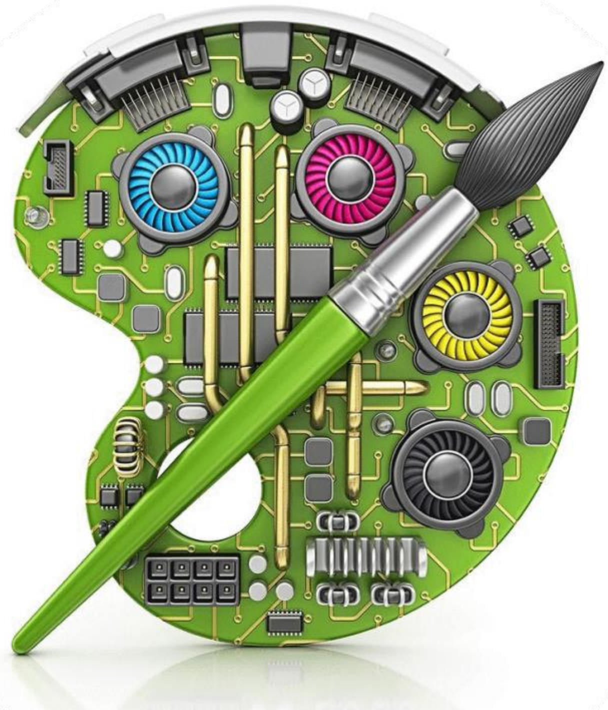
Using Arts and Crafts to teach Computer Science at the YMCA After-School Program



by
Metasebia Dejene
Matthew Figueroa
Neer Jariwala

Using Arts and Crafts to Teach Computer Science at the YMCA After-School Program

An Interactive Qualifying Project
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the
Degree of Bachelor of Science

by
Metasebia Dejene
Matthew Figueroa
Neer Jariwala

Date:
12/12/2019

Report Submitted to:

Pamela Suprenant
Sarah Levy
Central Community YMCA

Gillian Smith and Katherine Chen
Worcester Polytechnic Institute

Abstract

Computer Science (CS) is a popular, flourishing field. To prepare young students, CS is now being taught in middle school. Our team created a curriculum that delivers CS education using arts and crafts. We crafted a multi-week program that taught CS concepts including pseudo code, debugging, functions, and algorithms using art activities involving navigating a maze and drawing pixel art. We piloted the curriculum with 6th/7th graders at a YMCA after-school program, where we observed the students' perspective, interest, and knowledge of CS. After the pilot program, we analyzed the results to measure the curriculum's effectiveness and found the students better understood CS. The project resulted in a curriculum and set of recommendations for future groups conducting similar projects.

Acknowledgments

We would like to thank Sarah Levy and Pamela Suprenant from the Central Community YMCA branch for working with us through this project. We would not have been able to pilot our program without their support.

Authorship

Section	Primary	Secondary
Abstract	Neer Jariwala	
Executive Summary	Matthew Figueroa	Neer Jariwala
Introduction	Neer Jariwala	
Background	Matthew Figueroa	Neer Jariwala, Metasebia Dejene
Curriculum	Metasebia Dejene	
Methodology	Matthew Figueroa	Neer Jariwala, Metasebia Dejene
Results and Analysis	Neer Jariwala	
Discussions and Recommendations	Matthew Figueroa, Metasebia Dejene	Neer Jariwala
Conclusion	Neer Jariwala	

Executive Summary

Through our project, our group created a set of activities in the form of a curriculum that taught a young audience basic computer science concepts in an informal classroom setting using a secondary tool, arts and crafts. We wanted our audience to leave our program understanding more about computer science and a potential interest to explore the field as a possible future study or career path.

Our background research focused heavily on teaching methods and previous programs that have been run by other schools to teach computer science. The problem that this project addresses is not only a general lack of understanding of what computer science is, but also a lack of interest by young kids. The lack of interest, we hypothesized, is formed from how the topic is taught in schools and afterschool programs. Computer Science is generally taught in a formal setting using lectures and projects, but there is a lack of connection within those projects and the students' lives. In a visit with the STEM Education Center at WPI, it was suggested that in order to make an effective curriculum of activities, the activities and topics need to be tied into the students' lives. This is done by rooting the curriculum in something they already understand or something they can physically see. Our goal was to create a curriculum and analyze the impressions it made on a younger audience, specifically 6th and 7th graders. The second goal was to provide recommendations for future projects with goals like this project.

We piloted the program with a group of 6th and 7th graders at the YMCA in Worcester, MA. The YMCA provided extra staff to supervise the program as well as a computer lab. We garnered the student's interest and understanding by using surveys and a drawing activity, in which they drew what they believed a computer scientist was. The results were analyzed to assess whether the curriculum was effective.

The first deliverable we created was a curriculum in computer science that is based in the arts. The curriculum consisted of multiple after-school activities that ran an hour in length, twice a week, for three weeks total. Each day had a planned activity that used arts and crafts to teach some basic computer science concepts. These concepts included pseudo code, debugging, functions, and algorithms. The activities in the curriculum were called snowflakes, maze game, and pixel art. The snowflakes activity was designed to help the students understand the importance of clear instructions in coding. The maze game was focused on teaching the students debugging and writing clear and concise instructions, like coding. The pixel art activity touched upon cartesian coordinates and how to create and follow pseudo code.

Our second deliverable was a set of recommendations for future project groups that will be undertaking a similar project. The recommendations were made based off the experience of our group during this project.

- Do not expect everything to go as planned
- Know the environment where your program will take place
- Meet with your student before the program
- Build the curriculum to be adaptable to change
- Narrow down your scope as soon as possible to avoid unnecessary work
- Have a backup plan for various situations
- Make use of your resources as early as possible
    - Resources include the STEM Education Center and the Pre-Collegiate Outreach Program Office.

Our results overall were promising. Data was collected using a survey and a drawing activity. The survey investigated three categories: computer knowledge/interest, understanding computer science, and subject affinity. Subject affinity refers to how much the students enjoy specific subjects, such as math and art, and if they can possibly understand that computer science can be found in other fields of study as well. Before our program, the students self-reported that they were not very interested in computer science and did not know much about it. After going through our activities and program, they self-reported they understood some basic computer science concepts and had an increased interest in the field. For example, we saw the number of students self-reporting they somewhat or strongly agree they understood what Computer Science was double from two before our program and 4 after our program. In addition, we saw a decrease in the number of student's self-reporting that they did not have an interest in computers.

Our developed curriculum conveys computer science concepts using arts and crafts to a young audience in an informal setting. Although our group faced various obstacles along the way, we used each visit as an opportunity to adapt our program to counteract these obstacles. The data and our experience during the visits lead our group to agree that our goal was mostly reached. Due to the various problems we faced along the way, we were not able to complete everything as planned, but were still able to complete most of our planned curriculum. With our recommendations, future project groups will be able to further improve upon our pilot program and continue to educate youth about computer science.

Table of Contents

List of Figures

List of Tables

1. Introduction

Computing, and more specifically computer science is “part of just about everything that touches our lives” [7]. We find computers in cars, homes, and even in the movies we see. Despite this constant interaction with computers, many people do not understand how impressive computers really are. A loss of interest in computer science is especially seen in middle school students [25]. This lack of interest in computer science at younger ages is the area our group chose to address with our project. Our research indicated that multiple programs, schools, and educational standards have addressed this problem. Using their ideas as an inspiration, we created our own method.

Art is one of many fields full of creativity and inspiration. Much like the field of Computer Science, art gives its creators a large range of tools to use and multiple outlets to use them. Both fields give the opportunity to develop and create something interesting and truly amazing. In addition, art and design influence computer science research and development [21]. This idea raises the question: What if we were to use the hands-on aspect found in making art and use it to teach the concepts and skills found in computer science?

Teachers commonly use lecture-based curriculums and homework assignments to teach computer science in classrooms during school, a formal learning environment. The formal form of education, as referred to in this paper, is learning in lectures, doing homework, working individually, taking exams, and having a rigid structure. Alternatively, with after-school programs, the method of teaching is more informal with a larger variety of activities. This informal learning environment usual takes place outside of school hours and often by choice. Conducting activities such as using an Arduino to teach basic programming is one example of one possible after-school activity [11].

The goal of our project was to create a short curriculum that would use the hands-on aspect of arts and crafts to teach basic concepts of computer science to a middle school audience in an informal learning environment. Through our project, we express that computer science is enjoyable and a possible study/career path in the future. We focused on concepts such as functions, algorithms, debugging, and pseudo code. Our audience was a group of students in an afterschool program of the Young Men’s Christian Association (YMCA) in Worcester, MA. We wanted to teach computer science in a manner that did not involve assignments but rather used hands-on activities. Our team used a variety of technology and art activities to build a 3-week program rooted in computer science concepts.

During the initial development of our curriculum, we researched previous programs, games, and activities that had the goal of teaching computer science (Chapter 2). While creating activities and learning objectives, we made sure to align our lessons to current state teaching standards (Chapter 2.3). The curriculum was made to fit in three-weeks with flexible timing and engaging main as well as back-up activities. In order to make sure our curriculum was safe to implement, we had to go through the IRB process and make changes in order to create a curriculum that was worthy of being piloted. In order to measure the impact of our curriculum, we collected and analyzed data from self-assessment activities completed by the YMCA program’s students. The activities were a survey and a drawing activity. This method was used to remove any bias that could occur if our group self-assessed our own program’s success. Our survey results and drawing activity analysis showed an overall increase in the students’ interest in Computer Science and their willingness to learn more about computers. In the end, our group has produced a 3-week curriculum and a set of recommendations for future groups hoping to complete a similar project.

2. Background

The problem addressed in the project is the way computer science is being presented, and the target audience are middle schoolers.

2.1 Computer Science Education

There has been previous research done to address including Computer Science in education. Adams Nager and Robert Atkinson published a journal about improving U.S. computer science education. Only a quarter of high schools have a computer science program in place, but often they lack the principles and concepts that make up computer science. Most of the focus of computer science education is in affluent schools [1]. We focused our computer science education on middle schoolers rather than high schoolers. Our thought process was to expose younger children to computer science and teach them core ideas such as algorithms, debugging, and other methods of computational thinking. In order to teach these concepts, a method had to be chosen. According to Swaid, the use of computational thinking workshops involving hands on activities and utilizing computer science concepts instead of directly using coding to teach is a beneficial method [22]. This method provided a direction for our group to build a curriculum.

In terms of teaching the program and activities, we adopted a co-teaching model. This model refers to the process of having multiple teachers working with the same group of students. Co-teaching is effective for being able to cover content in less time and a greater number of students. It allows for students to have more individualized instruction and develop their skills [14] A high school craft program included the use of Arduinos and sewing them into fabric to make interactable clothing. Interviews done with the students afterwards asked them how they felt about the challenge and if they saw themselves more as computer scientist after the program. The results spoke on how the program widened their perspectives of what computer science can involve [26].

Inspiration for our activities and the structure of the curriculum came from a previous after school program. This program piloted with 52 students over the course of 9 weeks with surveys and reflections to collect data about the impressions of computer science [11]. Our group took inspiration from this program when designing our own curriculum and how we would collect data.

2.2 Inspiration Projects

In researching computer science teaching activities, we started with inspiration projects and focused on expressing computer science work through arts and media.

The current projects in this section are more instruction and acting out code to show w the flow works. One example of these activities is the *Coding a Lego Maze* activity [20]. The activity provides commands for the child to give a character in a maze so that they can "program" their character to move around the maze. The first level of the challenge is understanding the ideas of giving commands and working with a new frame of reference - the frame of reference being their character in the maze. As they work up the levels, they learn the use of loops and conditional statements in order to use fewer commands and be more efficient [20]. More code intensive activities, such as the Lego Mindstorms[1] show the idea that keeping the learning being visual and hands-on was essential. We worked our education concepts around this idea of visual and hands-on learning. The Superhero Coding Game Without A Computer is also a similar activity. Using poster board, tape, and pens to create "code" and obstacles to work around [17].

These activities focus primarily on coding and visualizing. The Arduino has large blinking lights to give information, and the Lego Mindstorm activity focuses on parking a car. The Blink Master is an Arduino project where the child creates a series of blinking lights [19]. The activity breaks into steps that teach children how to code the Arduino. We used the instruction breakdown as inspiration for how we divided our lesson plans into simpler steps. The Autonomous Parking Lego activity is also another coding activity we read into [8]. The children build a Lego car out of small motors and Lego pieces, and they attach it to a Lego Mindstorms. The project comes with pre-made to show examples of the project as well as plenty instruction on what each section does. The provided manual gives problems and shows multiple solutions to each problem. We wanted to follow this model in our maze activities. The handwritten code the students make has multiple solutions for each problem as to how they can write it, including loops and conditionals to automate some control [20].

These upcoming projects are more craft oriented. They're working with cloth, paper, and circuits to create art with computer science. Sew Electric has a design your own bracelet activity in which children can use the LilyTiny[2] computer to create a LED pattern [23]. The computer can be sewn into fabric using conductive thread with a small circuit attached to it. The instructions are quite simple. Another craft-based project made by Left Brain Craft Brain is the Paper Circuit Card. It comes with a template and instructions to create a firefly using LEDs and a simple circuit. The project is meant to expand on creativity with circuits and electricity to connect it to art [4].

These projects provided us with ideas to apply into our curriculum and became activities in them as well that you will see in chapter 3 with curriculum design.

2.3 Turtle Graphics

Turtle Graphics is a type of vector graphics. It uses a cursor on a Cartesian place to create drawings. The cursor, in this case, is displayed as the "turtle", which moves around the screen, leaving a line wherever is has been. Commonly used to create patterns, turtle graphics can create simple designs as well as more complex designs, as seen in Figures 1 and 2.

[1] A platform developed by Lego to allow for the programming of robots. The system uses Lego building blocks and a brick computer to control the motors, sensors, etc. on the robot.

[2] A small arduino with an array of LEDs that can be used as an educational tool. It is a piece of e-textile technology.

