The Integration of Computational Thinking Activities within K-12

Methods, Research, and Projects to Facilitate Interest in Computer Science in Post-Secondary Education

Parima Sahbai
Department of Computer Science
University of Virginia
Charlottesville, VA
ps4ww@virginia.edu

ABSTRACT

Computational Thinking is a method of problem solving which draws heavily upon computer science, but can be applied to many other disciplines. Computational thinking has been proposed as a solution to address the minority gap in secondary-education, through sparking interest in computer science early-on in K-12 education. I will discuss ways to integrate computational thinking, and a project that presents a method to do so in this paper.

KEYWORDS

Computational Thinking, Algorithmic Thinking, Jeanette Wing, CT, Abstraction, Pattern Recognition, Decomposition, Computer Science, Education

1 Introduction

Jeanette Wing is a professor of computer science and the director of the Data Science Institute at Columbia University. She received her Undergraduate and PhD degrees from MIT in 1979 – 1983 and is a strong advocate for minorities in Computer Science. The Term computational thinking was coined by Wing in 2006 in a paper by the same name and describes thinking and approaches to solving difficult and complex problems through a systematic process which includes gathering data, abstraction and using a computer or technology to analyze, automate, understand, and draw conclusions from said information.

In her paper Wing strongly argues to establish the importance of what she considers computational thinking. She reasons that the main purpose of computational thinking is to solve problems that we are unable to tackle as humans through computers. However, she is quick to point out that the relationship between computers and humans is codependent in a manner in which a computer, although computationally "perfect" is unable to use information, data, and results creatively and effectively in the same way which a human being can reason. Humans make up for what computers cannot do and vice versa.

Wing argues that computational thinking is a way of thinking and a fundamental skill for everyone which simply draws upon computer science concepts but is not just limited to, and for computer scientists. In her words, "computational thinking involves solving problems, designing, systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science." [1]

In short, computational thinking is a heuristic skill and method of learning that goes beyond programming. It is being able to plan ahead, learn from data, and use available tools like computers and technology to solve problems in a clever way that is inherently algorithmic; computational thinking takes into consideration future problems, worst case scenarios, optimization, and strategical thinking using four main components:

- Algorithmic Design
- Decomposition
- Pattern Recognition
- Abstraction

1.1.1 Algorithmic Design. Algorithmic Design, or thinking is "...a method to solve a problem that consists of exactly defined instructions" [2]. It is a method og thought which involves breaking down problems into steps in order to solve them logically. Algorithmic Design is a key component of computational thinking which embodies many of its' key ideas such as the ability to analyze and specify a problem, take logical and basic steps to provide an adequate solution, devise a correct algorithm, and consider edge cases, and optimization.

1.2 Decomposition. Decomposition defines the ability to break down large and seemingly difficult to solve problems into smaller and more manageable ones. Often times, this breakdown is used to analyze and provide a better solution to the more complex underlying problem.

1.3 Pattern Recognition. Pattern recognition is the capacity to recognize patterns within a problem and use discovered patterns to propose a solution, or predict an outcome which will also aid in providing a better solution if needed. This ability is considered an innate ability by many, but one which can be improved on upon proper education to harness its powers towards computational thinking [4]. Pattern recognition is a central element in Machine Learning where one is able to analyze datasets with results, learn patterns, and predict future results based on known patterns.

1.4 Abstraction. Abstraction is removing "unnecessary" details irrelevant to the big picture of a problem to more easily detect, and analyze said problem free from distractions. Abstraction can be a simulated, simpler model of some problem, or defining relationships between available elements in a problem, or simply clearing out "distractions". Abstraction is another central part of Machine Learning and computer science. For example, cleaning a dataset, and breaking down and abstracting a problem using a model often makes problems easier to solve, and allows computers to detect important patterns more easily.

As it is a method of thinking, many argue that computational thinking must be taught as early as possible to develop students into better problem solvers. Furthermore, early education in computational thinking can spark interest in computer science, a field with a major gap between white men, and minorities. Some of these minorities include women, African Americans, Latinx, Asians, and other minority races. I will be discussing methods, research, and other efforts that have been made to improve computational thinking in general education, as well as the efforts to close the computer science gap between minorities and spark interest in the major early on in this paper. I will also be conducting a small experiment to assess opinions of the general public on computational thinking education and the effectiveness of activities which are aimed to teach computational thinking.

