What Attracts Children to Computer Science?

Jessica Marie Yauney, Brigham Young University
jessica.yauney@gmail.com
Scott Bartholomew, Brigham Young University
scottbartholomew@byu.edu

ABSTRACT

Significant effort is being committed internationally to promote computer science (CS) learning in K12 classrooms. Career & Technical Education and Design & Technology courses are two of the most common targets for increased CS instruction. "Hour of Code" (HoC) is one example of the tasks teachers are asked to implement, devoting one hour annually to complete pre-developed CS activities with their students. Researchers collected data from students before and after engaging with an HoC activity and investigated students' motivation, or lack thereof, around coding. Specifically, all students were asked why they would or would not like to learn more about coding following their participation in the HoC activity. Several key findings emerged from the analysis of the student comments. These findings, as well as practical classroom implications, will be shared with an emphasis on trends in student's preconceptions and future interest in CS. Additionally, our examination of students' interest in coding as it relates to "fun" and "job prospects" was explored, as well as students' associated concerns. The role of K12 education as it relates to career preparation is one that can provide greater insight for all technology teachers as they approach CS and other subjects like engineering and design. These trends are aligned with the integration and implementation of the HoC activities in classrooms. Thus, this research has practical significance and can inform future efforts aimed at increasing student interest.

Key Words: Computer Science Education, Hour of Code, Student Perceptions.

1. INTRODUCTION

Educators, policymakers, and other leaders are currently working to expand computer science (CS) education for various reasons. CS is a growing career field where demand exceeds supply (New Data: AP Computer Science Principles Course Bringing More Diverse Set of Students into Computer Science Pipeline, 2020) and where diversity is a priority. "Hour of Code" (HoC) is one of the largest programs created toward this goal with over $90 million USD in funding (Code.org 2018 Annual Report, 2019) and millions of classroom hours devoted per year (Code.org 2018 Annual Report, 2019).

CS education programs and their effectiveness are of interest to Design & Technology (D&T) educators and other leaders for two reasons. First, D&T educators are increasingly being asked to include CS concepts in their courses. Second, many attitudes and motivations that influence students in CS are mirrored in other career focused education programs like D&T.

1.1. Hour of Code

The primary objectives of the Hour of Code (HoC) initiative include the promotion of awareness and interest in the field of CS (Majumdar, 2018). HoC offers a diverse range of one-hour activities explicitly designed to facilitate students' learning in computer science, eliminating the requirement for teachers to possess technical expertise. These activities encompass various themes, difficulty levels, and programming languages. Although HoC serves as a significant catalyst for fostering student interest, it primarily serves as an introductory platform.

The Hour of Code program curates an extensive catalogue of activities that can be utilized by teachers and students alike. These activities can be carried out using computers, tablets, or even without electronic devices. The available activities include self-led tutorials, as well as comprehensive lesson plans that can be incorporated into a teacher's instruction. Each activity is categorized by grade level and difficulty and can be sorted based on popularity or recommendations. Notably, at the time of this research, activities entered around the popular game Minecraft were particularly popular. Four distinct Minecraft-themed tutorials were made available to students as part of this research, allowing students to choose their preferred activity with some students completing a tutorial and proceeding to embark on a second activity and other students only partially finishing a single tutorial.

In-house data collection efforts by HoC creators claimed that engagement in HoC activities leads to enhanced student attitudes towards computer science and heightened interest in the subject (Phillips & Brooks, 2017). Nevertheless, a dearth of evaluation pertaining to the effectiveness of this initiative remains evident. Out of the 64 identified papers focusing on the Hour of Code, merely 12 of them involved research experiments that investigated the outcomes of HoC, with the majority of these studies concentrating on non-K12 audiences (Yauney, Bartholomew & Rich, 2021). Although some research has begun examining the knowledge acquisition facilitated by students' participation in HoC activities, preliminary findings suggest limited development of programming skills (Du & Wimmer, 2015).

1.2. Computer Science and Design& Technology

Computer science and Design & Technology (D&T) education share several fundamental similarities, as both disciplines emphasize problem-solving, creativity, and practical application of knowledge. Firstly, both computer science and D&T involve a problem-solving approach that requires students to identify, analyse, and devise solutions for real-world challenges. In computer science, students are trained to break down complex problems into smaller, manageable components and develop algorithms or programs to address them. Similarly, in D&T, students engage in the design process, where they identify design problems, generate ideas, and create prototypes to solve those problems.

