HCI education reflects the continual evolution of HCI, embracing the changing landscapes of technology, infrastructure, and technology use. This forum aims to provide a platform for HCI educators, practitioners, researchers, and students to share their perspectives, reflections, and experiences related to HCI education. **— Sukeshini Grandhi, Editor**

'Back off, man. I'm a scientist.' Using Fiction to Teach Beginners HCI

Jennifer Golbeck, University of Maryland

Introductory human-computer interaction (HCI) classes have a few common learning objectives: developing the ability to understand users, their environments, and their tasks; learning to design interfaces to support them; and learning to evaluate the interfaces' usability. A final project in such a class should leverage all these skills. This supports the core goals of HCI education laid out in 1992, that HCI is "a discipline concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them" [1]. It also allows for the exploration of new technologies, topics, and environments, as described in [2].

Projects in beginner HCI classes tend to include the following: Analyze a type of user and choose a particular problem or set of tasks that is important to them; design an interface to support those users and tasks; prototype, evaluate, revise, test, and present. Having graded these assignments and watched presentations in other classes, there is a common issue. Overall, students fulfill all the requirements and show that they have a general mastery of these skills. However, the majority of projects end up being straightforward variations on common systems. Another website. Another mobile interface. Another database front-end. Whether students have open-ended projects with total freedom to choose an application space or guidance to specific users with real problems, most students take what they know from the interfaces they use, adjust one or two features, and call it a day. The results are often… uninspired.

This variations-on-a-theme project can be a perfectly reasonable and successful way to learn HCI. It may be especially comfortable for those lacking confidence and wanting a safe, straightforward path to show they have mastered the material. But the projects I love to see as an instructor—the ones I tell colleagues about and the ones that show students have deep understanding—are the more creative ones: a WiFi-enabled play set for grandparents and grandchildren to use over distance. The iPhone app that, strapped to your arm, counts reps in the weight room. A T-shirt with attached organs and sensors to teach kids about anatomy. When students look beyond the interfaces and environments they know, they think more deeply about their users, designs, and evaluation.

This drove me to develop a term project for our Fundamentals of Human-Computer Interaction class. The project had the same skeleton as previous projects: pick a problem for a specific user type; propose an interface solution; develop a list of tasks; prototype and get feedback; refine the design; conduct a usability analysis; and present the results. But in this project, I would require students to design for a specific group of users: the Ghostbusters.

The Ghostbusters are a team of scientists and paranormal experts who capture ghosts that torment the citizens of New York City. They first appeared in the classic 1984 film and were rebooted in the summer of 2016. They use their science backgrounds to build equipment that can capture and contain troublemaking spirits, all the while encountering obstacles, physical challenges, and slime. They represent a set of users working in a challenging, unusual environment with a unique but clearly defined set of tasks. Because their work is documented in detail in movies, there is an easy-to-access user reference for students.

Insights

→ Fiction can be a useful tool for teaching intro to HCI because it pushes students to more deeply analyze users and their environments.

→ Running a class with a *Ghostbusters*-themed term project led to more creative projects, supported students with no programming background, and improved the depth of user analysis.

CREATING THE PROJECT

A final project should allow students to practice and integrate all the major skills they learned during the semester. In many courses that teach HCI outside of a computer science department, students have limited programming expertise; in my classes, only about 30 percent of students have a CS background. As a result, this project cannot require programming. At the same time, it is critical that students build prototypes that they can test and demo.

Students can certainly choose to build software interfaces. However,

IMAGE BY SHUTTERSTOCK.COM / TINSELTOWN

by designing equipment for the Ghostbusters, the project supports non-programmers in a few important ways. Students can create *physical* devices. They can use existing technology and modify its normal functionality for their designs (e.g., mounting movable ghostbusting technology on the base of a remote-controlled car). For software interfaces, students who do not program can build intricate prototypes with tools like Invision or even Powerpoint. The project also offers programmers opportunities to push themselves in creative ways, dipping their toes into the worlds of virtual reality, app development, or graphic design.

