Printy3D: In-Situ Tangible Three-Dimensional Design for Augmented Fabrication

Amanda K. Yung
Rochester Institute of Technology
Rochester, NY, USA
aky2875@rit.edu

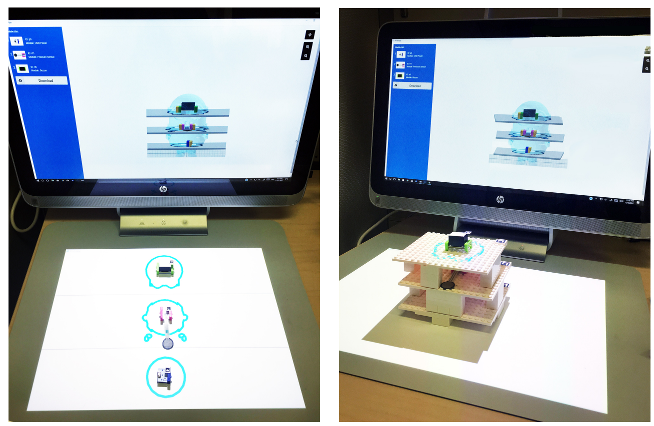
Zhiyuan Li
Rochester Institute of Technology
Rochester, NY, USA
zl7904@rit.edu

Daniel Ashbrook
University of Copenhagen
Copenhagen, Denmark
dan@di.ku.dk

ABSTRACT
Three-dimensional design software is challenging for novices and non-experts; when working with objects that already exist, the task becomes even more difficult. We present *Printy3D*, a system that enables children to design customized containers for electronic modules using tangible interaction and spatially augmented reality feedback. Our system allows users to position physical objects in three dimensions relative to a virtual container, providing feedback on placement location and validity. We implemented two different interaction styles and conducted a user study with 26 participants, 23 of them children. We detail the results of our study and suggest implications for design as well as opportunities for future research for systems of this kind.

CCS Concepts
•**Human-centered computing** → **Mixed / augmented reality; Interaction techniques;**

Author Keywords
Augmented Fabrication; 3D Printing; Augmented Reality



(a) (b)

Figure 1. Printy3D's two interfaces: flat (a) and platform (b).

INTRODUCTION
The maker culture has seen a boost in popularity, with the establishment of makerspaces as public spaces for creating and sharing together. There has also been a push for more creative learning in schools, with 3D printers and other fabrication devices being introduced in classrooms [9]. The use of 3D printing and other maker activities in educational settings has increased in part because of the potential for more engaged and active learning; 3D printing in particular can provide an ecosystem for improving independent learning through construction and physical manipulation [15], and can also provide an avenue for children to become more interested in STEM fields [20].

However, not all technologies in this emerging maker culture are easy to learn. While 3D printing has become a popular fabrication technique, the modeling software available to create 3D models can be complex for a non-expert user due to the high learning curve. One of the main difficulties with 3D modeling software is the need to visualize and interact in a virtual 3D space on a 2D screen. This is in part because the cognitive requirements to conceptualize physical objects in 2D space is more demanding than interacting with the physical objects themselves [48]. It is only after thorough practice that one may become adept at this particular skill set [46]. As non-expert users initially interact with 3D modeling software, they can become frustrated and lose interest in their creations [11]. Difficulties such as these might turn away individuals who would otherwise want to learn how to transform their ideas into physical objects.

Further issues emerge when working with existing objects. Previous research has shown that many potential users of 3D printing want to fabricate new items to interface with existing objects, for example to upgrade or repair them [56], an activity that has been termed "augmented fabrication" [5]. Undertaking this design task requires three actions of the user: 1) *capturing* the physical characteristics of existing objects, and their spatial relationships to each other; 2) *designing* geometry to be fabricated in relation to the physical characteristics of the existing object; and 3) *understanding* how the fabricated object will interact with the existing artifact. Current design tools generally support these tasks *indirectly*, requiring the

IDC '18, June 19–22, 2018, Trondheim, Norway

ISBN 978-1-4503-5152-2/18/06...$15.00
DOI: https://doi.org/10.1145/3202185.3202751

user to manually translate between the real world object and the digital model on the screen. This requirement imposes large *gulfs of execution* for capture and design, and a large *gulf of evaluation* to understand the potential output [32].

Tangible interaction offers a way to bridge these gulfs: physical user interfaces provide in-context interactions and rich physical affordances. Previous research has illustrated the advantages of tangible UIs. Hurst et al. discussed the potential of tangible interaction to develop more-accessible design hardware and software [31] and Fails et al. found advantages of physical over digital environments for young children [16].

In this paper, we present a first step towards combining the advantages of direct manipulation from tangible interfaces with the rich capabilities of 3D design software: rather than forcing users to use a two-dimensional interface for three-dimensional design, our research investigates the potential to incorporate spatial augmented reality into a tangible design environment. Our goal is to allow the user to work *directly with the existing object as part of the design process*.

Our system, *Printy3D*, contributes to the growing research in providing non-expert fabricators with accessible tools to design in 3D. Rather than creating a complex, full-fledged 3D design environment, Printy3D focuses on the simple yet compelling use case of enabling children to design and fabricate customized enclosures for electronics projects created with the commercial modular electronics toy littleBits[1]. With Printy3D, users position littleBit modules in a physical 3D space relative to a virtual container. They are supported by a spatial augmented reality environment which provides projected feedback to the user as they design inside a digital container. Projected visual aids guide the user in valid module placement, while a second screen also displays a full 3D rendering of the container for an additional view of how the littleBits will be configured inside the model.

We explored two different interaction techniques to enable non-expert users to position physical objects in a virtual 3D space. The first interface spreads out the 3D container into a 2D surface, allowing the user to more-quickly move the modules around, but separates the form of the design from the eventual output. The second interface incorporates platforms as support material to allow littleBit placement in three dimensions. The platforms also act as a projection surface for visualizing the virtual container co-located with the physical electronics modules in a one-to-one correspondence.

In the remainder of this paper, we discuss the design rationales behind Printy3D, detail our implementation, and share the results of a preliminary study we performed with three adult and 23 child participants. Finally, we discuss how the knowledge gained from creating and testing these proof-of-concept interfaces will help guide the development of future tangible and 3D interaction spaces used in computer-aided design (CAD) tools.

RELATED WORK

Printy3D pulls inspiration from several topics of research within human-computer interaction, including 3D tangible interaction, spatial augmented reality interfaces, 3D design for novices and children, design for augmented fabrication, and mixed reality interfaces for augmented fabrication.

Tangible Interaction in 3D

Research in tangible interaction has started to address the challenge of physical interaction in three dimensions. Much of the work to date has focused on information visualization and physical output, for example with acoustically levitated particles [44, 45]. Researchers have also investigated allowing users to interact with these displays. Several projects have created hovering balls that users can directly interact with, using magnets [41] or air jets [2, 66], but these objects must be magnetic or very lighweight and spherical. Another air-based project used small quadcopters to represent information in 3D space [24], but would not work on the small scale of littleBits modules. Some work has directly supported objects in 3D, with a movable "stalk" for 2.5D interaction [57] or with objects mounted on the end of a robot arm [4]. These projects only support one object at a time. In Printy3D, we only track the location of the objects as the users manipulate them, rather than attempting to actuate them, which greatly simplifies our system design.

Spatial Augmented Reality Interfaces

Spatial augmented reality is defined as augmenting an environment "with images that are integrated directly in the user's environment, not simply in their visual field" [52]. The use of spatially augmented workspaces for design and interaction is not novel. Tangible Viewports used a projector positioned behind the user to display realtime changes of graphical skins onto objects held up to and near the desktop workstation [23]. Unlike Printy3D, however, the projections represented decal-like additions that do not attempt to portray a three-dimensional environment. Whiteley et al. also made use of a hybrid environment for iteratively designing 3D-printed objects from built-up primitives [63]. In their Tangible-Tango system, they used a Microsoft PixelSense tabletop screen for tracking printed objects with tags placed on their bottoms. Users constructed objects level-by-level with visualizations of the lower levels displayed on the screen underneath. A full rendering of the combined primitives was shown on an adjacent vertical screen. However, because visualizations were presented under the objects, this meant that they could only be visible around the objects. Printy3D's projector-based interface enables feedback to be displayed on top of objects, including the bounds of the container and various checks on valid littleBit placement.