Secondly, both computer science and D&T foster creativity and innovation. In computer science, students are encouraged to think critically and creatively to develop novel solutions and optimize existing systems. They engage in algorithmic thinking and computational creativity to devise efficient and elegant solutions to problems. Similarly, in D&T, students are involved in the design and creation of functional and aesthetically pleasing products. They explore innovative design concepts, materials, and manufacturing techniques to bring their ideas to life. Both disciplines promote a mindset of exploration, experimentation, and the generation of original ideas.

Furthermore, both computer science and D&T emphasize the practical application of knowledge and skills. In computer science, students not only learn theoretical concepts but also apply them in coding and programming projects. They build software applications, develop websites, or work on data analysis tasks to see the direct outcomes of their coding skills. Similarly, in D&T, students engage in hands-on activities, using tools, materials, and processes to create physical artifacts or systems. They learn about structural integrity, mechanisms, and materials properties by designing and constructing prototypes. Both disciplines provide students with opportunities to apply theoretical concepts in practical contexts, enabling them to develop tangible outcomes and gain a deeper understanding of the subject matter.

Computer science and D&T share many of the same educational challenges as they seek to teach complex computational thinking skills, balance career application versus general education and preserve creativity while introducing a complex novel skill. By recognizing these similarities, educators can identify synergies between computer science and D&T education, fostering interdisciplinary connections and promoting learning for students.

1.3. Computer Science Motivation

Educators and other adults have formed many beliefs around what can and should attract students to computer science. Many people believe the increasing prominence of technology in students' lives has heightened the appeal of computer science. The rationale follows that the allure of being able to create, manipulate, and innovate with technology motivates students to explore the field of computer science. Others believe the chance for lucrative career opportunities leads students to computer science. They believe students are aware of the rewarding job prospects and financial stability that can come with a computer science degree. Still others believe the creative and problem-solving aspects of computer science are intrinsically appealing to students as computer science offers a platform for individuals to exercise their creativity and turn innovative ideas into reality. Others believe that, at least for some students, the inclusive and collaborative nature of the computer science community is attractive as computer science encourages teamwork, knowledge sharing, and open-source collaboration.

These ideas while commonly held, are not all supported by research finding. This report of a quasi-experimental study seeks to present the motivations claimed directly by students to support or question these assumptions to improve educators' ability to attract and retain students to CS fields and potentially apply these findings to D&T classrooms.

2. METHODS

2.1. Hour of Code Intervention

Over 2000 7th grade students participated in an HoC activity facilitated by the authors. After collection of associated permission forms, pre-test data and post-test data were successfully matched, and data was successfully analysed from 719 students (72 classes, nineteen 7th grade teachers, fifteen schools, four school districts) in the Western United States. To ensure students were representative of the wider school population, research was done with students enrolled in a required middle school D&T course which covers multiple D&T topics including computer science. Given that the teachers have historically covered CS topics variety of ways, Hour of Code was selected as the intervention for this study because it was the most common tool previously used by teachers. HoC is seen as particularly useful because it requires limited specific CS knowledge and training to facilitate. Many locations, including these districts, rely heavily on D&T teachers without specific CS training to provide computer science instruction (Yauney, 2022). The author who led these activities is a full-time professional software developer, former high school CS teacher and middle and high school CS teacher trainer.

All students were given one class period to complete the pre-survey, HoC activity, and post-survey. This time of these classes ranged from 50 minutes to 70 minutes based on the school schedule. Some students only completed the online HoC activities, while others completed online activities and then participated in a group discussion ranging from 5 to 15 minutes before completing their post-survey.

2.2. Student Survey

In student surveys, administered before and after completing the activities, students were asked on a 5-point Likert scale from strongly disagree to strongly agree if they “would like to learn more about coding in the future.” Students who agreed with the statement were then asked, “Why would you like to learn more coding in the future?” and students who disagreed were asked, “Why would you prefer not to learn more coding in the future?” For the sake of these analyses, only students’ agreement or disagreement with the statement was evaluated because it triggered this difference in secondary questioning. Further analysis of students’ full Likert responses, which is beyond the scope of this paper, would be possible using more advanced statistical models like the McNemar-Bowker Test.