2 Problem

Computer science is a fairly new major with growing interest over the past few years. However, a glaring issue lies within the demographics of the major. According to the National Center for Women & Technology, 57% of women earn all bachelor degrees within the United States, but occupy only 18% of all computer, and information science degrees. Moreover, according to the National Science Foundation, Bachelors computer science degrees awarded to women has decreased by 10% between 2000-2009 and has retained itself at a consistent 18% in 2015, down from 28% in 2000 **(Figure 1)**.

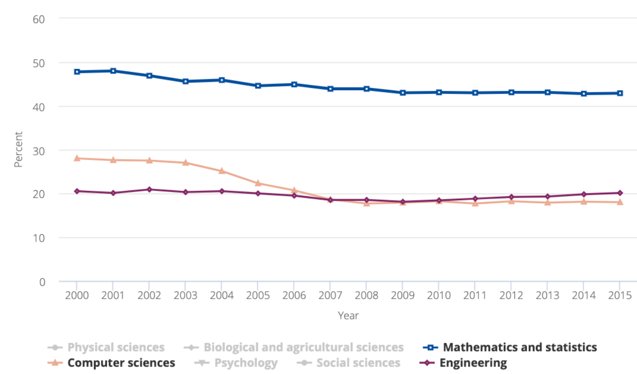

Figure 1: Bachelor's Degrees awarded to Women in Science and Engineering, Computer Science (orange) with a 10% drop since 2000 vs. 2015. [5]

Other minorities suffer similar complications and low rates with a national average of 17% of the awarded computer science degrees to Latinx students and 13% to black students (the National Center for Education Statistics).

Computational thinking provides a plausible solution to sparking interest in computer science and building confidence in minorities by teaching students how to "think like a computer scientist" and become better problem solvers through other coursework. The hope is that computational thinking will provide an alley way to make STEM, specifically, computer science less intimidating, and more appealing to students. Nevertheless, the *approach* to the integration of computational thinking is a place where much of the research lays.

3 Related Work

There has been a plethora of work done to research the effectiveness of computational thinking in the development and growth of student learning patterns and interest in computer science. Below I will describe some of which work inspired my project.

Settle, in her paper titled "Infusing Computational Thinking into the Middle and High-School Curriculum" suggest that "Having outsiders inject computational thinking into an already overburdened curriculum is not a formula for success and has been suggested that inviting teachers into the process is a promising technique" [6]. Settle suggests teaching teachers in a retreat-like seminar, and goes on to propose ways in which CT can be integrated into the following subjects in middle, and high school: computer science, Latin, graphic arts, and history. Some of her work included, a year-long monitoring of CT activities in history classes with goals such as: Improving students note taking, active listening, and recall skills, including learning to recognize the difference between homework and studying. Settle conducted her experimental activity in history class by introducing the idea of hash tagging, and by asking students to choose the term that most represented what they were studying. The hashtags were meant to serve as a tool to help students retrieve information more easily from their studies. Settle reported increased enthusiasm in the classroom by the students, and a change in how teachers teach due to the professional development workshop she taught as well.

Similar to Settle, Cho, Pauca, Johnson and James report work in computational thinking education in their paper "Computational Thinking for the Rest of Us: A Liberal Arts Approach to Engaging Middle and High School Teachers with Computer Science Students" where they conducted a workshop that would satisfy a broad distribution of teaching disciplines at once. The workshop consisted of 35 teachers from diverse backgrounds and spanned two days within two years. In the workshops, they developed short presentations which highlighted key concepts of computational thinking along with related hands-on interactive activities for teachers. Within the two years, they conducted pre, and post surveys and the results were promising. Teachers reported that many of the misconceptions about computer science they once had, had changed. For instance, "Computer science jobs are often solitary and involve very little social interactions with an average change" in the survey, saw a change of -0.7 after the workshops. A survey of the students of these teachers showed that students gained valuable experience and found the changes computational thinking brought to the classroom instruction to be worthy. [8]

Irene Lee, Fred Martin, and Katie Apone's approach to developing computational thinking in K-8 included a set of activities within regularly offered courses which integrate and embody computational thinking. Some of these activities included puzzles and independent tasks within computer "games". One example of this approach is creating a story telling tool program in which a student is required to tell a story in a short animation and within the scope of the given tool, satisfying the idea of abstraction within computational thinking. Although they did not conduct any "hard" experiments, they reported seeing an increase in confidence within the students in regards to their studies [7]. Their simulation is what inspired me to conduct my experiment with a very similar "activity" which I will describe in the following section.

