses—all traits that make for powerful prototyping tools.

Hackerspaces

While the open source hardware movement is creating new models for stimulating and profiting from technological innovation, hackerspaces are creating new models for collocated creativity and invention. (Similar spaces may also be called maker spaces or fablabs. Though there are subtle differences in connotation, all share the same open vision of a collective whole.) Several years ago, rapid prototyping tools like CNC mills, laser cutters, and 3-D printers were affordable only to big-budget corporate and university research labs. Now open source versions of these tools are beginning to proliferate, and to be affordable to grassroots communities of ordinary people. A major barrier to democratic invention has crumbled.

The idea of hackerspaces stemmed originally from Germany's Chaos Computer Club, which in 1995 arguably created the first hackerspace, c-base in Berlin, which is still operating today. The goal of a hackerspace is to provide a community-oriented space, sustainably funded by members, that supports creation and exploration. These facilities tend to be open 24/7 to members and have high-speed Internet and plenty of seating. Additional facilities can vary widely, but as costs come down, they increasingly include fabrication tools like DIY 3-D printers, wood shops, and laser cutters. Most space in a hackerspace is communal; however, many allow members to have a shelf, locker, or cubby to store their personal projects. Membership in a hackerspace works (and is priced) similarly to a gym membership. Hackerspaces generally follow the Hackerspace Design Pattern Catalogue [9] to get started.

The authors of this article hail from New York City, Montreal, and the San Francisco Bay Area. We all have our own local hackerspaces, all of which share certain traits, but each with its own personality. Noisebridge in San Francisco and Hacker Dojo in Mountain View are representative of the larger hackerspaces in California; Dojo membership costs $100 per month, and Noisebridge contributions are on a voluntary donation basis. NYCResistor, New York City's first hackerspace, charges $75 per month, and Montreal's Foulab charges $50; both also cover expenses by offering classes or workshops in their spaces. Besides providing lab equipment and a friendly community, these workshop-friendly hackerspaces also have outreach and educational activities that are available to the general public.

On a typical evening at a hackerspace, you might encounter hardcore computer programmers, designers, technology novices, weavers, biologists, and Roomba tinkerers. Any one of those people might happen to encounter your project and offer an interesting idea or even the solution to a problem that has had you stymied. This intermixing creates dialogue and exposes participants to designs and concepts from many different arenas. And with a selection of helpful people around, you have lots of folks willing to test out a new interface or product idea. So it's exciting, but not terribly surprising, that the successful Makerbot Industries (which makes inexpensive, open source 3-D printers) got its start at the NYCResistor hackerspace.

Due to the large number of people, high churn, and wide variety of projects, hackerspaces are great places to observe some of the rapid developments currently occurring in interaction design, in terms of both software and hardware interaction. Within these spaces, members share best practices, software tool kits, APIs, github projects, components, and tools. Most hackerspaces have a sharing ethos similar to that of the open source movement—NYCResistor's motto is "we learn, share, and make things."

We do not view hackerspaces as direct competition for traditional corporate or university research labs. Both informal and formal types of research spaces are part of a growing ecosystem of technological innovation. They share members, and interesting projects move between them. Readers are vigorously encouraged to look up a location near them on hackerspaces.org and drop by for a tour. Noisebridge and Foulab, like many hackerspaces, host an open house on Tuesday nights; Hacker Dojo has regular "happy hours" on Fridays. NYCResistor is open on Thursday evenings and typically hosts a couple events per week that are open to the public.

Enabling a Tangible Turn

While much of the hype around open source projects and hackerspaces centers around individuals, grassroots communities, and small businesses, researchers can certainly benefit from getting involved in DIY communities. As our opening example suggests, the practices of DIY and open source hardware—and the tools they produce—are powerful enablers of exactly the kind of quick, iterative prototyping that is considered a best practice in interaction design.

Designers and researchers do not just benefit from the existence of cheap, usable open source hardware—we are contributing to these projects more and more. A number of popular open source projects have academic origins: Arduino, Fritzing, and TinyOS are just a few that have taken on a life of their own. The open source hardware community can be a great channel for bringing a research project to maturity in the wider world.

Moreover, these tools may be fueling a "tangible turn" for hobbyists and small businesses. The availability of rapid prototyping tools and free modeling software for hobbyists now enables sites like Thingiverse (see sidebar), where people share free and modifiable models that others can print at home. Conversely, businesses like Ponoko and Shapeways use their high-end prototyping equipment to allow makers to order small batches of made-to-order physical items, or even to sell their designs on their website. Unlike older forms of manufacturing, such as molding or vacuum forming, 3-D printing and laser cutting do not have high fixed costs; making one widget is about as worthwhile as making several thousand. As this trend continues, we may encounter more and more general-public interest in the field of tangible interaction design.

ENDNOTES:

1. The definition of "hacker" we employ here is akin to that of "maker," with an additional connotation of reappropriating objects for other than their intended purpose. While the term can also connote someone who uses computer expertise to gain unauthorized access to data, this article does not refer to those practices.

2. Brewer, J., Williams, A., and Dourish, P. A handle on what's going on: Combining tangible interfaces and ambient displays for collaborative groups. *Proc. TEI'07* (Baton Rouge, LA). 2007.

3. Studios are six-hour workshops in which participants make tangible prototypes around specific topics of interest, e.g., game controllers, interactive books, wearables, etc.

4. The OSH definition can be read in full at http://freedomdefined.org/OSHW

5. http://www.ohwr.org/projects/cernohl/wiki

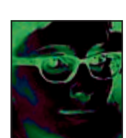
6. Cern Bulletin. Open hardware for open science; http://cdsweb.cern.ch/journal/CERNBulletin/2011/28/News%20Articles/1357331?ln=en

7. Kushner, D. The making of Arduino: How five friends engineered a small circuit board that's taking the DIY world by storm. *IEEE Spectrum.* Oct. 2011; http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino/0

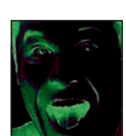
8. http://www.adafruit.com/blog/2010/05/03/million-dollar-baby-businesses-designing-and-selling-open-source-hardware-making-millions/

9. Weiler, L. and Ohlig, J. Building a hacker space: A hacker space design pattern catalogue. *The 24th Chaos Communication Congress* (Berlin, Germany). 2007; http://events.ccc.de/congress/2007/Fahrplan/events/2133.en.html

ABOUT THE AUTHORS
Amanda Williams is a cofounder of Wyld Collective Ltd., an independent design/research consultancy based in Montreal. Her research focuses on tangible and mobile interaction design.



Alicia Gibb, formerly of Bug Labs, does independent research and development. She is a member of NYCResistor and a pioneer in the open source hardware movement, co-chairing the Open Hardware Summit.



Born a nerd, David Weekly, after getting a CS degree, realized that hackers could be social and has since been working on getting them together by starting companies, and nonprofits, and parties. He is the founding director of Hacker Dojo in Mountain View, California.

DOI: 10.1145/2090150.2090156