2.3. 2.3. Response Coding

Student responses were coded using thematic coding techniques following recommendations from Saldaña (2013). Initially each of the pre-survey responses was summarized by a series of single word themes. For example, “it looks boring”, “Because coding isn't my thing and I kind of think it’s boring” and “because I just kinda think it is boring” were all coded as “BORING.” Other statements were coded as several topics. For example, “Because it seems boring and takes forever. Even coding a Sphero was difficult and a long process” was coded as “BORING” and “HARD.” After initially coding all responses, themes that were highly similar were collapsed into one category (e.g., “stress,” “anxiety,” “pain” and “headache” were collapsed into one category, “PAIN”. While many themes were unique to either positive or negative responses, some themes were used in both categories. This single-word thematic coding process allowed researchers to draw broad conclusions about students’ overall thoughts. The use of such an open-ended question in a context where students have limited time and are motivated to complete the task quickly does however have limits on its validity given that some students engaged in inauthentic behaviours including copying the responses of their peers. Researchers heard students ask peers for their answer and then copy it but chose not to interfere with this behaviour. Additionally, while researchers acknowledge the potential for deeper qualitative analysis, this was not completed as part of this research effort.

3. RESULTS

Understanding students’ initial motivation before interacting with the HoC activity provides valuable information regarding students’ independent motivations and will be discussed first. The shift in student responses following engagement in the activity is then presented. This analysis both provides information about the impact of this activity and the potential for similar activities to impact student motivation.

3.1. Initial Thoughts

Initially 169 students (24%) expressed interest in learning more coding in the future and 540 students (76%) expressed a lack of such interest. All categories are presented in Table 1. Sadly, several types of responses cannot be interpreted. Some students answered YES, NO, or MAYBE; others said they did not know (IDK); others explained they had answered the previous question WRONG. The five most common explanations for student interest in order were that it was fun (FUN), provided a good “JOB” opportunity, allowed for the creation and playing of games (GAME), allows for the building of products (BUILD), and that it is “INTERESTING”. The five most common explanations for lack of interest are stronger interest in an “ALTERNATIVE” skill, not finding coding interesting (NOT INTERESTING), believing it is “HARD” or “BORING”, and disliking working with a “COMPUTER”. Other students expressed motivation or lack of motivation based on the impact coding would have in the “FUTURE”, its usefulness (USEFUL, USELESS, HELP, NOT NEED), applications like robotics (ROBOT), applications (APP) and websites (WEBSITE), specific programming languages (LANGUAGE), and prior knowledge (ALREADY, NO EXPERIENCE) and successes (GOOD, NOT GOOD). Surprisingly some students provided difficulty (HARD) as an explanation for their interest and ease (EASY) as an explanation for lack of interest. Other explanations given for interest were a general interest in learning (LEARN), a desire to be “SMART”, a belief coding is “COOL”, the positive influence of a “MENTOR”, and the potential to make “MONEY”. Other explanations for lack of interest include the “TIME” requirement to learn, a lack of desire for a coding job (NOT JOB), physical “PAIN”, a preference for outdoor environments (NATURE), a dislike of “MATH”, or a belief that other jobs pay more (POOR PAY). Of additional note is the explanation given by some students that coding either matched or did not match their self-image (IDENTITY).

Table 1:
Number of Student Comments in Each Category of Explanation in Pre-Surveys

Why?			**Why Not?**		
Category	**Number of Comments**	**Percentage**	**Category**	**Number of Comments**	**Percentage**
FUN	44	26%	ALTERNATIVE	108	20%
JOB	32	19%	NOT INTERESTING	93	17%
GAME	22	13%	HARD	70	13%
YES	20	12%	BORING	63	12%
BUILD	14	8%	NO	51	9%
INTERESTING	11	7%	COMPUTER	45	8%
FUTURE	10	6%	NOT FUN	31	6%
USEFUL	9	5%	NOT GOOD	26	5%
LEARN	9	5%	USELESS	25	5%
ROBOT	4	2%	YES	20	4%
SMART	4	2%	TIME	20	4%
MAYBE	3	2%	NOT JOB	17	3%
WEBSITE	3	2%	MAYBE	16	3%
LANGUAGE	3	2%	ALREADY	12	2%
COMPUTER	3	2%	IDENTITY	10	2%
COOL	3	2%	FUN	9	2%
ALREADY	3	2%	PAIN	6	1%
APP	2	1%	IDK	6	1%
HARD	2	1%	NATURE	5	1%
MENTOR	2	1%	WRONG	5	1%
MONEY	1	.6%	MATH	5	1%
HELP	1	.6%	USEFUL	2	.4%
GOOD	1	.6%	INTERESTING	2	.4%
WRONG	1	.6%	NO EXPERIENCE	2	.4%
IDENTITY	1	.6%	NOT NEED	2	.4%
			EASY	1	.2%
			GAME	1	.2%
			JOB	1	.2%
			WEBSITE	1	.2%
			APP	1	.2%
			POOR PAY	1	.2%