3D Design for Novices and Children

While expert users of computer-aided design tools become very competent, newcomers to these pieces of software suffer from a variety of difficulties. Standard CAD packages have been shown to be difficult to learn [25], while even simplified software such as TinkerCAD[2] can still be hard for novices. One fundamental issue stems from the difficulty novices experience in understanding the idea of navigation in CAD programs: in their study of novice makers, Hudson et al.

[1] littlebits.cc

[2] tinkercad.com

found that some users were confused by TinkerCAD's two-dimensional representation of three-dimensional forms [30], and McNally et al. discovered multiple problems encountered by children using simplified 3D-modeling software [42].

Using physical visualization may be an effective way to enhance users' spatial understanding [58], and researchers have explored various methods to allow design input via tangible manipulation. One method explored in the literature is using building blocks to enable users to create 3D objects by physical manipulation. Many of these examples involve augmenting the blocks with circuitry to track their locations relative to each other [3, 10, 37, 50]. While promising, these approaches are expensive due to the custom electronics required in each building block, and the resolution tends to be low for the same reason. Other approaches use computer vision approaches to capture block positions [7, 28, 34]. Printy3D, while using computer vision to track objects, focuses on positioning physical objects relative to already-created 3D models, using building blocks in one interface as a support for the third dimension.

Previous research has also investigated ways to create more-tangible user interfaces for children when using modeling software, both for educational purposes and for creative endeavors—including some work in augmented fabrication. KidCAD emphasized using existing physical objects as inspiration for remixing new objects via a clay-like interface that tracks impressions made in the surface [18]. Easigami provided a tangible way of building geometric structures by connecting simple polygons together to create a matching virtual model [29]. Like Printy3D, these interfaces incorporate physical interactions to address the cognitive conflicts faced by children when attempting to use solely 2D software to create 3D objects.

Design for Augmented Fabrication

Designing for augmented fabrication can be challenging: as discussed earlier, it requires capturing information about existing objects, designing new geometry to be fabricated, and understanding how the new objects will work with the existing ones. Many domain-specific interfaces exist in the literature that address this problem [1, 12, 13, 14, 17, 19, 21, 26, 38, 51, 59, 60, 67]. Most relevant to our work are those that address combining existing electronics with fabricated enclosures. Weichel et al.'s Enclosed provided a CAD environment for designing laser-cut enclosures in conjunction with hobbyist prototyping boards [61], Pineal enabled embedding phones or smartwatches into 3D-printed objects to add interactivity [39], and Printy provided an interface to embed littleBits into 2D extruded containers [5]. All of these interfaces are screen-based, providing some of the advantages of CAD such as automatic error-checking, but require simulation of the electronics to provide feedback to the user on how the final product will function.

Mixed-Reality Interfaces for Augmented Fabrication

One method for making augmented fabrication more intuitive is through mixed-reality interaction, where virtual and physical content is seamlessly placed side-by-side. Several works used see-through augmented reality to capture existing objects [40] or to model new objects in relation to existing ones [38, 65]. NatCut enabled the creation of 3D laser-cut containers for electronics by allowing the user to place components on a Microsoft PixelSense table [55]; however, its table-based nature required a user to design a 3D object in 2D space. MixFab used see-through AR to allow users to manipulate scanned objects relative to new models in 3D [62], but does not offer the ability to place the physical objects themselves. Makers' Marks "mixes reality" via fiducial-based stand-ins for electronic components attached to user-built physical models; these models are later scanned to create 3D-printable objects [54]. Peng et al. created an augmented reality design interface that used a robot arm to fabricate during pauses in the design process [47]. Printy3D combines aspects from all these projects to create a unique interface that enables tangible interaction in three dimensions for designing containers for electronics.

PRINTY3D

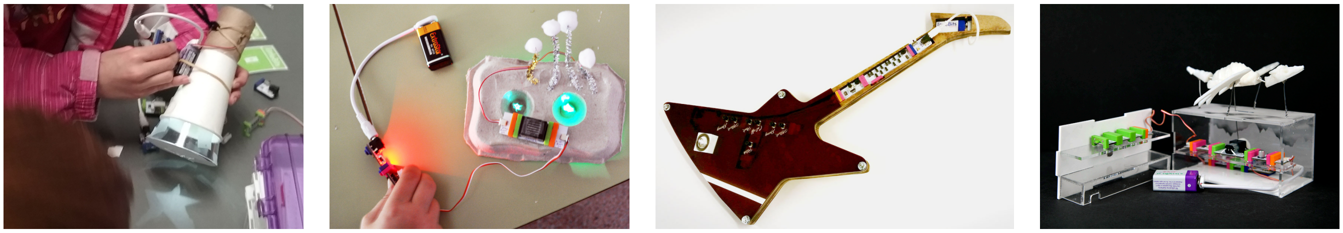
As evidenced by a wealth of shared photos on the littleBits site[3] and other websites, child makers often personalize their littleBits-based project by building custom containers for their circuits. These containers are typically constructed out of basic found items such as cardboard boxes and tape (Figures 2a and 2b). While flexible and allowing for creativity, these containers can be unfinished-looking and flimsy. Some adult makers have made littleBits containers with digital fabrication methods (Figures 2c and 2d). These methods, however, require in-depth knowledge of CAD programs and fabrication processes: the littleBits need to be measured, modeled with CAD software, placed virtually within the container's 3D model; the model must be modified with cavities, mount points, and button and light holes for the littleBits; and the final product must be checked for 3D-printing soundness. All of these steps are technically challenging for a novice hobbyist.

The original Printy application addressed this issue by providing a digital interface with graphical renderings of littleBits in the circuit [5]. Users could then move around these representations along one plane on the screen to position them within a 2D container (Figure 3). The system then extruded the container into 3D, modified it with fittings and holes for the littleBits, and output a printable file.

Printy3D takes these interactions further by including both tangible interaction and the ability to design in 3D space. We opted for physical interaction rather than fully screen-based due to both prior research findings [30] and our experience with the original two-dimensional Printy suggesting that a tangible interface would be easier to use, avoiding problems both in understanding 2D representations of 3D models and in understanding the onscreen representations of the electronic modules.

By working directly with the objects in a physical space in three dimensions, users have a more direct way of seeing and experiencing the container they are designing. This strategy decreases the gulfs of execution and evaluation [33]: the gulf of execution shrinks because the user physically manipulates the littleBits they want to use, rather than having to imagine

[3] See littlebits.cc/projects



(a) Flashlight *(b) Robot* *(c) Keytar* *(d) Dragon*

Figure 2. Publicly-shared littleBits-based inventions by children ((a) and (b)) with cardboard-based cases, and by adults ((c) and (d)) with fabricated cases. Photos ⓒ by Flickr users *marysvillelibrary* and *Ultra-lab* , and Thingiverse users *Quixotic* and *littleBits*, cropped and color-corrected.

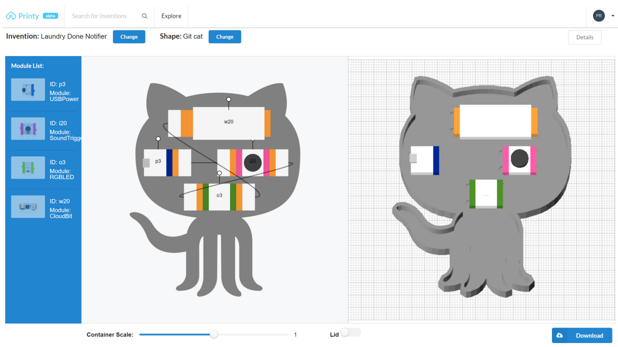


Figure 3. Original Printy user interface. littleBit drag-and-drop placement is limited to one plane for creating a 2.5D container.

how the circuit would connect together in the real world based on the digital renderings; the gulf of evaluation also diminishes because after the user places them, the objects reside in physical 3D space with a direct one-to-one correspondence with where they will appear in the finished container. Printy3D strives to provide a physical 3D environment where users can feel they are naturally able to design a container around a littleBits circuit. Using this system also ensures the littleBits will fit inside the designed container without needing to print, test, and redesign the container.