The project has four major parts:

- *Analyze the users, their environment, and their tasks.* We do this by watching *Ghostbusters* in class. We review specific scenes that show the Ghostbusters using technology or working in a typical ghost-filled environment to highlight important environmental attributes. These include things like omnipresent slime, details of their uniforms and equipment, and the active, strenuous activities they perform.
- *Identify a technology to build and explain what tasks it will support.* I offer suggestions, like building a new ghost trap, containment unit, interface, or ghost detector, but students are free to pick whatever they like.

It was a valuable exercise for teaching students about designing for special populations.

- *Create, evaluate, and iterate with low- and high-fidelity prototypes.* We use heuristic evaluations, cognitive walkthroughs, personas, and user testing (where classmates act as Ghostbusters since they have domain expertise). I bring in uniforms and equipment to force students to test their tech in "realistic" conditions.
- *Present.* Create a final prototype/demo, present it, and write a paper describing the whole process of analysis, design, and evaluation.

We spend the final six weeks of the 15-week semester on this project, with intermediate deadlines every week or two. By this point in the term, students have had lectures and practice with design techniques, task analysis,

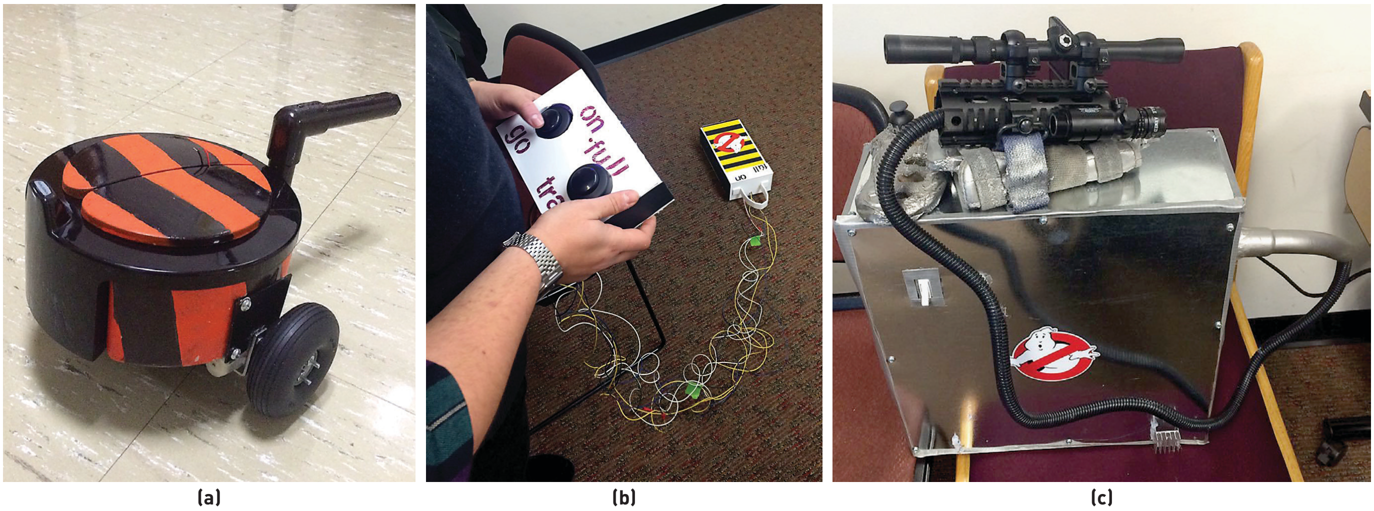


Figure 1. A few project prototypes: ghost traps (a, b, d), a redesigned proton pack (c), and a virtual-reality ghost-information system (e).

and usability testing. We conduct evaluations and demos in class, which keeps the students unified and allows them to leverage one another as testers. Class meetings continue with parallel lectures, but the progress of the project does not rely on the content of the ongoing material.

