Challenging Stereotypes and Changing Attitudes: The Effect of a Brief Programming Encounter on Adults' Attitudes toward Programming

Polina Charters, Michael J. Lee, Amy J. Ko, Dastyni Loksa
Information School
University of Washington
{polinac, mjslee, ajko, dloksa}@uw.edu

ABSTRACT
Computer programming is now used broadly across many industries, with a diversity of working adults writing programs and interacting with code as part of their jobs. However, negative attitudes toward programming continue to deter many from studying computer science and pursuing careers in technology. To begin understanding adults' attitudes toward computer programming and how we can improve them, we used an educational video game to give 200 adult participants a concrete programming experience via the web, and then collected their self-reported opinions about programming. We found that adults initially had poor attitudes toward programming, believing that it was difficult, boring, and something they generally could not learn. After the online learning experience, their attitudes improved significantly, regardless of gender, population density, or level of education. These results demonstrate that adult attitudes toward programming, while initially negative, can be quickly changed with a brief, positive exposure to programming.

Categories and Subject Descriptors
K.3.2 Computer Science Education: Introductory Programming

General Terms
Human Factors, Experimentation

Keywords
Programming attitudes; adult programming; novice programming.

1. INTRODUCTION
Programming in the workplace is becoming more commonplace for many of today's careers. In fact, research based on U.S. Bureau of Census and Bureau of Labor data shows that while there are about 3 million professional programmers in the United States, over 13 million more people say they do programming at work, and over 90 million use spreadsheets and databases [5,26]. American workers program using spreadsheet systems, web authoring tools, business processing tools such as Visual Basic, graphical languages, and even professional languages such as Java [18]. These workers, or end-user programmers, are primarily adults who write code in order to complete other tasks, like designing a web page, doing office work such as accounting, and conducting scientific research [18]. Recent surveys have found that the computer literacy requirements have skyrocketed in almost every end-user category with one showing that over 70 percent of the surveyed companies now require computer competency in their middle and senior management positions [6]. As more adults continue or switch their careers, they will likely need to become end-user programmers to sustain gainful employment in many fields.

Even though computer programming has become a technical skill of millions, workers sometimes may not recognize what they do as "programming" [19]. For example, despite their considerable programming skills, many system administrators [2] and research scientists [5] view their programs as only a means to facilitate their other work, such as keeping a network or other services online. Sadly, many Americans even hold negative perceptions of computer science and people who work with code [23]. Some believe that programming is too difficult to learn, associating the ability to do so with intelligence, logical thinking, and genius they do not possess [3]. Moreover, many believe that programmers lack interpersonal skills [1], are socially awkward [23], and are so obsessed with technology that they have no other interests [23]. These views are especially more pronounced in younger girls and women [7]. These stereotypes can create negative attitudes toward programming, discouraging individuals from learning and accepting programming as a skill necessary for their professional development. Attitudes are important because they, in combination with perceptions of expected behavior, determine how one is likely to act towards specific activities [14] such as learning computer programming. For example, positive attitudes toward computers have been shown to increase computer use and understanding of subsequent computer-related skills in youth and older adults [11,26,32].

In an effort to reduce negative stereotyping and improve attitudes toward programming, educators and researchers have targeted elementary, middle school, and high school students with interventions such as computing camps and educational games [32] that facilitate exposure to basic programming concepts. While these efforts are necessary for helping students and other young adults to view computer science and programming as something within reach, little research has investigated adult attitudes towards programming [29]. In his mainstream 2012 magazine article, Wisniewski urges adults who are interested in programming to " . . . swallow [their] pride and seek out resources aimed at kids," which demonstrates the lack of content designed for consumption by adults [33]. Unfortunately, we know little about the adults who would find programming appealing and how they react to initial programming experiences.

SIGCSE '14, March 05 - 08 2014, Atlanta, GA, USA

http://dx.doi.org/10.1145/2538862.2538938

Most up-to-date version: 06/25/2021

To begin exploring adult behavior and attitudes towards programming, we exposed 200 adult participants to a simple online educational programming environment that asked them to complete a series of basic programming tasks [20,21,22]. We then asked them to describe their views toward programming. We also explored how these views vary by individual demographic factors – such as age, gender, population density, and education – and how these demographic factors may influence adults' attitudes. In the rest of this paper, we discuss prior work on efforts to improve computer-related attitudes in youths, elderly adults, and end-user programmers; we then describe our methodology, present our results, and conclude with a discussion of our results' implications on computing education.