The main audience for Printy3D is children, as they may find the most benefit from designing with this system: children are typically novices to the world of 3D modeling in a digital environment, but are very familiar with physically playing with objects. Children also learn through experimentation: they tend to design through trial and error, also known as epistemic action [36]. As such, a physically manipulable interface may be more accessible for them to engage in creating in 3D.

Interface Design

Printy3D's setup centers around a physical spatially augmented reality interface, with a supporting role played by a computer monitor interface. The augmented reality interface tracks the littleBits in physical space, and provides a projected overlay that visualizes both where the littleBits are in relation to the virtual container, as well as if each module is placed in a valid location. Because the development of tangible interfaces for modeling is still in the early stage, we explored the possibilities of what the interactions in our system could look like, and how they would be perceived by users. Consequently, we developed two different versions of the physical interface: 1) the flat interface; and 2) the platform interface (Figure 1).

Both interfaces enable the user to move the physical littleBit modules with respect to a projected virtual container. As the user does so, the monitor shows a preview of the modules inside a 3D rendering of the completed container (Figure 7). The difference between the two interfaces is in how they handle the third dimension: while both divide the model's height into multiple slices to allow littleBit placement, the flat interface—inspired by Tangible-Tango [64]—separates these slices into "floor plan"-like segments spread out on the projection surface, while the platform interface creates the slices with a series of stacked platforms.

The flat interface (Figure 1a) consists of one plane of augmented reality space. Sections of the container are presented in parallel directly on the projected workspace, assisting with the placement of littleBits. Using this interface allows the user to simultaneously work on all levels of a container; however, it does not present a one-to-one mapping of where the littleBits would appear in physical space within the container. As such, it allows for quick adjustments of circuit placement, but it could be less intuitive with no physical correspondence in the height direction.

The platform interface (Figure 1b) solves this correspondence problem, mimicking how littleBits would be positioned in the actual container by using building blocks as scaffolding material. By building up layers of blocks, the user can position the bits in 3D space in the exact position they would want the bit to be. As such, this design provides the most direct correlation between what the user sees and what the resulting container would look like.

With the physical augmented reality interface and the monitor interface combined, they have the potential to give the user a deeper understanding of how the final container will look before it is even printed, as compared to a solely digital interface. We next describe the design of Printy3D's interfaces and interactions in further detail.

Spatial Augmented Reality Interface

To achieve the goal of enabling users to undertake design tasks using the existing object as an integral part of this process, we implemented the flat and platform interfaces. We wanted to explore different facets of these interaction styles: the flat interface emphasizes speed and fluidity, while the platform interface focuses on an accurate representation of the relative positions of the existing and virtual objects. Aside from their treatment of the height dimension, the two interfaces share the same visualizations and interaction styles.

Flat Interface

The flat interface was influenced by the Tangible-Tango interface [63]. This interface presents the levels of a container like a building floor plan: each level is separated out and fully visible along one flat surface (Figure 1a). The lowest level is projected closest to the user, and higher levels are presented at distances further away. We chose this order due to its spatial similarity to the worldspace displayed on the accompanying touchscreen monitor (described in the *Monitor Interface* section).

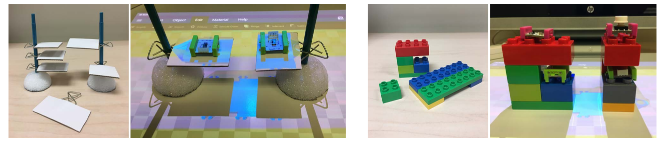
This interface uses a projector to display each level separated by a bold gray line, with an outline of the slice of the model represented by that section. Currently, these heights are equally spaced apart from each other by 32 mm (1.26 in)—the same separation as between the platforms—although in a future iteration we plan to make their placement user-configurable. One limitation of this interface is that because we display the height dimension by splitting it into 2D segments, we can only show three levels. More levels cause the display to become too cramped, while placing them side-by-side disrupts the spatial mapping between the physical workspace and the visualization on the monitor interface.

Figure 1a illustrates the flat interface, with littleBit modules placed within the projected outlines of three slices of a model. Users can quickly move the modules within and between levels, but need to maintain a mental mapping of which projected level on the horizontal workspace corresponds to which slice of the model on the vertical monitor interface.

Platform Interface

The idea behind the platform interface is to enable the user to physically place modules in a one-to-one correspondence to the virtual model. To physically build in 3D requires some kind of support to hold the littleBits at a particular height. Ideally, such a support will provide not only a stable, adjustable surface to allow users to quickly place modules at different heights, but a surface on which the projected feedback can be displayed. We first experimented with "floating" platforms which freely slide along a rod (Figure 4a), but found they offered *too much* freedom, being difficult to place in the exact desired position (e.g., putting two modules at exactly the same height position). We also tested large-format Duplo building blocks (Figure 4b) but found they were too bulky and low-resolution to support building more than the simplest structures. We ultimately settled on Lego building blocks, which support more adjustability while allowing precise adjustment, and are available in white, which allows the projected interface to be visible.

The platform interface (Figure 1b) uses standard Lego bricks and baseplates. The blocks act as supporting pillars, allowing the baseplates to be positioned at the height the user wishes to place the littleBit module within the container. Users can place one or several modules on a platform, and can stack the platforms on top of each other to build multiple levels in the container for the circuit. The platform heights directly correlate with where the littleBits will be positioned within the container, so users can physically place littleBits exactly where they want them to be housed in the model. The platform levels can overlap each other in the height dimension, allowing for precise placement of electronics.



(a) "Floating" platforms *(b) Duplo blocks*

Figure 4. Platform interface scaffolding prototypes.

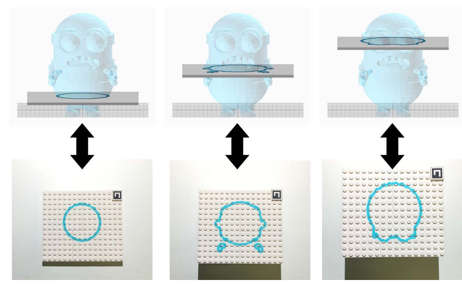


Figure 5. Projected container cross-sections for three different levels of the Minion model.

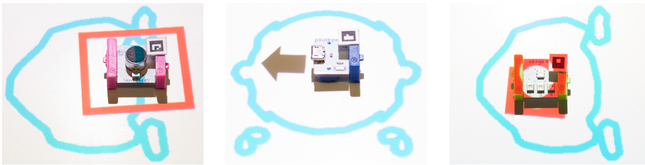
The system projects corresponding slices from the 3D model onto the top-level platform to indicate where in relation to the model the platform is located (Figure 5). The virtual model is fixed in space relative to the horizontal work surface, so moving the platforms gives the sense of a magic-lens interaction [8], with the platform as a "lens" into a virtual model located on the work surface.

Projected Feedback

Both the flat and platform interfaces provide projected feedback to guide littleBit module placement within the container. Informed by the principles of direct manipulation [32], we want to keep feedback minimal and intuitive, while at the same time helping the user understand how their physical placement actions interact with the virtual container. Doing so assures the user that their final container can successfully hold the modules required by their circuit *before* they print it, avoiding the print-adjust-reprint cycle typical of many augmented fabrication 3D printing projects.

Printy3D currently performs three types of validity checks around module placement and provides projected feedback for each:

Container enclosure: in general, littleBit modules should be fully enclosed in the container. This validity check inspects each module to ensure that no part of it is on the outside of the container; if this placement error is detected, the system projects a red rectangular outline around the given module (Figure 6a) until its placement is corrected.