4 Method

Our initial approach to computational thinking in the classroom was conducting a generalized workshop similar to the work done by Cho and others. We created a presentation geared towards a generalized group of student-teachers at the University of Virginia where we present the basics of computational thinking, its importance, and demonstrate hands-on activities to conduct in the classroom. However, due to time constraints, issues with hosting, and time to conduct experiments, the workshop turned out to be out of the scope of this course project.

Taking a different approach, inspired by the "in-class computer activities" approach mentioned briefly before, I decided to develop a program which embodies an activity which nurtures computational thinking within an Art classroom. The application was also inspired by the method suggested by CodeNC [9]. The project was designed with the following steps in mind. After explaining the background behind abstract art:

1. Show examples of good abstract art
2. Show participants a photographic image and ask them to quietly think about the patterns they see within it
3. Ask participants to recreate the image using simple shapes and patterns to create an abstract version of it

A small art program named "Artistic Groove" (**Figure 2)** which follows the protocols above was developed in Unity, using C#. In the program, participants are presented with a few shapes which are available in six different colors, and are given a canvas to recreate the reference picture presented on the bottom of the screen. Participants are able to turn the shapes, change their color, delete them, and place them wherever they see fit.

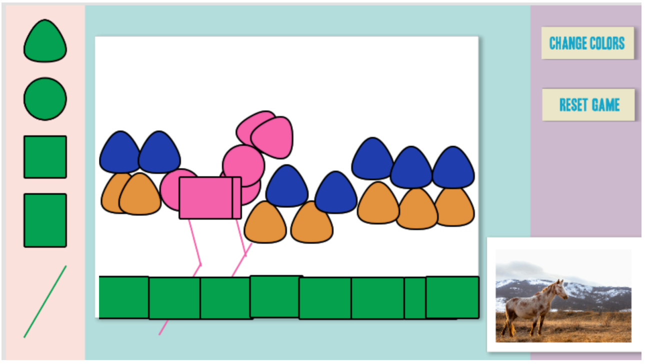

Figure 2: Artistic Groove, Canvas Activity

5 Experiments and Results

Due to limited time and resources, a volunteer-based survey was hosted online which introduced computational thinking, and used the activity I created as a supplement to further explain the main concept behind CT. The results from 24 participants are summarized below.

Presurvey:

- 58.3% of participants reported not knowing what computational thinking is, with 25% unsure.
- 62.5% of participants thought that computational thinking was important to learn, with 29.2% reporting "Maybe", and with 8.2% selecting "No".

Post activity:

- 87.5% of participants reported understanding what computational thinking is better after going through the activity, versus 4.8% answering "No", and 8.3% answering "Maybe".
- 66% of participants thought computational learning to be important to learn vs. 29.2% reporting "maybe".
- 75% of people agreed that the activity does a good job of conveying some properties of computational thinking, with 20% reporting "Maybe".
- 66.7% agreed that they learned something from this activity, with the remaining responses reporting "Maybe".
- 37.5% of Participants thought this activity would be suitable for children aged 10-15, 33.3% said ages 5-10, and 25% thought that it was suitable for all ages.
- Participants believed that the activity embodies the abstract, and the decomposition properties of computational thinking the most **(Figure 3).**

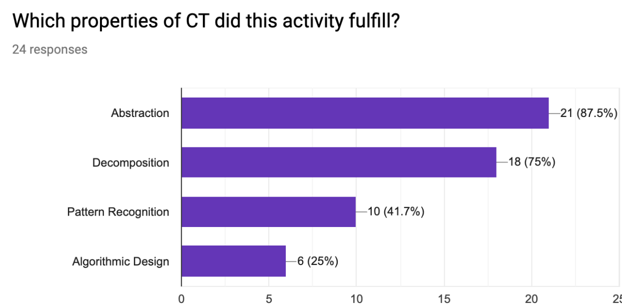

Figure 3. Participants agreed that the activity fulfilled Abstraction with 87.5%, Decomposition with 75% and Pattern recognition with a surprisingly low score of 41.7%

6 Discussion

Overall the activity seems to do a relatively good job of conveying the ideas of computational thinking. We saw an increase in participants' responses when questioned about whether they knew what the term meant from 16.7% to 71.5%, along with a 10% increase in participants who thought computational thinking was an important skill to learn. 66.7% of participants were able to answer that the activity taught them something confidently, which is great indicator that activities like Artistic Groove are on the right track in computer science, and computational thinking education. The activity was simple enough that most participants agreed that students aged K-12 would be able to enjoy, and learn something from it which is the target demographic that the initial workshop was trying to reach.
On the other hand, there were some minor bugs and issues with the program which made it difficult for some users to recreate certain shapes. Many people commented that a lack of directly controllable layers made the program difficult to use and reported that instructions should be provided during the game instead. Moreover, I was surprised to see that pattern recognition, which I thought to be the most important element that the program embodied came in 3rd place, behind abstraction and decomposition. Although I cannot guarantee the reasoning behind this phenomenon, it is probable that the program did not provide enough support, or suitable pictures for pattern recognition to take a more central role.

