Are Engineering Students Motivated by Interacting With Simulations They Program? A Controlled Study

John Bacher
jtbacher@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Thomas Price
twprice@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

James Skripchuk
jmksripc@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Wengran Wang
wwang33@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Yang Shi
yang.shi@usu.edu
Utah State University
Logan, Utah, USA

Keith Tran
ktran24@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

ABSTRACT

When instructors want to design programming assignments to motivate their students, a common design choice is to have those students write code to make an artifact (e.g. apps, games, music, or images). The goal of this study is to understand the impacts of including artifact creation in a programming assignment on students' motivation, time on task, and cognitive load. To do so, we conducted a controlled lab study with seventy-three students from an introductory engineering course. The experimental group created a simulation they could interact with – thus having the full experience of artifact creation – while the control group wrote the exact same code, but evaluated it only with test cases. We hypothesized that students who could interact with the simulation they were programming would be more motivated to complete the assignment and report higher intrinsic motivation. However, we found no significant difference in motivation or cognitive load between the groups. Additionally, the experimental group spent more time completing the assignment than the control group. Our results suggest that artifact creation may not be necessary for motivating students in all contexts, and that artifact creation may have other effects such as increased time on task. Additionally, instructors and researchers should consider when, and in what contexts, artifact creation is beneficial and when it may not be.

CCS CONCEPTS

• **Social and professional topics → Adult education**.

KEYWORDS

assignment Design, motivation, novice programmers, interactive artifact

ACM Reference Format:
John Bacher, Thomas Price, James Skripchuk, Wengran Wang, Yang Shi, and Keith Tran. 2024. Are Engineering Students Motivated by Interacting With Simulations They Program? A Controlled Study. In *Proceedings of (SIGCSE Virtual '24).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

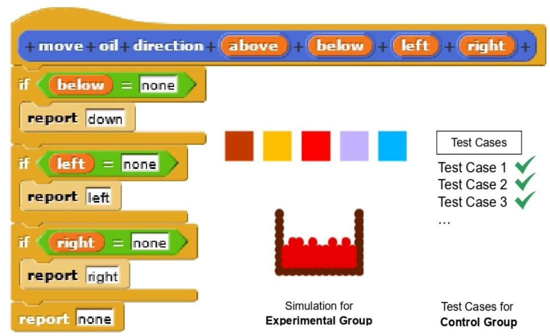

Figure 1: Example code and output for the experimental group (left) and control group (right)

1 INTRODUCTION

Many computer science instructors try to design programming assignments that will be motivating to their students. When assignments successfully motivate students, this can bring many benefits, such as increased academic performance [4] and academic engagement [3]. Computing education research venues host many examples of motivating assignments. These include experience reports [1, 9, 42] and the SIGCSE Nifty Assignments collection [30] where instructors describe assignments they believe were effective at motivating students. They also include whole curricula, like media computation [10] and the beauty and joy of computing curriculum [8] which emphasize interesting assignments as a valuable way to teach novice programmers. Research studying the impact of these curricula has found them to be successful at improving student grades [35, 36] and increasing retention of students [27]. However, while examples of motivating assignments are important, it is also important to understand *why* they work – what specific instructional design choices influence motivation?

There are many choices an instructor makes when designing an assignment (e.g., what narrative context to provide, what scaffolding to provide, how it will be assessed, etc.). This paper focuses on one specific design choice that is common to many nifty assignments and motivating curricula: having students write code to create an artifact that they can *experience and interact with*, such as a game, story or simulation – what we call *artifact creation*. For example an assignment could involve designing a game that the student can then play and share [14], developing a creative story in a block-based programming environment [15], manipulating images using programming [10], using programming to produce music [17], or designing programs to produce simulations [31]. All of these assignments have, as their output, artifacts which motivates students to complete the assignment because of increased intrinsic motivation from interest in the artifact [24] or increased ownership from the agency of making an artifact [34].

Despite the prevalence of artifact creation in CS assignments, there is little work investigating the effectiveness of this specific design decision in isolation, using a controlled experiment, to better understand the role it plays in motivation. There exists evidence that courses and curricula which use assignments with artifact creation lead to increased motivation [7], retention [27], and grades [35, 36]. However, these studies examine the effect of many design decisions simultaneously in the context of changing curricula; therefore, it is difficult to attribute the assignment's effectiveness to the presence of artifact creation specifically. While it seems likely that artifact creation *can* increase student motivation, it is important to understand *when* this design choice is beneficial, and *for whom*, since it may not be important in every context. One could imagine in such cases it may be better for instructors to have students write code without artifacts. Additionally, it is important to understand if this choice has additional impacts on student outcomes, beyond just motivation. Similar controlled experiments have played an important role in guiding how best to design instructional content, and in aggregate these experiments can build to larger theories, such as multimedia learning theory [37].