FIRST-RUN RESULTS

After running this project in 2015 and 2016, there were real, tangible results of using fiction for this exercise:

- It forced a *deeper analysis* of users, environment, and usability. Because the project is so unusual, students can't rely on what they have seen or on standard "best practices" guidelines. While students could look deeply at any population, the *Ghostbusters* universe was strange enough that most students felt they had to put a lot of time into this analysis. The only projects where I didn't see this depth were those that opted for a more standard technology (e.g., designing a ghostbusting scheduling system). I eliminated that option in 2016 for this reason.
- The projects were much *more creative*. Students seemed less fearful of making mistakes in their design. There was no "normal" way to build ghostbusting technology, and this reduced the perceived risk in trying something crazy and different.
- Students *learned a lot of tech skills* on their own, much more than when I give a non-fiction-based project. Students taught themselves Arduino programming, built voice-controlled devices, and found ways to combine existing technologies in creative ways. Students who could already program sought out more challenging applications, such as building a VR space for ghost education (see Figure 1e).
- It was a valuable exercise for teaching students about *designing for special populations*. Students learned to analyze a population's needs with no ability to rely on stereotypes or assumptions. The lessons from the Ghostbusters carried into our discussions of other special populations.

WHY NOT 'REAL' USERS?

Educators teaching these introductory HCI classes often believe that it is important to give students experience in designing for real-world users. I believe strongly that, at least for beginners, there are many reasons fictional scenarios can be superior.

It makes students think harder. In this crazy world, students cannot build on existing technology. This precludes projects that produce incremental improvements, additional features, or re-implementations of interfaces from one domain in another.

The Ghostbusters are often covered in slime. Smartphone touchscreens have no place in this environment.

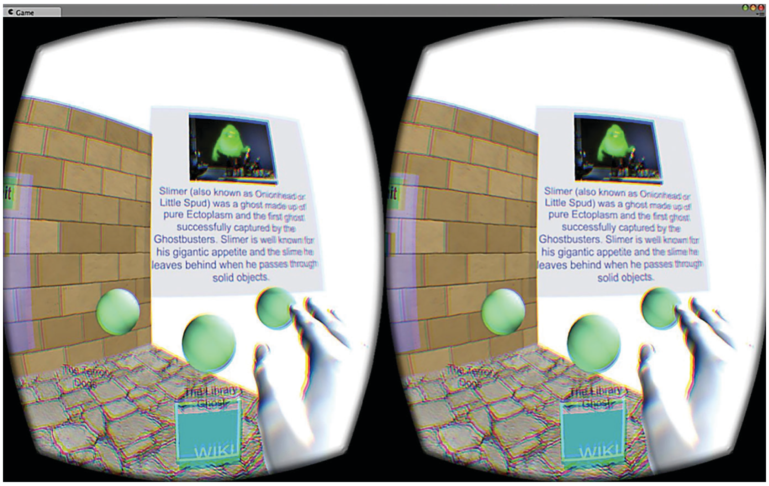
Students also need to think much more deeply about the users and their environment. Ghostbusting is physical, with unusual challenges. The ability to deeply analyze users and their environment is a critical skill for any HCI professional, and fiction forces more of it than a familiar environment would.

The Ghostbusters are always available. True domain experts aren't often available to spend a lot of time with students. With fiction, the students have a small, easy-to-access set of information to work with in place of an expert. They can take one two-hour movie, play it back any time, analyze scenes, watch the users in different scenarios, and judge their own interfaces against the action they see in the film. While they cannot ask questions, they can point to specific scenes to justify their choices (and the instructor can point to specific scenes to challenge them).

Fiction thwarts assumptions. Even if a real-world expert is available and engaged with students, I have seen students assume many things about the users, environment, tasks, or the usability of their designs. Sometimes this is because they don't know the right questions to ask the experts. Sometimes the experts don't really know how to answer the students' questions. Sometimes the students simply commit to an idea and don't discuss it.

