Animated Hints Help Novices Complete More Levels in an Educational Programming Game*

Michael J. Lee and Joseph Chiou
Department of Informatics
New Jersey Institute of Technology
Newark, NJ 07102
{mjlee,jcc45}@njit.edu

Abstract

Many people are learning programming on their own using various online resources. Unfortunately, learners using these resources often become disengaged or even quit when encountering an obstacle they cannot overcome without additional help. Teachers in a classroom can provide this type of help, but this may be impractical or impossible to implement in online educational settings. To address this issue, we added a visually-oriented hint system into an existing online educational game designed to teach novices introductory programming concepts. We implemented three versions of the hint system, providing equivalent information for each level of the game, adjusting the amount of interactivity between versions. The first version consisted of a static image with text showing how to solve a level in a single panel. The second version included a series of images that allowing users to scroll through hints step-by-step. The final version showed a short video allowing users to play, pause, and seek through animated hint(s). In total, we had 150 people play the game, randomly assigned to one of these three versions of the hint system. We found that users had a strong preference for the video version of the hint system, completing more levels. Based on these findings, we propose suggestions for designers of online educational tools to better support their users.

1 Introduction

In recent years, there has been an increasing interest in enhancing computing literacy. Rising availability of online learning resources, such as tutorial sites (e.g., Codecademy.org, Kahn Academy), block programming environments (e.g., Scratch), and educational games (e.g., Swift Playgrounds), are popular choices for people to acquire programming knowledge [21]. However, despite these numerous online educational resources for learners to choose from, there continues to be high dropout/attrition rates. Researchers and educators attribute this to a lack of support [17, 18, 21], user frustration, a lack of motivation to continue studying, and no direct interaction with an instructor [23].

One potential way to address high dropout rates is to provide users with additional help, or hints [22]. Textual hints are often used to provide important information to users [5, 13]. Studies have shown that textual hints can positively influence users' in-system behavior and time spent on tasks [5, 27]. In addition to textual hints, many non-educationally focused systems, such as games, use visually-oriented hints to help their users overcome obstacles [2]. To the best of our knowledge, we have not found any research examining the effectiveness of visually-oriented hints in educational programming games.

To address the challenges of high dropout/attrition rates in online programming resources, we explore the use of interactive graphics as an alternative form to present hints. In this study, we examine how three different visualized hints—a static image, a carousel (series) of images, and a video clip—affects user retention in an educational game. Based on the success of certain types of text-based [5, 13] and visually-oriented hints [2] we hypothesized that a hybrid approach—the carousel hints—would provide a good combination of interactivity, text, and visual aids to assist users overcome obstacles and ultimately complete more levels than the other two conditions.

2 Related Work

Engagement, and how it affects learning, has been widely studied in educational contexts. Student engagement has been shown to be key for student success at all grade levels [6, 9, 19]. As students become more engaged in learning, they improve learning outcomes and academic achievements [14, 19]. Engagement is essential for learning challenging topics such as computer programming [4], and educational games for teaching introductory programming concepts have shown to be successful at attracting a wide range of learners [8, 16, 22]. However, even with the success of these resources, users of these systems who encounter difficulties and not do not receive the support they need to overcome difficulties may become frustrated and quit the activity/topic.

To address these types of obstacles and frustrations, teachers often provide personalized, directed feedback to their students in classrooms. Online learning contexts—where teachers are unavailable or at a premium—such as Massive Open Online Courses (MOOCs) and tutorial websites often utilize help and/or hint systems of varying sophistication to help their users. Research shows that students perform better in learning environments when hints are provided [3]. Moreover, studies have found that the content of hints may have different effects, where high level hints tend to lead to long term positive effects, and detailed hints tend to be more useful immediately [5, 22].

The visual and interactive aspects of hints may also be important factors to consider when evaluating the usefulness of hints for online learners to overcome obstacles. According to Presmeg, visualization aids one in understanding a problem or a concept in a different modality and perspective, enabling them to better seek solutions [24]. Similarly, Gangwer suggests that students combine visualizations with active learning strategies to develop better mental models of problems and actively work on different approaches to solve them [7]. Finally, using visualization/graphics has shown to promote student learning and create opportunities for them to apply what is taught [12, 15, 20]. However, there is little consensus of how interactive graphics (e.g., static vs. animated images) compare in their utility for helping learners overcome obstacles [1, 11, 25, 26], especially in different online learning environments. In this study, we aim to explore this space, specifically examining how different types of hint visualizations, ranging from static images to animations, may affect learners' motivation to continue playing an educational programming game.