2. RELATED WORK

Besides working adults, researchers have identified many other groups of end-user programmers, including young children [25] and middle-school students [17]. Although contemporary youth sometimes engage with environments such as video game design where they unwittingly learn end-user programming skills, they still hold negative views toward programming [25]. Some think that code is too difficult to learn [17], that it is boring [16], "geeky" [1], and is ultimately a solitary occupation that does not generalize to other fields of interest [23]. Fun and engaging learning environments, such as Scratch, Alice, and Greenfoot seek to improve kids' perceptions of programming [32].

Working adults have been mostly studied in the fields of Information Systems and Education in the context of end-user computer training; there is little research focusing on working adults' attitudes towards or experiences with programming. The goal of end-user training is to produce a motivated user who has the skills needed to perform a job-related task [30]. Studies show that positive attitudes play a critical role in learning outcomes and successful progress in end-user training [12]. For example, Szajna and Mackay concluded that attitudes toward computing and achievement were related to adults' performance with basic Information Technology skills [30]. Another study showed that providing more training to database users led to more positive attitudes, which led to better understanding of the material [28]. These studies indicate the importance of positive attitudes in the context of training, but do not offer insights about adults' views toward programming specifically.

A large amount of literature is dedicated to improving elderly adults' attitudes toward computers and increasing their basic computer use, such as Internet navigation [6]. This research has reliably shown that exposure to and experience with computing tasks lead to improved computer attitudes. An intervention to learn how to use the computer and the Internet for one year resulted in elderly adults demonstrating increased participation in social activities and hobbies and an improvement in self-reported feelings of being more in control of their lives [30]. Similarly, Czaja and Sharit demonstrated that even brief exposure to a limited set of computer tasks resulted in more positive attitudes [8]. Jay and Willis found that elderly adults' attitudes toward computers are modifiable through direct computer experience emphasizing comfort and efficacy [14].

3. METHOD

Previous research has established the potential for adults to make positive changes in their attitudes and self-perceptions related to basic computer use. Fewer works have investigated if this can also apply to computer programming. The purpose of our study is to investigate how a concrete learning experience, in the form of an educational programming game, affects adults' attitudes of computer programming. Our hypotheses were:

1. Adults will initially have negative attitudes toward programming.
2. Adults' views on computer programming will positively change after a basic educational programming experience.
3. Adults' demographic characteristics of age, gender, population density, and level of education will influence their attitudes towards computer programming.

Table 1. Demographic Factors of the Sample.

Gender	***N***	***Percent***
Male	113	56.5%
Female	87	43.5%
Population Density		
Urban	28	14.0%
Suburban	103	51.5%
Rural	40	20.0%
Education		
No High School Degree	2	1.0%
High School Degree	25	12.5%
Some College	85	42.5%
College Degree or Higher	88	44.0%

3.1 Participants

We targeted self-reported non-programmers, defined as individuals who said that they had never written computer code. To recruit participants, we used Amazon's Mechanical Turk (MTurk), an online marketplace where individuals can receive micro-payments for doing small Human Intelligence Tasks (HITs). The task we posted explained that participants would be asked to play a game and then complete a survey, that they could quit at any time after the first level, and that they would receive $0.30 USD for starting the HIT, and an additional $0.10 for each level completed. We used an initial pricing and validation model from studies involving the same educational programming game [20,21], setting the base reward high enough to attract workers, and the bonus low enough so participants would not feel obligated to play the game.

A total of 200 respondents, including 113 men and 87 women ranging in age from 18-64 years participated in the study, as detailed in Table 1. The median self-reported age was 27.

3.2 Procedure

Although traditional survey measures are commonly used to understand views toward computing [19], they often ask about attitudes abstractly, or without exposing participants to actual programming. In our study, adults played an educational programming game called Gidget (described in detail in [20,21,22]), which asked participants to complete a series of debugging puzzles, finding and fixing errors in a virtual robot's programs in order to proceed. The game taught basic programming concepts including conditionals, variables, loops, and functions. Participants completed a median of 8 levels (*min* = 2, *max* = 37). Median playing time was 34 minutes, with a minimum of 6.88 minutes and a maximum of 142.29 minutes.

