25

PHYSICAL COMPUTING, EMBODIED PRACTICE

Nina Belojevic and Shaun Macpherson

In recent years, the increasing public availability of inexpensive, powerful electronics—such as microcontrollers and radio-frequency identification (RFID) tags—and the proliferation of do-it-yourself-themed websites and online communities have fostered the emergence of physical computing, a set of tools, technologies, and practices used by artists, technologists, academics, and hobbyists. Physical computing pushes human–computer interaction (HCI) beyond the logics of the screen by facilitating computer interaction with bodies and objects off-screen, in the physical world. While HCI can be broadly historicized—even insofar as considering cultural anxieties over the replacement of human workers by automated machines or the development of the computer mouse by Douglas Engelbart in 1967 as early examples—physical computing is typically defined as the practice of combining hardware design with computer programming to create networked, interactive devices and environments.

A core effect of physical computing is the creation of new devices and networks that challenge conventional ideas about interfaces and interactions between bodies, technology, and the environment. In *Physical Computing*, Dan O'Sullivan and Tom Igoe describe the practice as making a "computer for the rest of you" (2004: xvii). Such interactions expand not only the computer's range of actuation (or expression) beyond modes of HCI, but also the ways that computers "sense" physical matter and environmental behaviors. What results are novel modes of expression and interaction, achieved by programming sensors to translate input (e.g., temperature, touch, sound, or wireless data transmissions such as Twitter feeds) into machine-readable data. Algorithms then transform that data into output, both on- and off-screen (e.g., light, sound, rotating a servomotor, or triggering some digital behavior).

Such expanded capabilities offer nearly limitless potential for the development of technologies across a range of social, cultural, and political applications, from social justice to surveillance to data collection. For example, the Autonets project (2015), started by a group of artists led by micha cárdenas, uses small, wearable microcontrollers, which are programmed and sewn into pieces of clothing, to create autonomous, local networks that enable women, LGBTQI people, people of color, and others to stay connected and locate each other. With Autonets, physical computing enables its network of wearers to anticipate and thereby avoid potentially violent situations, while also increasing safety among communities susceptible to violence or discrimination. cárdenas describes the project as "fashion hacking for social re-organization, recoding the meaning of fashion symbols such as hoodies that have associations ranging from Trayvon Martin to the Black Bloc, or femme fashion elements like dresses and bracelets, into symbols of connectivity and autonomy" (cárdenas 2015).

On the other hand, the novel modes of HCI afforded by physical computing are also used to develop corporate interests. Google's Advanced Technology and Projects Group (ATAP) recently released Project Tango, a smartphone and tablet designed to "give mobile devices human-scale understanding of space and motion" (Google 2014). Project Tango combines computer vision with geolocation sensors inside the device to track its motion in three dimensions while geometrically mapping the space around it. Such behavior opens a range of interactive possibilities: from rapid three-dimensional mapping of an indoor environment to augmented reality apps that integrate their physical surroundings to assist people with disabilities. Of course, since Google is funding the project, it is also reasonable to speculate how the information gathered by these devices will serve the company's grander project of data collection—that Project Tango's slogan is, "The future is awesome. We can build it faster together," hints that the company's next era of world-mapping will be crowd-sourced. Though developed at a much different scale than Autonets, Project Tango similarly works to simulate a mode of perception, process sensory information algorithmically, and convey that information to larger networks. In so doing, both projects draw attention to different social issues that frame the interfaces between bodies and technology.

Since it networks digital and nondigital environments, physical computing requires a mix of programming, electronics, and mechanical knowledge on the part of the operator. This knowledge is gained through hands-on experience—code must be written and de-bugged, schematics must be translated into circuit-building, and parts must be hammered, cut, drilled, soldered, glued, sewn, and otherwise manipulated. As such, physical computing's investment in handicraft often corresponds closely with "maker culture," a term that broadly encompasses the practices of people interested in building their own tools, devices, and interactive technologies. Maker culture shares with physical computing practitioners an interest in making the practices of software programming, electronics prototyping, and manufacturing more accessible to nonspecialists.