3 Method

The goal of our study was to determine how different types of visual hints in an educational computing game affects engagement and task completion rates in self-directed learners, and to identify the extent of these effects. To do this, we modified *Gidget* (see Figure 1-A; www.helpgidget.org), a freely available online game, adding new types of hints. The game has a total of 37 levels, where each level teaches a new programming concept (e.g., variable assignment, conditionals, loops, functions, objects) using a Python-like language [16, 18]. The goal of each level is to debug existing code to pass 1-4 test cases (i.e., statements that evaluate to 'true') upon running the code. After code execution, the game displays which test cases were successful and which ones failed. The game already includes a set of help features to assist players overcome obstacles while coding on their own [17]. These include popup bubbles explaining different code components, a dictionary explaining keywords [17], and a context-aware, implementation of the Idea Garden [13] to assist with programming anti-patterns.

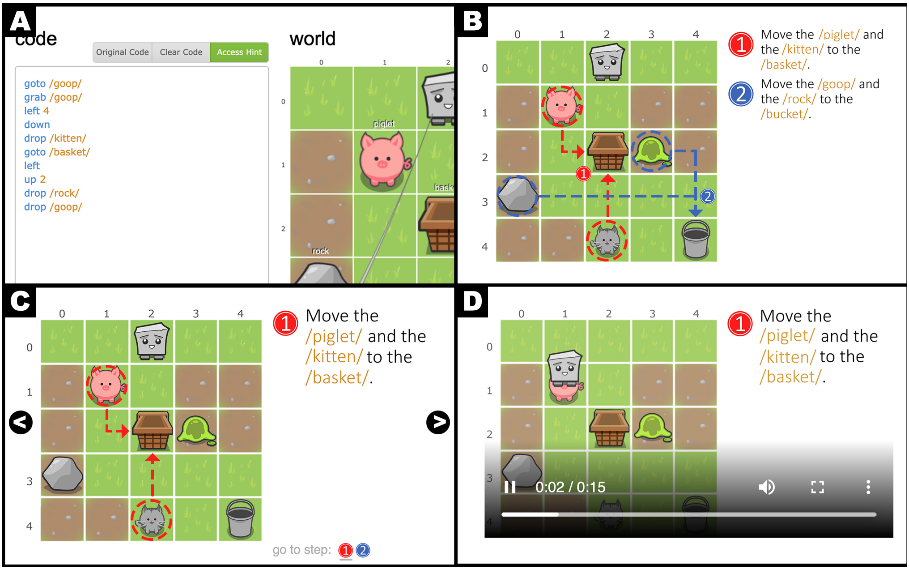

Figure 1: *A: the "Access Hint" button, displayed above the main code pane in green; B: the static image hint condition; C: the carousel hint condition; and D: the video hint condition.*

Users could access the new hint system by clicking on a button labeled *Access Hint*, which was prominently displayed above the game's coding pane (see Figure 1-A). This button opened one of three different kinds of visual hints, which served as the independent variable we manipulated in our experiment: (1) a ***static*** image with the entire hint for the level displayed in one panel (see Figure 1-B); (2) a ***carousel***, or sequence of images showing one step of a hint at a time, allowing the player to scroll left or right through each hint panel (see Figure 1-C), or (3) an ***animation*** hint in the form of a short, 5-50 second video clip showing one step of a hint at a time, with controls allowing users to pause, play, and scroll through the clip (see Figure 1-D). We created customized hints for each of the 37 levels in the game, ensuring that each level's three visual hint systems conveyed equivalent information so that we could make a fair comparison among them. We used a "divide and conquer" approach to breaking down each level's hints into 1-5 smaller tasks (depending on the complexity of the level) which were organized into a numbered list on the right side of the hint, along with a graphical representation of the state of the system on the left. We also did not want players to exploit the hint system to get the exact answer(s) to complete the level [13], so we did not provide actual code. Instead, the hint system presented one possible path and actions that the character could take through the level to complete it successfully.