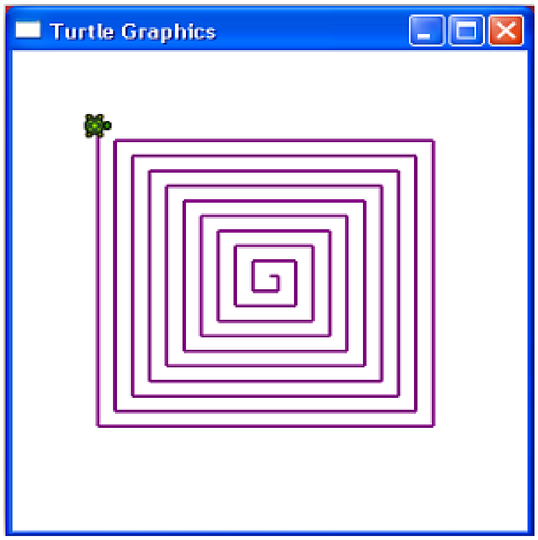


Figure 1: Turtle Graphics Simplistic Example [12]

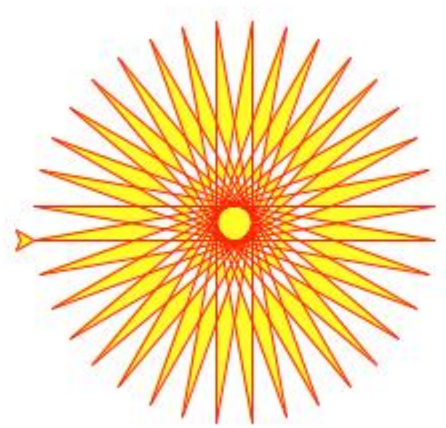


Figure 2: Turtle Graphics Complex Example [5]

The turtle graphics software is a major component of the Logo Programming Language[3]. It is also an available library within the Python Programming language. It comes with simple commands that allow the user to move the cursor in a certain direction by a specified number of pixels. Due to the easy to learn nature of this library, turtle graphics is a great tool for introducing programming to a younger audience, which is part of the goal of this project. In addition, the artistic purpose of the software allows for programming and art to be mixed into a single package.

[3] The Logo Programming Language, a dialect of Lisp, was designed as a tool for learning (Logo Foundation, 2014). It is a high level programming language known for its graphics capabilities.

2.4 YMCA School's Out Program

The group of students initially were sixth graders. The seventh graders that joined our group are from their teen section of the School's Out Program. Students are offered a variety of services from homework help to arts and crafts activities in different rooms. For example, one of the facilities the students used during this program was a computer lab. The program caters to each of the student's needs, considering the diversity of the population. One of the ways this is done is through the staff knowing multiple languages, allowing them to can communicate with a more diverse set of families. This safe and educational environment functions in a way to expose children to new concepts, making it a great location for the first test of our curriculum.

2.5 Curriculum Frameworks

We built our curriculum with help from the STEM Education Center at WPI. To combine STEM education and crafts, Donna Taylor, Assistant Director of Professional Development assisted with two major points from our discussion. The engineering process and tying the education and activities to the student's world. The engineering process can be seen in Figure 3.

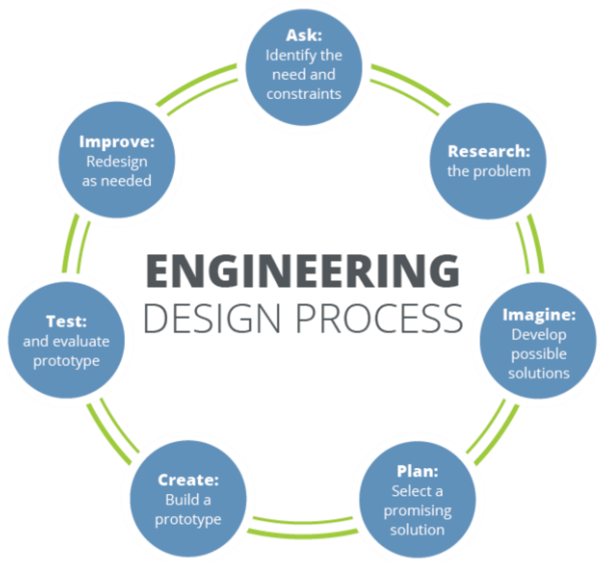


Figure 3: The Engineering Design Process

The projects are meant for the students to create something they can relate to. We described the engineering process visually using the game development process due to interest. We created lessons using the Massachusetts Department of Education's Curriculum Frameworks [18]. The main topics involve understanding how computers represent and manipulate information, analyzing outputs, and debugging to get the desired output, as seen in Table 1.

Table 1: Massachusetts Department of Education Framework Computer Science Algorithms Standards

6-9.CT.b	Algorithms
6-8.CT.b.1	Design solutions that use repetition and conditionals.
6-8.CT.b.2	Use logical reasoning to provide outputs given varying inputs.
6-8.CT.b.3	Individually and collaboratively, decompose a problem and create a sub-solution for each of its parts (e.g., video game, robot obstacle course, making dinner).
6-8.CT.b.4	Recognize that more than one algorithm can solve a given problem.
6-8.CT.b.5	Recognize that boundaries need to be taken into account for an algorithm to produce correct results.

Since we tied arts and crafts into our curriculum, we chose to include arts standards as well. Specifically, theatre and visual arts became our point of focus due to the activities we implemented. One activity involves youth acting out characters in a computer trying to process commands from a user. Our main art activity involves pixel drawing, so our art standards involve working with restrictions and maintaining them. Our chosen theatre and art standards are in Tables 2 and 3.

Table 2: Massachusetts Department of Education Frameworks Theatre Standards

Learning Standards: Theatre	Acting
1.7	Create and sustain a believable character throughout a scripted or improvised scene.
1.12	Describe and analyze, in written and oral form, character's wants, needs, objectives, and personality characteristics.

Table 3: Massachusetts Department of Education Frameworks Visuals Arts Standards

Learning Standards: Art	Methods, Materials, and Techniques
1.6	Create artwork that demonstrates an awareness of the range and purpose of each tool such as pens, brushes, markers, cameras, tools and equipment for printmaking and sculpture, and computers.
1.7	Use appropriate vocabulary related to methods, materials, and techniques students have learned and used in grades PreL-8.

These standards were used as guidelines for our curriculum and helped specify what we wanted to teach (i.e., learning objectives). They're also what Massachusetts' Department of Education deems necessary for the 6th-grade level in these topics. Each of our curriculum activities involved these standards.

3. Curriculum Design

The curriculum is the foundation of the pilot program. To succeed, it had to best-fit the criteria and condition of the setting and the audience. The curriculum had to be flexible enough to fit different timeframes to fit conditions. These include the variability in the winter weather, different obligations the students may have and any other scheduling complications that could occur. To combat time problems, we created three similar timeframes. This included a full three week, two and a half weeks, and two-week curriculum. For each timeframe, we kept the main points of the activities but simplified them to work with smaller allocated time.

When students arrive at their afterschool program, they have already sat in a classroom for several hours. This can cause them to be less attentive and tiresome. Instead of having a curriculum that required the students to learn in a lecture-based type, we made it have a project-based and a hands-on program filled with many activities. Having hands-on activities allows students to get up and engage with the material. It let them experience the lessons firsthand, thus allowing them the opportunity to cultivate their curiosity, creativity, and imagination. Students were able to work in groups, which grew their problem solving, communication, as well as conflict resolution skills. The activities allow them to see different perspectives and answer each other's questions.

The students were in middle school, so we tailored the curriculum towards their grade level. This was to ensure that they would understand and retain the material better. These students also come from diverse backgrounds and different levels of knowledge. The curriculum fits the student's abilities when it comes to using computers as well as any experiences with computer science and/or art. The activities had extra time, if need be, to ensure that all students had the time they need to complete them. Because of the variation in levels of students' knowledge, we made sure to have a variety of content. This was to ensure that those students who finished the material or activities did not have to sit and wait. It was best to keep going and to push the students to maintain their interests as well as gain new knowledge in the subject matter. To keep challenging the students, we used new and exciting material that cultivated their already gained knowledge and skills.

The curriculum also had art and computer science standards incorporated to help with the structure and learning objectives. The art standards adopted for the curriculum were the theater (table 2) and art (table 3) sections. Because we wanted the students energized and up on their feet, the theater aspect was a very important addition. It allowed the students to engage firsthand with and embody the material. The art aspect allowed the students to release their imaginations and express their creativity. For computer science standards, the algorithmic aspect was utilized. The algorithmic aspect allowed the students to learn algorithmic thinking to solve problems. Algorithmic thinking allows students to break down the material, focus, as well as learn how to recognize different patterns. This skill is not only useful and important inside the classroom, but in everyday life as well.

3.1 Main Curriculum Activities

Multiple activities were designed to encompass the standards and goals of the curriculum. These includes the art (Table 2-3) and the CS (Table 1). Certain concepts like algorithms, debugging, functions, pseudocode, Cartesian coordinates and more were incorporated into the activities. The main three activities were the Draw A Computer Scientist, Maze Game and Pixel Art.The description and objectives of all these activities follow.

3.1.1 Draw a Computer Scientist Activity

The drawing activity evaluated the student's thoughts and understandings of computer scientists. It helped gauge students' initial thoughts to see what we can do to improve or add to their impressions. The activity came from the draw-a-scientist-test used to draw out stereotypes about scientists [10].

Students received an 8' by 11 ½' piece of paper with the phrase "Draw what you think a Computer Scientist looks like. Give them a name," (Appendix B). Instead of having him or her, the phrase included the word them. This prevented the phrase from influencing the students to draw a specific gender. The students drew for about 30 minutes with markers, pencils, and pens. Once they finished, we collected the drawings and pulled out common themes seen using an assessment rubric (Chapter 5.4 Table 11). At the end of the pilot program, the students completed this activity again so see if their views changed.

3.1.2 Maze Game

We designed the maze game to teach and enhance student's abilities to write detailed and well-oriented instructions. We also hoped to improve their capabilities in following said instructions (Appendix C). The allotted time for this activity is a total of 50 minutes. The needed materials for this activity. The instructors used the instructions to allow them to guide the students and stay on task. They were to break down the tasks piece by piece as the students go through the process.

During the activity, students split into groups of three. After the students get settled, they got a maze for which the students had to write instructions to guide a person through. Within the groups, the students got assigned to become either a planner, designer, or an executor. Since the maze had many ways of going from start to finish, the planner's task was to choose the path the team will take through the maze. They were to come up with different strategies on how to face the obstacles found in the maze. The designer's job was to move forward with the plan created and to write the instructions that the executors would use to complete the maze. The executors would have to follow the steps to the best of their abilities to go through the maze. Each student, regardless of their assigned roles, had about 15 minutes to complete their tasks. To succeed, each student would have to rely on their group mates.

While the students are completing their tasks, the instructors had to ensure that the students were staying on task and on time. Each part of the activity was done on a step by step basis. Instructors must keep in mind of time adjustments that may be necessary for certain portions of the activity.

3.1.3 Pixel Art

Pixel art is a type of digital art in which each piece is created and edited at the pixel level (Figure 4). They're made using digital software or by hand using a paper grid. During this activity, Students got tasked with designing either an inspired or self-thought pixel art. Students wrote instructions that would allow someone to recreate their work of art. They combined their writing, problem-solving and algorithmic thinking skills learned from different activities. This activity took about 3 hours split into three days (Appendix D).

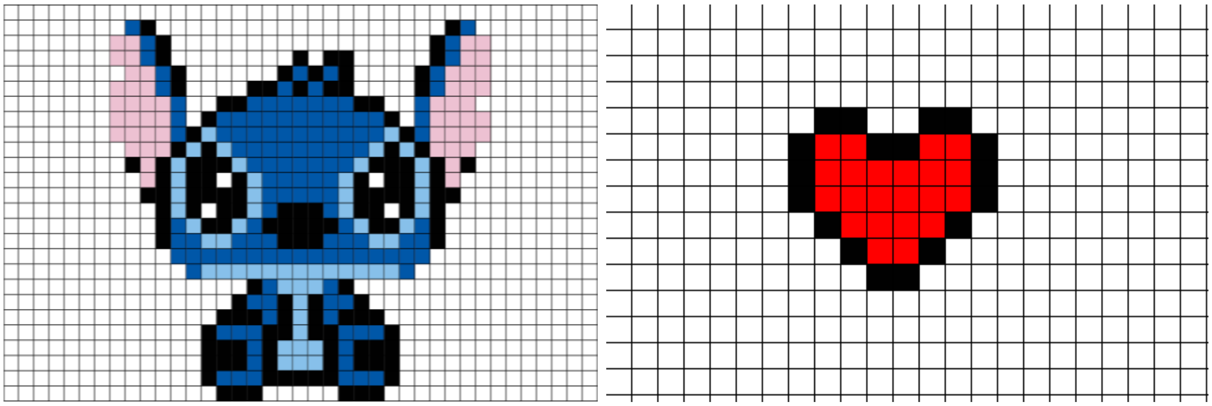


Figure 4: Different Examples of Pixel Art

For the first half, students learned about 2-dimensional cartesian coordinates. This gave them foundations on creating a plot in for their pixel art. The students first turned their worksheet grids into an X, Y plot. From there, the students plotted the sequence of coordinates provided. Since the plots only created half of the pixel art, students wrote the remaining sequence points to finish the symmetrical pixel art. The students exchanged their coordinates and completed the image. Next, the students researched different inspirational patterns and created their patterns. The pieces they create range from a cartoon character to a simple shape, like a heart. Once they had an idea of what they wanted, they created a pixel version and plotted the pattern on the grid paper provided to them.

From there, students learned how to translate their coordinates into pseudo-code. This utilizes the instruction writing skills they learned before the maze game. For example, when a student has the coordinate (1,1), they translated it to move 1 block up and 1 block to the right. If the X coordinate is negative, it meant going left and if positive, they would be going to the right. For the Y coordinate, negative would mean moving down and positive would mean moving up. Between every coordinate plotted, students put "pen down" and "pen up" to mimic the mechanisms of the AxiDraw (Chapter 3.4). In the end, students combined math and computer science to create art.

Instructors kept in mind the variation of background knowledge the students had on cartesian coordinates. They also ensured that the patterns chosen by students were somewhat simple. This helped keep the time frame activity from varying too much. The time allocated also was flexible enough to change depending on the area's students need more time in.

In the end, students combined math and computer science to create art. To give them something to keep at the end of the projects, we turned their pixel art a bitmap using a program called Inkscape (Chapter 4.3). This allowed the Axidraw to make a physical copy of their pixel arts which the students received.

3.2 Back-up Activities

Depending on the level the students are at and how much time it takes them to go through the activities, there might be extra time left over. To prevent the students from becoming disengaged from the program, we made back-up activities. Both activities teach the students about specificity when it comes to writing and following instructions. Make a Snowflake (Chapter 3.2.1) incorporates the art standard (Table 3) by allowing students to create varies types of snowflake art using different materials. The Peanut Butter and Jelly Sandwich (Chapter 3.2.2) touches upon the computer science standard (Table 1) by allowing the students to use algorithmic thinking to solve a problem. The description and objectives of all these back-up activities follow.

