Investigating the Role of Purposeful Goals on Novices' Engagement in a Programming Game

Michael J. Lee and Andrew J. Ko
The Information School | DUB Group
University of Washington
Seattle, Washington, USA
{mjslee, ajko}@uw.edu

***Abstract* — Engagement is a necessary condition for learning, and previous studies have shown that engagement can be significantly affected by changing the presentation of game elements within an educational game. In a three condition controlled experiment, we examined how changing the presentation of the data elements referred to in a game's goals would influence the purposefulness of the goals and thereby affect players' motivation to achieve them. A total of 121 self-described programming novices were recruited online to play the game. We found that 1) those using vertebrate elements completed twice the number of levels compared to those using inanimate elements, 2) those using vertebrate and invertebrate elements spent significantly more time playing the game overall compared to those using inanimate elements, and 3) those using inanimate elements were more likely to quit the game, especially on difficult levels. These findings suggest that the presentation of game elements that influence the purposefulness of goals can play a significant role in keeping self-guided learners engaged in learning tasks.**

Keywords – Programming, education, engagement, games

I. Introduction

Engagement is a necessary condition for learning [8], especially for challenging topics such as computer programming [4]. Such engagement may only occur, however, when objectives are meaningful (i.e. have a purpose) to the learner. Although these effects have been examined in formal educational settings [11,14], much less is known about their effects in informal contexts, especially in the space of educational games. For example, dropouts in CS1 courses are often attributed to students feeling that their programs did not solve meaningful problems [14] or were lacking any practical context (e.g. sorting a list of meaningless numbers) [11]. In our prior work, we investigated this effect in the our debugging game, Gidget, and found that players who worked with a robot that used personal pronouns and had a face were significantly more likely to report wanting to help it and completed twice as many levels in a similar amount of time as the other group [12].

Whereas our previous study investigated the effect of the visual presentation of the program interpreter, in this study we investigate the effect of game goals, manipulated by the presentation of data elements. Recent work has demonstrated that humans have evolved to empathize with animals [3], suggesting that players may attribute more purpose in the goals working with animate data objects, particularly vertebrates [1]. In Gidget programs, data are the objects that the robot scans, analyzes, and moves, such as those in Table 1, and these objects are directly tied to the goals that the player is trying to accomplish. Goals in the game include transferring spilled chemicals into containers, checking attributes of objects, and moving animals to safety. We hypothesized that changing the presentation of the data referred to in these goals would influence the purposefulness of goals, thereby affecting players' motivation to achieve them, especially as goals become increasingly difficult to accomplish.

TABLE I. Conditions, Goals, and Images of the First Level

Condition	Goal (Level 1)	Respective Game Images
Inanimate	*block on bin*	
Invertebrate	*beetle on jar*	
Vertebrate	*kitten on basket*	

Our experiment, which involved 121 rank novice programmers, asked participants to play the game until they wished to quit, enabling us to measure engagement as play time and the number of levels completed. To manipulate the purposefulness of the goals, we designed three versions of the game involving the three different kinds of objects shown in Table 1: inanimate, invertebrate, and vertebrate.

II. Related Work

Educators use engagement to improve learning. According to engagement theory, engaged students learn at high levels, better grasp what they learn, and retain that knowledge [9]. Experts agree that increasing student engagement in educational topics is key to success [5]. Since engagement in learning activities is connected with tasks perceived to be meaningful [9], it is closely related to motivation.

Several studies demonstrate that working towards *meaningful goals* positively affects engagement in gaming and learning contexts. Bowman encourages learners to play an active role in their engagement by having them pursue goals they find personally meaningful [2]. Similarly, Malone argues that a key element for creating enjoyable educational games is to provide clear goals that students find meaningful [13]. Meaningful goals have been found to be particularly important for women, minorities, and millennials (those born after 1982, including men) [7,14,15]. However, Layman et al. find that current CS1 courses lack meaningful projects [11]. Our work extends these studies in formal learning settings, exploring how the representation of particular game elements affects engagement in achieving goals in an informal learning context.

Other works support the constructionist approach to learning, promoting children's engagement by having them work on relatable and personally meaningful projects [16]. Kelleher et al. [10] were one of the first to demonstrate that opportunities and affordances for storytelling can significantly improve learners' motivation to program by making projects more personally relevant. Our work follows these traditions, but provides learners with the story, allowing them to contribute to its progress by helping one of the characters.