(a) Enclosure *(b) Side access* *(c) Top access*

Figure 6. Projected feedback examples. Examples are shown with the flat interface but are identical for the platform interface.

Side access: some modules have components that necessarily must be exposed through the walls of the container. For example, the power module must have a hole through which the USB power cable can be plugged into the module, and the pressure sensor module should allow the pressure sensor to exit the container. For each of these special-case modules, the system projects a black arrow (Figure 6b) to indicate the side of the module that needs access; moving the module such that the arrow intersects with a wall will result in a valid placement.

Top access: similar to the side access case, some modules such as the button and LED require access through the top of the container. For these special-case modules, a solid red rectangle with a hole is projected on top of the given module (Figure 6c); moving the module vertically closer to a top surface of the container corrects the error.

Monitor Interface

Printy3D's touchscreen monitor, located vertically adjacent to the horizontal workspace (Figure 1), performs several functions: it walks the user through the selection of a circuit to build, presents container models for the user to select from, and provides a digital rendering of the 3D container with virtual littleBits that correspond with the physical locations of the physical modules; this allows the user to verify the design of their final container.

During the selection phase, the user selects the littleBits to use in the circuit, as well as the container for the circuit. At the start of the application, the monitor view presents the user with a library of littleBits that can be used with the system. After selecting the bits in the order that they need to be assembled for the circuit to work, the user then selects a container to house the circuit. In the system's current form, users can select from a list of provided STL files with corresponding preview images. These models were obtained from Thingiverse[4], a public repository of CAD models which people are able to freely use and modify.

Once a model is selected, the application takes the user to the editor view (Figure 7). A list of littleBits required in the circuit appears on the side of the screen, and a 3D digital rendering of the model appears at the center. Users can rotate the model either via the touchscreen or the mouse to view different angles during the build process. Buttons on the side of the screen also allow the user to zoom in and zoom out, in addition to resetting the camera to its default view.

[4] thingiverse.com

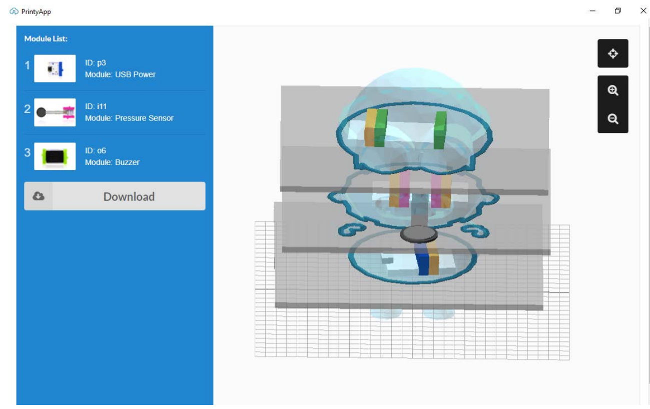


Figure 7. Editor view of monitor interface, with littleBit and platform renderings.

As the user places littleBits in the workspace, using either tangible interface, virtual 3D renderings of the bits appear at their corresponding positions within the rendered model. Because the model is semi-transparent, users can see where the modules are positioned within this digital container. These digital modules provide an accurate representation of where the bits will end up in the final product. The system continually updates the locations of the littleBits as the user manipulates them.

Digital renderings of the levels from the flat interface, or of the platform locations in the platform interface, also appear in the monitor view to further guide littleBit placement. Outlines at the point of intersection with the container are displayed on top of these platforms; these outlines match those in the augmented reality environment so that the user can better understand the connection between the two views. As with the littleBit modules, as a user manipulates the platforms on the workspace, their locations are updated in the rendering.

Once a user has placed all littleBits inside the container, a completion button activates on the screen. The user can continue to redesign the placement of their circuit if they wish, or click on the button to indicate they are done. Because our focus was on prototyping different interaction modalities and understanding their tradeoffs, Printy3D does not currently export the completed models; however, this functionality has been demonstrated in previous work [5, 27, 35].

IMPLEMENTATION

Printy3D runs on an HP Sprout computer, an all-in-one Windows-based PC with a touchscreen monitor, high-resolution camera, depth camera, DLP projector, and multitouch mat. The cameras and projector are factory-calibrated to view and project on the touch mat, offering a convenient setup for our system. Printy3D is implemented in two parts: a backend computer vision component, and a frontend UI component. The backend communicates with the frontend via a JSON file interface; the entire system runs at about four frames per second.

Backend

We use the HP Sprout's built-in color camera to track the littleBit modules. We initially tested feature-based tracking but found the modules are too small to be accurately tracked at interactive speeds; therefore, we place small (1 cm square) fiducial markers on each module and slightly larger ones (1.5 cm square) on each platform. We used the OpenCV C++ computer vision library with ArUco markers [22]. Each littleBit has a unique marker to identify the type of the bit, while all platforms share the same marker. The backend application detects each bit and platform, and calculates its pose.

In order for the frontend to correctly render the placed littleBit modules, the system must determine when two modules are magnetically connected to each other. To do so, we compute the coordinates of the four corners and the center of of each module from the coordinates of the fiducial markers and the known size of each module. We then sort each by their centers from top-left to right-bottom. Next, we compute the Euclidean distance between the corners of every two adjacent littleBits in sorted order; if less than an empirically-determined threshold, we consider the two modules to be connected.

Frontend

The frontend interface is implemented as a web application using the Vue.js[5] Javascript framework and three.js[6] to render the 3D graphics. The frontend displays both the monitor interface as well as the projected feedback. It accomplishes this via two browser windows which communicate with each other via web storage. When the backend updates the locations of the littleBits modules and platforms, the main window process of the frontend re-renders the preview, performs error checking on module placement, and projects feedback.

We generate the container cross-sections in the projected display by taking slices with boolean operations using the OpenJSCAD library[7]. The slices are static for the flat interface, but we dynamically update them whenever platforms are moved in the platform interface, displaying only the portion of the slice that intersects the platform.

We also check for placement errors (Figure 6) using boolean operations. To check for enclosure, we extrude each slice by the height of a level and perform an intersection test with each littleBit module represented as a rectangular solid. To check for side and top access, we cast a vector from the module in the direction of needed access (e.g., to the side for the power module and up for the button), and compare the distance to the nearest surface with a predefined threshold.

EVALUATION

To evaluate the usability and user preferences of the two physical interaction methods, we conducted a pilot study and a subsequent two-stage exploratory user study. Rather than to robustly evaluate metrics such as completion time, the goal of the study was to obtain initial impressions of the usability and enjoyability of our interfaces from child participants,.

[5] vuejs.org [6] threejs.org [7] joostn.github.io/OpenJsCad

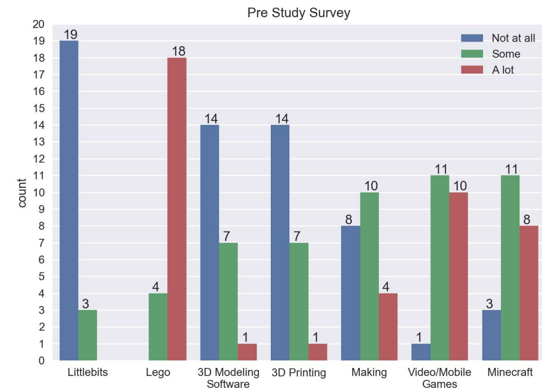


Figure 8. Non-pilot participant responses to the pre-study survey.

Participants

We recruited 3 adults (ages 24–28, two female) and 23 children (ages 8–15, nine female) via email advertising and snowball sampling. Participants received a 3D-printed toy as a gift. We gave all participants a pre-study survey to understand their levels of experience with 3D printing and modeling, littleBits, Lego blocks, video games in general, and Minecraft in particular. Most participants had little knowledge of making-related topics, although some of them had played with littleBits and 3D printers in school. Figure 8 summarizes the responses of the non-pilot participants. All participants (and their guardian in the case of child participants) agreed to be recorded during the study. Refer to Table 1 for details on the participants.