3.2.1 Make a Snowflakes

We developed the snowflakes activity to students about the importance of having specific and detail-oriented instructions. In this activity, students got different kinds of instructions on how to create snowflakes. Students had scissors, paper, instructions, and about 25 minutes to complete this activity. To begin, students got split into a control group and an experimental group. The control group got specific and detailed instructions to follow to create a snowflake. The instructions would contain details including how to fold the paper and how to cut the shapes on the folded paper. The experimental group got a set of instructions that are not detail oriented. For example, the non-detail-oriented instruction told students to fold a piece of paper three. A detail-oriented instruction, however, told the students to fold the paper the long way three times making sure the tips of the paper touch before making the crease at each fold.

After the activity is completed, students moved onto comparing the different results they have gotten. This allowed the students to learn the importance of specificity and understanding the perspectives of others. Instructors kept in mind that some students might have a harder time following either instruction. When asked questions, instructors were careful in answering them to not influence the outcome of the activity.

3.2.2 Peanut Butter and Jelly Sandwich

Attention to detail and being able to write clear instructions is an important skill to have. The peanut butter and jelly sandwich activity aimed to improve the students' communication and instruction writing skills. It thought them to be more detail-oriented, attentive, and specific to not only get the right results in life but while coding as well. Without the right lines of code that are specific and concise, the program written could do something unpredicted.

For this activity, students gave detailed instructions on how to make a peanut butter and jelly sandwich. They believed that the instructor was a robot that did not know what a peanut butter and jelly sandwich was and needed instructions on how to make one. The students gave instructions that had items that represent bread, a toaster, a butter knife, peanut butter and jelly on a table. These items were a piece of paper that represents bread as well as jars of peanut butter and jelly and a pen that represents a butter knife. Students had to be very specific and clear when giving the necessary steps. These instructions included moving arms to pick up items, which specific item to pick up, how-to pick-up bread, how to toast bread and more.

When completing this activity, instructors ensure students were being as clear and concise as possible. If the instructions given were vague action items, the outcome of their actions differed from the intention of the students. For example, when the action was "pick up the bread", the instructor, unless taught otherwise, was to say "Error; unknown item". The instructor also asked to define what bread is, to understand the instructions given.

3.3 AxiDraw

The AxiDraw (Figure 5) is a drawing machine that can be used to write, draw and plot using multiple kinds of writing instruments. It uses an algorithm to assess a vector given to replicate different types of simple and complex artwork onto any flat medium. The AxiDraw combines the art and the computer science portions of the program and serves as a real-world application. Since the AxiDraw is a physical device that can be seen in action, it will engage students.

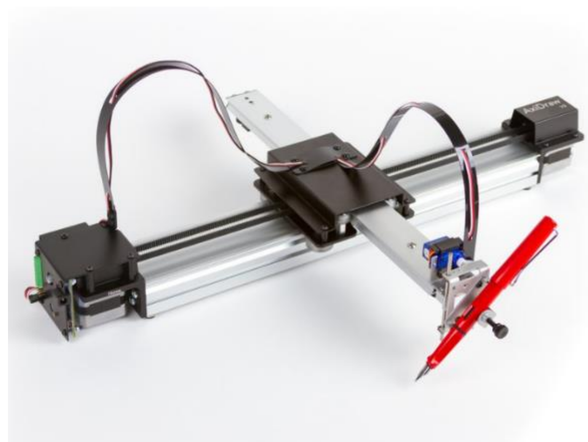


Figure 5: Image of an AxiDraw

At the end of the program, the AxiDraw will be used to replicate the student's Pixel Art so students can have a physical version to take home.

3.4 Planned Curriculum

In the final curriculum, each day was planned carefully with activities, learning objectives and materials needed (Chapter 3.1-3.2). The curriculum timeline was a three-week curriculum that would run for the initial pilot period. Tables 5-9 show the planned-out curriculum broken down and in detail. Table 4-5 shows the introductory period of the curriculum. It includes the activities that will slowly emerge the topics that will be covered during the main activities. Table 6-7 shows the introduction and continuation of the main programs planned. The main activity, pixel art, is broken down into connecting pieces to allow the students to smoothly pick up the content. Table 8-9 shows the continuation of the main programs planned as well as the end of the program. It breaks down the finishing pieces of the pixel art activity as well as the plan for the field trip.

Week 1

Day 1: Introductions and Art Assessment

- **Standard**: Art (Table 3)
- **Activity**: Draw a Computer Scientist
- **Objectives:**
  - Spend time with students to get them comfortable with us
  - Get the students interpretations of what they believe a Computer Scientist looks like

Table 4: Day 1 Breakdown of Planned Curriculum

Breakdown	Brief Description	Materials	Time allotted
Introduction	Instruction and Student introductions with quick icebreaker.	N/A	15 minutes
Draw a Computer Scientist	Students will be given the activity sheet that asks them to draw what they believe a Computer Scientist look like.	Activity sheet, markers, color pencils, and pencils	30 minutes

Day 2: Theater Activity

- **Standard**: Art Theater (Table 2)
- **Activity**: Maze game
- **Objectives:**
  - Students will be able to work effectively in groups
  - Students will write clear, detailed and concise instructions to get through a given maze
  - Students will have effective communication amongst each other

Table 5: Day 2 Breakdown of Panned Curriculum

Breakdown	Brief Description	Materials	Time allotted
Introduction	Introduction of the maze game and it's components	Instructor Instructions	15 minutes
Planner	Assigned team member plans out the path of the maze	Paper and writing utensil	10 minutes
Designer	Assigned team member takes the plans and writes the instructions for the maze	Paper and writing utensil	10 minutes
Executor	Assigned team member, follows instructions written	Paper and writing utensil	10 minutes

Week 2

Day 3: Pixel Art Introduction

- **Standard**: Art (Table 1) and Computer Science Algorithmic Thinking (Table 2)
- **Activity**: Pixel Art
- **Objectives:**
    - Students will be able to use the engineering design process to create and design their own pixel art
    - Students will able to effectively use cartesian coordinates to plot their pixel art
    - Students will be able to use algorithmic thinking to create Pseudo code that creates their pixel art

Table 6: Day 3 Breakdown of Planned Curriculum

Breakdown	Brief Description	Materials	Time allotted
Introduction to Pixel Art	Students are introduced to Pixel Art and its concepts	Worksheet	10 minutes
Introduction to Engineering Design Process	Students are introduced to the Engineering Design Process and its application to real life	Engineering Design Process Diagram	15 minutes
Pixel Art Planning and research	Students will use different pixel art's they have found to create their own	Grid paper, and pencil	25 minutes

Day 4: Theater Activity

- **Standard**: Art (Table 3) and Computer Science Algorithmic Thinking (Table 1)
- **Activity**: Pixel Art
- **Objectives:**
    - Students will be able to use the engineering design process to create and design their own pixel art
    - Students will able to effectively use cartesian coordinates to plot their pixel art
    - Students will be able to use algorithmic thinking to create Pseudo code that creates their pixel art

Table 7: Day 4 Breakdown of Planned Curriculum

Breakdown	Brief Description	Materials	Time allotted
Recap of Day 3	Introduction of the maze game and it's components	Activity Worksheet	5 minutes
Continuation of Pixel Art	Students will move forward with plotting their pixel arts and creating Pseudo code	Grid and plain paper, pencil, and activity worksheet	10 minutes

Week 3

Day 5: Pixel Art Continuation

- **Standard**: Art (Table 3) and Computer Science Algorithmic Thinking (Table 1)
- **Activity**: Pixel Art
- **Objectives:**
  - Students will be able to follow pseudo code and replicate a Pixel
  - Students will be able to debug errors
  - Students will effectively use Engineering Design process to come up with solutions for the errors

Table 8: Day 5 Breakdown of Planned Curriculum

Breakdown	Brief Description	Materials	Time allotted
Testing Pseudo	Students exchange their pixel art pseudo code to replicate	Grid and plain paper, pencil, and activity worksheet	5 minutes
Continuation of Pixel Art	Students will assess the replicated pixel art and fix any errors that occur	Grid and plain paper, pencil, and activity worksheet	10 minutes

Day 6: Field Trip and Closing Remarks

- **Standard**: Art
- **Objectives:**
  - Make connections between the activities they have completed during the program and apply it to the real work
  - Get a sense of what an engineering school is like and see different projects complete
  - Once again identify what they believe a Computer Scientist looks like

Table 9: Day 6 Breakdown of Planned Curriculum

Breakdown	Brief Description	Materials	Time allotted
WPI Campus Tour	Students will explore and see different sights of WPI	N/A	30 minutes
Draw a Computer Scientist	Students are once again asked draw what they believe a Computer Scientist look like.	Worksheet	30 minutes
Real world Applications and Connections	The program activities are broken down and shown in real world applications	Grid paper, and pencil	25 minutes
Closing Remarks	Students are given their AxiDraw Pixel Arts as keepsakes from the programs.	AxiDraw Pixel Art	15 minutes

4. Methodology

The goal of this project was to create and implement a computer science curriculum that involves all the standards we chose (Chapter 2.5, Tables 1, 2, 3)..We found a place to hold the curriculum and an audience to teach to. We created an initial timeline for curriculum implementation that would allow us to divide our planned activities and have a plan for each time we worked with our audience. The IRB (Institutional Review Board) ensured that our project plans would not endanger student's safety and privacy. Lastly, in order to test the effectiveness of our curriculum, we analyzed the data gathered from the IRB approved surveys and activities.

4.1 Setting and Audience

In order to implement our curriculum, we needed to find a place and group willing to allow us to preform our program at their facilities. We investigated nearby locations and programs that worked with middle school children. In looking through WPI's available Pre-Collegiate Outreach Programs, we noticed none of them run during the time frame of our project. The goal was to pilot the program in April, but most of the WPI programs occurred during the summer, outside our scope. The other option was looking for nearby middle schools that were possible to visit or find a group of students that would be able to travel to campus for the program. However, with advice from the project advisors, our decision was to recruit students from the local YMCA branch. The Central Community Branch YMCA in Worcester takes part in the School's Out Program. It's an afterschool program for kindergarteners all the way to through sixth graders. Students are offered a variety of services from homework help to fun afterschool activities involving sports and art.

The YMCA was extraordinarily helpful in recruiting the children. Since the target audience was specific on the grade levels we wanted to work with, we had a small group of seven to eight children, which worked well due to our inexperience in working with large groups of young students. The audience chosen to work with were 6th and 7th graders.

We spoke with Sarah Levy, the School Age Child Care Director, about bringing the children to WPI, working in the YMCA art classrooms, or working in the YMCA learning lab. Sarah presented us with a large art classroom and a smaller computer lab called the Learning Lab. Our audience was a small group, so we used the Learning Lab. There were computers ready and available to work with. There was a staff personnel assigned to us each day as well, in order to help facilitate.

4.2 Legal

In order to work with children on this project, our group went through the IRB approval process for minimal risk research with a vulnerable population. The process consisted of completing an application and having multiple sessions with Ruth McKeogh, the Administrative Assistant at the International and Global Studies Department at WPI, to ensure all documents were appropriate. The group also completed online training for minimal risk research on top of the background checks already needed to work with the YMCA.

We used surveys approved by the IRB to assess the children's impressions of computer science. The IRB checked for details, such as name, age, ethnicity, and gender, to make sure we weren't asking for personal information. Ruth also assisted in making the questions easier to understand with simpler questions and benchmarks. The benchmarks measured the students' understanding of computer science, math, and art. The curriculum approval was done with the IRB after a full review of each activity.

There's an approved consent form that would be given to the students and their parents, so that the parents can provide consent for their child to participate (Appendix E). The forms guarantee the safety of the students by explicitly stating what activities happen at the YMCA and WPI. They also inform the parents that the curriculum is voluntary and that they may withdraw their child at any point and confidentiality. Each group member needed to fill out a Criminal Offender Record Information (CORI) and have our background check approved, in order to work at the YMCA.

4.3 Curriculum Development Process

4.3.1: Changing Machinery (2/13/2019)

In the beginning of the project, we aimed to incorporate an embroidery machine that created stitches and patterns from a written algorithm. The students were to use a simplified version of the existing algorithm to create their own pieces. After working the embroidery machine, we noticed that we had a lot of challenges we would face if we continued. For example, understanding the necessary code for the embroidery machine was difficult. Trying to understand as well as teach this code would be hard especially with the time constraint. It would also make it difficult for our audience to grasp and understand. We also knew that we did not want students to be focused on learning a coding language with one activity.

In order to run the necessary code, we needed to install software as well as download all previous code written on each computer that would be used. This would have been very difficult to do on the YMCA computer and the written code would also occupy a lot of space. When running the code, it would take a lot of time to compile if the computers are being overworked. Another challenge was working with the tension of the machine. The tension would often get in the way of how the patterns looked when they were sown into the fabric.

To combat these problems, we decided to switch to the AxiDraw. The AxiDraw was ready, available to use and shared similar characteristics to the embroidery machine. They both provided us with a product we could provide the students. Both the Embroidery Machine and the AxiDraw can be controlled with written code. The AxiDraw, however, was more portable and easier to use. It can be controlled by a simple software that was accessible. We also found that the AxiDraw has an open source turtle graphics library that we can utilize.

4.3.2: Switch from Computer Based to Paper (2/20/2019)

After doing much research about the open source turtle graphics code for the AxiDraw, we were unable to download the necessary library. When trying to add turtle graphics to the AxiDraw component, we had issues with accessing the necessary libraries needed to create code for the AxiDraw to interpret. Furthermore, we did not have time to be able to create a coding library that would simplify the code to control the AxiDraw. Even if we were able to complete the library, it would be difficult to install the program and the library needed to run the code on the YMCA computers. We were aiming for too much and needed to change our scope.

Because of time constraints, we decided to move our activities to be more paper based and hands-on activities than utilizing computers. Instead of coding on the computer, students would be writing pseudo code on paper. This would further help students understand what they are learning since pseudo code is more English based than code. It also allows the students to immerse more into the activities instead of typing code on the computer.

4.4 Analyzing the Effectiveness of the Project

To analyze the effectiveness of our curriculum on the students' interest, we used two data methods, one qualitative and one quantitative. The qualitative was the Draw a Computer Scientist Activity (Chapter 3.2.1), and the quantitative was the survey the students were given (Appendix F). The survey itself consists of eight total questions, as shown in Table 10. A Likert scale[4] was used for each statement with 1 being "strongly disagree" to 4 being "strongly agree". We collected data using identical methods, pre and post program, in order to show the change over time. The quantitative data came from the survey and the qualitative data came from the "Draw a Computer Scientist" activity (Appendix B). The survey asked eight questions from three categories: knowledge of computers, knowing of computer science, and subject affinity (Table 10). These categories and questions were self-devised. We wanted to see if students with little computer science background or interest participate in our curriculum, then afterwards, they will be more aware and have an interest in Computer Science. We looked at the percentages of the answers, comparing the number from before and after the program. The drawing activity was analyzed in a similar method, using points of interest to compare the pre-program and post-program drawings. The data itself will be discussed in more detail in Chapter 5.