Spurred by ever-increasing access to materials and online communities of practice, maker culture represents itself as a shift in the way individuals and small groups explore the materiality of HCI, with the independence and entrepreneurial energy of the Whole Earth Catalogue subscribers of the 1960s, the Silicon Valley garage-programmers of the 1970s, and the open-source enthusiasts of the 1990s and 2000s. As was largely common with these communities, the culture's ethos is bound up in a technocultural narrative of open access and instrumentalist responses to social and economic issues. To this end, the virtues of individual empowerment and egalitarianism are often foregrounded in maker culture, with the belief that DIY practices constitute forms of radical dissent in the face of black-box, corporate technologies. While some of these practices gesture toward the centrality of human bodies in computing, their rhetoric is more invested in outlining a socioeconomic subject that is at once communitarian (sharing with and borrowing from the open-source community) and libertarian (practicing resistance against the cultural hegemony of proprietary goods and regulated services). Yet, as is the case with other grassroots technology movements, making and physical computing remain heavily implicated in discourses of late capitalism and commercialism, as evidenced by the heavy branding campaign of the O'Reilly-operated *Make* magazine and its how-to imprint as well as the rising ubiquity of "maker faires," which were started by *Make* and involve heavy trademark licensing fees and requirements. Hence, what is first framed as a liberatory practice by its champions is susceptible to naïveté and complicity with neoliberal ideologies.

What typically get overlooked by such criticism—absent of attention to process as it often is—are the ways in which the practices and outcomes of physical computing do not simply naturalize technology, but rather emphasize a material understanding of the digital and physical processes taking place all around us. Eschewing the black-box opacity of screen-based, proprietary technologies, physical computing emphasizes an experimental approach to better understanding and creating new technologies: it both provides an alternate framework vis-à-vis the proprietary nature of consumer tech culture and facilitates a broader understanding of how and where networked technologies influence our bodies, as well as the things and events near us.

By foregrounding the materiality of electrical operations and computational processes, physical computing also pushes against the increasing naturalization of digital interfaces—that is, the seeming "disappearance" of digital devices due to their ubiquity across many elements of contemporary life. For example, a line of code that causes electricity to pulse through a light-emitting diode (LED) to turn it on and off at regular intervals demonstrates the relationship between a digital process (the binary digital command that determines when the electrical signal will fire through the LED) and the physical object (the LED) itself. The conversion of signal data into observable behavior foregrounds the transductive processes that underlie any interface between digital and physical things.

Transduction, broadly defined, is the conversion of one form of energy into another. While this term originates in the sciences, it has been taken up in the context of computation as a way to understand, as Matthew Fuller puts it, the process of "how this becomes that" (2007: 85). Many technologies easily demonstrate the cause-and-effect principles of transduction: for example, a light bulb transduces electricity into light and heat, while a microphone transduces sound waves into fluctuating currents of electricity. Even the transductive processes of early analog computers could be partially observed and thus more easily grasped—latch relay switches were large enough that their switching mechanism could be seen or heard (the click of the switch was audible), and thus the operator was able to understand the material way a computer could perform, say, sequential logic operations.

With the continuing miniaturization of digital technologies, physical transduction has become increasingly complex and harder to observe without specialized equipment. Thus the ways that contemporary electronics and data processing actually work is relegated to what is graphically expressed on screens, an effect of computing Nick Montfort calls "screen essentialism" (2004). Innumerable critical discourses have taken shape that indicate the profound effect screen essentialism has on contemporary technoculture, from the ecstatic communication theorized by Jean Baudrillard (2012) in the 1980s to the vibrant discussions surrounding digital labor and cognitive capitalism among scholars such as Jonathan Beller (2006), Nick Dyer-Witheford (1999), Maurizio Lazzarato (1997), Tiziana Terranova (2004), McKenzie Wark (2004), and many others. Physical computing's emphasis on the material aspect of digital technology offers both a contribution to these critical discourses and a potential for new approaches to critical analyses and computational practices.

A Brief History of Physical Computing

Physical computing first emerged through academic channels as a means to explore how HCI could facilitate studies of computation by pushing beyond the screen and in part reviving the concept of the feedback loop. In 2001, Casey Reas and Benjamin Fry—then graduate students in the Aesthetics and Computation Group at MIT's Media Lab—wrote an open-source, integrated development environment (IDE) called Processing, a simplified coding environment for generating interactive digital visual graphics and designs. Visual artists themselves, Reas and Fry set out to create a programming language that could be easily learned by designers and artists, with an emphasis on writing "sketches" (or programs) for interactive graphics (Reas & Fry 2010: vii). They articulate this idea in *Getting Started with Processing* (2010): "Processing offers a way to learn programming through creating interactive graphics. There are many possible ways to teach coding, but students often find encouragement and motivation in immediate visual feedback" (1).