In this work we investigate the impact of the artifact creation design choice in isolation, to better understand its impact for one student population on motivation (RQ1), time on task (RQ2), and cognitive load (RQ3). We did so through a controlled lab study of 73 students enrolled in an introductory engineering course, most of whom go on to take an introductory programming course. We isolated the impact of artifact creation by comparing student outcomes on two versions of the same assignment. In the assignment, both groups used the coding interface in Figure 1 to code elements, such as oil and gas, for use in a simulation modeling the interaction of those elements. However, only students in the experimental group could *see and interact with the simulation* – central properties of an artifact creation assignment. Students in the control group saw a series of test cases evaluating their code, but did not experience the simulation in any way. Our hypothesis is that students in the experimental condition would find the assignment with the artifact to be motivating. We additionally measure cognitive load, as changes in design of instructional content often affect cognitive load [39], and time on task, to determine how active time is affected by artifact creation.

We found that both groups were motivated to complete the assignment and that there were no significant differences in their motivation and cognitive load. However, students in the experimental group spent significantly more time on the assignment. Our results suggest that artifact creation may not be necessary to motivate our students, and our survey data supports the idea that students in the control group may have been motivated by other factors such as an interest in programming or problem solving. Additionally, artifact creation in assignments may impact the amount of time students need to complete an assignment. Our work demonstrates the potential for investigating individual design decisions in novice programming assignments. This study emphasizes the importance of considering population and context when designing assignments, and suggests that artifact creation may be less motivating in certain contexts.

2 RELATED WORK

2.1 Motivation

In education motivation is a drive that moves students to engage with educational content [13]. Motivation can come from external rewards, a sense of value, or a goal the student has that education helps them achieve [29]. One theory of motivation, self-determination theory, recognizes two distinct kinds of motivation, intrinsic and extrinsic motivation [34]. Intrinsic motivation is the drive of an individual to do an activity because of an existing interest or an internal desire to do the activity. Extrinsic motivation, in self-determination theory, is motivation from factors other than intrinsic motivation, and includes things like social pressure to do well or anxiety [34]. Artifact creation could increase a student's intrinsic motivation either by increasing the value the student sees in the assignment or the interest they have in the assignment.

2.2 Instructional Design & Motivation

When making curricula, assignments, and learning activities, motivation is an important goal as research on various theories of motivation have found that motivation is associated with higher achievement [29]. Linnenbrink-Garcia et al. reviewed existing literature on instructional design for motivation and extracted five design principles for supporting different motivation with one principle being making lessons personally relevant to students [21] which is one goal of artifact creation assignments. One key step in instructional design for motivation is the evaluation of the results of the motivational content [29], further it is suggested that a formal evaluation process is essential for best understanding the effects on motivation [16].

One successful method for evaluation is controlled experiments, which have played an important role in guiding how best to design instructional content, and in aggregate these experiments can build to larger theories, such as multimedia learning theory [37]. Previous work in multimedia learning theory has investigated the impact of various individual design decisions including whether the medium of the multimedia representation matters [23], whether assignments that evoke positive emotions are more effective [41], and whether decorative illustrations lead to situational interest or distract students [22] . These decisions were evaluated using controlled experiments that only varied the individual decision in question, thus allowing educators to select the best approach for their context.

2.3 Cognitive Load Theory

Cognitive load is the burden imposed on a student's working memory to process information as it is transferred to long term memory during the learning process [40]. Cognitive Load Theory (CLT) identifies that a student's working memory is limited, and working memory must be spent on various tasks during learning. CLT identifies three types of cognitive load: intrinsic, extraneous, and germane [20, 38]. Intrinsic Load is the cognitive load associated with the difficulty of an individual task or assignment. Intrinsic load is largely immune to instructional intervention [39]. Extraneous load describes cognitive load generated by the learning activity based on the way information is presented or instructions are organized [20]. Extraneous load is the cognitive load instructors have the most control over [39], and modifying assignments to reduce complexity in instructions is key to reducing extraneous load. Germane load is the type of cognitive load associated with learning as germane load is the excess cognitive load that is allocated for intrinsic cognitive load, not extraneous cognitive load [39].

3 METHODOLOGY

We conducted a controlled lab study, approved by our institution's ethics board, where students were required to complete an introductory programming assignment aimed at having students practice conditional programming logic and learn to create a simulation in Snap*!*. Students were assigned randomly into two groups: the experimental group with artifact creation and the control group without artifact creation. Both groups were provided with the same narrative context in the task description – simulating how elements interact – but only the experimental group could interact with (and, we hypothesize, be motivated by) the simulation they created. By evaluating how students performed on quantitative measures in the two groups, and by qualitatively analyzing responses to open-ended questions, we investigated how artifact creation affects student motivation, time on task, and cognitive load.