Table 10: Program survey questions to measure student's computer knowledge/interest, computer science knowledge, and subject affinity using Likert scale

Category	Question
Computer Knowledge/Interest	I am comfortable working with a computer.
	I want to know how a computer works.
Knowledge of computer science	I know what computer science is.
	I know what it means to write a computer science program.
	I think art and computer science do not have very much to do with each other.
Subject Affinity	I enjoy doing art activities more than normal classes.
	Learning something new can sometimes be fun.
	I like math and science.

[4] A rating scale used in survey forms/questionnaires that gives people several balanced responses to choose from.

The pre-program survey and drawing activity were conducted after introductions and before the start of any of the planned activities to understand what the students self-reportedly thought at that point in time. Meanwhile, the post-program survey and drawing activity were conducted after the completion of all activities to show the full effect of the program. The students who took the survey before the program were the same who took the survey after the program, except for one student missing during the post-program survey. To adjust for this, that student's data has been removed the data from the pre-program survey. Thus, all pie charts are using a sample size of six. Also, all students who participated in the surveys were present for all days of the program, except for one student being absent on Day 3.

5. Results and Analysis

In order to test the effectiveness of our curriculum and see the progress we had made in raising awareness and interest for Computer Science, we collected data before and after the program using two methods. We used a survey to gather quantitative data and the "Draw a Computer Scientist" drawing activity to gather qualitative data. The goal was to observe a positive change in the students' interest in Computer Science. The results and an analysis of those results are presented in this chapter along with a day by day reflection of our visits to the YMCA.

5.1 Day-by-Day Implementation Reflection

The program consisted of a pre-program visit, six visits to the YMCA, and one day in which the students visited the WPI campus. In the following subsections, the events of each day are discussed in comparison to the original curriculum (Chapter 3.1) along with a reflection of those events. The audience for our pilot program at the YMCA were 6th and 7th graders.

5.1.1 The Pre-Program Visit (3/21/19)

Our first visit occurred before the start of the program; it was not a part of the original six-day curriculum plan. The purpose of this visit was to allow our group to meet the students and go through introductions. Since the curriculum planned to complete introductions on the first program day, there was no formal plan laid out for this visit. Instead, we chose to complete introductions, give an overview of the program, and get to know the students. Only Neer and Matthew were present for this visit.

On this day, the audience consisted of seven students, five girls and two boys. The program overview explained what kind of activities and what the overall goal of the curriculum was. We also introduced ourselves to the students, giving our names and a brief explanation of our field of study. A few students showed interest when Matthew introduced himself as a game designer. The students also introduced themselves individually. They also discussed their interests and hobbies with our group, creating more of a bond between our group and the students. Some students were a lot more eager to talk and participate while others were more reserved. While having conversations with the students, we asked them some simple questions such as "Ever heard about Computer Science?" Since all the students answered no or not really, it was apparent the overall level of knowledge regarding computer science was almost nonexistent. This gave us the proper set-up to begin the program to introduce the students to computer science.

5.1.2 Day 1 - Initial Data Collection and AxiDraw Demo (3/25/19)

On this day, Neer and Metasebia were present. Our audience was a total of eight students. The planned activities for this day were introductions, the survey, and the draw a computer scientist activity (Chapter 3.2.1). The main goal was to collect the pre-program data and introduce everyone to each other. However, since we completed introductions during the pre-program visit, we turned our focus towards collecting data.

On this day, most of the same students were present from the previous visit and a few new students chose attended the program. As a result, after Metasebia introduced herself, another brief overview of the program occurred. In order to collect pre-program data, we conducted the pre-program survey and "Draw a Computer Scientist" activity. During the drawing activity, the students said words such as "I do not even know what a computer scientist is!". Some students took time to put in detail into their drawings while others used stick figures to make more simplistic illustrations. We answered any questions, such as what a computer scientist looks like, after collecting the drawings.

Since introductions only took a short amount of time and we collected data a lot faster than expected, there was a significant amount of time remaining in the visit. Due to the large amount of interest the students were showing in the AxiDraw, Metasebia and Neer chose to provide a demonstration (Chapter 3.4). In order to demonstrate the robot's capability, each student had a chance to request a drawing from the AxiDraw. The drawings were created using paper and markers. Watching the robot draw caused the students to be excited, amazed, and start taking videos the robot in action. Some comments included "That the most perfect circle I have ever seen!" Seeing how enthusiastic the students were, our group started a short discussion about how the AxiDraw works using vectors. On this day, we were able to gather important pre-program data and greatly raise the interest in the upcoming program days using the AxiDraw.

5.1.3 Day 2 - Maze Activity (3/28/2019)

During the second planned program visit, Neer and Matthew were present with Metasebia absent. The students consisted of six students. The plan on this day was to complete the maze activity (Chapter 3.2.2) with the goal being to teach the students how to plan, write pseudo code, and execute it to get through the maze. Since the audience totaled to six on this day, we decided to split the room up into two groups of three with the groups being self-chosen. Neer and Matthew each took one to work with personally through the activity.

The group Neer was working with was very energetic. They came up with characters, chose their own roles, and helped each other throughout the process. The student with the designer role in this group first chose to write down their instructions in paragraph form. While doing so, Neer took the chance to introduce bug-fixing. By the end of this visit, this student specifically was able to use the term in the right context. In addition, the same student started to ask questions about how the AxiDraw works, leading into a small discussion about the robot's programming.

With minor help, Neer's group was able to complete a path through the maze. The students abandoned the instructions written on paper due to errors and, because of the lack of time remaining, the students gave the maze instructions verbally. Neer held a few extra discussions regarding how computer science was present in other fields such as game design and used examples from games that the students were familiar with, such as Fortnite [3]. Overall, Neer's group saw a great amount of success during this visit and was able to complete the activity as planned. However, Matthew's group had a different experience.

Matthew's group didn't have the same energy. They did not want to participate in the activities due to some friction between each other, something we did not expect. Matthew tried to resolve the conflicts quickly and gather them together to complete the activity. However, there was not much change in the student's behavior. The students persisted to not cooperate until the staff member fully stepped in to calm them down. The staff member also attempted to help by providing Matthew with some advice on how to work with the students, such as talking in a calmer voice and trying to shut down side conversations when they came up. The group eventually began the activity but still required heavy assistance and constant encouragement. Eventually, the students completed the first few steps of the activity and understand how methods work. Matthew explained the connection between the maze game and running methods in code using loops and conditions. While there was still more for this group to complete, there was not enough time due to the earlier complications.

5.1.4 Day 3 - Review and Short Pixel Art Overview (4/1/19)

Neer and Metasebia conducted this visit and had an audience of seven students, with one student leaving part way through due to pick up. Planned activities included an introduction to pixel art and the engineering design process and Pixel Art planning (Chapter 3.2.3). The main goal was to introduce the students to what pixel art is. However, due to delays, the students showed up 30 minutes late and took an additional 15 minutes to get settled, reducing the time remaining to about 15 minutes, instead of our usual hour.

Keeping the main goal for this day in mind, our group had to improvise their plans. Instead of doing the first pixel art activity, Metasebia explained what pixel art is, gave an overview of the planned pixel art activity, and provided a couple of examples. In addition, we held a short review of the maze activity from the prior visit. Four of the students were able to correctly recall most of the terms and the importance behind the maze activity.

5.1.5 Day 4 - Pixel Art and Start of Final Project (4/8/19)

On this day, both Neer and Metasebia conducted the visit with an audience of six students. The students were late by about 10-15 minutes on this day. The plan for this visit was supposed to be a continuation of Day 3, where the students would work on their pixel art activity and write the pseudo code for it (Chapter 3.2.3). However, the deviations that occurred on Day 3 caused this visit to almost completely changed from the original plan.

Building off the brief overview of pixel art from the prior visit, Neer and Metasebia spent the first half of the visit talking about Cartesian coordinates and doing plotting activities to create pixel art (Chapter 3.2.3). During this math/art-based activity, our group gave an explanation where we established a connection between math and computer science. The remainder of the class was used to start the pixel art projects. The students used the computers in the room to research some pixel art that interested them enough that they wished to draw it. Our team walked around from student to student talking about topics such as symmetry and a program that could autonomously draw the pixel art.

While Neer was helping a student look up an image, the student asked how Neer knew how to use a computer so efficiently. We realized, at this point, that our ability at doing a google search seemed very impressive to the students. As a result, Neer talked a little about how, through coding and hours of using a computer, he was able to become adept at using one. In addition, on this day, some of the more reserved students started to be more open towards us. A few students chose to draw more complicated pixel art, but with some help and encouragement, each student was able to draw the art. Before the end of the visit, we recorded each student's choice of pixel art so Metasebia could draw them using the AxiDraw.

5.1.6 Day 5 - Relating back to Computer Science (4/8/19)

Day 5 was the first visit in which all group members were present. Seven students attended this day with two students leaving part-way due to possible illness and being picked up. According to the curriculum, the plan for this visit was to have the students try and use another student's pixel art code to re-create the other student's pixel art. By doing so, we could explain to the students how the pixel art activity relates back to computer science (Chapters 3.1 and 3.2.3). The main goal for this plan was to show the students how all the activities previously completed were related to computer science. Due to the amount of deviations from the original plan that had already occurred, the schedule for this visit was quite different.

Metasebia took the role of lead with Neer and Matthew providing support and jumping in when possible. We went more in depth with the pixel art activity, including explaining how turtle graphics was able to draw pixel art. The plan was to give the students their AxiDraw pixel art drawings from last time, but Metasebia was not able to have them finished before this visit. Instead, we delved a little deeper into some computer science topics, such as variables, functions, and a little more on bug-fixing. In addition, terms such as coding language, loops, and libraries were also touched upon. In order to further cement these topics, we talked about the various activities we had done throughout our program and related them back to these topics and terms. To build even further, each group member talked about how computer science is found in their area of study (Figure 6x). The students responded with their own interests, which included game design, engineering, and being a doctor.

Figure 6: Matthew talking about Computer Science in Interactive Media and Game Design

While having these discussions, Metasebia came up with the idea of doing the "How to make a Peanut Butter and Jelly Sandwich" activity (Figure 7). In short, this activity requires the participants to give very precise, step-by-step instructions on how to make a peanut butter and jelly sandwich. The catch is that everything from how to pick up something to putting a piece of bread on a plate must be described in detail. When the students tried to instruct us how to make the sandwich, they got aggravated since every one of their instructions were being taken literally and the result was not what they wanted it to be. For example, the instruction "move the bread right" would result in the bread moving right, but not the correct amount. However, they were also really enjoying the activity thanks to Metasebia's exaggerated acting of moving the bread (i.e., paper) around. After some time, we took a break and explained how those specific instructions were just like coding; each line must be exact and specific so that the program can run correctly.

Figure 7: Metasebia conducting the “How to make a Peanut Butter and Jelly Sandwich” activity

Overall, interest and excitement were high compared to other days. To our eyes, the students seemed highly interested and were listening attentively to the discussions. In addition, we received the consent and travel forms approved by the student’s parents on this day, giving our group permission to conduct the field trip to WPI.

5.1.7 Day 6 - Post-Program Data Collection (4/22/19)

Day 6 was originally supposed to be the day of the field trip to WPI’s campus, where we would gather all the post-program data and wrap up the program (Chapter 3.1). However, due to large delays and upcoming deadlines, our group decided to go to the YMCA to complete the necessary post-program data collection instead of waiting for the students to come to campus. The trip itself, however, was still planned to take place on an alternative date.

The data we needed to collect, specifically, was the post-program survey and the second “Draw a Computer Scientist” activity. Our audience this day was six students, with one student leaving after taking the survey and doing the drawing activity. Neer and Matthew took charge of conducting the survey and drawing activity with the students while Metasebia wrapped up the pixel art project with the students. Metasebia used the AxiDraw to create some simple drawings for the students in person and planned to bring back more detailed drawings to the students at a later date. Some of the drawings were a rose, dragon, and a yin yang symbol.

5.1.8 Field Trip to WPI (4/26/19)

Our last day of interaction with the students was the WPI campus field trip and tour (Figure 8). As mentioned in Chapter 5.1.7, the tour plan included the post-program data collection. However, since all necessary data was already collected, we were able to fully focus on giving the tour.

On this day, Metasebia, Matthew, and our advisor Katherine Chen were the tour guides. The trip started off a bit rough due to miscommunication with the arrival location, causing a small delay. Only three students were able to attend the trip with a chaperone. The students were able to visit places such as the robot pits in the Rec Center, the MakerSpace in Foisie Innovation Studio, the Collab lab, the Campus Center, and various places in Fuller Laboratories. In the MakerSpace, each student was given a 3d printed duck to take home with them. In Fuller 222, an IMGD art student was kind enough to give a small art demonstration using Maya[5]. The student showed how a substance painter looks and did a quick run in Mixamo[6] with showing the animation plugins. The students asked questions throughout the tour about building robots and creating video games. When asked about their favorite activity from the program, all three students responded with the pixel art activity. After the tour was over and goodbyes said, the program was officially complete.

Figure 8: Students at WPI Campus for Field Trip

[5] A 3D computer graphics program that allows for modeling, animation, and rendering of 3D sculptures.
[6] A 3D computer graphics software that allows for 3D modeling, rigging, and animation.

5.1.9 General Observations from Visits to the YMCA

During our visits, we came across a few general observations. The students were more interested in interactive activities, such as the pixel art activity and the demonstration with the AxiDraw, shown in Chapters 5.1.2 and 5.1.5. We suspect that by getting the chance to be active or see something new helped increase the student's desire to participate. In addition, after completing these types of activities, the student's asked more questions and seemed to be overall more interested in the topic (Chapter 5.1.3). We also found that most of the students claimed to not like subjects such as math, science, etc. However, those same students showed interest in our activities and teaching topics. Based on this, it may be true that the students actually did have an interest in a STEM field. But, due to their inexperience and lack of exposure to STEM, the students believed the field was something different from what it is.

5.2 Survey Data Results

Figures 9-16 are the pre-program survey and post-program survey bar chart for each of the questions by category. The questions are in Table 1 (Chapter 4.5).