Study	PN	Age	M/F	Study	PN	Age	M/F
Pilot	P01	24	F	1	P14	9	M
	P02	28	M		P15	11	F
	P03	25	F		P16	10	F
	P04	9	M		P17	8	M
1	P05	9	M		P18	9	F
	P06	12	F		P19	12	F
	P07	13	M		P20	10	M
	P08	13	M		P21	8	M
	P09	8	F	2	P22	14	M
	P10	9	F		P23	10	M
	P11	9	M		P24	11	M
	P12	11	F		P25	10	M
	P13	8	M		P26	15	F

Table 1. Participant statistics for the pilot study and two stages. PN indicates participant number and M/F indicates gender.

Procedure

We designed four circuits for this exploratory study: two simple circuits containing three littleBits in total, and two larger circuits containing five littleBits in total. We also paired each circuit with a particular container. For example, a circuit for detecting noise sets off an alarm inside a T-Rex container to deter unwanted guests in a room (Figure 9). During the study, we presented these inventions as scenarios to lead users to build the circuits inside the containers with a purpose. The provided scenarios, as well as the size constraints of the selected containers, also encouraged participants to use all levels of a model so that their use of the 3D building space could be

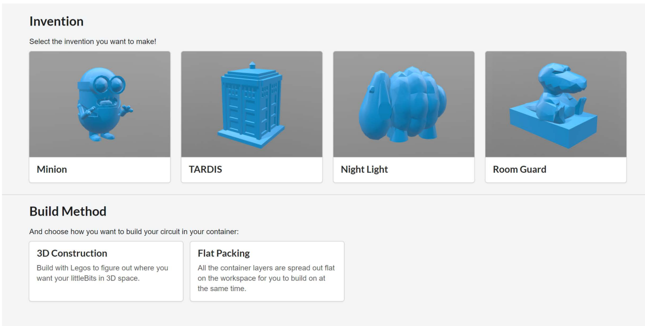


Figure 9. Selection view during the exploratory study: four circuits and containers were designed for scenarios given to the participant. Models ©; Thingiverse IDs respectively 82480, 352308, 2490980, and 913069.

observed. We tested all circuits before the study to ensure they could be successfully built using both interfaces.

We conducted our study in three stages. The first stage study was a pilot with three adults and one child (recruited via convenience sampling). The goal was to get preliminary results and validate our study methods. The second stage was a study with 17 children. In this stage, all participants used the same circuit and container model. By controlling these variables, we were able to focus on comparing the two interfaces. The third stage was a study with five children. Unlike the second stage, participants tested one simple circuit and one complex circuit, each with its corresponding container model. In this stage we focused on the design opportunities our system provided for different levels of design goals.

To equally compare the two interaction techniques, the platform locations in the platform interface were matched in height to the slice locations in the flat interface. This meant that the same container design could be created in both interfaces.

We gave each participant a brief demonstration for each interface to help them understand how they worked. We did not explain the meaning of the projected feedback initially; instead, when the user triggered a new type of feedback, we asked them what they thought it meant. Only if the participant became frustrated did we explain its meaning.

We asked participants to think aloud while interacting with the interfaces, and asked probing questions to understand what the user was doing during the build process. We also conducted a post-study interview to debrief participants and collect more of their thoughts about both interfaces. Additionally, participants filled out a post-study survey to indicate their preferred interface, as well as rate the usability and enjoyability of each interface, using Read et al.'s "smileyometer" Likert scale [53]. Throughout the study, we video- and audio-recorded each participant to use for later analysis.

Pilot Study

We recruited four participants (three adults, ages 24–28, two female; one child, age 9, male) to conduct an initial pilot study. We first introduced participants to littleBits and provided them a few minutes of free time to learn how they worked and how to use them. Afterwards, we explained the purpose of the Printy3D application and how it could be used to build custom containers for littleBit circuits. We then presented participants with one of the physical interfaces, selected at random, and one of the circuits with its corresponding container. After successfully placing the littleBits within the container and pressing the Done button (or verbally indicating they were done), the participant repeated this process with the other interface and another circuit. Due to time constraints, participants were only able to build two circuits in total during the study.

First Stage Study

In this study, we recruited 17 child participants (ages 8–13, 8 female). We gave participants the same introduction and study procedure as in the pilot. However, participants only used one circuit paired with its container model. The container was a "Minion" model (Figure 9) and the circuit contained three modules: a power module, a pressure sensor, and a buzzer. The scenario was that pressing the pressure sensor makes the buzzer sound. We did not specify a predetermined position for these electronics, and participants could put them wherever they wished inside the container. Each participant used both interfaces, presented in random order.

Second Stage Study

In this study, we recruited five child participants (ages 10–15, one female). We gave participants the same introduction as in the pilot. However, in this study, participants used two circuits and containers. One circuit/container combination was identical to that used in the first stage (the "Minion"), while the second was a more-complex circuit containing a power module, a sound trigger, a cloud module, a light module, and a buzzer. These five modules were to be placed into a dinosaur model. The scenario was a "room guard": upon detecting sound, the circuit causes a sound and light alert; it also sends a message via the littleBits web API. Each participant tried the simple circuit first, followed by the complex circuit. The order in which they used the interfaces was randomized.

FINDINGS

As discussed earlier, our focus with the study was to understand whether the two interfaces were usable by children for our simple augmented fabrication task, and what differences there were in interaction between them. Although we hypothesized that the second stage study, with more-complex circuits and containers, would yield different participant behavior, we did not observe significant changes; as such, in the remainder of the paper we report the results for all studies together.

Module Placement

All participants understood the concept of placing littleBits within the digital container for both interfaces. For example, using the flat interface, P12 said "This lower part is like the body and this [pointing to top layer] is the head." Also using the flat interface, designing the T-Rex room guard, P26 commented, "…and then it'll have a light shining through its eyes, and it'll look scary!" P25, using the platform interface, placed a second layer of platform, then looking at the monitor said, "so that's now the neck…that's actually the smallest part, so…since the neck is the smallest part, we should have the smallest [littleBit]." Other participants, while less verbal, demonstrated their understanding of the correspondence between physical littleBit placement and the location of the littleBits in the model.

However, the platform interface did exhibit some placement issues: several participants had difficulties with understanding that the platforms occluded the camera's view of lower levels, and manipulated modules between platforms. Experiencing this issue, P5 said that "it is harder to switch parts at the bottom if you want to keep the [arrangement at the] top."

Projected Feedback

Some participants struggled with understanding the provided feedback when littleBits were positioned in an invalid location. Most understood the red outline when a littleBit was out of bounds, such as when one participant (P22) noted, "it doesn't fit inside" after seeing the feedback. Fewer participants understood that the arrow meant the littleBit needed to be moved, and those who did were not sure of exactly how to move the bit to fix the problem. After some prompting of the form, "what do you think this means?" participants could usually correct the error. The participants who triggered the top-access error feedback were unsure what it meant.

Interface Preferences

Participants were split on their physical interaction preferences. Eleven preferred the flat interface, five liked the platform interface better, and ten liked both equally.

Those who preferred the flat interface thought it was more straightforward and simpler to use than the platform interface, and most participants didn't think it was difficult to understand the 3D models because they could reference the rendered model on the monitor. Regarding understanding the relationship between the projected slices, P19 said, "it's not harder for me because you also see on the computer." Most participants felt that it was easier to manipulate a littleBit in the flat interface. P5 said that with the flat interface, "it is easier to know where to put the [littleBits]." P4 stated that he "liked experimenting with the position" of littleBits when using the flat interface, and also appreciated that the flat interface shows all the layers at the same time. This allowed him to easily move littleBits between sections of the container without needing to disassemble or reassemble anything. P8 said that with the flat interface "it is easier to put on littleBits at specific points." P21 said that "I choose the [flat] one because it's much quicker and you don't need to take layers off and put them on." P26 said that "first I like the Lego interface...because it's nice to see the actual 3D model, but then [the flat one] is a little easier to see all layers all at once."