3.1 Population

We recruited 73 undergraduate students from an introductory engineering course focused on teaching students how to use technology that is used throughout their degree program like UNIX, HTML, spreadsheet programming, and some computer hardware. We recruited students from this course because they included students who had the ability to major in computer science, but who had not yet selected a major – a time at which motivation is particularly important. Students also self-reported their prior programming experience and the vast majority (61/73) of students in our study indicated that they had some prior programming experience (e.g. taking a programming course, using programming for class, completing an independent tutorial, or using programming for a personal project). Students in the course were emailed information about our study, and those that participated were provided a small amount of extra credit. Of those recruited, 34 students identified as White, 23 identified as Asian, 5 identified as Hispanic or Latino, 5 identified as Black or African-American, and 6 identified as other or did not specify. Additionally, 53 students identified as male, 18 identified as female, and 2 did not specify.

3.2 Materials

We searched the SIGCSE Nifty assignments repository for an assignment that should be motivating for our population of students to create and we selected the Falling Sands' artifact [32]. We chose from SIGCSE Nifty assignments because they are assignments designed by practitioners with the goal to motivate students, and many involve artifact creation. We implemented Falling Sands in Snap*!* [8], a block-based programming environment. The original version of the Falling Sand assignment focuses on teaching loops and two-dimensional arrays. Given our goal of assessing the impact of artifact creation assignments on student motivation it was important that all our participants, regardless of prior programming experience, could complete the assignment. Therefore, we modified the *programming tasks* of the original Falling Sands assignment to focus on conditional logic as we expected this topic to be accessible to students regardless of their prior programming experience. However, the *output of the program* – simulated elements interacting – is the same as the original assignment, and it is that artifact creation element of the assignment that we are investigating in this work. Our modified version requested students to build the simulation by coding the behavior of different elements (e.g. sand, oil, gas) represented using custom blocks, akin to functions, in Snap*!* using conditionals. In our modified version of the assignment, students only implemented the behavior of each element (e.g. sand falls, oil floats on water), and the rest of the simulation was provided as starter code. For example, students were required to write code, seen in Figure 1 that caused oil to move left, if the variable `left` is empty, or right if the variable `right` is empty. This simplification and scaffolding was necessary to allow novices to create the complex simulation, and we discuss possible implications in Section 5.1.

3.3 Conditions

Students were randomly assigned to a condition upon agreeing to participate in this study. In the experimental condition, students were provided access to the simulation to test their code. In the control group, the portion of the code that renders the simulation was removed and instead replaced with a series of test cases that students used to test their implementation. The test cases were displayed to the user using Snap*!*'s 'say' function. The correct solutions and the instructions were the same between the conditions; therefore, the only difference is whether the student had access to the simulation or test cases. Removing the simulation reduces the potential increase in motivation from student interest in the simulation or from the ownership of creating the artifact, while also changing the way that students test their code. Test cases are often not provided to students programming artifacts as students are expected to use the artifact to test their code.

3.4 Procedure

Before any procedures began, students were provided with an overview of the research and a consent form that they completed prior to taking part in this research and a short tutorial on how to use Snap*!*. This tutorial demonstrated how to implement the sand element and the students were also shown how to test their code. The tutorial was followed by a short demographics survey and a pre-test.

There were four tasks that the students must complete. Students were provided with the implementation of the "sand" element as a reference. In the first task, the students implemented behavior for oil so that it is affected by gravity and falls to the ground; therefore, mimicking the behavior of sand. The second task required students to build on the first by implementing behavior in oil so that it flows to the left and right, filling up a volume like a liquid. In the third task, students implemented gas which behaves the exact same as oil, except that it floats rather than sinks.

In the fourth task, we provided students with three options. We added a choice in task four as another metric for potentially evaluating student motivation based on the assumption that more motivated students might select a choice that required more work. Choice in the assignment could also lead to increased motivation [21]. The first choice required the least amount of work, one to three blocks, and required students to implement behavior so that oil sinks below the sand. The second required four to eight blocks, and required students to have oil turn into gas when it touches sand. The final task required ten to twelve blocks and required the students to implement a new element, water, and create an interaction between sand and water.

After completing all tasks, students were asked to complete a post-survey, which asked students about their motivation and cognitive load as well as a post-test. These measures are explained in Section 3.5. The survey took about fifteen minutes to complete.