After the game, participants completed a short survey that included demographic questions and the five following free-response questions related to attitudes toward programming:

1. What motivated you to play the game?
2. What was the *most* motivating part of playing the game?
3. What was the *least* motivating part of playing the game?
4. Describe your attitude/view towards programming *before* playing the game.
5. Describe your attitude/view towards programming now, *after* playing the game.

Although it would have been valid to ask participants about their attitudes toward programming before they played the game, it was critical that the participants were not told that the game involved programming before they were asked to play. Thus, we attempted to mitigate the effects of negative stereotypes that influence the outcome of the experience, and the participants' decision to participate in the learning activity.

4. RESULTS

We provide both quantitative and qualitative evidence to support our hypotheses. We used the SPSS statistical package to analyze our data. Throughout the analyses, we used nonparametric tests as our data was ordinal and not normally distributed. Because we first coded the attitude responses on an ordinal scale, we used chi-square tests of independence to see potential relationships between nominal demographic factors (gender, population density, and levels of education) and ordinal categories of motivations and attitudes. To determine potential correlations between the continuous variable of age and attitudes labeled on ordinal scales, we used the Spearman R test. We then sorted participants' text-based responses into emerging themes, and used a multinomial regression model to observe the relationships between adults' motivations and attitudes with age.

4.1 Changes in Programming Attitudes

The overall programming attitudes score was computed by coding the sentiment of participants' text-based responses on an ordinal Likert scale from -2 to 2, where -2 = very negative ("scary," "intimidating, "difficult"), -1 = moderately negative ("don't care about it), 0 = neutral or no opinion, 1 = moderately positive ("interested in learning") and 2 = very positive attitude ("I love it"). This score was calculated so the results of this study could be discussed relative to other studies that used Likert scales.

In order to assess the validity of the scale, we selected a random, 12.5% subset of the data and independently coded participants' *before* and *after* attitudes across three researchers according. For attitudes toward programming before the game, researchers reached 81% agreement; for attitudes after the game, researchers reached 83% agreement. To account for some agreement by chance, researchers further analyzed their agreement via the Cohen Kappa statistic. A value above .60 is conventionally considered to be adequate [24]. For attitudes *before* the game, the obtained result of 0.89 indicates that inter-coder agreement was high. For attitudes *after* the game, obtained results of 0.64 indicate that inter-coder agreement for this variable was adequate.

In support of our first hypothesis, adults reported having an initial overall negative attitude toward programming (*Mdn* = -2.00, *min* = -2, *max* = 2). However, as shown in Figure 1, participants reported having significantly better attitudes toward programming after the game (X^2 (16, N = 197) = 83.542, $p < .001$).

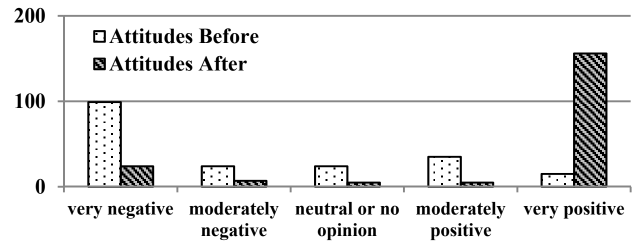

Figure 1. Number of participants exhibiting a particular sentiment toward programming before and after the game.

To better understand the reasons underlying these changes in attitudes, the first author categorized participants' *before* and *after* attitudes further into their emerging themes. As seen in Tables 2 and 3 (N = 188), most participants reported that prior to playing the game, they thought of programming as boring and unfamiliar, or something they wanted to learn but did not know how to approach. After playing the game, there was a distinct shift toward positive sentiments and a belief that learning to program was possible, even if difficult.