Participants who liked the platform interface enjoyed it because of its accurate and physical representation of how the container would be built. P1 noted that the platform interface "is the real thing. I can touch [it], so I like [it] more." She also felt that she had a better understanding of where to place littleBits when using the Lego block platforms because they "are more real, like the real world." P6 said that "I like the one with the Legos because you can see, like, where [a littleBit] would be...it's easier." P7 said that the platform interface "shows you how it really works physically." And P12 noted that with the platform interface "you can kind of see what, um, it's kind of going to look like...you can kind of see the Minion shape." P19 said that "The Lego one is good because it shows you, like, where exactly [a littleBit] is on the Minion because you have the layers in front of you."

Participants also had some criticisms about the platform interface. Although the actual projected model slices were the same size in both interfaces, post-study discussion revealed that the platforms gave P22 the feeling of limited interaction space. P14 found the Lego bumps made it difficult to see the projected feedback and suggested a smooth platform surface.

DISCUSSION

Our focus in developing and testing Printy3D was to take a first step towards enabling users to engage in augmented fabrication, using existing objects directly as part of the design process. We are particularly interested in what kind of interaction techniques best support a user's understanding and manipulation of the relationship between physical and virtual to-be-fabricated objects. With the development of two different physical interfaces, we had the opportunity to observe and compare user reactions to each one during the studies, and to draw some conclusions about what design strategies might best apply for future efforts of this kind.

Our first observation is that, as a specialized tool, Printy3D succeeded in taking a task that would be highly complex with traditional tools and made it simple enough for children. Printy3D exemplifies the classic trade-off between a low threshold and a high ceiling [43]: our system lowers both the threshold for learning and the ceiling of what is possible. Despite the perhaps non-surprising nature of this observation, it illustrates that we have taken a step in the right direction, creating a system that allows children to design cases for their electronics using the physical modules as an integral part of the design process.

We were surprised to find that a relatively small proportion of participants preferred the platform-based interface; our tentative hypothesis had been that the one-to-one spatial correspondence between the projection workspace and the rendered model would provide a more-intuitive method to interact with the system and therefore might be preferred. We see several reasons for this evident preference. First, Printy3D's platform interface suffers from technical limitations not present in the flat interface. Because the Lego block-based platforms are not optimized for projection, feedback was more difficult to see. Camera-based tracking prevents the system from observing littleBit modules moved outside of the view of the camera, especially in-between platforms; however, this limitation is not obvious to children and appeared to cause, if not confusion, at least a greater sense of the limitations of the system. Second, while modules on platforms are placed at the same three-dimensional location as in the virtual model, a *holistic* understanding of the state of the system is harder to gain. Just as the platforms occlude the camera's view of modules in lower positions, they also occlude the projector's ability to show slices and feedback indicators. Finally, our participants found moving modules between platforms more difficult than shifting them on the same horizontal plane.

The technological limitations can, no doubt, be overcome with sufficient work: non-visual tracking systems (e.g. magnetic trackers [49]) could allow freer manipulation of the modules, and constructing the platforms from active materials such as LCD panels. However, the fluidity of interaction is another challenge: in our simple three-platform scenario, the main concern for participants was correctly placing modules. In this case, the flat interaction method simplifies moving modules between levels of the model. A more-complex scenario, with multiple overlapping levels, might be easier to understand with the platforms and overcome the flat interface's advantage for moving modules around.

It was clear from participant feedback that the preview rendering on the monitor was key to their understanding of the system. As a user moves a physical module around the workspace (whether in the flat or platform interface), the preview updates in realtime to reflect the module position. This correspondence especially helped participants understand the relationship between the flat interface and the 3D model.

Although most participants understood the red outline feedback given when littleBits were not placed completely inside a container, they did not fully understand the feedback given for bits that needed to either be closer to a side wall or to the surface of the model (Figure 6). One reason could be a lack of experience with littleBits leading to some participants being unclear as to what the modules did and how they interact with the environment; for example, it was not immediately obvious that the LED module required a hole in the container in order to be visible. However, there is clearly room to improve our projected feedback: one participant in the flat interface condition interpreted the arrow feedback as an instruction to move the module to a different level rather than closer to a container wall. One possibility could be to animate a suggested fix by showing the littleBit module moving closer to its needed destination, either in projection for an in-plane movement or on the monitor for a height movement.

Overall, we observed that the projected feedback *did* help participants correct their designs. Even with the feedback that was less obvious, once they understood its meaning participants were quick to move the modules to address the issue.

IMPLICATIONS AND CHALLENGES

Printy3D as a system allows creating containers for littleBit modules; however, we also intended it as a test-bed for systems that allow a user to work directly with existing objects as part of an augmented fabrication design process. Our findings allow us to make a number of recommendations for future systems that use tangible manipulations of existing objects with respect to existing objects; we also identified several challenges for future research in systems of this type to address.

Implications for Design

In-situ information and feedback: Printy3D provides in-situ feedback via its spatial augmented reality interface. This feedback proved effective with our participants, allowing them to quickly understand how the module they were manipulating was situated within the container, as well as to understand—in most cases—if an error in placement had occurred. Like other forms of augmented reality-based annotation [6], Printy3D's feedback has the advantage of being co-located with the objects of interest.

Provide previews: While we found in-situ feedback and information to be useful, our spatial AR technology could not give the entire picture to our participants. In the flat interface, the slices of the model were physically separated from each other, while the platforms in the platform interface prevented display on any but the topmost surface. Providing a preview display on the monitor gave participants a way to confirm the understanding they formed from the projected interface.

Interface and complexity of interaction: As discussed earlier, participants largely found the flat interface to serve their needs best, but this may have been in part due to the simplicity of the task. Our supposition—to be tested in future work—is that more-complex layouts might provide an advantage to the platform-based interface.

Challenges

Positioning physical objects in 3D: The core interaction in Printy3D is positioning the physical littleBits modules in three dimensions relative to the virtual container. Our two interfaces—flat and platform—were two different approaches to this problem. This is a core challenge for future research: while there are an increasing number of examples in the literature of physical *visualization* in three dimensions, there are very few examples of fully-3D tangible *manipulation*.

In-situ design: Printy3D presents one, relatively small step towards in-situ design with existing objects. How can users—especially children—take the next step? Can they be enabled to modify the containers themselves at the same time, or even take the step to designing the containers?

CONCLUSIONS

Our research explores the possibilities of what future tangible design interfaces for augmented fabrication could look like, focusing on enabling the user to work directly with existing objects as part of the design process. As a first step towards this goal, we implemented Printy3D, a system designed to enable children to create containers to hold littleBits electronic modules based on existing 3D models. With the support of digital information, the system benefits from physical interaction with existing objects: augmented overlays and feedback projected over the littleBits provide a connection to the virtual world for a complete picture of the finished build. In Printy3D, tangible and virtual elements complement each other for an even richer and more-engaging experience in the design process. While enabling designers to work fully in 3D with tangible interfaces is challenging, our system was successful, and points the way towards future systems for more-complex designs.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1464377, and HP. Also many thanks to Siyu Li for helping organize user studies.

SELECTION AND PARTICIPATION OF CHILDREN

We recruited 3 adults (ages 24–28, two female) and 23 children (ages 8–15, nine female) via email and social media advertising and convenience and snowball sampling. We gave both children and their parents the description of our study and had them sign consent forms. The study description and consent form for child participants used simplified language in order to ensure informed consent, but contained the same information as the adult consent materials. All study materials were approved by our institution's IRB.