Reas and Fry's approach to computation—that engagement with a user-friendly, programmable system is reinforced through "immediate visual feedback"—is echoed by Tom Igoe in *Making Things Talk* (2007). Igoe cautions his readers to remember the operator end of the interaction: when creating interactive design projects, it is vital to "give some indication as to the invisible activities of your objects" and build indicators such as "an LED that gently pulses while the network transfer's happening, or a tune that plays" (2007: 47). According to Igoe, people using a device or interacting with a system do not need to know what is being communicated—or how this becomes that—at all points. But they do need to be aware that communication is taking place.

At the core of what Reas, Fry, and Igoe suggest is that operators remain invested in the *process* of computing—that, when a given function's invisibility is reified as a physical mechanism, people become aware of (and presumably invested in) the network's communications. Such mechanisms are rendered both knowable (in that we are aware that they are taking place) and unknowable (in that we do not actually know how the process is taking place). This making visible the invisible perpetuates a sort of fetish: the physical indication of an otherwise invisible software process constitutes in operators a sense that the unknowable functions of software have tacitly exposed themselves in a way that surpasses the mode of representation (be it light, text, sound, or something else).

Reas and Fry later adapted Processing to work as the IDE for Arduino, an open-source microcontroller platform that has become one of the primary tools in physical computing communities. Arduino was developed by a team at the Interaction Design Institute Ivrea in Italy as a means to provide students with an inexpensive microcontroller platform. Led by Tom Igoe, Massimo Banzi, David Cuartielles, Gianluca Martino, and David Mellis, the development team made Arduino an open-source hardware platform: designs and parts lists for the board were shared freely, meaning people could assemble their own boards for the price of parts rather than buy a manufactured one. This low-cost, open-source approach to hardware was intended to make hardware as accessible to new users as Processing was for software. This accessibility was important, as Arduino provides interoperability with Processing, extending the program's interactive elements beyond the realm of on-screen graphics by providing input and output pins that enable an operator to connect sensors and actuators to the board and program behaviors using the modified IDE.

While, through microcontroller platforms such as Arduino as well as the Raspberry Pi, process indicators can ostensibly free operators from the visual domain (e.g., a "tune that plays" makes processes knowable via auditory conveyance), the insistence of screen-centric HCI persists: such expanded sensory diversity is tied to the visual insofar as what is knowable is expressed in visual terms (e.g., Igoe's "invisible activities of your objects"). When programming a microcontroller, this continued adherence to the visual persists because the construction of a schema for translation and communication is written in the IDE, which, as most interfaces do, functions according to a metaphorical relationship—anchored in visual paradigms—between operator and machine. Although such paradigms are conducive to rendering technologies friendly to operators without expertise in computing or manufacturing, they also reduce the complexity of technological processes, mask or reify them, and curb the range of critical or creative approaches.

Physical Computing and the Graphical User Interface

The proliferation of such interfaces resulted in what Wendy Chun calls the empowerment of operators, whose ability to directly manipulate and engage with computational processes afforded by the graphical user interface (GUI) "offers [them] a way to act and navigate an increasingly complex world" (2011: 176). Interestingly, empowerment remains tied to the ability to manipulate and engage with process, though the process has grown increasingly mediated by automation or digitization over time, from manipulating analog computation (e.g., flipping relays in the 1950s) to attending to screens (e.g., writing a sketch in Processing). In other words, the emphasis has gradually shifted from a hands-on interface with electronics to a visual and arguably abstract mode of HCI. Such screen essentialism results in a mode of production that privileges the visual display of information while obscuring the balance of a platform's processes, hardware and electronic circuitry included.

However, programmable microcontrollers—emblematic of physical computing's primary "purpose"—also represent a departure from GUI-based HCI. O'Sullivan and Igoe write that "we need computers that respond to the rest of your body and the rest of the world. GUI technology allows you to drag and drop, but it won't notice if you twist and shout" (2004: xvii). Though they adhere to the parlance of conventional HCI—falling back on the metaphor of visuality that the computer "sees" our physical, embodied expression—they nevertheless shed light on the inherent constraints and limitations of the desktop computer's interaction with nondigital environments (interactions that are typically limited to the computer screen, mouse, and keyboard, and tools that allow operators to manipulate symbolic representations of data on a screen). Their point is that screens are for people, and physical computing is an exploration of HCI for the computer.