3.5 Measures

Students completed a pre/post-test and two surveys. The pre-survey was focused on demographic information. The post-survey consisted of three measures: To measure **motivation (RQ1)**, we used the Intrinsic Motivation Inventory (IMI) [33] asks students to answer likert scale questions regarding their motivation in working on the assignment they just completed. This measure consists of six subscales that relate to different factors that contribute to an individual's motivation (shown in Table 1). The IMI is task-specific and is validated as a measure of motivation on a specific task [5, 25] and has been used previously in education [6]. To measure **cognitive load (RQ3)**, we used the Cognitive Load Component Survey (CLCS) [26], which assesses a student's intrinsic, extraneous and germane cognitive load with subscales for each during an assignment [26].

To measure **time on task (RQ2)**, we used log data gathered from each student during the programming task. Each student did eventually complete the programming task successfully, as verified by a researcher during the study, so time on task serves as our best measure of the impact of artifact creation on students' programming experience. We note that increased time on task can be considered a positive outcome (e.g., if additional learning occurs during that time) or a negative one (if the student is struggling or demotivated), as we discuss in Section 5.2. Specifically, we measure a student's *active time*, as in prior work [19, 28], which only includes time when students have taken an action in the past 3 minutes (e.g., modifying code).

Students also took a short isomorphic pre/post-test, which assessed students' knowledge of variables, conditionals, and loops. The pretest was included to verify that the control and experimental groups had similar levels of prior knowledge; therefore, the pre-test included eleven questions: three questions on variables, four on conditionals, four on loops in Snap*!*. Our results, shown in Table 1, confirm this assumption (control mean 86.67; experimental mean 89.87; no significant difference). Since our primary research interest was students' motivation, and we would not expect to see much learning given students' high pretest scores and the short duration of the assignment, our analysis does not focus on these tests, but we report them for completeness.

We added a set of five open-ended questions, listed below, after the first fourteen students in order to better understand student perception of the assignment. As a result of including this measure late only fifty-nine students saw these questions. We added the open-ended questions to provide better insight into student motivation on this assignment with the goal of understanding what parts of the assignment led to student interest.

(1) What was your favorite part of this assignment?
(2) What was your least favorite part of this assignment?
(3) Think of a moment in this session where you were unsure what to do. Describe what happened.
(4) Think of a moment in this session where you were excited. Describe what happened.
(5) Think of a moment in this session where you were frustrated. Describe what happened.

3.6 Analysis

For the IMI we analyzed the results by summing each question in a subscale to determine the student's motivation. We additionally average each subscale by the number of questions related to that subscale in the IMI given that subscales varied in the total number of questions from five to seven. The resulting score indicates how motivating each factor was on this specific assignment for the student. The CLCS consists of three subscales for three kinds of cognitive load and we summed the values on those subscales to create a single value representing that subscale. Similar to the IMI, we averaged each subscale total by the number of questions in that subscale given that there are three questions for intrinsic and extraneous, and four questions for germane. For the pre- and post-test we report the average percentage of questions the students answered correctly. For each measure, and for each subscale in a measure, we divided the students by condition and performed a one-way analysis of variance to determine if there are significant differences in motivation and cognitive load.

For the open-ended questions we focused our analysis on the response of each student to questions one and four as both questions focused on parts of the assignment that may interest or motivate students, and therefore could help provide context for RQ1. As a result, the coding process focused on answering the question: what did the student find exciting during the assignment. Two researchers engaged in an adapted thematic analysis process [2] beginning with an open coding phase where each researcher read the same five responses and identified sentences from students that indicated their interest. The researchers each assigned a candidate code to the sentence and then met and discussed their codes. This process was repeated until no new codes appeared. The researchers then discussed the existing codes and constructed a closed list of codes. Researchers then identified similar codes and merged those codes into a closed codebook which we discuss in Section 4.4. Using the closed list of codes, each researcher coded every response, given the short length of many of the responses, and discussed any conflicts that arose during the closed coding process. The researchers reached consensus for each response, so we do not report inter-rater reliability.

Instrument	Max	Exp. Mean (SD)	Ctrl. Mean (SD)	Exp. Median[1]	Ctrl. Median	Cohen's d	p-Value
IMI Interest	7	4.16 (0.61)	4.05 (0.72)	4.14	4.00	0.17	0.50
IMI Effort	7	3.83 (0.65)	3.79 (0.52)	3.80	3.90	0.07	0.67
IMI Choice	7	2.97 (0.68)	3.04 (0.62)	2.71	2.71	-0.11	0.52
IMI Competence	7	4.57 (0.96)	4.36 (0.85)	4.83	4.50	0.23	0.25
IMI Pressure	7	3.27 (0.64)	3.46 (0.54)	3.40	3.40	-0.33	0.22
IMI Value	7	5.11 (1.25)	5.04 (1.21)	5.08	4.83	0.05	0.82†
Pre-Test	100	86.67 (16.01)	89.89 (15.76)	90.91	100	-0.01	0.72
Post-Test	100	92.79 (12.86)	91.72 (12.21)	100	100	0.08	0.78
Assignment Time	N/A	26.92 (11.26)	22.08 (7.68)	26.35	20.79	0.49	**0.04**†
CLCS Intrinsic	10	1.82 (1.06)	2.20 (1.57)	1.83	2.00	-0.30	0.39
CLCS Extraneous	10	8.22 (1.25)	7.96 (1.49)	1.33	2.00	-0.27	0.25
CLCS Germane	10	1.52 (2.42)	1.83 (2.45)	6.13	6.25	0.10	0.69†