4.2 Categorized Attitudes and Demographics

To better understand how these improvements in attitudes might be affected by individual differences, we considered how demographic factors might influence participants' views. Chi-square tests of independence were performed to examine the relationships between all demographic factors and adults' attitudes. As shown in Table 4, differences in gender, population density, and level of education were generally not related, although adults from suburban neighborhoods (n = 35) were marginally more likely to report that the difficulty of programming concepts discouraged them from continuing to play the game, (X^2 (16, N = 196) = 25.57, p = 0.06). There was also a significant relationship between Level of Education and Ordinal Attitudes *after* the game: those with some college education (n = 66) and those with a bachelor's degree (n = 47) were significantly more likely to report positive perceptions of programming, (X^2 (24, N = 197) = 43.68, $p < .05$). However, these results may not be widely generalizable due to the fairly large number of tests, since at the 5% level it's likely that 1 out of 20 results will be significant by chance. A correlational analysis also revealed that as age increased, initial attitudes towards programming worsened, (r (200) = -0.181, $p < .05$).

4.3 Motivations and Demographics

To better understand participants' reasons for continuing to learn and their reasons for eventually quitting, the first author categorized all text-based responses from the three motivation questions in the survey. In response to the question of *"What motivated you to play the game?"*, participants reported that they thought the game might be *fun* (27.5%, n = 55), because they wanted to *learn* (25.5%, n = 51), out of *curiosity* (16.0%, n = 32), because they wanted a *challenge* (9.5%, n = 19), to play a *game* (8.5%, n = 17), due to *novelty* (5.0%, n = 10), *boredom* (3.0%, n = 6), and because they liked *solving puzzles* (1.5%, n = 3). A few participants played the game just for monetary compensation (2.5%, n = 5). As seen in Table 4, there was no relationship between these categories and any of the demographic variables.

Table 2. Positive and negative categories of attitudes toward programming *before* the learning experience.

Programming is…	Frequency	Example Quotation
too difficult to understand	**31%,** *n* = 62	*"Difficult to learn, hard to understand how it works."*
something I've wanted to learn	**16%,** *n* = 32	*"I've always wanted to learn how to program, I just haven't found a good resource that I can focus on to do so."*
boring	**12%,** *n* = 24	*"I always thought it was very complicated and somewhat boring."*
something I did not know about	**8.5%,** *n* = 17	*"Didn't know much about it. You occasionally see it in various places online and in popular culture. I was pretty neutral on it before...just didn't really understand all the intricacies of it."*
something I did not want to learn	**7%,** *n* = 14	*"I thought it was too complex to learn."*
a foreign concept	**5.5%,** *n* = 11	*"I think my view towards programming was very negative. I didn't think it was something that I could learn. It's like a foreign language to me."*
enjoyable	**5.5%,** *n* = 11	*"I thought programming might be enjoyable."*
important to know	**4%,** *n* = 8	*"It is a useful but boring tool that I'd like to know more about."*
an innate ability	**2.5%,** *n* = 5	*"I felt that it was something that was difficult for individuals who had a different mind than myself."*
too time consuming	**1%,** *n* = 2	*"I thought it was complicated and not something I had time to learn."*
nerdy	**1%,** *n* = 2	*"I thought the people who did it were super nerdy geniuses. And I thought that programming was probably extremely complicated."*

Table 3. Categories of self-reported attitudes toward computing *after* the learning experience.

Programming is…	Frequency	Example Quotation
something I can or want to learn	**32%,** *n* = 64	*"I'm definitely more interested in taking a course in programming or trying to learn more about it on my own."*
fun	**25%,** *n* = 50	*"I now know that programming can be fun and easy, also anyone can do it."*
easy to start	**20.5%,** *n* = 41	*"It's not as difficult to learn the basics as I thought."*
difficult	**11.5%,** *n* = 23	*"I think some programming might be manageable for me to do although the more complicated stuff I will be leaving to experts."*
an innate ability	**3%,** *n* = 6	*"I'll leave it to the pro's. I'll stick to my field and let them people good at this do their jobs."*
takes practice	**2%,** *n* = 4	*"The basics are fun and easy. Its still like a language I don't understand but at least now I can understand a little bit of what is going on in the code. I also feel like with practice I could grasp the more complicated concepts."*
something I don't want to learn	**1%,** *n* = 2	*"Something I could never learn."*

In response to the question *"What was the most motivating part of playing the game?"*, participants primarily mentioned *accomplishing goals/finishing levels* (35%, *n* = 70) and *learning* (23%, *n* = 46). Some were motivated by the *challenge* (9%, *n* = 18), *gameplay* (7%, *n* = 14), *puzzle solving* (6%, *n* = 12), *fun* (5%, *n* = 10) and *curiosity* (1%, *n* = 2). Some participants were only motivated by the monetary compensation (12.5%, *n* = 25). Quotes such as this one were representative of participants' responses:

> *"The most motivating part of the game was knowing that the levels got harder as the game went on and I felt like I was accomplishing a goal." (female, 29 years old, 19 levels completed)*

As seen in Table 4, these motivations did not vary by the demographic variables that we measured.