We encountered these assumptions in my class's first run of this project.

(d)



Slimer (also known as Onionhead or Little Spud) was a ghost made up of pure Ectoplasm and the first ghost successfully captured by the Ghostbusters. Slimer is well known for his gigantic appetite and the slime he leaves behind when he passes through solid objects.

Slimer (also known as Onionhead or Little Spud) was a ghost made up of pure Ectoplasm and the first ghost successfully captured by the Ghostbusters. Slimer is well known for his gigantic appetite and the slime he leaves behind when he passes through solid objects.

(e)

Several students designed touchscreen interfaces on smartphones as the main interface component of their design. That simply will not work in the Ghostbusters scenario. The Ghostbusters wear rubber gloves on the job, and they are often covered in slime. They tend to run around carrying heavy equipment. Smartphone touchscreens have no place in this environment. To correct these students' assumptions, we watched a short clip that showed the Ghostbusters in uniform. Then I had the students put on a set of rubber gloves and use their interfaces. Revisions were clearly necessary.

In fact, basically every scene in the movie can be reenacted. This allows students to test their technologies doing the same things the Ghostbusters would do. Being able to immerse themselves in the environment and act like Ghostbusters allows for a more visceral understanding of the users and their environment. This stopped many assumptions before they got too far.

Reduce tech intimidation. Students can become intimidated when building prototypes for systems that may require a particular technology when deployed for real use. For example, if they were designing a system for firefighters, there would inevitably be some students who couldn't get past the fact that their prototype was not fireproof. Even in more common environments, students can get stuck on certain technology requirements, like building a functional database to support an interface project. While that's not necessarily a bad thing, these students often sacrifice their interface design and testing for the work on non-interface elements.

Fiction helps with this. Since there is no way for students to build functional spirit-related technology, they feel free of the constraints that come with something that might eventually be a product. Once they internalize that they don't need a fully working system, students let go of unnecessary details throughout the project.

LESSONS LEARNED

In the second run of this project, I implemented changes based on these lessons learned from the first offering:

- Some projects were too simple. They met the requirements but could have been implemented as a one- to two-week assignment. On the plus side, this suggests that the project could be modified to serve as a midterm or long homework. On the other hand, the requirements for a major term project needed clarification. I strengthened the intermediate deliverables to require more writing and analysis, and gave closer attention to the "initial ideas" phase of the project. I spent more time vetting projects with students to make sure they were ambitious enough.
- Some students became too focused on building hardware, distracting them from the main goal. This opportunity to build hardware prototypes was very exciting for some students, but some were so enthusiastic about their hardware ideas that they lost sight of the larger design goal. In the second offering, the assignment included a stronger emphasis on building from the user context, environment, and tasks at every step. I asked students to connect each design decision back to those fundamentals to keep them focused on the target users.

Is fiction the best option for *all* HCI classes? Absolutely not. Working on projects with real users; building a portfolio with recognizable, useful applications; and mastering the skills associated with designing and testing in a realistic environment are incredibly important. However, when students are starting out, learning the basics of HCI and working on their first project ever, fiction has benefits! It circumvents many issues that lead to uninspired students and ordinary projects while forcing students to pay more attention to the user analysis skills they should be mastering early on.

ENDNOTES

1. Hewett, T. et al. *ACM Curricula for Human-Computer Interaction.* 1992; http://old.sigchi.org/cdg/index.html
2. Churchill, E.F., Bowser, A., and Preece, J. Teaching and learning human-computer interaction: Past, present, and future. *Interactions 20*, 2 (2013), 44–53.

Jennifer Golbeck is an associate professor in the College of Information Studies at the University of Maryland. She studies social media and builds algorithms that understand people from their data.
→ jgolbeck@umd.edu

DOI: 10.1145/3029599 © 2017 ACM 1072-5520/17/03 $15.00