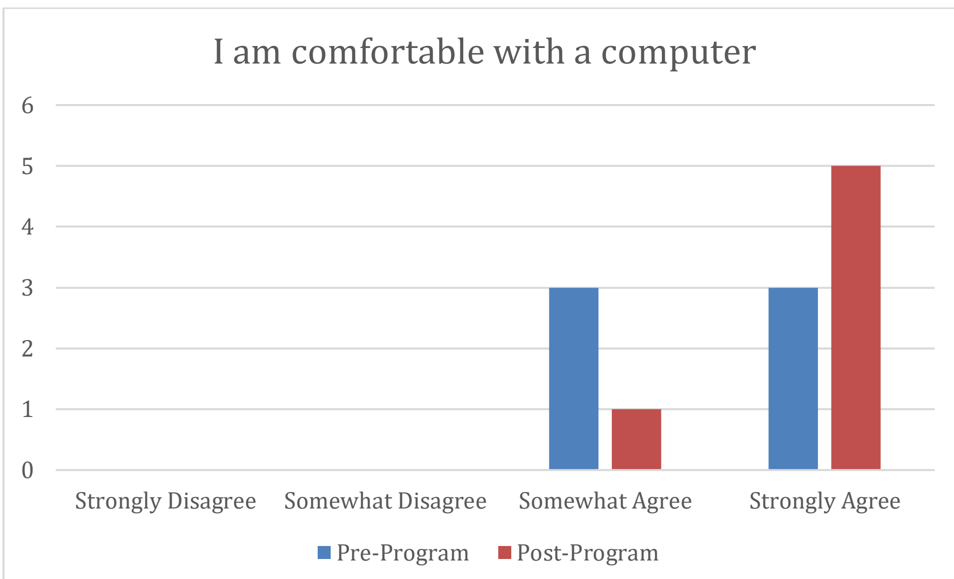


Figure 9: Pre-program and Post-program survey responses to comfort with computers

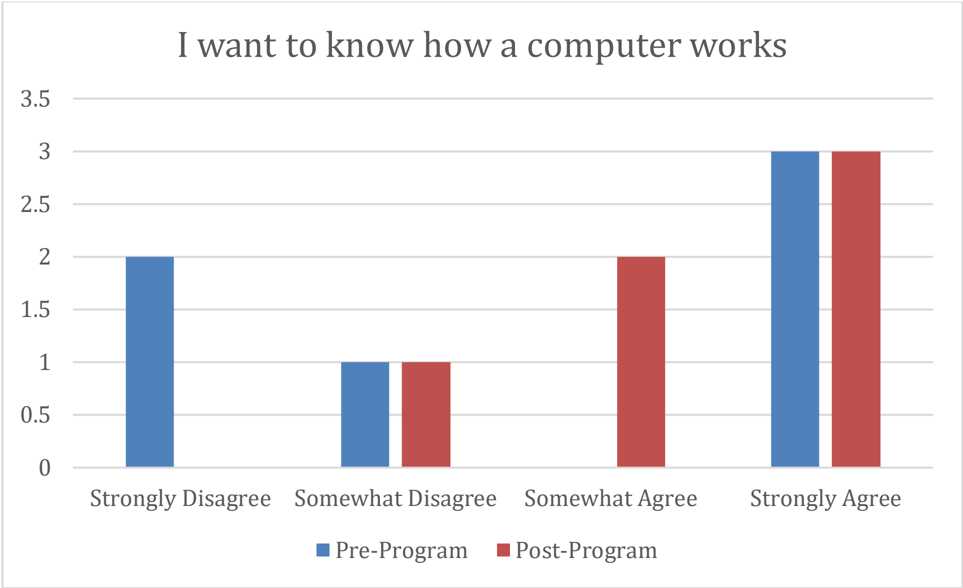


Figure 10: Pre-program and Post-program survey responses to wanting to know how a computer works

The first category of questions inquired about the level of knowledge the students had with computers in general and their interest in computers. Figure 9 represent how comfortable the students felt with computers before and after the program. Figure 10 show the student's interest in learning more about computers.

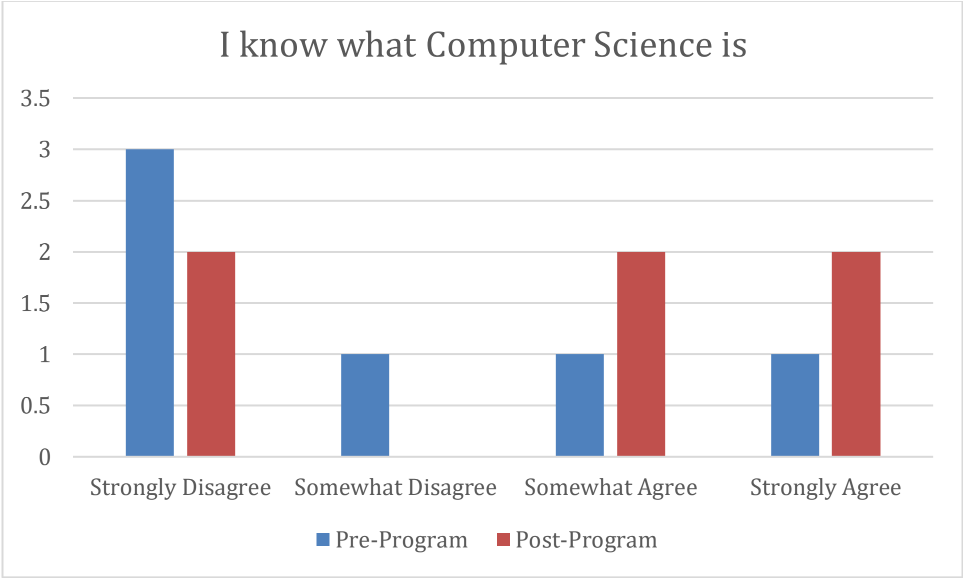


Figure 11: Pre-program and Post-program survey responses to knowing what computer science is

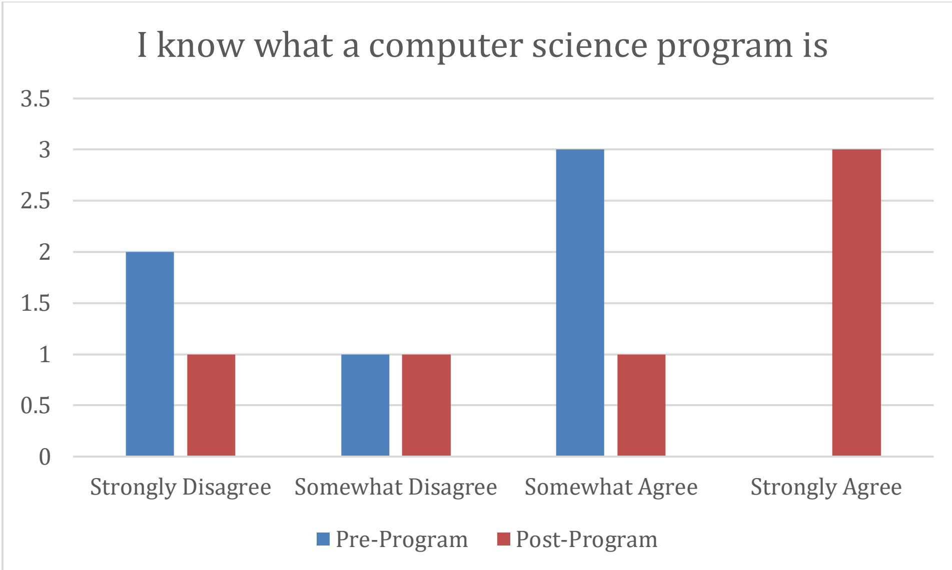


Figure 12: Pre-program and Post-program survey responses to knowing what a computer science program is

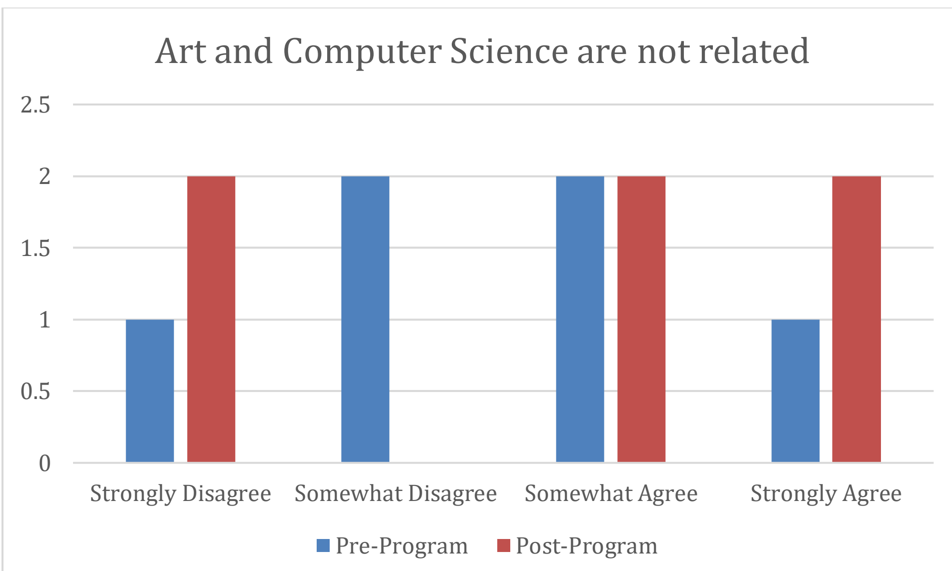


Figure 13: Pre-program and Post-program survey responses to connection between art and computer science

The second category, knowledge of computer science, asked 3 questions with a direct connection to computer science. Figure 11 visualize the amount the students believed they knew what computer science is. Figure 12 explain how greatly the students believed they knew what a computer science program is. Program in this context talks to a computer science program and not a computer science education program, which we explained to the students while they were taking the survey. Lastly, Figure 13 show how much the students believed there was a connection between art and computer science.

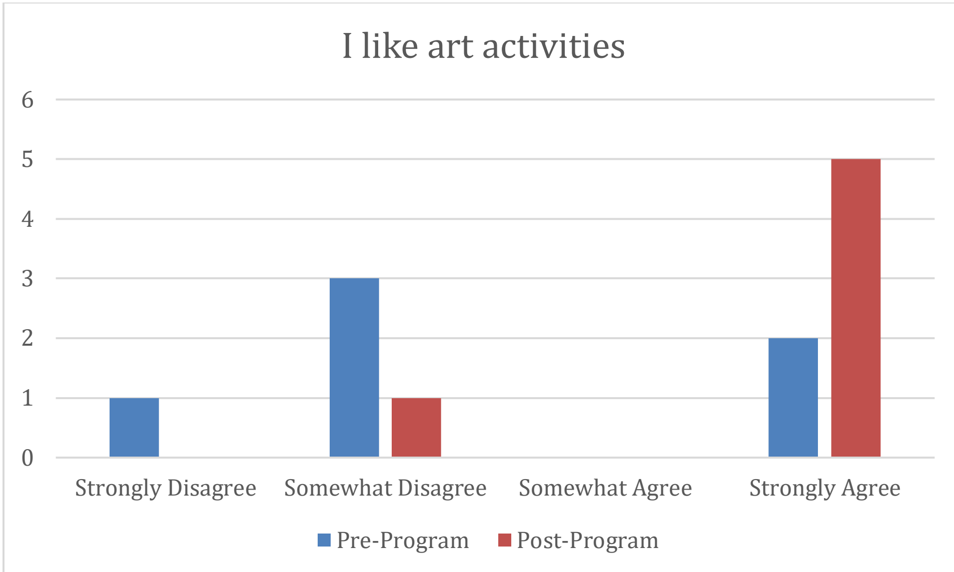


Figure 14: Pre-program and Post-program survey responses to affinity for art

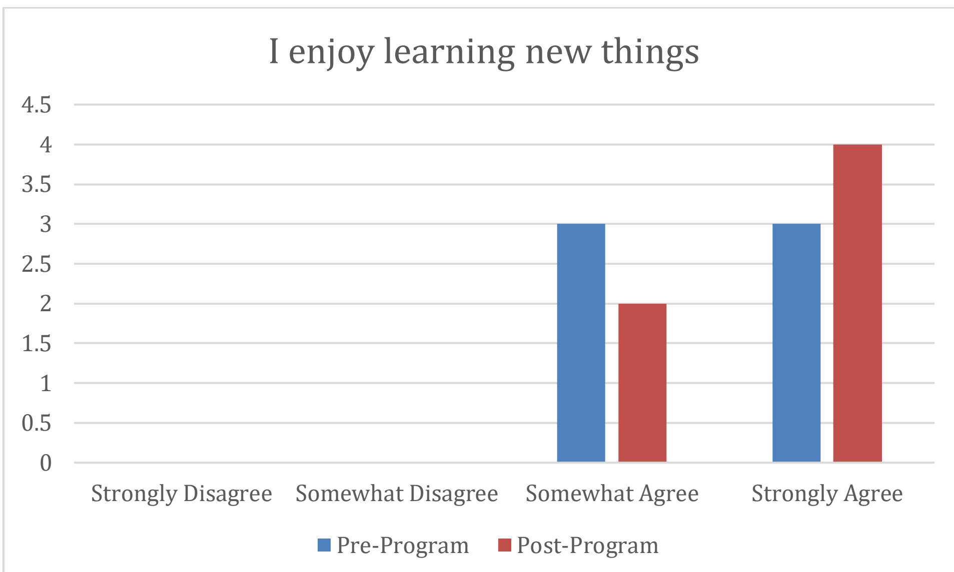


Figure 15: Pre-program and Post-program survey responses to enjoyment to learn

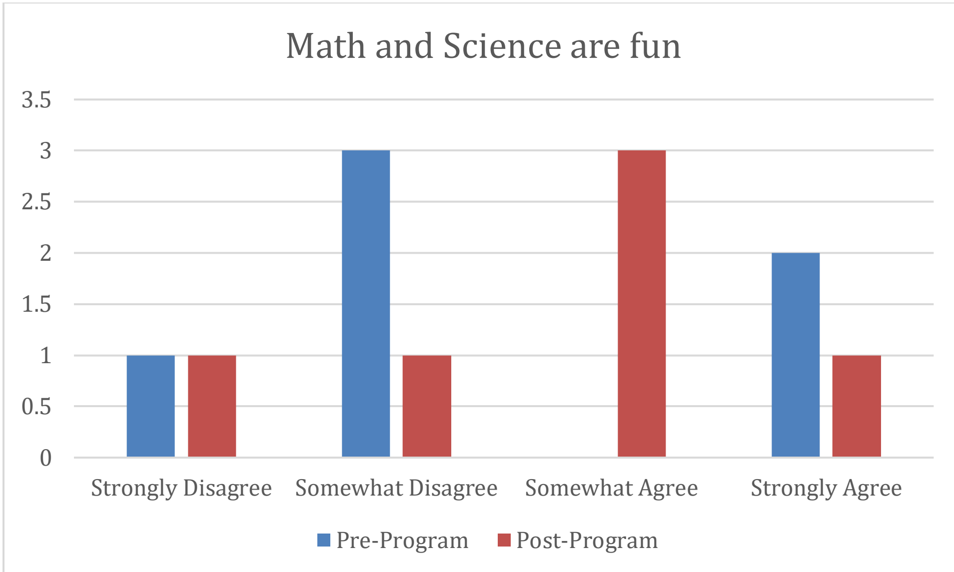


Figure 16: Pre-program and Post-program survey response to affinity for math and science

The last set of questions asked about the student's affinity toward certain subjects. Figure 14 show about the student's affinity towards art, Figure 15 measure how willing they were to learn new material, and Figure 16 measure the student's affinity towards math and science.