In response to the question *"What was the least motivating part of playing the game?"*, participants mentioned many experiences that give insight into why they quit learning, including that the programming itself was *too difficult* (29.9%, *n* = 59) and too *tedious* (16%, *n* = 32). Participants also did not like making *mistakes* (13.5%, *n* = 27), did not like the *gameplay* (11%, *n* = 22) and could not commit more *time* (10%, *n* = 20). Some had problems with *instructions* (8%, *n* = 16) and *system failure* (2%, *n* = 4). A respectable number of participants (8.5%, *n* = 17) did not find anything demotivating about the game. The following is a representative response for those who got discouraged by programming:

> *"I just got stuck…wanted to play more but knowing nothing about programming I guess I needed more guidance." (male, 30 years old, 9 levels completed)*

As with the other motivations, none of the demographic factors that we measured were statistically related to these categories of demotivating aspects of the learning experience.

4.4 Threats to Validity

Our study has several limitations that may restrict its generalizability. First, we solicited attitudes from adults who self-reported they did not have any prior programming experience, which may explain many participants' initial negative or apathetic attitudes toward programming. Next, rather than simply surveying participants, we gave them a specific learning experience in which

Table 4. Demographic Factors and Participants' Attitudes and Motivations.

	Gender				Population Density				Level of Education			
	df	*N*	X^2	*p*	*df*	*N*	X^2	*p*	*df*	*N*	X^2	*p*
Ordinal attitudes *after* the game	4	196	0.77	*n.s.*	8	196	5.33	*n.s.*	**24**	**197**	**43.68**	***< .05****
Ordinal attitudes *before* the game	8	196	0.99	*n.s.*	8	196	4.51	*n.s.*	24	197	17.08	*n.s.*
Categorized attitudes *before* the game	10	188	16.69	*n.s.*	20	187	22.20	*n.s.*	60	188	63.93	*n.s.*
Categorized attitudes *after* the game	6	196	0.98	*n.s.*	12	189	7.32	*n.s.*	36	197	29.31	*n.s.*
What motivated you to play the game?	8	198	11.23	*n.s.*	16	200	12.45	*n.s.*	48	198	36.61	*n.s.*
What was the *most* motivating part of playing the game?	7	197	4.75	*n.s.*	14	198	19.54	*n.s.*	42	197	39.88	*n.s.*
What was the *least* motivating part of playing the game?	8	200	9.04	*n.s.*	**16**	**188**	**25.57**	*n.s.*	48	197	31.24	*n.s.*

***Note: Starred p-value is significant, with 28 cells (80.0%) of expected count less than 5.** *"n.s" p-value stands for "not significant."*

to ground their attitudes before asking them about their initial attitudes toward programming. Participants may have misremembered their attitudes prior to playing, or at least reported them differently than if asked beforehand.

Additionally, MTurk allows participants to self-select into HIT's, given that they meet certain qualifications. Our HIT only required that participants were living in the USA and had no programming experience. Additionally, filtering HITs for certain payouts could have affected participant recruitment. These limitations introduce a sampling bias, which may limit the generalizability of our results to the particular population found on MTurk. MTurk workers generally make up a diverse group, including a range of ages, education levels, and socio-economic status, although the workers are primarily from highly industrialized countries [13].

Our results are also grounded in the reactions of participants to Gidget, which is a particular learning technology designed for explicit instruction via the web [20,21,22]. Had participants been asked to learn by using other popular environments, such as Scratch or Alice, the changes in attitudes might have been different in magnitude and direction.