III. Method

We aimed to investigate how the purposefulness of goals, manipulated by the visual representation of data elements and their labeling, affects learners' voluntary engagement. Our study had three conditions involving inanimate, invertebrate, and vertebrate data elements, within a game we designed called Gidget. We used a between-subjects design with 41 participants in the vertebrate condition, and 40 each in the inanimate and invertebrate conditions. The key dependent variable in our study was engagement, which we operationalized as the number of levels completed, the time spent on each level, and the use of different UI elements.

A. The Game

Our game, called Gidget (Figure 1), is a web-based, HTML5 application. Learners are guided through a sequence of levels that teach the design and analysis of basic algorithms in a simple imperative language designed specifically for the game (further details about the game can be found in [12]). A simple story motivates the game: a small robot capable of identifying and solving problems with programs has been deployed to clean up the area and shut down a factory that has gone awry. Unfortunately, the robot was damaged, and now struggles to complete its missions, generating programs that almost accomplish its missions, but not quite. The learner must help the robot by fixing its problematic code. In this sense, the learner and the robot are a team, working together to complete levels and ultimately shut down the hazardous factory.

In the game, players learn how to communicate with the robot via commands to help it accomplish a series of goals. The levels, goals, language, and user interface (UI), however, were designed to teach specific aspects of algorithm design. Levels contain syntax and/or semantic errors that learners must understand and correct by inspecting the program, executing it, and optionally reading Gidget's explanations of his actions at each step in the code. Each level includes one or more goals, which are executable expressions that must all be true after program execution. Each goal is on a single line predicate, with corresponding references to the data elements in the world.

The robot was given facial expressions (neutral, happy, and sad) shown upon error states and goal completions. It referred to itself and the player using personal pronouns in its feedback such as *"I don't know what this is..."* and *"I never could have done it without you!"* This was based on research showing that personified feedback by an interpreter with agency significantly increased learners' engagement [12].

To aid the players with debugging, the game includes four execution controls for the code: *one step*, *one line*, *all steps*, and *to end*. The *one step* button evaluates one compiled instruction in the code, as a breakpoint debugger does, but also displays text describing the execution of the step. The *one line* button evaluates all steps contained on one line of the code, jumping to the final output of that line immediately. The *all steps* button evaluates the entire program and the goals in one button press, animating each step. The *to end* button does the same as *all steps*, but jumps to the final output immediately.

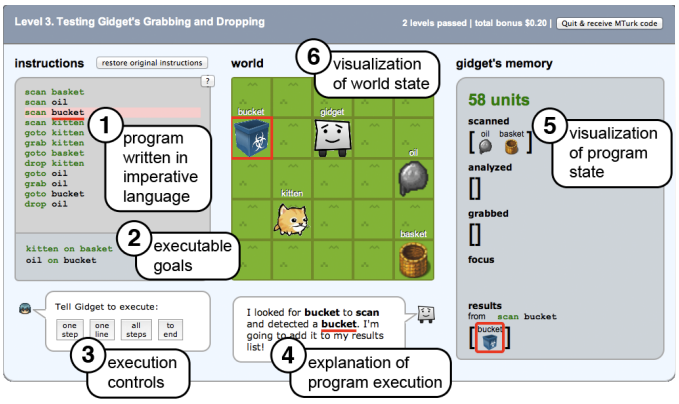

Figure 1. The Gidget game (with annotations), where learners help a damaged robot fix its programs by debugging its code.

B. The Three Level Conditions

The independent variables we manipulated in our experiment were the labels and visual appearance of the objects referred to in the level goals (e.g. Table 1). The data elements in the inanimate condition were colored blocks, and were intended to diminish the purposefulness of the goals, separating them from the context of the story. In contrast, the other two conditions' data elements were designed to be specific, animate objects. In the invertebrate condition, the data elements included beetles, flies, ladybugs, bees, termites, butterflies, and spiders. In the vertebrate condition, the data elements included cats, birds, dogs, kittens, puppies, piglets, and rats. These conditions were intended to increase the purposefulness of the goals, tying them to the context of the story. Supporting objects across the conditions did not follow these categories, but were modified to be consistent with the game's story (i.e. cleaning up an *oil* spill) and type of object (e.g. *block::bin*, *beetle::jar*, *cat::basket*, respectively). Our hypothesis, based on prior work showing that humans empathize and attribute more positive attitudes towards vertebrates [1,3], was that players would ascribe more purpose in saving vertebrates than inanimate objects, and therefore complete more levels.

