DebugIts: Designing for Learning through Debugging

Debora Lui
University of Pennsylvania
Philadelphia, PA
U.S.A.
deblui@upenn.edu

Deborah Fields
Utah State University
Logan, UT
U.S.A.
deborah.fields@usu.edu

Gayithri Jayathirtha
University of Pennsylvania
Philadelphia, PA
U.S.A.
gayithri@gse.upenn.edu

Yasmin B. Kafai
University of Pennsylvania
Philadelphia, PA
U.S.A.
kafai@upenn.edu

ABSTRACT

In this demo, we will share the development of *DebugIt* challenges—pre-designed electronic textiles projects that have built-in 'bugs' or mistakes for students to discover and fix. Over the course of several years, we developed different versions of these challenges, including different kinds of bugs (e.g. programing errors, messy sewing causing short circuits), as well as different kinds of construction (e.g., sewn projects, electrical blueprints, modular pieces). We will not only share our three DebugIt versions, but also discuss how we have used them within research on students' computational learning.

Tools, Skills and Materials

• Tools➞Microcontrollers • Tools➞Arduino • Skills➞Debugging • Materials➞Felt, Conductive Thread, Sewable Electronics.

Keywords

Debugging, electronic textiles, productive failure

1. INTRODUCTION

In the context of designing, fabricating, programming, and making activities, students will invariably encounter bugs and problems that need resolving. Bugs can come up at many times in the design process—from planning to creating to testing to sharing final products. Yet instead of thinking of these debugging situations as problematic, we see these moments of troubleshooting as rich opportunities for learning, particularly contexts for "productive failure". This concept, introduced by Manu Kapur, highlights the counterintuitive notion that failure precedes later success in learning. His idea for designing effective learning activities was to provide students with tasks in which early failures could help them later on in completing other problems successfully.

While most productive failure implementations have focused on getting students to solve well-defined canonical problems in areas such as mathematics, we propose focusing on the role that failure plays in solving open-ended design problems more common in software and engineering applications. This includes, for instance, the context of electronic textiles where circuits, sewing, and coding combine to lead to challenging, overlapping problem domains . In more open-ended design activities, failure plays a constant and prominent role in the overall learning process. In projects that involve designing software, building car ramps, or engineering bridges, students working in teams or alone constantly run into challenges as they iteratively cycle through design and implementation on their way to product completion.

Yet we also see opportunities in explicitly designing for productive failure rather than waiting for the inevitable but highly variable bugs that will come up in students' designs. To this end, over the past several years we have explored ways to introduce buggy projects, or *DebugIts*, to students in the midst of their designing e-textiles. In this demo, we will share several versions of DebugIts that we designed to test with high school youth in the midst of working on e-textile designs in their classes. We share some of the design decisions we made in creating these challenges—i.e., how many problems to design and what kinds of problems—as well as students' responses to the activities. Although we have designed DebugIts for the domain of e-textiles, we are excited to discuss with others how DebugIts might be used in other areas of making and design.

2. DEMO DESCRIPTION

2.1 Description of the Product/Project

In this demo, we present our development of DebugIt challenges—pre-designed computational projects that have built-in 'bugs' or mistakes for students to discover and fix. Our DebugIt challenges are based in electronic textiles, which are electronic components that can be sewn into circuits using conductive thread onto fabric objects (e.g., clothing, stuffed animals) and programmed in Arduino to perform particular behaviors. This includes actuators (e.g., LEDs, buzzers), sensors (e.g., touch sensors, light sensors), and a microcontroller. Over the course of several years, we developed different versions of these challenges, including different kinds of bugs (e.g. programing errors, messy sewing causing short circuits) in different formats (e.g., sewn projects, electrical blueprints, modular pieces). All versions of the DebugIts were tested with high school students with prior experience in e-textiles. Either in pairs or small groups, students were asked to identify and fix as many bugs as possible within a specified amount of time (usually a class period). For the demo, we will not only share these DebugIt versions, but also discuss our preliminary findings from these studies.

Sewn DebugIts. The first version of DebugIts we designed were pre-sewn, pre-programmed e-textiles projects that contained a series of coding and physical bugs, which prevented them from functioning as intended. The initial iteration of these projects included flat felt images with electronic components sewn into different aesthetic designs, such a map, a flying pig, and an anime character (Figure 1, left two images). While they all looked distinct, they contained identical circuit connections and were intended to work with the same Arduino programs. Included within all these DebugIts were sewing mistakes (e.g., short circuits, reversed polarity, parallel versus series circuit issues), as well as coding mistakes (e.g., constant versus variable connections, control flow issues, and end-state definition mistakes).