In other words, the cybernetic function of microcontrollers in physical computing interfaces—specifically the feedback loop of sensors, software, operators, and actuation—facilitates a broader discussion about access between humans and objects, objects and objects, and so on. Such interfaces resist the screen essentialism at work in the GUI. Jef Raskin, who created the interface for Apple's Macintosh project (arguably the first consumer-level GUI), echoes O'Sullivan and Igoe's critique. Acknowledging his own culpability, he claims that GUIs are not conducive to the cognitive processes at play in the way people work: "Human adaptability has its limits and . . . GUIs have many features that lie outside those limits, so we never fully adapt but just muddle along at one or another level of expertise" (Raskin 1993). Physical computing functions as a response to GUI reliance by expanding the range of interaction. For instance, an operator can communicate with a digital environment via a motion sensor, an electret microphone, or a photoresistor, thereby exposing the screen-based paradigm and ableist norms of embodiment that inform most interface designs while also increasing the scope and capacities of computer perception.

Physical Computing and Critical Practice: Highlighting Embodiment

Just as physical computing expands the range of possible interactions between digital and analog environments—and, in so doing, offers a renewed attention to materiality in the context of technocultural critique—so, too, does it open a range of potentially interesting and valuable methodologies for research across media studies and critical practices, particularly in the context of the humanities, arts, and social sciences. Scholars and artists, such as Leah Buechley, Critical Art Ensemble, Electronic Disturbance Theater, Jennifer Gabrys, Garnet Hertz, Tom Igoe, Natalie Jeremijenko, Kim Knight, Kari Kraus, Matt Ratto, Jentery Sayers, and William J. Turkel, among many others, apply a critical lens to physical computing projects and techniques toward an awareness of the cultural, social, and political factors at work in technologies. For critical practitioners, physical computing can be applied as a form of close study that complicates scholarly objectivity and employs a more experimental approach to theoretical analysis. This approach results in highlighting embodiment (of both humans and nonhumans) as a fundamental part of HCI. Because physical computing requires elements to be taken apart and assembled, circuits to be designed and built, and code to be programmed and debugged, it is an intrinsically embodied practice—one that relies on the practitioner's combined physical and mental attention in more tactile ways than forms of observation through other senses commonly used for humanities analysis, such as reading, looking, or listening. In this way, physical computing offers a form of practiced media study that not only observes but also moves inside the black boxes ubiquitous across contemporary life.

Such an approach also provides a purposeful way to trouble (or augment) a kind of top-down, theory-first approach to studies of technology. While transductive processes in the production and perception of, say, sound waves (for example) can be theoretically mapped, an embodied perspective—akin to Donna Haraway's "situated knowledge" paradigm (1988)—allows the researcher to explore the cognitive or social impacts of such information in ways not necessarily afforded by a theory-first approach. A consciously embodied approach provides a purposeful way to raise important questions for cultural critique: where do these materials come from? How are they produced? How does transduction occur? What cultural, social, economic, or political factors are obfuscated under the screen of user friendliness? Why is this technology built in this way? What gets foregrounded is not only the materiality of transduction (for instance, when we feel a small electric shock while touching a circuit board attached to a battery), but also the practitioner's place in a multitude of ecologies, ranging from operational processes to histories of technologies to the labor, modes of production, resources, and socioeconomic relations that are always circulating under the veneer of digital technologies. Importantly, recognizing this embodiment draws the user's attention to their own partial perspective and lack of access to certain knowledge, a point that is particularly important for media studies.

Example Projects: *BodyPlay* and *Ghost Tree*

The possibility for critical insights that may not be as apparent via a top-down, conceptual approach can be practically demonstrated through creative projects. For instance, consider several physical computing art pieces by Nina Belojevic, a co-author of this chapter. First, her interactive wall installations for the *BodyPlay* series (see Figure 25.1) depict body parts using a combination of mixed media visuals and microcontroller-based interactivity to encourage critical inquiry into the audience's relationships with technologies, interfaces, art, and bodies. The process of creating these pieces required not only the conceptualization and development of visual elements, but also the exploration of sensors and electronic triggers that enact and foreground these relationships. One piece, *Connect the Wires*, triggers different colored light cycles when two wires that emerge from a silicone nipple are connected.