C. Participant Recruitment, Compensation, & Demographics

We recruited participants using Mechanical Turk (MTurk), an online marketplace where individuals can receive micro-payments for doing small tasks called Human Intelligence Tasks (HITs). Our pricing model and validation method was carried over from a previous study [12], which was designed to minimize the effect of monetary compensation on players' motivation to start and continue playing the game. Participants were given $0.40 for completing the mandatory first level, and an additional $0.10 per level completed thereafter. These decisions were validated to attract participation in a pilot test consisting of 29 players from MTurk, and 6 people in-person.

We focused on self-reported, rank novice programmers. Our HIT description deliberately did not mention programming to prevent people from self-selecting out of the task. A total of 121 participants met our criteria for being novice programmers, which were 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."

Participants were distributed proportionally among our three conditions by demographics, with no statistically significant differences in age ($F(2,117)=1.46$,MSE=111.3,n.s.), gender ($\chi2(2,N=121)=1.1$,n.s.), level of education ($\chi2(14,N=121)=4.0$,n.s.), or country of residence ($\chi2(32,N=121)=30.7$,n.s.). The median age was 26, ranging from 18 to 66 years old. Our sample included 63 females and 58 males. Fifteen countries were represented in our study, with participants primarily from the US (61.6%) and India (14%). Our sample was well-educated, with 80.9% reporting that their highest level of education was some college or beyond.

D. Procedure & Dependent Measures

On game load, each participant was randomly assigned one of the three conditions. Once a participant chose to quit, they were given a post-survey asking about gender, age, country, education, programming experience, and asked to select their agreement to the following attitude-measurement statements on a 5-level Likert scale: 1) "I enjoyed playing the game," 2) "I would recommend this game to a friend wanting to learn programing," 3) "I wanted to help Gidget succeed," and 4) "I enjoyed interacting with the objects in Gidget's world."

In addition to the survey responses, we collected a time-stamped activity log of all participants' attempted levels including: (1) Each ***press*** of the execution buttons and a copy of the code at the time of execution; (2) ***Level start*** & ***level end***: events marking when a player started, completed, or quit a level; (3) ***Idle start*** & ***idle stop***: events marking mouse or keyboard inactivity (of 30 seconds or more), and where in the UI the idle time occurred. Events were also recorded marking resumption in activity; (4) ***Edit time*** (***edit in*** & ***edit out***): events marking when the player clicked inside the code pane to edit code or clicked elsewhere to leave the editing pane; (5) ***Pane time*** (***time in*** & ***time out***): timestamps of mouse cursor movement over or out of the major UI panes.

From these, we calculated the following dependent measures for each participant: (1) ***Time on level***: how long individual participant was actively engaged with the code and interface of each level overall, adjusted by subtracting ***idle time***. This was calculated for each level by first taking the difference of ***level end*** and ***level start***, then subtracting ***idle time*** for that level; (2) ***Time overall***: how long each participant played the game overall, adjusted by subtracting ***idle time***. This was calculated by summing up the all of the time on level data per participant and subtracting the sum of their ***idle time***.

Finally, each participants' *number of levels completed*, *time to complete* or *quit* a level, and logs of *execution buttons* and *UI pane activity*, were used to compute dependent measures of activity proportional to overall time spent on levels.

IV. Results

We used non-parametric statistical tests (at $\alpha=0.05$), as our dependent measures were not normally distributed.

A. Vertebrate Condition Players Complete More Levels

All participants completed at least one level. The maximum number of levels completed in the inanimate, invertebrate, and vertebrate conditions were 9, 16, and 18, respectively. There was a significant difference in the number of levels participants completed between the three conditions ($\chi2(2,N=121)=7.3$,$p<.05$). Further post-hoc analysis with a Bonferroni correction shows that the significantly different pair was the inanimate vs. vertebrate conditions ($W=1380.5$,$Z=-2.5$,$p<.01$), with the vertebrate group completing more levels. Comparison of the inanimate vs. invertebrate ($W=1669.5$,$Z=0.5$,n.s.) trended towards significance with the invertebrate group completing more levels. Finally, comparing the vertebrate vs. invertebrate ($W=1427.5$,$Z=-2.0$,n.s.) conditions showed no difference.