3.2. Change of Opinion

In the post-survey, 174 students (27%) expressed interest in learning more coding in the future and 471 (73%) students expressed a lack of such interest (See Tables 2 and 3). While more students responded on the pre-survey than on the post-survey (the same number of student responses were recorded in the pre- and post- surveys, but more students chose to skip the question on the post-test), a higher percentage of students expressed interest in continued learning after the activity. Only student responses that did not skip the question are presented. Using a chi-squared test, a p-value < .003 was calculated for the observed responses on the post-survey, suggesting that the activity had a statistically significant effect.

Table 2:
Number of Student Positive and Negative Responses

	PRE	**POST**	**TOTAL**
YES	169	174	343
NO	540	471	1011
TOTAL	709	645	1354

Table 3:
Percent of Student Positive and Negative Responses

	PRE	**POST**
YES	23.84%	26.98%
NO	76.16%	73.02%

Also, of interest were shifts in student explanations as seen in Table 4. While more students expressed interest in continued learning after the activity, the only explanations that increased in popularity were FUN, BUILD, INTERESTING, COOL, HARD, MONEY, and HELP. Fewer negative responses were given in the post-survey than the pre-survey, so the majority of explanations decreased slightly. However, several categories dropped more than expected with fewer students claiming they were NOT GOOD at coding. Additionally, several responses were more common than before with a larger number of students claiming coding was NOT FUN, they were not interested in a coding job (NOT JOB), it did not match their IDENTITY, or they do not like MATH.

Table 4:
Number of Student Comments in Each Category in Pre- and Post-Survey

Why?			**Why Not?**		
Category	**Pre-Survey**	**Post-Survey**	**Category**	**Pre-Survey**	**Post-Survey**
FUN	44 (26%)	69 (40%)	ALTERNATIVE	108 (20%)	74 (16%)
JOB	32 (19%)	25 (14%)	NOT INTERESTING	93 (17%)	54 (11%)
GAME	22 (13%)	17 (10%)	HARD	70 (13%)	55 (12%)
YES	20 (12%)	19 (11%)	BORING	63 (12%)	58 (12%)
BUILD	14 (8%)	17 (10%)	NO	51 (9%)	15 (3%)
INTERESTING	11 (7%)	16 (9%)	COMPUTER	45 (8%)	20 (4%)
FUTURE	10 (6%)	6 (3%)	NOT FUN	31 (6%)	46 (10%)
USEFUL	9 (5%)	5 (3%)	NOT GOOD	26 (5%)	0 (0%)
LEARN	9 (5%)	8 (5%)	USELESS	25 (5%)	12 (3%)
ROBOT	4 (2%)	1 (.6%)	YES	20 (4%)	10 (2%)
SMART	4 (2%)	3 (2%)	TIME	20 (4%)	14 (3%)
MAYBE	3 (2%)	0 (0%)	NOT JOB	17 (3%)	33 (7%)
WEBSITE	3 (2%)	2 (1%)	MAYBE	16 (3%)	5 (1%)
LANGUAGE	3 (2%)	0 (0%)	ALREADY	12 (2%)	0 (0%)
COMPUTER	3 (2%)	3 (1%)	IDENTITY	10 (2%)	19 (4%)
COOL	3 (2%)	4 (2%)	FUN	9 (2%)	7 (1%)
ALREADY	3 (2%)	1 (.6%)	PAIN	6 (1%)	4 (1%)
APP	2 (1%)	0 (0%)	IDK	6 (1%)	6 (1%)
HARD	2 (1%)	4 (2%)	WRONG	5 (1%)	9 (2%)
MENTOR	2 (1%)	0 (0%)	NATURE	5 (1%)	0 (0%)
MONEY	1 (.6%)	3 (2%)	MATH	5 (1%)	7 (1%)
HELP	1 (.6%)	7 (4%)	USEFUL	2 (.4%)	3 (.6%)
GOOD	1 (.6%)	0 (0%)	INTERESTING	2 (.4%)	1 (.2%)
WRONG	1 (.6%)	1 (.6%)	NO EXPERIENCE	2 (.4%)	0 (0%)
IDENTITY	1 (.6%)	0 (0%)	NOT NEED	2 (.4%)	0 (0%)
			EASY	1 (.2%)	0 (0%)
			JOB	1 (.2%)	0 (0%)
			GAME	1 (.2%)	0 (0%)
			WEBSITE	1 (.2%)	0 (0%)
			APP	1 (.2%)	0 (0%)
			POOR PAY	1 (.2%)	1 (.2%)