Table 1: Comparison of outcomes for the experimental and control groups across measures, with effect size of differences and significance of statistical test. All measures are non-normally distributed and a Mann-Whitney U test was used to calculated p, except for those marked with †, where a t-test was used.

4 RESULTS

4.1 RQ1: Motivation

Student responses to the IMI are our primary measure of motivation. The results of our analysis for the IMI are presented in Table 1. Starting with the interest subscale, we see there is marginally higher motivation in the experimental group over the control group, but the effect size is quite small, and the difference is not significant. The results are not significant for all other subscales in the IMI. The effort, perceived competence, and value scale are slightly higher in the experimental group, while the perceived choice and pressure scales are higher in the control group. For each subscale, the effect sizes are quite small. We therefore see little evidence that the artifact creation impacted students' self-reported intrinsic motivation.

Student selection on Task 4 is another potential indicator of motivation, as students who are more motivated may be more likely to select more challenging, but more interesting tasks. The proportion of students who selected either of the more difficult choices for Task 4 was slightly higher in the experimental group (13/34 = 38%) than the control group (11/39 = 28%), but a χ^2 test shows that this difference was not significant ($\chi^2 = 0.83; p = 0.36$). Overall, most students opted for the easiest choice (49/73 = 67.12%). Together these results do not strongly support our hypothesis for RQ1, and we cannot reject the null hypothesis.

4.2 RQ2: Time on Task

We also examined the amount of active time that students spent during each task. This helps us understand whether artifact creation may affect students *as they are programming*. As shown in Table 1 the experimental group took *more* time on average (26.9 minutes) than the control group (22.1 minutes), and this effect is significant, with a medium effect size ($d = 0.49$), suggesting that having access to the artifact may have caused students to complete the assignment more slowly.

4.3 RQ3: Cognitive Load

Responses on the CLCS indicate slightly lower intrinsic and extraneous cognitive load (negatively associated with learning) for the experimental group while there was slightly higher germane load (positively associated with learning) for the control group. There were no significant differences between the two groups on any of the subscales, and effect sizes for each subscale were quite small. Given the consistency of these results across subscales, our results may suggest the potential for artifact creation to reduce cognitive load, but if so the effect is relatively small.

4.4 Qualitative Analysis

We found evidence that the experimental group found the artifact itself to be motivating in their answers mentioning the *output* of the assignment (Exp.: 18/27, Ctrl.: 6/32). One student in the experimental group expressed that, "getting to watch the particles move after I completed my code," was enjoyable, while another wrote "I was excited during the water and sand portion when I completed the code and got to play with the simulation to make sure it works correctly." Students in the control were less likely to mention outputs, but were more likely to mention *coding* (Exp.: 9/27, Ctrl.: 12/32) and *problem solving* (Exp.: 7/27, Ctrl.: 15/32) in the absence of an interesting output. One student in the control group explained that it was exciting "being able to actually perform coding myself," while another expressed that they enjoyed "visualizing the walkthroughs of how the code works." Additionally, students in the control group were more likely to mention the *completion* (Exp.: 7/27, Ctrl.: 18/32) of the assignment, or completing specific tasks, as motivating. One student in the control group wrote "whenever I passed the test cases, I was happy." Finally, a small number of students in both groups mentioned *choice* (Exp.: 4/27, Ctrl.: 4/32) as motivating. One student in the control group expressed that they, "enjoyed getting to choose which of the next tasks I'd work on for that one part," while another in the experimental group wrote, "my favorite part of the assignment was the final task, in which we could choose what task to pursue."