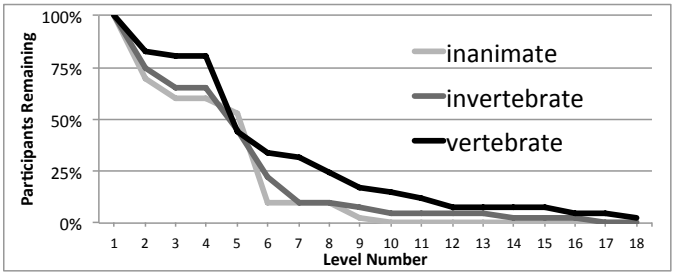

Figure 2. Percent of players remaining for each condition at each level. By level 10, all in the inanimate condition had quit.

Investigating further, Figure 2 shows that approximately 25% of the participants from each group quit the game after completing only the first level. Next, many participants quit on level 4, which required them to use the command learned in the previous level with a new command. Finally, participants quit again in large numbers on level 6, which introduced conditional statements. This is consistent with others' findings that novice programmers have difficulty with conditional logic [6]. Here, the inanimate condition had the most drastic drop with 90% of its participants quitting, followed by a drop of 77.5% and 67.5% of participants in the invertebrate and vertebrate conditions, respectively. All of the inanimate condition's participants quit by level 10, whereas the other conditions had a few participants complete or nearly complete all the levels.

Since all participants were novice programmers with no statistical difference in demographics, these results suggest that interacting with goals that use animate data elements had a significant positive effect on participants' engagement with the game, particularly on levels introducing difficult concepts.

B. Invertebrate & Vertebrate Condition Players Play Longer

There was a wide range of overall play times for the inanimate, invertebrate, and vertebrate conditions (4.9 min to 1.3 hrs, 8.3 min to 1.9 hrs, and 6.9 min to 2.8 hrs, respectively). There was a significant difference in the length of time participants played the game overall by condition ($\chi2(2,N=121)=10.2$,$p<0.01$). A post-hoc analysis with Bonferonni correction reveals that two conditional pairs were significantly different: inanimate vs. vertebrate ($W=1330$,$Z=-2.9$,$p<.016$) and inanimate vs. invertebrate ($W=1889$,$Z=2.6$,$p<.016$). In both cases, the inanimate condition players spent significantly less time playing the game than the other conditions. Play time between the animate conditions did not differ ($W=1620$,$Z=-0.2$,n.s.).

Next, we investigated how quickly players completed levels by comparing participants' ratio of total play time to number of levels played, finding no significant difference ($\chi2(2,N=121)=3.7$,n.s.). In particular, the median times to complete the first 5 levels were very close across conditions.

C. No Significant Differences in Code Execution Strategies

One possible explanation for the differences in levels completed was a different use of the game UI. Therefore, we investigated the proportion of execution button presses per unit time on completed levels, for each of the four execution buttons, finding no significant differences in usage (one step: (χ2(2, N=121)=2.2,n.s.), one line: (χ2(2,N=121)=0.5,n.s.), all steps: (χ2(2,N=121)=1.6,n.s.), to end: (χ2(2,N=121)=0.1,n.s.)). These results show that the differences in success were likely not due to one condition executing the program more frequently or stepping through it differently.

D. No Differences in User Interface Usage

Another possible explanation for the disparity in levels completed was differences in how participants used the various panels in the UI. We examined the proportion of interface pane usage to overall time on levels played, again finding no significant differences among conditions (Code: (χ2(2,N=121)=2.5,n.s.), Goals: (χ2(2,N=121)=4.0,n.s.) Execution: (χ2(2,N=121)=3.7,n.s.), Feedback: (χ2(2,N=121)=0.3,n.s.), World: (χ2(2,N=121)=4.3,n.s.), Memory: (χ2(2,N=121)=1.0,n.s.), CheatSheet: (χ2(2,N=121)=5.8,n.s.)). We also tested the proportion of time spent editing code (computed from the logs) to overall time on levels and found no significant differences among conditions (χ2(2,N=121)=0.9,n.s.). All of these results suggest that the differences in success and play time were not due to players' variations of user interface usage in the game.