3.1 Participants

We recruited our participants on Mechanical Turk (MTurk), specifically sampling adults who self-reported that they had no experience with programming—those who responded "never" to all of the following statements: 1) "taken a programming course," 2) "written a computer program," and 3) "contributed code towards the development of a computer program." We also required participants to be U.S. residents to minimize English language barriers with the instructions and activities. We followed our previous work's pricing model from a similarly scoped study [16], adjusting the payment to US$5 to better reflect the task difficulty and other similar HITs and prices on MTurk at the time. To help participants make an informed decision about the time commitment required to participate in our study, we told them that they would be playing a puzzle game for as long (or as short) as they wanted, over a maximum of seven days so they could have flexibility in their play time(s). Our HIT was labeled as "5 hours" for the task time, but emphasized that this was an estimate, and that they could quit the task at any time without negative repercussions.

Once an MTurker accepted the HIT, they were required to fill out the form certifying they were a novice programmer, and to read and digitally sign the informed consent form agreeing to participate in the experiment. Once they did so, they were redirected to the game website to make an account (requiring an e-mail address, password, gender, and age). Each participant was randomly assigned to one of the hint conditions, and this information was saved so that they would always see their assigned type of hint whenever they played the game. The introductory tutorial for the game (shown automatically the first time someone logs in) highlighted the *Access Hint* button and included text encouraging players to use it when they needed help. For the purposes of this study, we logged the total number of levels the participant completed, how many times they pressed the *Access Hint* button per level, and the cumulative time they had the hint window open per level.

4 Results

We provide quantitative results comparing the outcomes from our three groups using nonparametric Chi-Squared and Wilcoxon rank sums tests with $= 0.05$ confidence, as our data were not normally distributed. For post-hoc analyses, we use the Bonferroni correction for three comparisons: ($/3 = 0.01\overline{6}$).

Our study was a between-subjects design, with an even split of 50 people each among the three conditions. Demographic data revealed that there were no significant differences between groups by age (range 18-54 years old; median 22) or gender (88 females and 62 males). The key dependent variable in our study was engagement, which we operationalized as the number of levels completed. We also examine the participants' use of the hint system (number of times accessed per level and total time open per level).

4.1 Animation Condition Participants Complete More Levels

All participants completed at least seven levels. The range of levels completed in the static, carousel, and animation conditions were 7-33 (median 10), 7-37 (median 13), and 7-37 (median 13), respectively. There was a significant difference in the number of levels participants completed between the three conditions ($\chi^2(2, N = 150) = 7.0276, p < .05$). Further post-hoc analysis with a Bonferroni correction shows that the significantly different pair was the static vs. animation conditions ($W = 14.64, Z = 2.541, p < .01\overline{6}$), with the animation group completing more levels. The static vs. carousel ($W = 11.52.5, Z = 1.996, p < .05$) comparison trended towards significance (p-value was less than $.05$, but not less than the correction threshold of $0.01\overline{6}$) with the carousel group completing more levels. Finally, comparing the carousel vs. animation conditions showed no significant difference ($W = 1.58, Z = 0.274, n.s.$).

Since all participants were novice programmers with no statistical difference in demographics, these results suggest that something about interacting with the animated hints (and to a lesser degree, the carousel hints), had a significant positive effect on participants' engagement and ability to complete more levels in the game compared to the other condition(s). This was surprising, as we had hypothesized that the carousel hints would help learners the most because it would allow quick, directed access to different parts of the hints (instead of having to look through a long list of hints, or seek through a video).

4.2 No Significant Differences in Accessing the Hint System

All participants used their respective hint system at some point during their gameplay. The range of access to the hint system in the static, carousel, and animation conditions per level were 0-11, 0-11, and 0-10, respectively. Because everyone completed a different number of levels, we calculated the average number of times each participant accessed the hint system per level (sum of number of times they pressed the hint button throughout their entire gameplay record, divided by the farthest level number they reached) for our analysis. There was no significant difference in the number of times participants accessed the hint system among the three conditions ($\chi^2(2, N = 150) = 0.036, n.s.$).

We originally expected to see a different number of hint access across the conditions because the information from each was conveyed so differently. For example, we thought that the static image would be accessed the least, since it showed the entirety of a hint in one image, and that the users of the carousel and animated hints would have to jump back and forth between their code and hints more often since the hints were broken down into smaller sections. However, this was not the case, with participants accessing their respective hints a similar number of times on average per level. Combined with the previous result, this suggests that animation participants made better use of their time when accessing a hint, as they clicked on their hint button a similar number of times as their counterparts, but ended up completing more levels.