4. DISCUSSION

Some of the student responses match generally held assumptions; many students were interested in the creative power of CS and the potential career opportunities, with jobs and building each being one of the top five responses. However, the intrinsic enjoyment or fun expresses by students was not considered as often. Additionally, no students referred to the collaborative nature of CS. While the activity's "fun" nature and lack of collaboration may explain these deviations from expectation in the post-survey, their presence and absence in the pre-survey suggests that some assumptions may not be entirely correct – at least among this group of students.

While considered less often, the reasons students were not interested in CS may also provide insight into ways to better support students. While some reasoning, like an alternate preference is expected and reasonable, some provided explanations also showed student misconceptions, like a lack of use and poor pay. These concerns could potentially be addressed to increase student interest and engagement.

Some explanations illustrated potential contradictions as more students expressed a lack of interest in learning because they were not interested in a job than students who were interested because of job prospects on the post-survey. This may suggest a value in deemphasizing the job potential in CS and D&T fields. The potential motivating power of collaborative work paired with the lack of student responses about collaboration, may point to the power of increasing collaboration to increase motivation or could suggest that collaboration is not as motivating as we believe.

Some of the shifts from the pre- survey to the post- survey raise questions about the accuracy of how potential careers are presented in the classroom. While an increase in students' beliefs that coding is fun was positive, it may also give students unreasonable expectations as video games are engaging but not representative of computer science generally, or a career in computer science specifically.

Overall, an analysis of students' reasoning for continued engagement in coding and CS learning provided additional insight into potential ways for attracting students to computer science. Activities like HoC have the power to influence students' motivation and thus it is important to consider how activities will influence motivation. As we better understand how motivation functions in CS, it is possible that these insights can be transferred to D&T classrooms due to the significant similarities in the courses.

5. REFERENCES

Code.org 2018 Annual Report. Code.org. (2019, February 12). https://code.org/about/2018.

Du, J & Wimmer, H. (2015). Impact of Hour of Code: A Five-Year Study. *AMCIS.*

Majumdar, A. (2018). The Hour of Code: An initiative to break the barriers of coding. *XRDS: Crossroads, The ACM Magazine for Students*, 24(3), 12–13. https://doi.org/10.1145/3186711

New Data: AP Computer Science Principles Course Bringing More Diverse Set of Students Into Computer Science Pipeline. (2020, December 9). Newsroom; College Board. https://newsroom.collegeboard.org/new-data-ap-csp-course-bringing-more-diverse-set-students-computer-science-pipeline

Phillips, R. S., & Brooks, B. P. C. (2017, January). The Hour of Code: Impact on Attitudes Towards and Self-Efficacy with Computer Science. Code.org. https://code.org/files/HourOfCodeImpactStudy_Jan2017.pdf.

Saldaña Johnny. (2013). *The coding manual for qualitative researchers*. SAGE.

Yauney, J., Bartholomew, S. R., & Rich, P. (2021). A systematic review of "Hour of Code" Research. *Computer Science Education*, 1–33. https://doi.org/10.1080/08993408.2021.2022362

Yauney, J. M. (2022). K-12 CS Teacher Licensing in the US. In *SIGCSE 2022: The 53rd ACM Technical Symposium on Computer Science Education*. Providence, Rhode Island; Association for Computing Machinery. Retrieved April 28, 2022, from https://doi-org.erl.lib.byu.edu/10.1145/3478432.3499202.