5.3 Survey Data Analysis

We preformed analysis on the survey results by taking the total number of responses we received and observing the percentages of those responses and their change between pre-program and post-program.

5.3.1 Computer Knowledge/Interest

Looking at the computer knowledge/interest category, as shown in Figures 9 and 10, all students had some level of comfort with using a computer and 50% had an interest in learning more how a computer works. Both pairs of questions show an increase in the strongly agree/somewhat agree categories on the post-program survey, suggesting the students have become more comfortable and have more interest in computers.

At the time of the pre-program survey, according to the YMCA staff, the students had mainly been using the computers to play video games, explaining why they all claimed to be comfortable with using a computer. By the time of the post-program survey, the students had used the computers look up images and do a little bit of research for the pixel art project. This gave the students an opportunity to use the computer for something other than games and become more comfortable with a computer to complete tasks such as simple research.

We conjecture that a couple of the most prominent ways our program helped increase the interest in how a computer works was through the improvised "How to make a peanut butter and jelly" mini activity and various demonstrations, such as the one with the AxiDraw on Day 1 (Chapter 3.3.2). The mini activity revealed how complex a computer can be but also how it was possible to get a program to do almost anything, as long as you knew how. The students, during this activity, wanted to know about the limits of what a computer knows. For example, one of the topics we discussed, based on a question by a student, was how a computer understands keyboard input. In addition, as the program went on, we started to receive questions such as "How does the robot work?" and "How did you make the computer do that?" that indicated increased interest in computers.

5.3.2 Knowledge of Computer Science

In the knowledge of computer science category of questions, the results were promising, as shown by the greater percentage of responses agreeing. At the time of the pre-program survey, a majority of students self-reported that they did not know what computer science is or much about the topics surrounding it, as seen in Figures, 11, 12, and 13. After the program, 33.3% more students answered positively when asked if they knew what computer science is, as seen in Figure 11. The students also self-reported that they think they better understand what a computer science program is (Figure 12). Only one more student self-reported that there is a connection between computer science and other fields of study, such as art (Figure 13).

As part of the goal of the project was to raise interest in computer science, seeing an increase in all three of these areas is a promising result. The students, by the end of the program's various activities, were able to self-report that they have a better grasp on what computer science is. For example, the maze activity gave a lesson in debugging and writing a function to get through a random maze. The students continued to use the term "debugging" on their own during the later days of the program. In addition, Neer showed the students code he had written and explained what some terms meant in terms of computer science. There was not much change between the pre-program and post-program data shown in Figure 13, showing that some the students did not see a connection between Computer Science and other fields, such as arts and crafts, by the end of the program. However, during the program, connections were explained between computer science and the medical field, game design, and biology.

5.3.3 Subject Affinity

In Figures 14-16, the pre-program survey results showed that about half the students had some form of interest in math and science, all of the students enjoyed learning something new, but only a couple had an interest in art activities. By the time of the post-program survey, the opinions about these subjects changed. The most significant change came from the enjoyment of art activities. Figure 14 shows that during the pre-program survey, only 2 students enjoyed art activities. Figure 14 also shows that, after the program, five students self-reported that they enjoyed art activities. In addition, each student still agreed by the end of the program that learning something new can be an enjoyable experience (Figure 15). When it came to math and science, there was a decrease in the number of students disagreeing the subjects were fun, although the change was not that significant (Figure 16).

The curriculum was heavily based around arts and crafts. As such, the students experienced a various assortment of art activities, most likely causing the increase in agreement that art activities can be enjoyable. While the activities were art-based, the concepts taught were based in computer science; and since math and science are an intricate part of computer science, the interest in these areas also increased. During the program, we made a connection between math and the computer science through our introduction to pixel art on Day 4. For example, the draw your pixel art activity helped the children learn about Cartesian coordinates, a topic some of the children seemed to really enjoy. Overall, our results show that our methods of teaching computer science allowed the classroom environment to remain enjoyable for the students to learn new topics.

The surveys helped us gather quantitative data to see a change over time in relation to the program. They helped us look at keys points of possible change that could occur after a student had gone through the program. In additional, it is a possibility that the student's interaction with us, as college students, helped change their attitude towards computer science. The data shows an overall improvement across all of the categories, suggesting the effectiveness of our curriculum.

5.4 Draw a Computer Scientist Activity Results

The second method of collecting data to measure program effectiveness was the "Draw a Computer Scientist" drawing activity [10]. The students had to draw what they imagined a computer scientist looked like, once before the program and once after. For the remainder of this section, they are referred to as the pre-drawings and post-drawings.

As the data from these activities was qualitative, we chose a few categories for defining aspects to look for in the drawings: the drawing's organization, the person drawn, if they had glasses, the type of technology in the drawing, the clothing they were wearing, the facial expression, and the drawing utensils used. We chose these categories so we could focus on the characteristics that would help us identify how the students viewed a computer scientist. For example, we chose to look for gender as it would show if the students believed a computer science was a male dominated field or not. In addition, looking at characteristics such as the presence of glasses, the type of clothing, and the facial expression would help us understand what the students thought about the environment and lifestyle of a computer scientist. A lab coat and smile would indicate a happy scientist while a t-shirt and frown would indicate a more relaxed but unhappy person. The results of this activity helped our group understand what the students imagined a computer scientist to be.

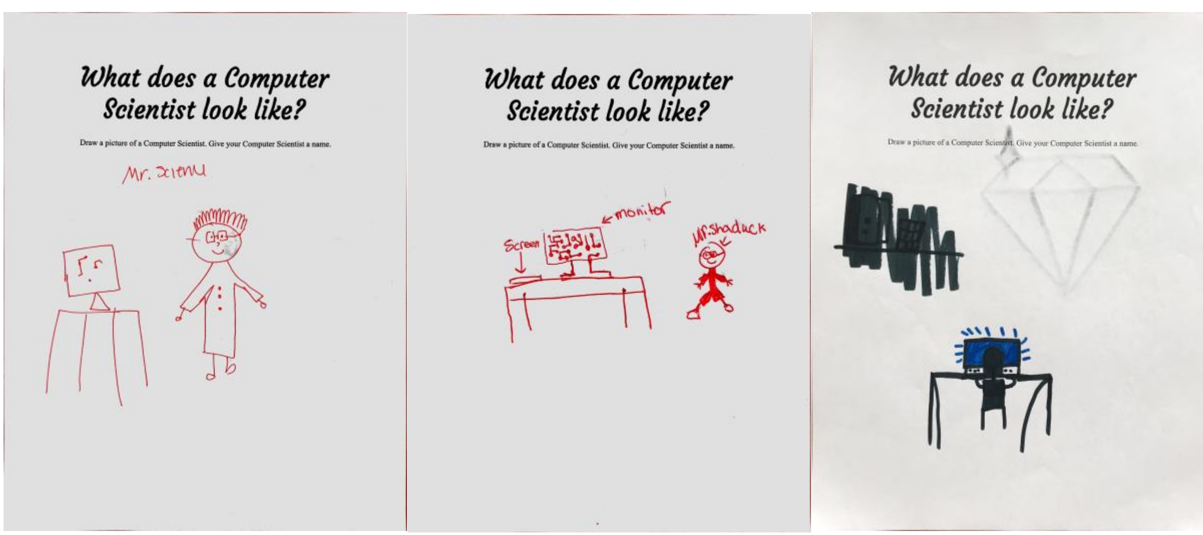


Figure 17: Pre-program drawings of what a computer scientist looks like a) with glasses, lab coat, and smiling b) with glasses, desk, and smiling
c) sitting at a desk

Figures 17a-17c are a sample of the results received during the pre-drawings. In the post-drawings, the illustrations were quite different compared to the ones from the pre-drawing. A sample of the drawings are shown in Figures 18a-18c.

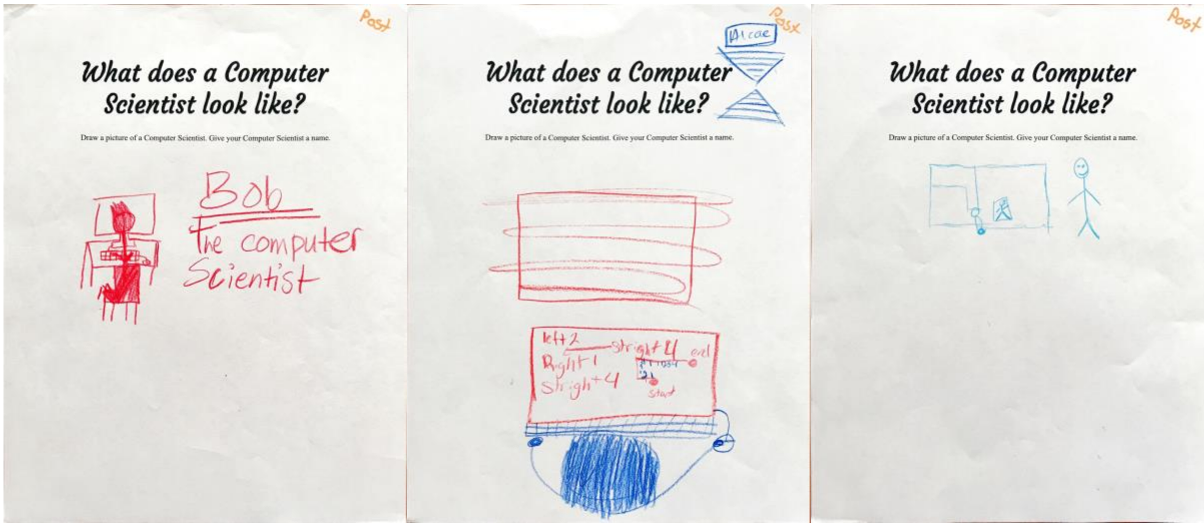


Figure 18: Post-program drawings of what a computer scientist looks like a) sitting at a desk b) writing maze activity code c) completing maze activity

Table 11: Pre-program and post-program "Draw a Computer Scientist" activity themes

Theme	Pre-Program	Post-Program
Drawing organization	Almost all had computer and person separate	4 had computer with person together
Person Drawn	4 were men	Variety: man, woman, stick-figures
Glasses?	4 with glasses	Only one
Type of technology	All had a computer or laptop, with 50% of them on a desk	All had some form of computer, 2 referencing maze activity
Clothing	Variety: Lab coats, t-shirt with word "science", casual, none (stick figure)	Not drawn, was stick figures
Facial Expression	Majority smiling	Half smiling. Cannot see faces of rest
Drawing utensil	Pencil/Pen or just one color	Went for crayons over markers and pencil, large variety of colors

In order to organize the results, a summary of the results from both the pre-program drawings and post-program drawings were compiled (Table 11). We decided if the drawing contained a man or woman based off either the length of the drawing's hair or the name the name the student wrote down, such as "Mr. Science". In addition, the drawing utensils that were available were the same during both the pre and post program drawing activities.

5.5 Draw a Computer Scientist Activity Analysis

One common trend observed across most of the pre-program drawings was a computer and a scientist being drawn separately. The separation of the term computer science into its two parts shows the understanding the students had of the words individually, but not together as a whole.

Each drawing had a form of a computer, ranging from laptops to desktops with monitors. Also, half of the computers were drawn on a desk, some with a person sitting at it, giving the idea that the students imagined a desk job when imagining a computer scientist. Most drawings were of men and had glasses, giving the idea that computer science is a male-orientated, nerdy field. The clothing chosen, if any, had interesting variety to it. They were lab coats, casual clothes, a t-shirt with the word "Science" printed across it, and the rest just did not have clothes due to the fact they were stick figures. For this part of the drawing, the students were most likely focused on the scientist portion of the computer scientist, explaining the lab coats and the word "Science".

The most surprising result was the facial expressions. All drawings in which a face was visible had the person drawn with a smile on their face. A smile or frown tends to indicate if the artist of the drawing thinks computer science is a field that is enjoyable (smile) or not (frown). Based on the research and results, it is possible the students did not think computer science was an unenjoyable field. When running this activity, we gave each student the option of pencils, pens, markers, and crayons as their drawing utensils. Most students either chose a pencil, pen, or just one color to complete their drawings. However, one student did was a variety of markers to decorate her page.

The collection of illustrations received during the post-drawings were quite different from the pre-drawings. Many drawings still did have the computer and person separated, but a majority showed them together. This shows that the students started to view Computer Science as one concept instead of two. A larger variety of people drawn were drawn, but the ratio of men to other kinds of people was greatly reduced. Instead, there were more gender-neutral drawings such as stick figures. The presence of glasses almost completely disappeared, suggesting the student's might not believe computer science is a so-called "nerdy" field. The type of technologies found were very similar to the pre-drawings, in that each illustration had some form of computer. However, references to previous activities can be observed in the drawings, such as references to the maze activity. There was a distinct lack of any form of clothing in these drawings. The previous articles of clothing such as the lab coat were not found, showing the idea that the students view Computer Science differently from the work of a typical lab scientist. It was observed that, within all the drawing that had a visible face, the students drew their person smiling. The post-program drawing had a larger variety of colors overall but individually were mostly mono color. Also, similar to the pre-drawings, a majority of students followed the idea of drawing a computer and then a scientist. Some, however, drew only a computer and one other student drew a team sitting at desks with computers.

This activity helped us see into the minds of the students and what they imagined a computer scientist was before and after the program. The results of the pre-drawings were mostly as expected, having the drawings be mostly men, wearing lab coats and glasses. The post-drawings were a little blander than desired, but still showed a good amount of change. Seeing remnants of the activities showing up the post-drawings indicates our program had left an impression/impact on the students.