Overall, I believe that this activity embodies the foundations of computational thinking well, and does a good job of conveying and simplifying concepts. There a few technical and software improvement areas that need to be worked on but the essence of the game seems to step in the right direction.

7 Future work

Future work in terms of the project may include pushing an update which fixes bugs and inconveniences as mentioned in the section above. Some of these bugs and improvements which include:

- Allowing users to directly control object layers
- Changing a button which seems to malfunction randomly
- Providing a transparent backdrop on the main canvas to guide users further, since the reference image is small
- Allowing users to save their images
- Providing more intuitive controls for object rotation

In regards to future work with computational thinking in K-12, I hope to create more hands-on online software for other subjects such as history, and psychology. Furthermore, I would like to see the workshop become a reality, and aid in the conduction of a year-long monitoring program to see the effects that educating teachers on computational thinking has, on students.

8 Conclusion

Computational thinking is a way of thinking which involves algorithmic design, decomposition, abstraction, and pattern recognition. Computational thinking revolves heavily on the principles of computer science but can be applied to a variety of subjects as it is a method of thinking and problem solving. The aim of computational thinking in my project has been to increase interest in computer science amongst minorities such as women, African Americans, and Latinx, to close the worrisome gap between bachelor's computer science recipients in the United States. In order to address this gap, we decided to focus on teachers and K-12 education, integrating computational thinking within a variety of subjects which does not necessarily directly involve STEM. One of the methods in which we used included preparations for a workshop. nonetheless, after complications, our main focus shifted onto designing a piece of game-like software targeted towards K-12 which promotes computational thinking ideas. We ran a quick, public survey to gather opinions, and perceptions on computational thinking and found that most people, although initially unaware of what computational thinking is, understood its concepts and were able to connect the activity to its underlying building blocks indicating that hands-on activities such as the one we created, may be a step in the right direction.

REFERENCES

[1] Wing, Jeanette M. (2006). "Computational thinking" (PDF). Communications of the ACM. 49 (3): 33. doi:10.1145/1118178.1118215.

[2] Snyder, L. Interview by F. Olsen "Computer Scientist Says all Students Should Learn to Think 'Algorithmically'," The Chronicle of Higher Education, May 5, 2000: http://chronicle.com/free/2000/03/2000032201t.htm/

[3] Gerald Futschek. 2006. Algorithmic Thinking: The Key for Understanding Computer Science. Informatics Education – The Bridge between Using and Understanding Computers Lecture Notes in Computer Science (2006), 159–168. DOI:http://dx.doi.org/10.1007/11915355_15

[4] Kyu Han Koh, Ashok Basawapatna, Vicki Bennett, and Alexander Repenning. 2010. Towards the Automatic Recognition of Computational Thinking for Adaptive Visual Language Learning. 2010 IEEE Symposium on Visual Languages and Human-Centric Computing (2010). DOI:http://dx.doi.org/10.1109/vlhcc.2010.17

[5] NSF | Higher Education in Science and Engineering. Retrieved May 9, 2019 from https://www.nsf.gov/statistics/2018/nsb20181/report/sections/higher-education-in-science-and-engineering/undergraduate-education-enrollment-and-degrees-in-the-united-states

[6] Settle, Amber. Infusing computational thinking into the middle- and high ... Retrieved May 9, 2019 from http://explore.appinventor.mit.edu/sites/all/files/teachingappcreation/unit2/p22-settle.pdf

Conference Short Name:WOODSTOCK'18

[7] Lee Irene, Martin Fred, and Apone Katie. Integrating computational thinking across the K--8 curriculum. Retrieved May 9, 2019 from https://dl.acm.org/citation.cfm?id=2684721.2684736

[8] Cho, Samuel S., Pauca V.Paul, Johnson Denise,and Yu'Vonne James. Computational Thinking for the Rest of Us: A Liberal Arts ... Retrieved May 9, 2019 from https://users.wfu.edu/~choss/docs/papers/27.pdf

[9] Nguyen, Rich. 2018. CodeNC - Computational Thinking in the Arts. (October 2018). Retrieved May 9, 2019 from https://www.youtube.com/watch?v=iq32rm6AVbA