4.3 Static Hint Participants Spend Less Time Looking at Hints

To further examine our last result, we examined how long participants looked at their respective hints (i.e., time the hint window was open). The range of looking at the hint system in the static, carousel, and animation conditions were 0-87 seconds (median 46), 0-81 seconds (median 62), and 0-96 seconds (median 65), respectively. Because everyone completed a different number of levels, we calculated the average time each participant spent looking at a hint per level (sum of the cumulative time they looked at a hint within each level throughout their gameplay record, divided by the number of the farthest level they reached) for our analysis. There was a significant difference in the time participants looked at their respective hints between the three conditions ($^2(2, N = 150) = 8.335, p < .05$). Further post-hoc analysis with a Bonferroni correction shows that the significantly different pair was the static vs. carousel conditions ($W = 14.28, Z = 2.462, p < .01\bar{6}$) and the static vs. animation conditions ($W = 14.40, Z = 2.482, p < .01\bar{6}$) with the static group spending less time on the hints. Comparing the carousel vs. animation conditions showed no statistically significant difference between the two ($W = 3.140, Z = 0.541, n.s.$).

Similarly to the reasoning we described above, we expected (and found) that participants spent the least amount of time looking at the static hints (since everything was displayed in one image), and more time looking at the videos (since each video hint had a specific pace and run time), with the carousel being somewhere in the middle (since the user could skip to specific, labeled parts of the hints on demand). Combining the last two results with this shows that animated hint users (and to a certain degree, the carousel hint users) spent a little more time looking at the same number of hints, but were more successful in completing levels than the other condition(s), suggesting something about the animated hints helped users better apply the information to complete levels.

5 Discussion & Conclusion

Our findings show that animated hints (and to a certain degree, carousel hints) can significantly improve users' performance in an educational game. The animated hint condition group participants completed significantly more levels than their static hint counterparts, while looking at their respective hints a similar number of times. Our results have several potential interpretations for better understanding hints in the context of educational games.

We tried to keep the information content of the three conditions equivalent—mainly manipulating the visual density and interactivity of each condition—but found significant differences in outcomes. A possible interpretation of our results is that showing users different visual states of a program (i.e., a screenshot of beginning state, some middle states, and an end state) can help users better understand the goals of a level. Both the animation and carousel hints showed the users what their program should look like in at least three different stages. On the other hand, the static hints presented everything in one picture, which may not have significant impact on helping understand the goal of a level and cause information overload. Information overload can be a factor that negatively affect users' information acquiring process [10]. To further this case, we observed that our static hint users spent less time with their hint windows open compared to the other two conditions, even though users from all three conditions opened their respective hint windows the same number of times. This may mean that users were able to use the time they had their hint window open more efficiently when the hints were broken down into smaller, more digestible chunks. Our results show that hints in educational games can be represented and interacted with in different ways, and that small changes can have a significant impact on user engagement.

We have several limitations to our study. First, we recruited participants on MTurk, which might not be representative of the larger population. However, our groups were similar to each other, with no significant differences by age or gender. Second, we provided an economic incentive for people to participate in our study, which may have affected their engagement and sense of obligation to complete levels. To counteract this, our payment was low compared to the estimated time to complete the task, and allowed participants to quit at any time. Despite this, we found that participants were engaged with the task, spending hours playing the game and everyone completing a minimum of 7 levels, suggesting that they were entertained and not playing the game for the monetary compensation. Third, all participants completed a different number of levels, making it difficult for us to get a consistent count of overall times people accessed and looked at hints across the same levels. In our analyses, we calculated the average count and time each participant took through their play of the game, which may have introduced some level of inaccuracy. However, this was intentional as we did not want to force all our players to complete the entire game, which might be difficult and/or unreasonable for some participants, and also because our main goal was to measure engagement as a function of how many levels users in each condition completed. Future studies may ask participants to complete all levels and/or collect qualitative measures from users, asking them how they felt about the hints they used. Finally, we plan to measure learning outcomes (using pre-post tests) to determine users' knowledge before and after playing the game using these different hint systems.

In conclusion, our study examined how different visualized hints affect users' engagement with an online educational game. We found that participants using animated hints—video clips that allowed users to pause, play, and seek through numbered hints—were more engaged with the game, completing more levels. Our findings suggest that interactive, visual hints that are subdivided into smaller parts showing different states of a program during execution assist learners in understand programming task goals. Researchers, educators, and designers of these online learning systems may benefit by utilizing these types of hints. Future work will explore these findings further, especially with different types of online resources to explore potential differences and similarities.

6 Acknowledgements

This work was supported in part by the National Science Foundation (NSF) under grants DRL-1837489 and IIS-1657160. Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the views of the NSF or other parties.