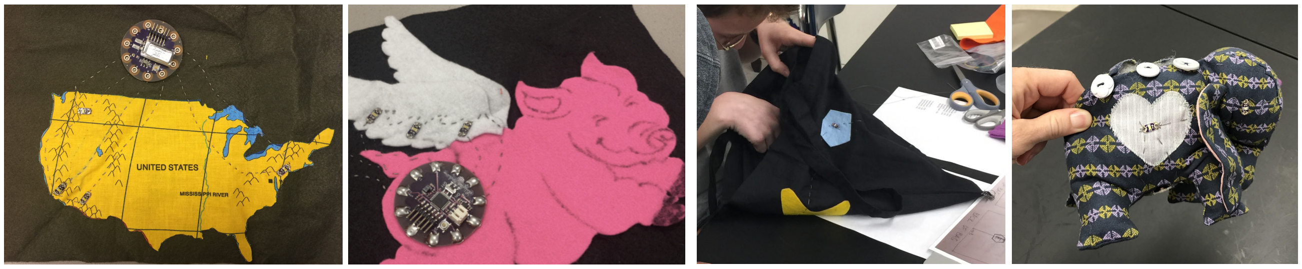

Figure 1. (left two images) The initial iteration pre-sewn DeBugIt challenges, which were flat felt images with codeable circuits sewn on top. (right two images) The next iteration of pre-sewn DeBugIts incorporated three dimensional elements including components sewn into a tote bag. (third from left) A student looks inside the bag for sewing/circuitry mistakes. (right-most) A stuffed elephant DebugIt with sewing/circuitry and code bugs.

In a subsequent study, we revised the DebugIts for more intermediate e-textiles students. While these challenges had similar types of sewing/circuitry problems, we added an additional layer of complexity. First, we incorporated three-dimensional design—something students had reported as challenging during their making experience. In one DebugIt challenge, we sewed LEDs onto both sides of a canvas tote bag (Figure 1, third from left). In order to fix the problems, students had to recognize that laying the bag flat created short circuits in and of itself since both sides of the bag touched each other. Another DebugIt challenge involved a stuffed animal, yet another context that students found difficult to navigate considering both the inside and outside of the project (Figure 1, right-most). These spatial dimensions added complexity to the DebugIt challenge. In addition, we provided more challenging coding errors, in particularly errors that could not be detected by the Arduino compiler. For instance, we mimicked problems that students often introduced to their code, such as mixing up variable names between the name of a switch (for data input) and the storage variable for the switch (to store digital readings from the switch). As with the initial DebugIts, we limited the number of bugs within each challenge.

Both iterations of the sewn DebugIts were appropriately challenging to students: despite having a full class period to solve the challenges, most students did not fix all the bugs. Still, interviews with students up to two weeks after this experience showed that this was a very positive experience, and students pointed to specific ideas that they learned and ways that they applied their learning from the DebugIt experience into their own design practices.

Circuit Diagrams DebugIts. We also sought to explore non-sewn DebugIts in the form of circuit diagrams. Here we presented students with a series of four circuit diagrams that included a number of 'bugs' or mistakes within the drawn electrical connections (Figure 2). We based these on actual student designs in the past (Figure 3, middle and right) where LEDs might be missing a connection to either a positive or negative power source, where lines were crossed (i.e., short-circuited), or where LEDs might not be able to be programmed since they were connected to a non-programmable pin. We made sure to include three-dimensional designs where multiple sides of the artifact had to be presented. In one case, there were differences between the diagrams from different views, and students had to recognize and reconcile those differences. Overall, students engagements with these DebugIts mostly involved redrawing the circuit diagrams completely, and working with the physical objects in order to understand more clearly how the circuits were laid out (e.g., the stuffed animal). Students tended to have an easier time with these drawn DebugIts than the sewn challenges since they only involved one domain (electrical circuitry, rather than programming and sewing). However, they were sufficiently difficult such that students were only able to get through two to three of the four challenges.