E. No Significant Differences in Attitudes

Although there was a trend in survey responses indicating a positive experience playing the game, there was no significant difference in participants' self-reported level of enjoyment comparing the three conditions (χ2(8,N=121)=5.8,n.s.) or whether they would recommend the game to a friend wanting to learn programming (χ2(8,N=121)=4.1,n.s.). Similarly, there was a positive trend in responses across conditions, but no significant difference in participants' self-reported desire to help Gidget succeed (χ2(8,N=121)=11.6,n.s.) or whether they enjoyed working with their data elements (χ2(8, N=121)=5.5,n.s.).

V. Discussion & Implications

Our findings show that goals involving vertebrate objects (and to some degree, invertebrate objects), rather than inanimate objects, significantly increase learners' engagement in a programming game, leading rank novices to play significantly longer and complete significantly more levels. Moreover, we showed that these effects were not due to differences in how players executed the programs, how they used the game UI, how long they attempted each level, or how much time they spent editing their code.

A possible interpretation of these results is that when reaching a difficult level, players saw a greater purpose in saving an animal or insect than in moving a block. Prior research on computer science recruitment shows that there are gender specific effects in motivation to enroll, specifically related to the reasons for computing (females are enticed when they see how computing can be used for a purpose) [14].

Our results have many potential implications for our understanding of online learning, the role of game elements in engagement, and computing education pedagogy. Our results show that purposeful goals may play a significant role in engagement in the context of self-guided, discretionary, educational games. These findings support prior works done in classroom settings [10,14], and broadens them to informal learning settings. Future work should investigate the effects of these factors on learning, both in formal and informal contexts.

In addition, this study demonstrated that small changes to the game elements can have a significant effect on engagement in educational games. Here, we had large effect sizes, with double the overall play time and level completion, as was the case in our prior study [12]. This suggests that in the growing amount of work in educational games research, game designers should be doing more on low-level factors that are predicted to be influential by research in learning, memory, and attention.

Our results raise many questions about the underlying mechanisms of this effect, which range from effects on motivation, learning, and attention. In our future work, we hope to investigate these possible mechanisms, leading to a deeper understanding of the effect of the presentation of a computer on a person's ability and desire to program and do so successfully.

Acknowledgements

This material is based in part upon work supported by the National Science Foundation (NSF) under Grant Number CCF-0952733. 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.

References

[1] Batt, S. (2009). Human attitudes towards animals in relation to species similarity to humans. *Bioscience Horizons,* 2(2), 180–190.

[2] Bowman, R.F. (1982). A Pac-Man theory of motivation: tactical implications for classroom instruction. *Educational Tech.*, 22(9), 14-16.

[3] Bradshaw, J.W.S., Paul, E.S. (2010). Could empathy for animals have been an adaptation in the evolution of Homo sapiens? *Animal Welfare*, 19(1), 107-112.

[4] Carter, L. (2006). Why students with an apparent aptitude for computer science don't choose to major in computer science. ACM *SIGCSE Bulletin*, 27–31.

[5] Corno, L., Mandinach, E.B. (2004). What we have learned about student engagement in the past twenty years. *Big Theories,* 297–326.

[6] Dahotre, A., Zhang, Y., Scaffidi, C. (2010). A qualitative study of animation programming in the wild. *ACM-IEEE ESEM*, 1-10.

[7] Freeman, P., Aspray, W. (1999). The supply of information technology workers in the United States. *Computing Research Association*

[8] Garris, R., Ahlers, R., Driskell, J.E. (2002). Games, motivation, and learning: a research and practice model. *Simulation & gaming*, 441–467.

[9] Kearsley, G., Shneiderman, B. (1998). Engagement Theory: a framework for technology-based teaching and learning. *Educational Technology*, 38(5), 20-23.

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

[11] Layman, L., Williams, L., Slaten, K. (2007). Note to self: make assignments meaningful, ACM *SIGCSE, 459-463.*

[12] Lee, M.J., Ko, A.J. (2011). Personifying Programming Tool Feedback Improves Novice Programmers' Learning. *ACM ICER*, 109-116.

[13] Malone, T.W. (1981). What makes computer games fun? *Byte*, 6(12), 258-277.

[14] Margolis, J., Fisher, A. (2002). Unlocking the Clubhouse. *MIT Press*.

[15] Oblinger, D., Oblinger, J. (2005). Educating the net generation. *Educause*.

[16] Papert, S. (1980). Mindstorms. *Basic Books*.