BLOG @CACM

The *Communications* Web site, <http://cacm.acm.org>, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.

Follow us on Twitter at <http://twitter.com/blogCACM>

DOI:10.1145/3105421

<http://cacm.acm.org/blogs/blog-cacm>

How Adults Ages 60+ Are Learning to Code

Philip Guo discusses his project studying older adults that have chosen to learn computer programming.

Philip Guo
Older Adults
Learning Computer
Programming:
Motivations,
Frustrations,
and Design Opportunities

<http://bit.ly/2rmGla5>

May 15, 2017

I recently published and presented a paper at CHI 2017 (the annual ACM Conference on Human Factors in Computing Systems, <https://chi2017.acm.org>) called “Older Adults Learning Computer Programming: Motivations, Frustrations, and Design Opportunities” (<http://bit.ly/2snS4LN>). This paper won an Honorable Mention award at the conference. Here’s a summary of the project.

There is now tremendous momentum behind initiatives to teach computer programming to a broad audience, yet many of these efforts (for example, Code.org, Scratch, ScratchJr, and Alice) target the youngest members of society: K–12 and college students. In contrast, I wanted to study the other end of the age spectrum: *how older adults aged 60 and*

over are now learning to code.

Why study older adults in particular? Because this population is already significant and also quickly growing as we all (hopefully!) continue to live longer in the coming decades. The United Nations estimates that by 2030, 25% of North Americans and Europeans will be over 60 years old, and 16% of the worldwide population will be over 60. There has been extensive research on how older adults consume technology, and some studies of how they curate and produce digital content such as blogs and personal photo collections. But so far nobody has yet studied *how older adults learn to produce new technologies via computer programming.*

Thus, to discover older adults’ motivations and frustrations when learning to code, I designed a 10-question online survey that asked about their employment status (such as working, semi-retired, retired), occupation, why they are learning, what resources they use to learn, and what has been the most frustrating part of their learning experience thus far.

The first challenge was finding a large-enough group of older adult

learners to fill out my survey. Fortunately, I created a popular learn-to-code website called Python Tutor (pythontutor.com), which has gotten over 3.5 million total visitors from over 180 countries throughout the past decade. Approximately 16% of its user base self-report as aged 55 and older, so there are plenty of older adults learning to code on there.

I deployed my survey to the Python Tutor website from March 2015 to August 2016 and collected 504 responses. Respondents were, on average, 66.5 years old, and came from 52 different countries. Unsurprisingly, most were highly educated professionals in STEM (science, technology, engineering, mathematics) fields, since they are amongst the most tech-savvy of their generation. Specifically, 18% of respondents were (either current or retired) scientists and engineers, 18% were K–12 and college teachers, 12% were software developers hoping to learn new technologies, and 8% were business executives and managers.

Motivations

Why were our respondents learning

programming? The most common age-related motivations were:

- 22% wanted to learn to make up for missed opportunities during their youth.

- 19% wanted to keep their brains challenged, fresh, and sharp as they aged.

- 5% were directly motivated by younger relatives such as children or grandchildren.

Here's a great anecdote about learning to make up for missed opportunities during youth. A 67-year-old retired chief information officer wrote in his survey response:

"I did a little programming when I was in school, and when I first started working. However, I got 'kicked upstairs' [into management] quite quickly, and was never able to program professionally. [...] I always wanted to be able to create programs but between work and family, never took the time. Now that I am retired, I am trying to fulfill the dream and learn."

Relatedly, a 64-year-old retired networking engineer wrote about his desire to keep his brain sharp and to create technologies that benefit peers in his generation:

"First, by endlessly learning new things, I hope to delay or reduce the effects of senility on my brain. [...] Second, to take advantage of data produced by the many health-related, sensor-based monitors, I want to help myself and other senior citizens maintain an independent living lifestyle that is affordable by the masses."

Frustrations

What got our respondents frustrated as they were learning to code? The three most commonly reported age-related frustrations were:

- 14% were frustrated by perceived cognitive impairments, such as memory loss and difficulty in concentrating.

- 11% were frustrated by lack of free time since they often had other duties, such as being a spousal caretaker.

- 10% were frustrated by lack of human contact with tutors or peers, since they must learn online and do not have convenient access to in-person classroom environments.

A 71-year-old retired IT technician

Older adults do not want to be patronized, to be talked down to, or made to play with "kids' toys."

humorously wrote about his own perceived cognitive impairments:

"Given that I was a VERY early adopter of microprocessor/microcontroller technology, I have NO fear of the equipment or the concepts. But things that were 'automatic' a few years back seem to take a lot more time and effort to digest and store than they used to. Early onset Alzheimer's? Probably not. ACS? (Advanced curmudgeon syndrome)—Probably some of that."

Design Opportunities

Inspired by the findings from this study, I applied the Learner-Centered Design framework developed by Mark Guzdial in his book *Learner-Centered Design of Computing Education: Research on Computing for Everyone* (<http://bit.ly/2seYahY>) to propose design ideas for improving the learning experience for this older adult population. Three main themes emerged from my design process:

- **Targeting:** Like everyone else, older adults want to feel that programming curricula and tools "look like they're for me"—that is, that they are properly targeted to the motivations, needs, and aesthetic preferences of this population. They do not want to be patronized, to be talked down to, or made to play with "kids' toys." Several survey respondents mentioned brain training games (for example, from Lumosity) as being popular with their peers, so perhaps framing programming education in terms of brain-training games could work well for this audience.

- **Contextualizing:** It is also important to ground learning materials

in contexts that engage this learner population, rather than trying to find a generic "one-size-fits-all" solution. Examples of relevant contexts here include structuring curricula around coding projects to help older adults curate digital media, to perform genealogical and historical storytelling, and to organize their personal healthcare data.

- **Universal Design:** The promise of universal design is that designing for the specific needs of a target population (such as older adults) can lead to designs that benefit *everyone*. In this case, we may want to design next-generation pedagogical programming environments that mitigate the effects of both cognitive and motor impairments, which will hopefully make it easier for older adults to learn to code without as many frustrations. If properly designed, these environments may actually end up benefiting learners of *all ages*.

Parting Thoughts

The tech world is notoriously youth-centered: popular conceptions of who learns and does programming are filled with images of young people, often under 30 years old. Also, age discrimination (see <https://bloom.bg/2qCdIv6>) is an all-too-common reality in the technology sector. To counteract these prevailing trends as people keep living longer in the coming decades, it is vital for older adults to have equal access to high-quality computing and programming education throughout their lives.

We have already made great strides in broadening participation of computing to traditionally under-represented groups ... but there is still much, much more work to be done. Efforts to spread the power and joy of *computing for all* should also include people of all ages.

That's it for now! You can read my paper for more details: *Older Adults Learning Computer Programming: Motivations, Frustrations, and Design Opportunities*, at <http://bit.ly/2snS4LN>.

Philip Guo is an assistant professor of cognitive science at the University of California, San Diego.

© 2017 ACM 0001-0782/17/08 \$15.00