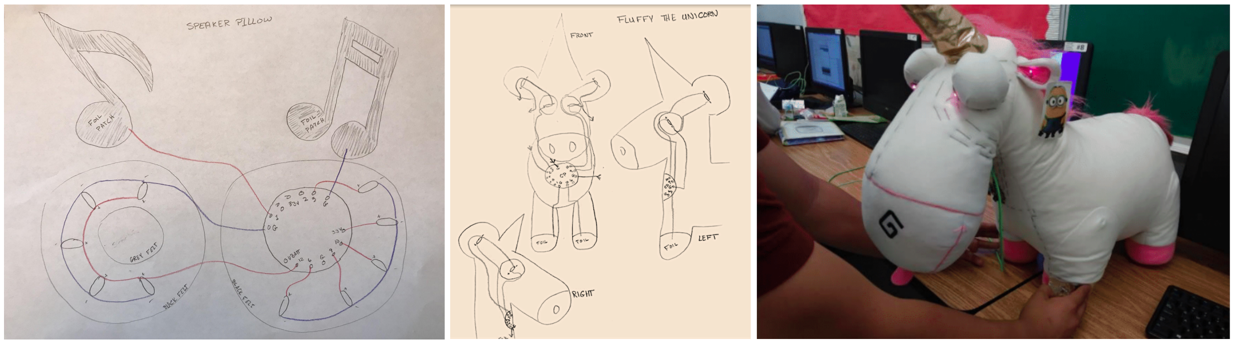

Figure 2. Two Circuit Diagram DebugIts, which included circuit drawings (left, middle) with mistakes such as short circuits, ungrounded LEDs, and sensor patches connected to the wrong microcontroller pins. (right) Shown here is the past project of an actual students, which inspired our circuit DebugIt (middle).

Modular DebugIts (the "Reconstruction Kit"). Our final version of the DebugIts was more modular, using flexible easy-to-attach connections rather than the sewn connections in the earlier versions of DebugIts . To bypass the usual steps of crafting an electronic textile project, we created what we call the 'Reconstruction Kit', which uses metal hooks and safety pins for quickly deattaching and reattaching components together to make functional circuits (Figure 3). The main reason why we developed this kit was because of the time and effort required for students to debug the pre-sewn DebugIts. There, even if students could quickly identify a circuitry/sewing issue, it might take them several minutes to cut and undo stitches, and several more to resew them into secure connections. Further, there is no way of testing out a solution before committing within this sewn format. While this experience is more authentic to the debugging that students go through while actually making an e-textile project, we sought to make this process easier using the Reconstruction Kit in order to allow students to test their solutions and experiment with fixes, thereby scaffolding their learning throughout the process of problem solving.

In order to test out this idea, we created two iterations of the Reconstruction Kit. The first iteration uses a foundation 'mat' upon which DebugIt challenges are built. The mat contains an affixed microcontroller with hooks, which can be connected to LEDS and buzzers using felt strips with conductive thread (Figure 3, left two images). As with the sewn DebugIts, student groups were given a pre-assembled circuit with uploaded code, that contained both programming and circuitry bugs. Unlike the sewn DebugIts, however, we saw that students were able to more easily test possible solutions because of the ease of taking apart and reassembling the circuit, something that supported the growth of their debugging skills.

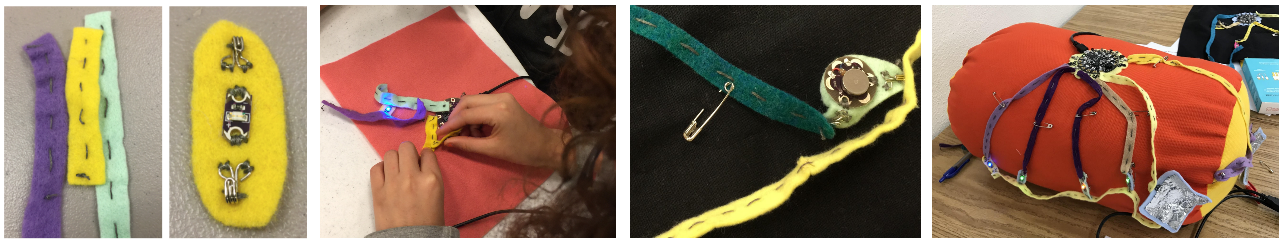
Figure 3. (left-most) The initial iteration of the Reconstruction Kit which includes modular pieces such as felt connection strips and hooked LEDs for easy connections. (second from left) The kit used foundation 'mats' for building DebugIts. (third from left) The next iteration of the Reconstruction Kit with safety pin connectors that allow attachments to any fabric element. (right-most). A solved DebugIt challenge using this reconstruction kit attached to a bolster pillow.

For the more advanced e-textiles students, we developed a second iteration of the reconstruction kit. This kit differed from the first because we eliminated the foundation mat, and added safety pin connections (see Figure 3, third from left image) so that the pieces could be attached to existing fabric objects such as a bag, a sweatshirt, and a pillow (see Figure 3, right-most image). We also added sensor elements including aluminum patches, which could be used as touch sensors. Student pairs were again given a pre-assembled buggy circuits and code, and asked to fix as many issues as possible within a class period. Because of the ease of connecting and disconnecting the circuits, students ended up going through more DebugIt challenges within the same time (e.g., instead of one DebugIt, two or more challenges). Ultimately, this meant we were able to expose students to a wider range of bugs during this time, that is, not only the issues within one DebugIt, but within multiple challenges.