6. Discussion and Recommendations

6.1 Curriculum Discussion

The original daily plans for the curriculum, as mentioned in Chapter 3, did not go according to plan. We improvised each day when necessary and came up with new activities on the spot. Due to time constraints, we created the "How to create a Peanut Butter and Jelly Sandwich" activity. While we did have multiple curriculum versions to handle anomalies, the deviations were much larger than we expected. As a result, we had to adapt to the situation in order to make the curriculum a success.

We completed a majority of the planned learning objectives for each day, even with the unaccounted improvisations. Students learned what a computer scientist was and what they did in the real world. The discussion afterwards focused on the results of computer scientist that affect their daily lives. Students were able to write clear instructions and execute them effectively during the maze activity. They were also able to retain and recall vocabulary words like debugging, Cartesian coordinates and point of origin. Despite the major schedule and time changes, each of our major planned activities were able to take place. The maze and pixel art activities, in particular, were able to be completed while still being fun for the students.

When creating our curriculum, we had identified a short list of standards from the Massachusetts Department of Education Frameworks in the computer science and Visual Arts and Theater sections. We were not able to touch upon all of the standards that we identified, but still managed to include most of them. The algorithms standards we addressed, as seen in Table 1, were 6-8.CT.b.1, 6-8.CT.b.3, 6-8.CT.b.4. These standards focused on the problem-solving skills involved with algorithms and providing multiple solutions for one problem. The art standards covered, according to Tables 2 and 3, were Acting 1.7, Acting 1.12, and Visual Arts 1.6. The theatre standards require the students to communicate about a character they created that is a computer scientist and what their jobs were. These standards were addressed in the "Draw a Computer Scientist" activity and the maze activity (Chapter 3.2.1 and 3.2.2). The visual art standard was the main objective of the pixel art activity (Chapter 3.2.3). The objective of the standard is to teach students to work with any medium provided and adjust to their tools. The students were only allowed to use a cartesian coordinate graph to create their image, and they needed to provide instructions on how to recreate them.

A major part of our curriculum was to teach a set of computer science concepts to our audience. Our mains concepts to teach were algorithms, debugging, functions, and pseudo code. Even with our activities not going exactly as we hoped, we were still able to teach these concepts. The students learned about debugging and pseudo code through our maze activity. Students wrote down instructions and worked to find errors in those instructions as they acted them out. Through this, the students learned how to find mistakes through practice and were able to adjust them. The pixel art activity helped the students learn about functions and algorithms by using code to create pixel art. The students also learned how the field of Computer Science is filled with the need for clear communication, teamwork, and concise but well thought out work.

6.2 Recommendations

Our group faced a variety of challenges and learned a great amount from our experience during this project. As such, as part of our project deliverables, we wish to provide a collection of recommendations for future projects that attempt to tackle a problem like the one described in this report.

6.2.1 Be Prepared to Adapt

A couple obstacles we encountered in running our curriculum was with reality not aligning with our expectations and understanding the environment we were working in. The curriculum was created to run an hour each day, and the YMCA did their best to make sure schedules were followed. However, some days lost up to 30 minutes or had some other unforeseen circumstances. We changed our curriculum each day to fit our program to the time that was available, so the children could still do the activity and learn what we hoped they would learn. The YMCA is an environment with a variety of elements that could change at a moment's notice; a student might have to leave the program early due to an unexpected parent pickup or some staff might call in sick. Sometimes the number of children at the program location might be unusually high on some days, causing delays. Hoping such an environment to be consistent at all times is an unreasonable expectation and instead should be observed beforehand to increase preparedness. For example, arriving at the program site ahead of time to help the staff organize the students and have them ready to participate in the program on time. Do not expect everything to work as hypothesized and think about the environment the project is being hosted in.

6.2.2 Meeting your Audience Earlier

This recommendation was taken from some of our conversations with Sarah Levy from the School Age Child Care Services at the YMCA. She enjoyed working with our group, but she mentioned wishing that we met with her and the children earlier. If we met them earlier, there would have been greater probability to develop bonds with the children, which would have made the learning process go smoother. They would already trust us, making the process more enjoyable and comfortable for both the students and ourselves. We would be able to learn about our audience's abilities and life experiences. Since we did not do this, the students called us Mister and Miss for the first couple of visits. In addition, according to both Sarah Levy and our advisor Katherine Chen, inspiration and impressions can also be made through the relationship between the teacher and the student and not just the curriculum and activities.

6.2.3 Curriculum Writing and Extra Activities

As mentioned previously, time constraints were a problem that we were not fully prepared to deal with. Our curriculum was designed so that, for each activity, there was a certain amount of time we expected to have. Each day was designed without much room for delays or errors, which ended up being the biggest mistake in designing the curriculum overall. Our recommendation to avoid this situation is to have a library of actives to choose from depending on the situation. Using this method, it would be possible to make up for lost time by giving shorter activities that teach the same concept. In terms of time lengths for activities, there should be a variety - from around five-minute quick games to 30 or 40 minutes of full-on activities with a debriefing. This method would allow a project team to adapt to situations on the spot and not loose valuable time during their program.

6.2.4 Scope and Backup Plans

Our first obstacle we encountered was shrinking our scope down to have a manageable project. Initially, our curriculum was going to use the embroidery machine with a python library we would create. However, we did not expect the amount of work that was required to have the machine working as needed. So, we went with our backup plan and decided to use the AxiDraw instead. Narrowing down on what the group wants to do, what is required to complete it, and how much time is available are all very important aspects that should be considered during the early stages of a project. Having a backup plan to cover for possible failures, like the embroidery machine, should be thought through and implemented at an early stage. Thinking ahead in this manner will allow for a much smoother execution of the curriculum.

6.2.5 Resources

Writing a curriculum with just research and without experience is difficult. In order to prepare for this, it would be best to make use of any and all resources available. For example, going to the library to check if different projects have already completed a similar project. The librarians can also provide excellent references and research to base your curriculum on. The STEM Education Center at WPI is a solid resource as well. The STEM Education Center focuses on helping teachers incorporate STEM education into their classrooms. They can help with picking education standards based on the audience and the setting. They can also help in narrowing down what activities would best fit the goal of an educational project. The Pre-collegiate Outreach Program Office (POP) puts on STEM programs year-round for K-12 students, especially middle school students. They have faculty, students, and volunteers that are available. Reaching out to the other faculty as well for their experience in creating programs can help a group expand their horizons are create a better curriculum.

7. Conclusion

Our developed curriculum was a series of activities intended to last three week that would teach our audience basic computer science concepts and help raise their interest in Computer Science. In order to do so, we used arts, such as drawing and theatre, as our method of teaching. As seen in our data, there was an overall positive change from our pre-program data to our post-program data. The students self-reported that they understood what computer science is, felt more confident around computers, and even began to enjoy the subjects of math, science, and art a little more! We collected data using a survey and the “Draw a Computer Scientist” activity. Based on our time with the children, we personally feel that they did enjoy a lot of our program. Through our informal setting, we were able to bond with the children and use our teaching approach to introduce them to the Computer Science field.

However, all our expectations and plans did not go as desired. We faced a variety of obstacles on the way and had to adapt almost daily in order to continue our program. We reflected after each YMCA visit on what plan of action we would need to take in order to have our curriculum aligned with out intended learning objectives, all the while keeping in mind our overall goal. While looking back on the obstacles we faced, the idea arose to make an assortment of recommendations that would assist future groups doing a similar project on what and what not to do (Chapter 6.2). Our recommendations will hopefully allow others to avoid potholes we personally experienced.

Our project worked to create an educational program that covered the basics of computer science to a group of younger children. However, we did not want to use a formal classroom method, but rather use a set of hands-on activities that would use a secondary tool, arts, to teach. In our informal learning environment, we were able to interact with the students on a more personal level and foster an enjoyable learning experience.

Our goal was to create a curriculum that would contribute to the effort of increasing interest in computer science in younger children. Based on our data and experience with the children, our group agrees we reached our goal. We hope that future groups will take our pilot program and our recommendations to expand, debug, and recursively run it to continue to affect the lives of younger students.

References

[1] Adams Nager & Robert Atkinson. 2016. The Case for Improving U.S. computer science Education, https://eric.ed.gov/?q=%22computer+science%22&id=EJ1121431

[2] Allen Tucker. 2003. A Model Curriculum for K-12 computer science: Final Report of the ACM K--12 Task Force Curriculum Committee. Computer Science Teachers Association. https://ftp.unpad.ac.id/orari/library/library-ref-eng/ref-eng-3/application/education/curriculum/k12final1022.pdf

[3] Aman Yadav, & John T. Korb. 2012. Learning to teach computer science: the need for a methods course. Communications of the ACM. 55(11), 31-33. https://www.researchgate.net/publication/262401690_Learning_to_Teach_Computer_Science_The_Need_for_a_Methods_Course

[4] Anne. 2018. How to Make a Lightning Bug Paper Circuit Card - Left Brain Craft Brain, https://leftbraincraftbrain.com/how-to-make-a-lightning-bug-paper-circuit-card/

[5] Anon. 24.1. turtle - Turtle graphics. The Python Standard Library. https://docs.python.org/3.3/library/turtle.html?highlight=turtle

[6]Anon. Fortnite - Play Free Now: Official Site: Epic Games https://www.epicgames.com/fortnite/en-US/home

[7] Association of Computing Machinery. Top 10 Reasons to Major in Computing. https://www.acm.org/binaries/content/assets/education/top-10-reasons-to-major-in-computing.pdf

[8] Autonomous Parking - EV3 Coding Activities - Lesson Plans - LEGO Education, https://education.lego.com/en-us/lessons/mindstorms-ev3/autonomous-parking

[9] Blink master. 2018. Arduino: Project Hub. https://create.arduino.cc/projecthub/rmbyers/blink-master-78aa7a?ref=tag&ref_id=education&offset=5

[10] C. Dianne Martin. 2004. Draw a computer scientist. In Working group reports from ITiCSE on Innovation and technology in computer science education (ITiCSE-WGR '04), 11-12. https://doi.org/10.1145/1044550.1041628

[11] Chrrystalla Mouza, Alison Marzocchi, Yi-Cheng, & Lori Pollock Pan . 2016. Development, Implementation, and Outcomes of an Equitable Computer Science After-School Program: Findings From Middle-School Students. Journal of Research on Technology in Education 48, 84-104. https://doi.org/10.1080/15391523.2016.1146561.

[12] Harold Sack. 2018. Seymour Papert's Logo Programming Language. http://scihi.org/seymour-papert-logo-programming-language/

[13] Hillary Swanson, Gabriella Anton, Connor Bain, Michael Horn, & Uri Wilensky. 2017. Computational Thinking in the Science Classroom. Northeastern University. https://ccl.northwestern.edu/2017/CTE2017_Paper_Final_2.0.pdf

[14] Hurd, Ellis, Weilbacher, & Gary. 2017. Developing and Using a Co-Teaching Model within a Middle Level Education Program. https://eric.ed.gov/?q=developing+middle+school+learning&ft=on&id=EJ1191655

[15] Leah Buechley, Mike Eisenberg, Jaime Catchen, & Ali Crockett. 2008. The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08). ACM, New York, NY, USA, 423-432. https://doi.org/10.1145/1357054.1357123

[16] Linda Godzicki, Nicole Godzicki, Mary Krofel, & Racheal Michaels. 2013. Increasing Motivation and Engagement in Elementary and Middle School Students through Technology-Supported Learning Environments. https://eric.ed.gov/?q=middle+school+learning&ft=on&ff1=subTeaching+Methods&ff2=subMiddle+School+Students&ff3=subTechnology+Uses+in+Education&id=ED541343

[17] Little Bins; Little Hands. 2016. Superhero Coding Game Without A Computer!, https://littlebinsforlittlehands.com/superhero-computer-coding-game-without-a-computer/
[1] Logo Foundation. 2014. Logo Programming. (2014). Massachusetts School of Technology (MIT). https://el.media.mit.edu/logo-foundation/what_is_logo/logo_programming.html

[18] Massachusetts Department of Elementary. Massachusetts Learning Standards. Retrieved November 18, 2019 from http://www.doe.mass.edu/frameworks

[19] Michael Mendicino, Leena Razzaq & Neil T. Heffernan (2009) A Comparison of Traditional Homework to Computer-Supported Homework, Journal of Research on Technology in Education, 41:3, 331-359. https://doi.org/10.1080/15391523.2009.10782534

[20] Michelle. (2016). Coding a LEGO Maze. Research Parent: Engineering the Future Through Calculated Parenting. https://researchparent.com/coding-a-lego-maze/

[21] National Research Council. 2003. Beyond Productivity: Information Technology, Innovation, and Creativity. Washington, DC: The National Academies Press. https://doi.org/10.17226/10671.