5 DISCUSSION

5.1 Research Question 1: Motivation

We hypothesized that students in the experimental group would report higher intrinsic motivation to complete the assignment on the IMI due to their ability to see and interact with the artifact they were creating. Contrary to our hypothesis, we found little evidence that the ability to see and interact with the artifact increased student motivation. There are a few potential reasons for this. First, it is possible that students in the experimental group did not find the Falling Sands simulation itself to be interesting, so interacting with the artifact did not increase motivation. We did not measure this directly as the IMI asked about the whole programming activity, not the artifact itself; however, the results from the qualitative analysis suggest that students in the experimental group did find the artifact interesting. As mentioned in Section 4.4, students in the experimental group expressed a variety of interest in the output of the program, for example, "I did enjoy watching the sand change to gas when it hit the oil and actually all the other visual displays of the tasks and watching the elements move throughout the display area." Second, it is possible that students found the simulation to be interesting, but did not feel a sense of ownership [34] when "creating," the artifact, either because they did not feel like the code they wrote meaningfully contributed to the simulation or the simulation itself was prescribed to them and therefore not their own design. It is possible that lack of ownership may limit the impact of artifacts on students' motivation. Third, it is possible that both groups were motivated by the coding task either because programming was relevant for the student or because of their interest in critical thinking and problem solving, and this programming-related motivation eclipsed any differences between the groups due to the experimental manipulation. This may explain the small, non-significant effect we did observe. This is supported by the control groups' free responses where they identified an interest in programming and problem solving as contributing to their interest in the assignment (Section 4.4). As we discuss below, our results for RQ2 and RQ3 leave open the possibility that artifact creation could bring benefits – and also costs – even in the absence of increased motivation.

5.2 Research Question 2: Time on Task

Our results show that the time students spent on the assignment was significantly higher in the experimental group; however, due to our many comparisons in this study, it is important to interpret this result with caution, and we do not make strong claims about this effect. In our study, students in the experimental group needed on average about 5 minutes (22%) longer to complete the assignment (including all four tasks). Students in the experimental condition may have had increased time as they had to come up with their own strategy for testing the correctness of their code using the simulation and interpret the results of the simulation, which could also contribute to the amount of time spent on the assignment. Increased time on task can be beneficial if there is an increase in student learning; however, we did not observe this effect in our results. This suggests that, in some situations, assignments without artifact creation may require less student time and lead to similar student outcomes. Future work should explore how students spend their time during artifact creation assignments, and also investigate the relationship between the time spent and the effects on learning.

5.3 Research Question 3: Cognitive Load

While we found a pattern of lower intrinsic and extraneous cognitive load across each subscale in the experimental group, these differences were not significant, and our results remain inconclusive as to whether artifact creation affects cognitive load. If so, the effect appears to be small. If the effect is real, one potential explanation for why we might observe lower extraneous cognitive load for students who are creating artifacts is that students can use their own prior knowledge (e.g., of how simulated materials should behave) to guide program development decreasing cognitive load. This interpretation aligns with constructionist [12] and discovery learning [11] theories of learning, in which students use what they already know to acquire more knowledge. However, there is evidence from cognitive load research that changes in instructional design, such as including an artifact, even when all else is equal can lead to an increase in extraneous load [39]. While inconclusive, our results raise the possibility that artifact creation may have benefits beyond motivation.

5.4 Implications

The primary implication of our study for instructors is that artifact creation is not a silver bullet that can provide motivation equally for all students in all contexts. For populations such as ours that already express high interest in programming itself, the added benefit of artifact creation may not outweigh the instructor effort required to design and grade such an assignment, or potential tradeoffs in terms of time on task or cognitive load. However, prior work [7, 17], suggests that there are likely many contexts where artifact creation *will* be motivating, and our results do not contradict these findings. For researchers, our work suggests the importance of further investigation to understand what differentiates these contexts; where is artifact creation most beneficial? Our findings emphasize the value of investigating domain specific instructional design choices in CS, and the need to better understand what factors facilitate or mediate motivation from assignment design decisions.

5.5 Limitations

One key limitation of our work is that we focused on only one artifact (a simulation) that may not have been the best choice for every student in our population; however, it was based on a popular SIGSCE Nifty assignment [32], and we argue it well represents the affordances of artifact creation, which is supported by students' open-ended comments. Additionally, for our population of engineering majors, simulations are highly relevant. Another limitation is that our assignment was relatively short and focused on only a few programming tasks. While it is possible that a longer assignment could lead to more changes in motivation previous work has identified that even a fifteen minute intervention is enough to affect student motivation [18].