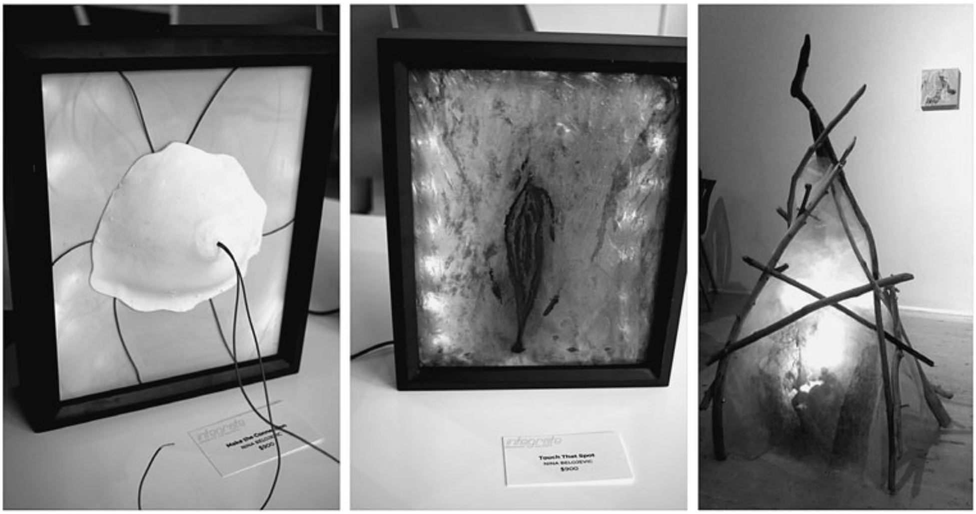

Figure 25.1 Left and center: *BodyPlay* wall pieces. Right: *Ghost Tree*.

Source: Images courtesy of Nina Belojevic.

Another, *Touch that Spot*, brightens up and displays various color patterns when conductive parts of the image are touched.

Both pieces use LED strips to create their responsive effects—colorful light patterns, movements, and cycles. While the capabilities of these LED strips are vast and allow for endless applications, they are relatively easy to program, even for someone new to physical computing. That said, even simple circuit construction such as this one necessitates specific material considerations for proper and reliable execution. For example, to regulate the power between the strips and the microcontroller, a capacitor, soldered in a specific configuration, is essential. Without it, the initial onrush of electrical current can damage the delicate LEDs inside the strip. If the wrong terminals are connected and a short circuit occurs, then any part of the circuit may be damaged. While recalling such a process might seem banal to media studies scholars, it highlights the cognitive, aesthetic, and material connections—or the interfaces and transductions—that must be made to create a computational art piece that conveys some meaning, be it social, political, or aesthetic.

When approached through critical media studies, physical computing projects can also highlight how digital technologies are not disembodied concepts or metaphors, but rather situated in different ecologies of place and context. Belojevic's sound-responsive piece, *Ghost Tree* (see Figure 25.1), was originally commissioned as a set piece for a local musician. Constructed from a combination of found materials and Arduino-based circuitry, *Ghost Tree* translates different sonic frequencies into preassigned color patterns; what results is an object that glows with particular colors based on the notes and chord arrangements performed by the musician. Belojevic was later invited to display *Ghost Tree* at a noise music event, where the decibel level was substantially higher than that of the initial performance. She was forced to make on-the-fly adjustments to the piece for its input (a small electret microphone) to function effectively in this different sonic environment. In the absence of the time and means to rebuild the circuit or reprogram the Arduino, she created an effective sound dampener over the input using tissue paper and electrical tape. Not ideal, but it worked, and *Ghost Tree* successfully interacted with the much louder music. The effectiveness of the device was mitigated by a complex ecology, which included different sonic environments, affordances of the materials, and the integration of further materials that altered the behavior of the device in a given context.