5. DISCUSSION AND CONCLUSION

This study investigated adults' attitudes and opinions toward computer programming and how demographic characteristics might affect these opinions. In line with literature on programming stereotypes among youth [1], we found that adults initially had poor attitudes toward programming, believing that it was difficult, boring, and something they generally could not learn. However, we found that even a brief, concrete learning experience can have a significant positive impact on many adults and their attitudes towards programming. After playing the game, adults viewed programming as fun, easier to begin learning than they had initially thought, and something they could potentially learn. These effects did not significantly vary by gender, population density, or level of education, although college-educated adults reported being more likely to want to continue learning. These results complement a growing literature about the effect of programming experiences such as camps on youth attitudes toward programming [32]; adults' attitudes can also be improved even with a brief exposure to programming.

These results, and the body of literature on which they build, have important implications for future work in online learning resources for computing education. For example, because we used an educational game, our results suggest the need for a much wider variety of learning experiences for adults that do not assume an existing motivation to learn and instead embed programming in compelling learning experiences that create and sustain motivation to learn. If the motivation to learn can be sustained, adults' positive attitudes toward programming may also influence their children's motivation to learn programming. Our results also show that with careful design, these learning opportunities can also be gender and age inclusive. Websites like Codecademy, Coursera, Kahn Academy, and Udacity, which currently market to those already interested in learning programming, might reach a much broader audience if their learning materials were designed to not just teach, but also to engage. Our findings also suggest that more traditional education settings, job training programs, and even ad hoc efforts to learn programming at work may just need a careful and controlled introduction to programming to have a positive, reinforcing learning experience

These results, of course, also raise many questions for future research. For example, what are the essential characteristics of learning technologies for computer programming that engender positive changes in attitudes? How sustainable are positive changes in adult attitudes through longer learning experiences? What else is necessary beyond positive attitudes to compel adults to engage in further learning? Do the positive changes in attitudes we observed occur in more formal learning settings, where learning might be compulsory? What else is necessary beyond positive attitudes to compel adults to engage in further learning? How can we motivate students or adults to learn programming languages without fearing them? With the rising interest in learning to code and the proliferation of resources to do so, answers to these questions will be a key part of developing effective pedagogy and learning technologies for making programming accessible to the growing millions who will program as part of their everyday lives at work and at home.

6. ACKNOWLEDGEMENTS

This material is based upon work supported by Microsoft and the National Science Foundation (NSF) under Grants CNS 1240786, CCF 0952733, CCF 1339131, and OISE 1210205. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

7. REFERENCES

[1] Anderson, N., Lankshear, C., Timms, C., & Coutney, L. (2008). 'Because it's boring, irrelevant and I don't like computers': Why high school girls avoid professionally oriented ICT subjects. *Computers and Education,* 50, 1304–1318.

[2] Barrett, R., Kandogan, E., Maglio, P.P., Haber, E.M., Takayama, L.A., & Prabaker, M. (2004). Field studies of computer system administrators: Analysis of system management tools and practices. *ACM CSCW,* 388–395.

[3] Beckwith, L., Inman, D., Rector, K., & Burnett, M. (2007). On to the real world: Gender and self-efficacy in Excel. *IEEE VL/HCC.* 119–126.

[4] Burnett, M. (2009). What is end-user software engineering and why does it matter? In *End-User Development*, 15-28. Springer Berlin Heidelberg.

[5] Carver, J., Kendall, R., Squires, S. & Post, D. (2007). Software engineering environments for scientific and engineering software: a series of case studies. *ACM/IEEE ICSE,* 550–559.

[6] Chu, A., Huber, J., Mastel-Smith, B., Cesario, S. (2009). Partnering with seniors for better health: Computer use and internet health information retrieval among older adults in a low socioeconomic community. *J. of the Medical Library Association, 97,*12–20.

[7] Creamer, E. G., Lee, S., & Meszaros, P.S. (2006). Factors associated with women's interest in computing fields. *Association for the Study of Engineering Education*.

[8] Czaja, S.J., & Lee, C.C. (2001). The Internet and older adults: Designs challenges and opportunities. In N. Charness & D. C. Parks (Eds.), *Communication, and aging: Opportunities and challenges for the future*, 60–78.

[9] Downey, G.L., Lucena, J. C., Moskal, B. M., Parkhurst, R., Bigley, T., Hays, C., Nichols-Belo, A. (2006). The globally competent engineer: Working effectively with people who define problems differently. *J. of Engineering Education, 95*(2), 107-122.