Despite the advantages of using these reconstructions kits within DebugIts, we witnessed a few limitations. While the reconstruction kits were useful in quickly testing solutions, they had spatial constraints that were different from actual sewn projects (e.g., lack of short circuits due to felt insulation of connector strips). Because of this, students sometimes created solutions using the reconstruction kit that would not properly translate to sewn contexts.

2.2 Target Audience

We believe our demo will be relevant for two audiences. First, teachers and educators working in the areas of "making"—especially hybrid modalities that use computational technology (i.e., coding)—should be able to benefit from this demo. Second, we would like to share our work with researchers who are not only interested promoting spaces of productive learning in hybrid (physical, digital) areas of making, but also studying how to support students' problem solving, computational thinking, and maker practices through debugging activities.

3. CONCLUSION

3.1 Lessons Learned

In developing and testing these DebugIts over time, we learned lesson about: 1) the material and conceptual design of DebugIt challenges, and 2) students' engagements and learning through debugging.

Experimenting with multiple versions of the DebugIts helped us to compare the affordances of different formats for supporting student learning. One question we had to ask was regarding the nature and quality of the bugs themselves—namely, what counts as a 'productive' bug? While some bugs were cumbersome to fix or required a lot of repetitive effort, we noticed that they yielded very little benefit for students in terms of pushing their conceptual understanding. Designing DebugIts therefore required carefully weighing the effort associated with fixing a bug and its ability to push student learning. We also had to think about how to control the scope of the DebugIt problem space without allowing for bugs to intersect each other in ways unproductive for learning. Because e-textiles includes multiple domains, bugs might involve coding errors, circuit connections, or faulty sewing—we had to be cautious of not overwhelming students. What are the appropriate constraints that will limit this problem space such that students are not just abandoning the buggy project and restarting but are grappling with the issues presented in the challenge? Yet another element we had to consider involved the material form of the DebugIts themselves and how that influences students actions. Above, we already spoke about the potential limitations of both pre-sewn and reconstruction kit Debugits. With either of these forms, we had to consider how to appropriately balance ease of engagement with authenticity of the making context.

Overall, we discovered more about students' experience of debugging, as well as what they learned through the process across the different formats of DebugIts. We were initially surprised to find out how much students enjoyed solving these challenges and how much they learned from dealing with a buggy project that was not their own. In this respect, we saw how DebugIts have game-like properties, in that they contain limited, designed problems with a clear endpoint (the end of the class period), and are thus intrinsically motivate students to participate. Additionally, through students' engagements across multiple forms of DebugIts, we were able to identify specific aspects of problem solving/debugging that was most challenging for them (e.g., isolating parts of the multi-domain system and testing these on their own before combining them back together). Finally, observing students debugging in real-time has given us ideas about what supports are needed throughout this process. In the coming year, we are working to incorporate these debugging activities into our existing e-textiles unit—where students normally only create projects—through collaboration with two exemplary high school CS teachers.

3.3 Broader Value

The maker education community tends to focus—understandably so—on the design and creation of new projects. Yet, our studies illustrated how working through mistakes and valuing the process of construction (and deconstruction) can be tremendously valuable in supporting students' learning and their ability to innovate over time. Even one hour of class time working on DebugIts seemed to help students better appreciate the importance of dealing with mistakes in the making process, and better articulate their problem solving processes to themselves. While we have designed DebugIts specifically in the domain of e-textiles, we believe the general concept and process could apply to many areas of making and computing, and hope that our lessons learned will provide a head start to others as they consider supporting greater learning through the process of debugging.

3.3 Relevance to Theme

When we think about maker education, we need to think beyond the overly consumeristic impulse to creating more and more things. Although it is not the emphasis of this demo, focusing on debugging and fixing as part of the creative process may encourage makers to reuse, fix, and adapt older or broken items (a car, a record player, a piece of clothing). Further, this approach provides opportunities for educators in these spaces to reuse existing artifacts and student work for furthering student learning. Makers and students in these spaces need to develop sensibilities for persevering in the face of problems and thinking creatively about problems solving, and this demo shows one way to focus on fixing and debugging as valid skills in and of themselves.

4. REQUIREMENTS

We require a small table to display samples of our different DebugIts versions for audience interaction, plus space to hang a poster highlighting our findings from the different studies.

5. BIOS

Debora Lui is a postdoctoral fellow at the Graduate School of Education at the University of Pennsylvania.

Deborah Fields is an Associate Research Professor of Instructional Technology and Learning Sciences at the College of Education and Human Services at Utah State University.

Gayithri Jayathirtha is a doctoral student in the Teaching, Learning, and Leadership Division at the Graduate School of Education at the University of Pennsylvania.

Yasmin Kafai is a Professor of Learning Sciences at the Graduate School of Education at the University of Pennsylvania.