References

[1] Erik Andersen, Yun-En Liu, Rich Snider, Roy Szeto, and Zoran Popović. Placing a value on aesthetics in online casual games. In *ACM CHI*, 2011.

[2] Erik Andersen, Eleanor O'Rourke, Yun-En Liu, Rich Snider, et al. The impact of tutorials on games of varying complexity. In *ACM CHI*. ACM, 2012.

[3] John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, 4(2), 1995.

[4] Lori Carter. Why students with an apparent aptitude for computer science don't choose to major in computer science. In *ACM SIGCSE Bulletin*, volume 38, 2006.

[5] Christa Cody, Behrooz Mostafavi, and Tiffany Barnes. Investigation of the influence of hint type on problem solving behavior in a logic proof tutor. In *International Conference on AI in Education*, pages 58–62. Springer, 2018.

[6] Lyn Corno and Ellen B Mandinach. What we have learned about student engagement in the past twenty years. *Big theories revisited*, 4:299–328, 2004.

[7] Timothy Gangwer. *Visual Impact, Visual Teaching: Using Images to Strengthen Learning*. Simon and Schuster, 2015.

[8] Nan Gao, Tao Xie, and Geping Liu. A learning engagement model of educational games based on virtual reality. In *IEEE ICIME*, pages 1–5, 2018.

[9] Rosemary Garris, Robert Ahlers, and James E Driskell. Games, motivation, & learning: A research & practice model. *Simulation & Gaming*, 33(4), 2002.

[10] Kyle J Harms. Applying cognitive load theory to generate effective programming tutorials. In *IEEE VL/HCC*, pages 179–180, 2013.

[11] Tim N Höffler and Detlev Leutner. Instructional animation versus static pictures: A meta-analysis. *Learning and Instruction*, 17(6):722–738, 2007.

[12] Jozef Janitor, František Jakab, and Karol Kniewald. Visual learning tools for teaching/learning computer networks: Cisco networking academy and packet tracer. In *IEEE ICNS*, pages 351–355, 2010.

[13] Will Jernigan, Amber Horvath, Michael Lee, Margaret Burnett, et al. A principled evaluation for a principled idea garden. In *IEEE VL/HCC*, 2015.

[14] Greg Kearsley and Ben Shneiderman. Engagement theory: Framework for technology-based teaching and learning. *Educational Technology*, 38(5), 1998.

[15] Daesang Kim and David A Gilman. Effects of text, audio, and graphic aids in multimedia instruction for vocabulary learning. *Journal of Educational Technology & Society*, 11(3):114–126, 2008.

[16] Michael J Lee. Teaching and engaging with debugging puzzles. *University of Washington, Seattle, WA*, 2015.

[17] Michael J Lee, Faezeh Bahmani, Irwin Kwan, et al. Principles of a debugging-first puzzle game for computing education. In *IEEE VL/HCC*, 2014.

[18] Michael J Lee, Amy J Ko, and Irwin Kwan. In-game assessments increase novice programmers' engagement and level completion speed. In *ACM ICER*, 2013.

[19] Helen M Marks. Student engagement in instructional activity: Patterns in the elementary, middle, and high school years. *American educational research journal*, 37(1):153–184, 2000.

[20] Richard E Mayer and Roxana Moreno. Aids to computer-based multimedia learning. *Learning and Instruction*, 12(1):107–119, 2002.

[21] Daniel Fo Onah, Jane Sinclair, and Russell Boyatt. Dropout rates of massive open online courses: behavioural patterns. *EDULEARN*, 1:5825–5834, 2014.

[22] Eleanor O'Rourke, Christy Ballweber, and Zoran Popovií. Hint systems may negatively impact performance in educational games. In *ACM LS*, 2014.

[23] Angie Parker. Identifying predictors of academic persistence in distance education. *Usdla Journal*, 17(1):55–62, 2003.

[24] Norma C Presmeg. Prototypes, metaphors, metonymies and imaginative rationality in high school mathematics. *Educational Studies in Mathematics*, 23(6):595–610, 1992.

[25] Fanny Ståhl and Hanna Holmgren. How does animated and static graphics affect the user experience in a game?, 2016.

[26] Barbara Tversky, Julie Bauer Morrison, and Mireille Betrancourt. Animation: Can it facilitate? *International J. of Human-Computer Studies*, 57(4), 2002.

[27] Kurt Vanlehn. The behavior of tutoring systems. *International Journal of AI in Education*, 16(3):227–265, 2006.