[10] Gupta, S., Bostrom, R.P., & Anson, R. (2010). Do I matter? The impact of individual differences on training process. *ACM SIGCPR*, 112-120.

[11] Hewner, M., & Guzdial, M. (2008). Attitudes about computing in postsecondary graduates. *ACM ICER*, 71-78.

[12] Huang, H. M. & Liaw, S.S. (2005). Exploring user's attitudes and intentions toward the web as a survey tool. *Computers in Human Behavior*, 21(5), 729-743.

[13] Ipeirotis, P. (2010). Demographics of Mechanical Turk, *CeDER-10–01 working paper*, New York University.

[14] Jay, G. M., Willis, S. L., Gerontol, J.B. (1992). Influence of direct computer experience on older adults' attitudes toward computers. *The J. of Gerontology Series B: Psychological Sciences and Social Sciences, 47,* 250-7.

[15] Jenkins, T. (2002). On the difficulty of learning to program. *Conf. of the LTSN Centre for Information and Computer Sciences*, *4*, 53-58.

[16] Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM CSUR, 37*(2), 83-137.

[17] Kelleher, C., Pausch, R., and Kiesler, S. (2007). Storytelling alice motivates middle school girls to learn computer programming. *ACM CHI,* 1455-1464.

[18] Ko, A.J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., et al. (2011). The state of the art in end-user software engineering. *ACM CSUR, 43*(3), 21.

[19] Kölling, M., & Utting, I. (2012). Building an open, large-scale research data repository of initial programming student behaviour. *ACM SIGCSE*, 323-324.

[20] Lee, M.J., & Ko, A.J. (2012). Investigating the role of purposeful goals on novices' engagement in a programming game. *IEEE VL/HCC*, 163-166

[21] Lee, M.J., & Ko, A.J. (2011). Personifying programming tool feedback improves novice programmers' learning. *ACM ICER*, 109-116.

[22] Lee, M. J., Ko, A. J., & Kwan, I. (2013). In-game assessments increase novice programmers' engagement and level completion speed. *ACM ICER*, 153-160.

[23] Margolis, J., & Fisher, A. (2002). *Unlocking the clubhouse: Women in computing.* Cambridge, MA: *MIT Press*.

[24] McHugh, M. L. (2012). Interrater reliability: The kappa statistic. *BiochemiaMedica, 22*(3), 276-282.

[25] Petre, M., & Blackwell, A.F. (2007). Children as unwitting end-user programmers. *VL/HCC,* 239-242.

[26] Scaffidi, C., Brandt, J., Burnett, M., Dove, A., & Myers, B. (2012). SIG: end-user programming. *ACM CHI, Extended Abstracts,* 1193-1996.

[27] Shayo, C., & Olfman, L. (2000). The role of training in preparing end users to learn related software. *J. of Organizational and End User Computing*, 12(1), 3-13.

[28] Slegers, K., van Boxtel, M.P., & Jolles, J. (2008). Effects of computer training and Internet usage on the well-being and quality of life of older adults: A randomized, controlled study. *The J. of Gerontology Series B: Psychological Sciences and Social* Sciences, 63(3), 176-184.

[29] Sung, K., & Shirley, P. (2004). Algorithm analysis for returning adult students. *J. of Computing Sciences in Colleges*, 20(2), 62-69.

[30] Szajna, B., & Mackay, J. M. (1995). Predictors of learning performance in a computer user training environment: A path analytic study. *Int'l. J. of Human Computer Interaction*, 7(2), 167-185.

[31] U.S. Department of Labor, Bureau of Labor Statistics. (2012). Occupational outlook handbook, 2012-13: Accountants and auditors. *Retrieved from http://www.bls.gov/cps/lfcharacteristics.htm#emp*

[32] Utting, I., Cooper, S., Kölling, M., Maloney, J., & Resnick, M. (2010). Alice, greenfoot, and scratch – a discussion. *ACM TOCE, 10*(4), 17.

[33] Wisniewski, J. (2012). Parlez-Vous Code? *Online*, 36(6), 57-60.

[34] Wong, J., & Hong, J. (2007). Making mashups with Marmite: Towards end-user programming for the web. *ACM* CHI, 1435-144