REFERENCES

1. Yoh Akiyama and Homei Miyashita. 2016. Fitter: A System for Easily Printing Objects that Fit Real Objects. In *UIST '16: Adjunct Proceedings of the 29th annual ACM symposium on User interface software and technology*. ACM, New York, New York, USA, 129–131.
2. Tobias Alrøe, Jonas Grann, Erik Grönvall, Marianne Graves Petersen, and Jesper L Rasmussen. 2012. Aerial tunes: exploring interaction qualities of mid-air displays. In *the 7th Nordic Conference*. ACM, New York, New York, USA, 514–523.
3. David Anderson, James L Frankel, Joe Marks, Aseem Agarwala, Paul Beardsley, Jessica Hodgins, Darren Leigh, Kathy Ryall, Eddie Sullivan, and Jonathan S Yedidia. 2000. Tangible interaction + graphical interpretation: a new approach to 3D modeling. *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), 393–402. DOI:http://dx.doi.org/10.1145/344779.344960
4. Bruno Araujo, Ricardo Jota, Varun Perumal, Jia Xian Yao, Karan Singh, and Daniel Wigdor. 2016. Snake Charmer: Physically Enabling Virtual Objects. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, New York, New York, USA, 218–226.
5. Daniel Ashbrook, Shitao Stan Guo, and Alan Lambie. 2016. Towards Augmented Fabrication: Combining Fabricated and Existing Objects. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, New York, New York, USA, 1510–1518.
6. Ronald T Azuma. 1997. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments* 6, 4 (Aug. 1997), 355–385.
7. Patrick Baudisch, Torsten Becker, and Frederik Rudeck. 2010. Lumino: Tangible Blocks for Tabletop Computers Based on Glass Fiber Bundles. *CHI '10: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2010), 1165–1174. DOI: http://dx.doi.org/10.1145/1753326.1753500
8. Eric A Bier, Maureen C Stone, Ken Pier, Bill Buxton, and Tony D DeRose. 1993. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. ACM, New York, New York, USA, 73–80.
9. P. Blikstein. 2013. Digital Fabrication and 'Making' in Education: The Democratization of Invention. In *FabLabs: Of Machines, Makers and Inventors*, J. Walter-Herrmann and C. Büching (Eds.). Bielefeld: Transcript Publishers.
10. Leah Buechley and Michael Eisenberg. 2007. Boda blocks: a collaborative tool for exploring tangible three-dimensional cellular automata. In *Proceedings of the 8th iternational conference on Computer supported collaborative learning*. International Society of the Learning Sciences, 102–104.
11. Erin Buehler, Niara Comrie, Megan Hofmann, Samantha McDonald, and Amy Hurst. 2016. Investigating the Implications of 3D Printing in Special Education. *ACM Transactions on Accessible Computing (TACCESS) - Special Issue (Part 2) of Papers from ASSETS 2014* 8, 3 (2016), 1–28. DOI:http://dx.doi.org/10.1145/2870640
12. Patrick Carrington, Shannon Hosmer, Tom Yeh, Amy Hurst, and Shaun K. Kane. 2015. "Like This, But Better": Supporting Novices' Design and Fabrication of 3D Models Using Existing Objects. In *iConference*.
13. Xiang 'Anthony' Chen, Stelian Coros, Jennifer Mankoff, and Scott E Hudson. 2015. Encore: 3D Printed Augmentation of Everyday Objects with Printed-Over, Affixed and Interlocked Attachments. In *UIST '15: Proceedings of the 28th annual ACM symposium on User interface software and technology*. ACM Press, New York, New York, USA, 73–82.
14. Xiang 'Anthony' Chen, Jeeeun Kim, Jennifer Mankoff, Tovi Grossman, Stelian Coros, and Scott E Hudson. 2016. Reprise: A Design Tool for Specifying, Generating, and Customizing 3D Printable Adaptations on Everyday Objects. In *UIST '16: Proceedings of the 29th annual ACM symposium on User interface software and technology*. ACM Press, New York, New York, USA, 29–39.
15. Michael Eisenberg. 2013. 3D printing for children: What to build next? *International Journal of Child-Computer Interaction* 1, 1 (2013), 7–13. DOI: http://dx.doi.org/10.1016/j.ijcci.2012.08.004
16. Jerry Fails, Allison Druin, Mona Leigh Guha, Gene Chipman, Sante Simms, and Wayne Churaman. 2005. Child's Play: A Comparison of Desktop and Physical Interactive Environments. In *Proceedings of the 2005 conference on Interaction design and children*.
17. Sean Follmer, David Carr, Emily Lovell, and Hiroshi Ishii. 2010. CopyCAD: remixing physical objects with copy and paste from the real world. In *UIST '10: Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, New York, New York, USA, 381–382.
18. Sean Follmer and Hiroshi Ishii. 2012a. kidCAD: Digitally Remixing Toys Through Tangible Tools. *CHI '12: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), 2401–2410. DOI: http://dx.doi.org/10.1145/2207676.2208403

19. Sean Follmer and Hiroshi Ishii. 2012b. KidCAD: digitally remixing toys through tangible tools. In *CHI '12: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

20. Simon Ford and Tim Minshall. 2016. *3D printing in education: a literature review*. Working paper. University of Cambridge. DOI: http://dx.doi.org/10.13140/RG.2.2.33008.05124

21. Madeline Gannon, Tovi Grossman, and George Fitzmaurice. 2016. ExoSkin: On-Body Fabrication. In *CHI '16: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. To appear.

22. S Garrido-Jurado, R Muñoz-Salinas, F J Madrid-Cuevas, and M J Marín-Jiménez. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47, 6 (June 2014), 2280–2292.

23. Renaud Gervais, Joan Sol Roo, and Martin Hachet. 2016. Tangible Viewports: Getting Out of Flatland in Desktop Environments. *TEI '16: Proceedings of the Tenth International Conference on Tangible, Embedded, and Embodied Interaction* (2016), 176–184.

24. Antonio Gomes, Calvin Rubens, Sean Braley, and Roel Vertegaal. 2016. BitDrones: Towards Using 3D Nanocopter Displays as Interactive Self-Levitating Programmable Matter. In *CHI '16: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, New York, New York, USA, 770–780.

25. Tovi Grossman, George Fitzmaurice, and Ramtin Attar. 2009. A survey of software learnability. In *CHI '09: Proceedings of the 27th international conference on Human Factors in Computing Systems*. ACM Press, New York, New York, USA, 649–658.

26. Anhong Guo, Jeeeun Kim, Xiang 'Anthony' Chen, Tom Yeh, Scott E Hudson, Jennifer Mankoff, and Jeffrey P. Bigham. 2017. Facade: Auto-generating Tactile Interfaces to Appliances. In *CHI '17: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM Press, New York, New York, USA, 5826–5838.

27. Shitao Guo. 2016. *Printy: UI for Non-experts to Customize "Skinnable" Internet-connected Devices*. Capstone report. Rochester Institute of Technology.

28. Ankit Gupta, Dieter Fox, Brian Curless, and Michael Cohen. 2012. *DuploTrack: a real-time system for authoring and guiding duplo block assembly*. ACM, New York, New York, USA.

29. Yingdan Huang and Michael Eisenberg. 2012. Easigami: Virtual Creation by Physical Folding. *TEI '12: Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction Pages* (2012), 107–112. DOI: http://dx.doi.org/10.1145/2148131.2148143

30. Nathaniel Hudson, Celena Alcock, and Parmit K Chilana. 2016. Understanding Newcomers to 3D Printing: Motivations, Workflows, and Barriers of Casual Makers. In *CHI '16: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, New York, New York, USA, 384–396.

31. Amy Hurst and Shaun K. Kane. 2013. Making "making" accessible. In *IDC '13: Proceedings of the 12th International Conference on Interaction Design and Children*. ACM.

32. E.L. Hutchins, J.D. Hollan, and D.A. Norman. 1985a. Direct manipulation interfaces. *Human-Computer Interaction* 1, 4 (1985), 311–338.

33. Edwin Hutchins, James Hollan, and Donald Norman. 1985b. Direct Manipulation Interfaces. *Human-Computer Interaction* 1, 4 (1985), 311–338. DOI: http://dx.doi.org/10.1207/s15327051hci0104_2

34. R Jota and H Benko. 2011. Constructing virtual 3D models with physical building blocks. *CHI'11 Extended Abstracts on Human Factors in Computing Systems* (2011).

35. Tiago Justino. 2016. *Wire Routing for 3D Printed Embedded Electronics*. Capstone report. Rochester Institute of Technology.