Embodied interactions and explorations such as the examples given above encourage unique ways of seeing and studying technologies—techniques that do not rely on a removed study of an object, but rather encourage the development of situated knowledge that experiences what is at hand and then follows a path of inquiry. It makes possible a reciprocal methodology whereby complex and inaccessible elements of technologies can be studied and communicated through experimental practice, which may make it possible to communicate new insights in words, create a discourse around them, and share perspectives that can then feed back into hands-on computing. This approach relies on much more than the interfaces we are provided with: images we see on screens, user-friendly interactions with metaphors, sounds we hear through speakers, or code we can read. It follows the flow of electricity through wires, resistors, and capacitors, out of sensors and into chips, from computer to actuator, and from air to the sensory receptors in our skin, eyes, and ears. The landscape of the circuit can be felt with our hands and seen with our eyes; and while the transduction of energy may not be fully visible to us, its operations are more easily felt and discerned when we become physically invested in them.

Further Reading

Banzi, M. (2009) *Getting Started with Arduino*. Sebastopol, CA: O'Reilly.

Belojevic, N. (2014) "Circuit Bending Videogame Consoles as a Form of Applied Media Studies," *New American Notes Online* 5, retrieved from nanocrit.com/issues/5/circuit-bending-videogame-consoles-form-applied-media-studies.

Fuller, M. (2007) *Media Ecologies: Materialist Energies in Art and Technoculture*, Cambridge, MA: MIT Press.

Macpherson, S. (2014) "A Computer for the Rest of You: Human-Computer Interaction in the Eversion," Master's Thesis, University of Victoria.

Nakamura, L. (2014) "Indigenous Circuits: Navajo Women and the Racialization of Early Electronic Manufacture," *American Quarterly* 66(4), 919–41.

O'Sullivan, D. and T. Igoe (2004) *Physical Computing: Sensing and Controlling the Physical World with Computers*, Mason, OH: Cengage Course Technology.

Scholz, T. (ed.) (2013) *Digital Labor: The Internet as Playground and Factory*, London: Routledge.

References

Autonets (2015) *Local Autonomy Networks*, retrieved from autonets.org.

Baudrillard, J. (2012) *The Ecstasy of Communication*, Cambridge, MA: MIT Press.

Beller, J. (2006) *The Cinematic Mode of Production: Attention Economy and the Society of the Spectacle*, Hanover, NH: Dartmouth College Press.

cárdenas, m. (2015) "Local Autonomy Networks: Post-Digital Networks, Post-Corporate Communications," *Media-N* (2013), retrieved from median.newmediacaucus.org/caa-conference-edition-2013/local-autonomy-networks-post-digital-networks-post-corporate-communications/.

Chun, W. H. K. (2011) *Programmed Visions: Software and Memory*, Cambridge, MA: MIT Press.

Dyer-Witheford, N. (1999) *Cyber Marx: Cycles and Circuits of Struggle in High Technology Capitalism*, Chicago: University of Illinois Press.

Fuller, M. (2007) *Media Ecologies: Materialist Energies in Art and Technoculture*, Cambridge, MA: MIT Press.

Google (2014) Project Tango, retrieved from www.google.com/atap/project-tango/.

Haraway, D. (1988) "Situated Knowledges: The Science Question in Feminism and the Privilege of Partial Perspective," *Feminist Studies* 14(3), 575–99.

Igoe, T. (2007) *Making Things Talk: Practical Methods for Connecting Physical Objects*, Sebastopol, CA: O'Reilly.

Lazzarato, M. (1997) "Immaterial Labor," in P. Virno and M. Hardt (eds.), *Radical Thought in Italy: A Potential Politics*, Minneapolis, MN: University of Minnesota Press, 133–50.

Montfort, N. (2004) "Continuous Paper: The Early Materiality and Workings of Electronic Literature," *MLA Convention*, 28 December. Philadelphia, PA.

O'Sullivan, D. and T. Igoe (2004) *Physical Computing: Sensing and Controlling the Physical World with Computers*, Mason, OH: Cengage Course Technology.

Raskin, J. (2013) "Down with GUIs!" *Wired* 1(06) (Dec. 1993), retrieved from wired.com.

Reas, C. and B. Fry (2010) *Getting Started with Processing*, Sebastopol, CA: O'Reilly.

Terranova, T. (2004) *Network Culture: Politics for the Information Age*, Ann Arbor, MI: Pluto Press.

Wark, M. (2004) *A Hacker Manifesto*, Cambridge, MA: Harvard University Press.