6 ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grant #1917885

REFERENCES

- [1] Bhavya, Assma Boughoula, Aaron Green, and ChengXiang Zhai. 2020. Collective Development of Large Scale Data Science Products via Modularized Assignments: An Experience Report. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. ACM. https://doi.org/10.1145/3328778.3366961
- [2] Victoria Clarke, Virginia Braun, and Nikki Hayfield. 2015. Thematic analysis. *Qualitative psychology: A practical guide to research methods* 3 (2015), 222–248.
- [3] Lyn Corno and Eric M. Anderman. 2015. *Handbook of Educational Psychology*. Routledge, 91–103.
- [4] Luc G. Pelletier Edward L. Deci, Robert J. Vallerand and Richard M. Ryan. 1991. Motivation and Education: The Self-Determination Perspective. *Educational Psychologist* 26, 3-4 (1991), 325–346. https://doi.org/10.1080/00461520.1991.9653137 arXiv:https://doi.org/10.1080/00461520.1991.9653137
- [5] Terry Duncan Edward McAuley and Vance V. Tammen. 1989. Psychometric Properties of the Intrinsic Motivation Inventory in a Competitive Sport Setting: A Confirmatory Factor Analysis. *Research Quarterly for Exercise and Sport* 60, 1 (1989), 48–58. https://doi.org/10.1080/02701367.1989.10607413 arXiv:https://doi.org/10.1080/02701367.1989.10607413 PMID: 2489825.
- [6] Lisa Facey-Shaw, Marcus Specht, Peter van Rosmalen, and Jeanette Bartley-Bryan. 2020. Do badges affect intrinsic motivation in introductory programming students? *Simulation & Gaming* 51, 1 (2020), 33–54.
- [7] A. Forte and M. Guzdial. 2005. Motivation and nonmajors in computer science: identifying discrete audiences for introductory courses. *IEEE Transactions on Education* 48, 2 (2005), 248–253. https://doi.org/10.1109/TE.2004.842924
- [8] Dan Garcia, Brian Harvey, and Tiffany Barnes. 2015. The beauty and joy of computing. *ACM Inroads* 6, 4 (2015), 71–79.
- [9] Nasser Giacaman, Partha Roop, and Valerio Terragni. 2023. Evolving a Programming CS2 Course: A Decade-Long Experience Report. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023)*. ACM. https://doi.org/10.1145/3545945.3569831
- [10] Mark Guzdial. 2003. A Media Computation Course for Non-Majors. In *Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education* (Thessaloniki, Greece) *(ITiCSE '03)*. Association for Computing Machinery, New York, NY, USA, 104–108. https://doi.org/10.1145/961511.961542
- [11] David Hammer. 1997. Discovery learning and discovery teaching. *Cognition and instruction* 15, 4 (1997), 485–529.
- [12] Idit Ed Harel and Seymour Ed Papert. 1991. *Constructionism*. Ablex Publishing.
- [13] Suzanne E. Hidi, K. Ann Renninger, K. Ann Renninger, and Suzanne E. Hidi. 2019. *The Cambridge Handbook of Motivation and Learning*. Cambridge University Press, 1–12.
- [14] Yasmin B Kafai and Quinn Burke. 2015. Constructionist gaming: Understanding the benefits of making games for learning. *Educational psychologist* 50, 4 (2015), 313–334.
- [15] Caitlin Kelleher, Randy Pausch, and Sara Kiesler. 2007. Storytelling Alice Motivates Middle School Girls to Learn Computer Programming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 1455–1464. https://doi.org/10.1145/1240624.1240844
- [16] John M. Keller. 2010. *Integrating Motivational and Instructional Strategies*. Springer US, Boston, MA, 255–265. https://doi.org/10.1007/978-1-4419-1250-3_10
- [17] Christian Köppe. 2020. Program a Hit – Using Music as Motivator for Introducing Programming Concepts. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* (Trondheim, Norway) *(ITiCSE '20)*. Association for Computing Machinery, New York, NY, USA, 266–272. https://doi.org/10.1145/3341525.3387377
- [18] Jeff J Kosovich, Chris S Hulleman, and Kenneth E Barron. 2017. Measuring motivation in educational settings: A Case for pragmatic measurement. *The Cambridge handbook on motivation and learning* (2017), 39–60.
- [19] Juho Leinonen, Leo Leppänen, Petri Ihantola, and Arto Hellas. 2017. Comparison of time metrics in programming. In *Proceedings of the 2017 acm conference on international computing education research*. 200–208.
- [20] Jimmie Leppink, Fred Paas, Cees P. M. Van der Vleuten, Tamara Van Gog, and Jeroen J. G. Van Merriënboer. 2013. Development of an instrument for measuring different types of cognitive load. *Behavior Research Methods* 45, 4 (2013), 1058–1072.
- [21] Lisa Linnenbrink-Garcia, Erika A. Patall, and Reinhard Pekrun. 2016. Adaptive Motivation and Emotion in Education: Research and Principles for Instructional Design. *Policy Insights from the Behavioral and Brain Sciences* 3, 2 (2016), 228–236. https://doi.org/10.1177/2372732216644450 arXiv:https://doi.org/10.1177/2372732216644450