36. David Kirsh and Paul Maglio. 1994. On distinguishing epistemic from pragmatic action. *Cognitive Science* 18, 4 (1994), 513–549. DOI: http://dx.doi.org/10.1016/0364-0213(94)90007-8

37. Y. Kitamura, Y. Itoh, T. Masaki, and F. Kishino. 2000. ActiveCube: a bi-directional user interface using cubes. *KES 2000: Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies* 1 (2000), 99–102. DOI: http://dx.doi.org/10.1109/KES.2000.885768

38. Manfred Lau, Masaki Hirose, Akira Ohgawara, Jun Mitani, and Takeo Igarashi. 2012. Situated modeling: a shape-stamping interface with tangible primitives. In *TEI '12: Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*. ACM, New York, New York, USA, 275–282.

39. David Ledo, Fraser Anderson, Ryan Schmidt, Lora Oehlberg, Saul Greenberg, and Tovi Grossman. 2017. Pineal: Bringing Passive Objects to Life with Embedded Mobile Devices. In *CHI '17: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 1–11.

40. Joohi Lee, Gentaro Hirota, and Andrei State. 2006. Modeling Real Objects Using Video See-Through Augmented Reality. *dx.doi.org* 11, 2 (March 2006), 144–157.

41. Jinha Lee, Rehmi Post, and Hiroshi Ishii. 2011. ZeroN: Mid-Air Tangible Interaction Enabled by Computer Controlled Magnetic Levitation. In *ACM symposium on User interface software and technology (UIST)*. 327–336.

42. Brenna McNally, Leyla Norooz, Alazandra Shorter, and Evan Golub. 2017. Toward Understanding Children's Perspectives on Using 3D Printing Technologies in their Everyday Lives. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. ACM, New York, New York, USA, 298–303.

43. Brad A Myers, Scott E Hudson, and Randy Pausch. 2000. Past, present, and future of user interface software tools. 7, 1 (March 2000).

44. Yoichi Ochiai, Takayuki Hoshi, and Jun Rekimoto. 2014. Pixie dust: graphics generated by levitated and animated objects in computational acoustic-potential field. *ACM Transactions on Graphics (TOG)* 33, 4 (July 2014), 85–13.

45. Themis Omirou, Asier Marzo, Sue Ann Seah, and Sriram Subramanian. 2015. LeviPath: Modular Acoustic Levitation for 3D Path Visualisations. In *CHI '15: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. New York, New York, USA, 309–312.

46. Bob Parslow and Geoff Wyvill. 2008. Seeing in 3D. *SIGGRAPH Asia '08: ACM SIGGRAPH ASIA 2008 course* (2008). DOI: http://dx.doi.org/10.1145/1508044.1508116

47. Huaishu Peng, Jimmy Briggs, Cheng-Yao Wang, Kevin Guo, Joseph Kider, Stefanie Mueller, Patrick Baudisch, and François Guimbretière. 2018. RoMA: Interactive Fabrication with Augmented Reality and a Robotic 3D Printer. In *CHI '18: Proceedings of the 36th Annual ACM Conference on Human Factors in Computing Systems*.

48. Sara Price and Paul Marshall. 2013. Designing for learning with tangible technologies. *Routledge Handbook on Design in Educational Computing* (2013).

49. Frederick Raab, Ernest Blood, Terry Steiner, and Herbert Jones. 1979. Magnetic Position and Orientation Tracking System. *IEEE Trans. Aerospace Electron. Systems* AES-15, 5 (1979), 709–718.

50. Hayes Solos Raffle, Amanda J Parkes, and Hiroshi Ishii. 2004. Topobo: a constructive assembly system with kinetic memory. In *CHI '04: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, New York, USA, 647–654.

51. Raf Ramakers, Fraser Anderson, Tovi Grossman, and George Fitzmaurice. 2016. RetroFab: A Design Tool for Retrofitting Physical Interfaces using Actuators, Sensors and 3D Printing. In *CHI '16: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, New York, New York, USA, 409–419.

52. R Raskar, G Welch, and H Fuchs. 1998. Spatially augmented reality. *First IEEE Workshop on Augmented Reality* (1998).

53. Janet Read, Stuart MacFarlane, and Chris Casey. 2002. **Endurability, Engagement and Expectations: Measuring Children's Fun**. In *Interaction Design and Children*.

54. Valkyrie Savage, Sean Follmer, Jingyi Li, and Björn Hartmann. 2015. Makers' Marks: Physical Markup for Designing and Fabricating Functional Objects. In *UIST '15: Proceedings of the 28th annual ACM symposium on User interface software and technology*. ACM Press, New York, New York, USA, 103–108.

55. Stefan Schneegass, Alireza Sahami Shirazi, Tanja Döring, David Schmid, and Albrecht Schmidt. 2014. NatCut: An Interactive Tangible Editor for Physical Object Fabrication. *CHI EA '14: Proceedings of the 2014 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (2014), 1441–1446. DOI: http://dx.doi.org/10.1145/2559206.2581189

56. R Shewbridge, A Hurst, and S K Kane. 2014. Everyday making: identifying future uses for 3D printing in the home. *ACM Conference on Designing Interactive Technology* (2014), 815–824.

57. Jungu Sim, Chang-Min Kim, Seung-Woo Nam, and Tek-Jin Nam. 2014. G-raffe: an elevating tangible block supporting 2.5D interaction in a tabletop computing environment. In *the adjunct publication of the 27th annual ACM symposium*. ACM, New York, New York, USA, 87–88.

58. Lei Sun, Tomohiro Fukuda, Toshiki Tokuhara, and Nobuyoshi Yabuki. 2014. Differences in spatial understanding between physical and virtual models. *Frontiers of Architectural Research* 3, 1 (March 2014), 28–35.

59. Joao Marcelo Teixeira, Gutenberg Barros, Veronica Teichrieb, and Walter Franklin Correia. 2016. 3D Printing as a Means for Augmenting Existing Surfaces. In *2016 XVIII Symposium on Virtual and Augmented Reality (SVR)*. IEEE, 24–28.

60. Nobuyuki Umetani and Ryan Schmidt. 2017. SurfCuit: Surface-Mounted Circuits on 3D Prints. *IEEE Computer Graphics and Applications* 38, 3 (2017), 52–60.

61. Christian Weichel, Manfred Lau, and Hans Gellersen. 2013. Enclosed: a component-centric interface for designing prototype enclosures. *TEI '13: Proceedings of the Seventh International Conference on Tangible, Embedded and Embodied Interaction* (2013), 215–218. DOI:http://dx.doi.org/10.1145/2460625.2460659

62. Christian Weichel, Manfred Lau, David Kim, Nicolas Villar, and Hans W Gellersen. 2014. MixFab: A Mixed-Reality Environment for Personal Fabrication. In *CHI'14: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press, New York, New York, USA, 3855–3864.

63. B Whiteley, R Blagojevic, and B Plimmer. 2015a. Tangible-Tango: Designing and Fabricating Tangibles with Tangibles. *AUIC 2015: Proceedings of the 16th Australasian User Interface Conference* 162 (2015). https://www.cs.auckland.ac.nz/{~}beryl/publications/AUIC2015Tangi

64. Brent Whiteley, Rachel Blagojevic, and Beryl Plimmer. 2015b. Tangible-Tango: Designing and Fabricating Tangibles with Tangibles. In *16th Australasian User Interface Conference*. 31–39.

65. Brandon Yee, Yuan Ning, and Hod Lipson. 2009. Augmented Reality In-Situ 3D Sketching of Physical Objects. In *IUI Workshop on Sketch Recognition*.

66. Toshiya Yui and Tomoko Hashida. 2016. floatio: Floating Tangible User Interface Based on Animacy Perception. In *UIST '16: Proceedings of the 29th annual ACM symposium on User interface software and technology*. ACM, New York, New York, USA, 43–45.

67. Kening Zhu, Alexandru Dancu, and Shengdong Shen Zhao. 2016. FusePrint: A DIY 2.5D Printing Technique Embracing Everyday Artifacts. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM, New York, New York, USA, 146–157.