[22] Samar I. Swaid. 2015. Bringing Computational Thinking to STEM. Procedia Manufacturing. 3, 3657-3662. Education,https://doi.org/10.1016/j.promfg.2015.07.761

[23] Sew Electric. Design your Bracelet. http://sewelectric.org/diy-projects/sparkling-bracelet/design-your-bracelet/

[24] Valerie Barr & Chris Stephenson. 2011. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. ACM Inroads 2, 1 (February 2011), 48-54. DOI=http://dx.doi.org/10.1145/1929887.1929905

[25] Vanessa L. Wyss, Diane Heulskamp, & Cathy Siebert. 2012. Increasing middle school student interest in STEM careers with videos of scientists. International Journal of Environmental and Science Education. 7(4), 501-522. https://eric.ed.gov/?id=EJ997137

[26] Yasmin B. Kafai, Eunkyoung Lee, Kristin Searle, Deborah Fields, Eliot Kaplan, & Debora Lui. 2014. A Crafts-Oriented Approach to Computing in High School: Introducing Computational Concepts, Practices, and Perspectives with Electronic Textiles. Trans. Comput. Educ. 14, 1, Article 1 (March 2014), 20 pages. http://dx.doi.org/10.1145/2576874

[27] Yizhou Qian & James D. Lehman. 2016. Correlates of Success in Introductory Programming: A Study with Middle School Students. Journal of Education and Learning 5, 73-83, https://files.eric.ed.gov/fulltext/EJ1097391.pdf

Appendix A – Curriculums for Various Time Frames (2-3 Weeks)

2 Week Curriculum

Week 1:

Day 1: Introduction of us and Art Assessment Tool

Activity	Description	Time Allocated
Drawing Activity	Draw what a computer scientist looks like If time runs out, take home and bring it for day 2	25 min
Introduction of Theater Activity	Explanation of activity. Students have to write instructions that gets a person through a maze. Maze has 1 simple path.	5-10 min
Writing instruction for Theater Activity	Students write pseudo code/ instructions to get through the maze.	15 min
Acting out Theater Activity	Each group gives the instructions they have written. Instructions are followed by one of us to take away bias	10 min

Day 2: Pixel art

Activity	Description	Time Allocated
Introduction of Activity	Introduction of pixel art. Students are given simple instructions to color a simple smiley face. :)	15 min
Introduction to the engineering process	Students are given a breakdown of the engineering process and planning their code and pattern	10 min
Pixel Art Planning	Students Start making their own Pixel art and writing a code for it	25 min

Week 2:

Day 3: Pixel art continuation

Activity	Description	Time Allocated
Pixel art continuation	Students plan pattern and finish their "code"	25 min
Art Assessment Activity	Students are given a breakdown of the engineering process and planning their code and pattern	30 min

Day 4: Tour of WPI

2 and ½ Week Curriculum

Week 1:

Day 1: Introduction of us and Art Assessment Tool

Activity	Description	Time Allocated
Introductions	Introductions us to the students as well as them introducing themselves to us. Name, Age, and ...	10 -15 min
Drawing Activity	Draw what a computer scientist looks like If time runs out, take home and bring it for day 2	30 min

Day 2: Theater Activity

Activity	Description	Time Allocated
Introduction of Activity	Explanation of activity. Students have to write instructions that gets a person through a maze. Maze has 2 paths they choose from.	5-10 min
Writing instruction	Students write pseudo code/ instructions to get through the maze.	30 min
Play	Each group gives the instructions they have written. Instructions are followed by one of us to take away bias	20 min

Week 2:

Day 3: Pixel art

Activity	Description	Time Allocated
Introduction of Activity	Introduction of pixel art. Students are given simple instructions to color a simple smiley face. :)	15 min
Introduction to the engineering process	Students are given a breakdown of the engineering process and planning their code and pattern	15 min
Pixel Art Planning	Each group gives the instructions they have written. Instructions are followed by one of us to take away bias	20 min

Day 4: Pixel art continuation

Activity	Description	Time Allocated
Pixel art continuation	Students plan pattern and finish their "code"	25 min
Art Assessment Activity	Students are given a breakdown of the engineering process and planning their code and pattern	30 min

Week 3:

WPI Tour and Presentations

3 full week Curriculum

Week 1:

Day 1: Introduction of us and Art Assessment Tool

Activity	Description	Time Allocated
Introductions	Introductions us to the students as well as them introducing themselves to us. Name, Age, and ...	10 -15 min
Drawing Activity	Draw what a computer scientist looks like If time runs out, take home and bring it for day 2	30 min

Day 2: Theater Activity

Activity	Description	Time Allocated
Introduction of Activity	Explanation of activity. Students have to write instructions that gets a person through a maze. Maze has 2 paths they choose from.	5-10 min
Writing instruction	Students write pseudo code/ instructions to get through the maze.	30 min
Play	Each group gives the instructions they have written. Instructions are followed by one of us to take away bias	20 min

Week 2:

Day 3: Pixel art

Activity	Description	Time Allocated
Introduction of Activity	Introduction of pixel art. Students are given simple instructions to color a simple smiley face. :)	15 min
Introduction to the engineering process	Students are given a breakdown of the engineering process and planning their code and pattern	15 min
Pixel Art Planning	Each group gives the instructions they have written. Instructions are followed by one of us to take away bias	20 min

Day 4: Pixel art continuation

Activity	Description	Time Allocated
Pixel art continuation	Students plan pattern and finish their "code"	50 min

Week 3:

Day 5: Pixel Art and Art Assessment

Testing of Pixel Art code	Students are given a different groups code to try to recreate the pattern/ pixel art	25
Art Assessment Activity	Students are given a breakdown of the engineering process and planning their code and pattern	30 min

Day 6: WPI Tour and Presentations

Appendix B - Draw a Computer Scientist Activity

What does a Computer Scientist look like?

Draw a picture of a Computer Scientist. Give your Computer Scientist a name.

Appendix C - Maze Activity

The Maze game

Instructors Script: You are computer scientists who are part of a team that solves the working world's greatest problems. The teams are broken down into three groups. The planning team, designing team and executing team. Each team has a task to complete and must do their individual parts in order to be able to get through the problems. If each team does not complete their tasks successfully, the other teams will not be able to complete their tasks.

Note: So, they can be "one with the characters", students are instructed to come up with a name, age, and story of their characters. They have 10 minutes to come up with their characters. Once the characters are picked, let them know what team they are a part of.

Planner:
As the planning team, your task is to choose the path the team will take to get through the maze. You must come up with different strategies on how to get through the different obstacles within the maze. (Have 10 minutes for planning)

Designer:
As the designing team, your task is to take the path the planning team has made and the strategies they have come up with to create a set of instructions. These instructions have to be detailed. (Have 10 minutes for planning)

Executer:
As the executing team, you are tasked with following the specific instructions the designing team has come up with to get through the maze. You must follow exactly what has been written on the instruction. (Have 10 minutes for planning)

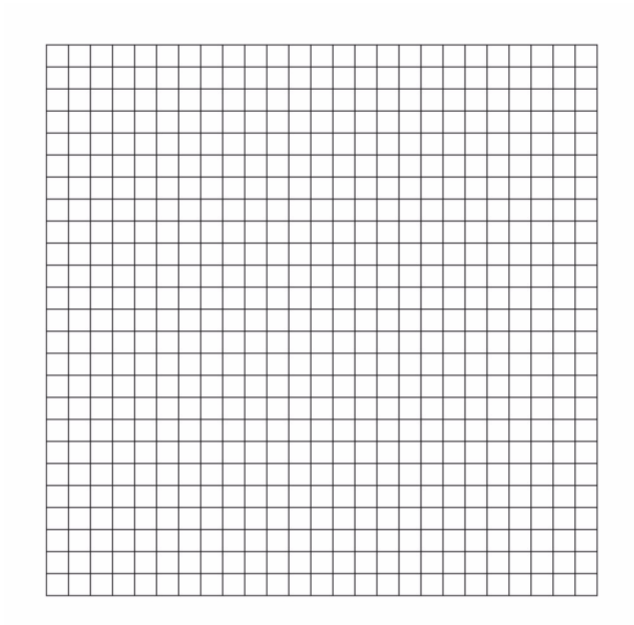
Note: Once the executing team is done, explain the reason behind the game.

Instructor Script (use as reference): In the computer science workplace, it is often that people are assigned to different tasks. For example, some people have to talk to customers who have problems or things they would like. It is their job to get all the things the customers want and give that information to a designing team. For example, if the customer is looking to make a website, the customer would tell the planning team what they would like. The planning team will take those details and put a website together. The designing team will make the basic website and how it looks like. It is the executing team's task to take those beautiful designs and make them work. For example, the planning team might have a navigation bar at the top of the website, the designing team will make it look nice and the executing team will make the buttons work. If they don't work well together, then the team can not successfully complete their assignment or tasks. They also will not be able to create a great solution.

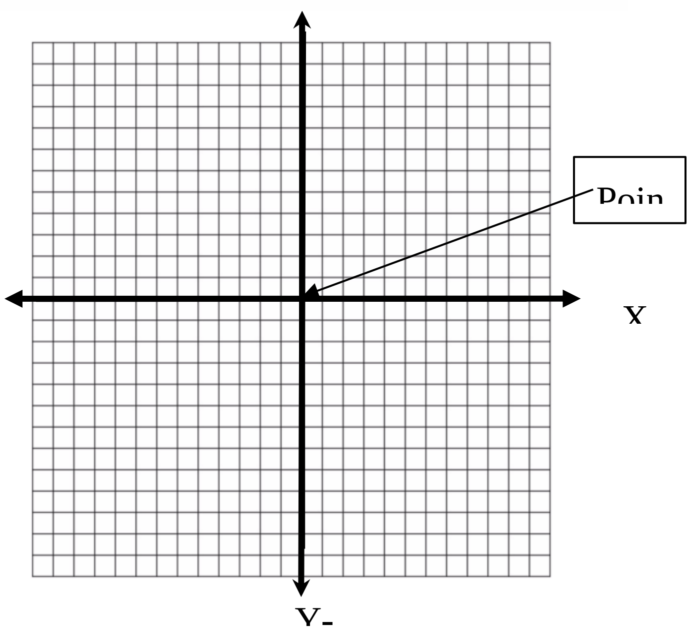
Appendix D - Pixel Art Activity

Pixel art

For this activity, you will be following a set of instructions in order to make an art piece. Please read carefully and ask one of your instructors any questions you have.



You are given a 25 by 25 grid. You will be creating the artwork on this grid. Treat it like it's an X and Y coordinate



First, identify the point of origin (0,0). This is the point at which you will start your drawings. It is the center point of the grid.

From there, you will be given instructions or coordinates. Follow those coordinates to be able to create your artwork.

Instructions and meaning

(1,1) Move **1 block up** and **1 block to the right** of the origin point

(-1,1) Move **1 block down** and **1 block to the right** of the origin point

(1,-1) Move **1 up down** and **1 block to the left** of the origin point

(-1,-1) Move **1 block down** and **1 block to the left** of the origin point

Pen down: color the block identified

Pen up: move to the next instruction, but do not color the block

Task

Your task today is to follow the set of instructions/coordinates to complete half of the pixel art. Once you complete that, your next task is to create instructions for the other half that is symmetrical to the one you have already drawn.

After you have completed your task, let your instructors know. After everyone is done, you will swap with another person so they can follow your instructions and complete your other half.

Challenge 1: (1,6) (2,7) (3,8) (4,9) (5,10) (6,10) (7,10) (8,10) (9,9) (10,8) (11,7) (11,6) (11,5) (11,4) (11,3) (11,2) (11,1) (10,-1) (10,-2) (9,-3) (8,-4) (7,-5) (6,-6) (5,-7) (4,-8) (3,-9((2,-10) (1,-11)

Challenge 2:
(1,1) (1,2) (1,3) (1,4) (1,5) (1,6) (1,7) (1,8) (1,9) (2,9) (3,9) (4,9) (5,9) (6,9) (7,9) (8,9) (9,9) (10,8) (11,7) (12,6) (12,5) (11,5) (11,4) (10,3) (10,4) (9,2) (8,1) (7,-1) (6,-2) (5,-3) (4,-4) (3,-5) (2,-6) (1,-6) (1,-7) (1,-6) (1,-5) (1,-4) (1,-3) (1,-2) (1,-1) (2,5) (3,5) (4,5) (5,5) (6,5) (7,5) (8,5) (9,5) (10,5) (2,-3) (3,-2) (4,-1) (4,1) (5,2) (6,3) (7,3)

Appendix E - Consent Form

Informed Consent Agreement for Participation in a Research Study

Investigators:
Matthew Figueroa mtfigueroa@wpi.edu (718)304-6780
Neer Jariwala njariwala@wpi.edu (508)446-3686
Metasebia Dejene madejene@wpi.edu (857)350-2422
Advisors: Gillian Smith gmsmith@wpi.edu and Katherine Chen kcchen@wpi.edu

Title of Research Study: Computational Craft Education

Introduction. We're a group of students from Worcester Polytechnic Institute working on our Interactive Qualifying Project, and we would like your child to participate in our computer science curriculum.

Purpose of the study. The purpose of this study is to see if this new way of teaching computer science is effective in showing the basics and promoting some interest in computer science in the future. Your child will be working with us in creating art projects and learning about technology with a new outlook.

Procedures to be followed. We have a curriculum set up for the next 2-3 weeks that will involve your child's participation in art activities that will be 1-hour long sessions twice a week. There will be YMCA staff on deck to help us in this, and their job is to help us teach and mediate. Your children will be asked to participate in the following task:

At the YMCA:

- Focus groups/group interviews as introductions
- Participate in art activities with computer science ideas
- Go through a breakdown of what they learned throughout the curriculum
- Post program group interviews/surveys

At WPI:

- Draw in a program called Inkscape
- Observe and analyze a robot creating their drawing

Risks to study participants. As for the risk in this study, there is no risk to your child participating in this study. The YMCA staff has volunteered to help us complete the study and will be facilitating the curriculum.

Benefits to research participants and others. The benefits for the study will be a fun art program that your child can participate in, and they will have a project to take home with them at the end. Your child will also learn some basic computer programming skills, such as algorithms and problem solving.

Record keeping and confidentiality. Records of your child participation in this study will be held confidential so far as permitted by law. However, the study investigators, the YMCA, and, under certain circumstances, the Worcester Polytechnic Institute Institutional Review Board

(WPI IRB) will be able to inspect and have access to confidential data that identify your child by name. Any publication or presentation of the data will not identify your child.
You do not give up any of your legal rights by signing this statement.

For more information about this research or about the rights of research participants, or in case of research-related injury, contact: study investigators (refer to the top of the form)
IRB Chair (Professor Kent Rissmiller, Tel. 508-831-5019, Email: kjr@wpi.edu)
Human Protection Administrator (Gabriel Johnson, Tel. 508-831-4989, Email: gjohnson@wpi.edu)

Your participation in this research is voluntary. Your child's refusal to participate will not result in any penalty to you or any loss of benefits to which your child may otherwise be entitled. Your child may decide to stop participating in the research at any time without penalty or loss of other benefits. The project investigators retain the right to cancel or postpone the experimental procedures at any time they see fit. If your child doesn't want to participate, the YMCA will be offering activities during this time because we will be running this simultaneously with everything else they do.

By signing below, you acknowledge that you have been informed about and consent to be a participant in the study described above. Make sure that your questions are answered to your satisfaction before signing. You are entitled to retain a copy of this consent agreement.

Please check the appropriate boxes for each location you consent for child to participate at:

- ❏ YMCA
- ❏ WPI

Parent/Guardian Signature:___________________________

Date: ___________________

Parent/Guardian Name (Please print): ______________________________

Signature of Person who explained this study: __________________________

Date: _______________

Appendix F - Survey

Please circle a number on the scale that shows you how much you agree or disagree with the statement

		Strongly Disagree	Somewhat Disagree	Somewhat Agree	Strongly Agree
1.	I am comfortable working with a computer.	1	2	3	4
2.	I know what computer science is.	1	2	3	4
3.	I want to know how a computer works.	1	2	3	4
4.	I know what it means to write a program.	1	2	3	4
5.	I think art and computer science do not have very much to do with each other.	1	2	3	4
6.	I enjoy doing art activities more than normal classes.	1	2	3	4
7.	Learning something new can sometimes be fun.	1	2	3	4
8.	I like math and science.	1	2	3	4