- [22] Ulrike I.E. Magner, Rolf Schwonke, Vincent Aleven, Octav Popescu, and Alexander Renkl. 2014. Triggering situational interest by decorative illustrations both fosters and hinders learning in computer-based learning environments. *Learning and Instruction* 29 (2014), 141–152. https://doi.org/10.1016/j.learninstruc.2012.07.002
- [23] Richard E Mayer. 1997. Multimedia learning: Are we asking the right questions? *Educational psychologist* 32, 1 (1997), 1–19.
- [24] Joseph E Michaelis and David Weintrop. 2022. Interest development theory in computing education: a framework and toolkit for researchers and designers. *ACM Transactions on Computing Education* 22, 4 (2022), 1–27.
- [25] Vera Monteiro, Lourdes Mata, and Francisco Peixoto. 2015. Intrinsic motivation inventory: Psychometric properties in the context of first language and mathematics learning. *Psicologia: Reflexão e Crítica* 28 (2015), 434–443.
- [26] Briana B. Morrison, Brian Dorn, and Mark Guzdial. 2014. Measuring Cognitive Load in Introductory CS: Adaptation of an Instrument. In *Proceedings of the Tenth Annual Conference on International Computing Education Research* (Glasgow, Scotland, United Kingdom) *(ICER '14)*. Association for Computing Machinery, New York, NY, USA, 131–138. https://doi.org/10.1145/2632320.2632348
- [27] Barbara Moskal, Deborah Lurie, and Stephen Cooper. 2004. Evaluating the Effectiveness of a New Instructional Approach. *SIGCSE Bull.* 36, 1 (mar 2004), 75–79. https://doi.org/10.1145/1028174.971328
- [28] Jonathan P Munson. 2017. Metrics for timely assessment of novice programmers. *Journal of Computing Sciences in Colleges* 32, 3 (2017), 136–148.
- [29] S Won Park. 2017. Motivation theories and instructional design. *Foundations of learning and instructional design technology* (2017).
- [30] Nick Parlante. 2024. *Nifty assignments*. http://nifty.standford.edu Accessed: 2024-05-26.
- [31] Nick Parlante, Julie Zelenski, Dave Feinberg, Kunal Mishra, Josh Hug, Kevin Wayne, Michael Guerzhoy, Jackie Chi Kit Cheung, and François Pitt. 2017. Nifty Assignments. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 695–696. https://doi.org/10.1145/3017680.3028255
- [32] Nick Parlante, Julie Zelenski, Dave Feinberg, Kunal Mishra, Josh Hug, Kevin Wayne, Michael Guerzhoy, Jackie Chi Kit Cheung, and François Pitt. 2017. Nifty Assignments. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 695–696. https://doi.org/10.1145/3017680.3028255
- [33] Richard M. Ryan. 1982. Control and information in the intrapersonal sphere: An extension of cognitive evaluation theory. *Journal of personality and social psychology* 43, 3 (1982), 450–461.
- [34] Richard M Ryan and Edward L Deci. 2020. Intrinsic and extrinsic motivation from a self-determination theory perspective: Definitions, theory, practices, and future directions. *Contemporary educational psychology* 61 (2020), 101860.
- [35] Beth Simon, Päivi Kinnunen, Leo Porter, and Dov Zazkis. 2010. Experience Report: CS1 for Majors with Media Computation. In *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education* (Bilkent, Ankara, Turkey) *(ITiCSE '10)*. Association for Computing Machinery, New York, NY, USA, 214–218. https://doi.org/10.1145/1822090.1822151
- [36] Robert H. Sloan and Patrick Troy. 2008. CS 0.5: A Better Approach to Introductory Computer Science for Majors. *SIGCSE Bull.* 40, 1 (mar 2008), 271–275. https://doi.org/10.1145/1352322.1352230
- [37] Stephen D Sorden. 2012. The cognitive theory of multimedia learning. *Handbook of educational theories* 1, 2012 (2012), 1–22.
- [38] John Sweller. 2010. Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational psychology review* 22 (2010), 123–138.
- [39] John Sweller, Jeroen JG van Merriënboer, and Fred Paas. 2019. Cognitive architecture and instructional design: 20 years later. *Educational Psychology Review* 31 (2019), 261–292.
- [40] John Sweller, Jeroen JG Van Merrienboer, and Fred GWC Paas. 1998. Cognitive architecture and instructional design. *Educational psychology review* (1998), 251–296.
- [41] Eunjoon Um, Jan L Plass, Elizabeth O Hayward, Bruce D Homer, et al. 2012. Emotional design in multimedia learning. *Journal of educational psychology* 104, 2 (2012), 485.
- [42] Tammy VanDeGrift. 2020. Applying the Design Process to Life Goals: An Experience Report from a Capstone Course. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. ACM. https://doi.org/10.1145/3328778.3366895