Spring 4-13-2020

Engagement and Computational Thinking through Creative Coding

Dana Hoppe
University of Nebraska - Lincoln

ENGAGEMENT AND COMPUTATIONAL THINKING THROUGH CREATIVE CODING

by

Dana Hoppe

A THESIS

Submitted in Partial fulfillment of

University Honors Program Requirements

University of Nebraska-Lincoln

Bachelor of Science

Computer Science

College of Arts and Sciences

April 10, 2020

Faculty Mentors:

Leen-Kiat Soh, PhD, Department of Computer Science and Engineering

Megan Elliot, Director, Johnny Carson Center

for Emerging Media Arts

Abstract

Rising enrollments in Computer Science pose an opportunity to engage students from diverse backgrounds and interests; and a challenge to deliver on positive learning outcomes. While student engagement is the driving factor for increased learning performance and retention, it has been declining to new lows for Computer Science students in recent years. In order to further explore the potential of contextualized computing as a tool for increasing engagement in computing and developing Computational Thinking aptitude in students, we have developed an introductory computing course contextualized with Art and Design with modules centered around guiding pedagogical principles and aimed at middle school and high school students. This Creative Coding course utilizes a principle and theory-based approach in its design in order to meet the pedagogical goals of this project. Additionally, we have developed data collection tools for analyzing the effectiveness of the course in increasing student engagement and Computational Thinking aptitude. Results coming from limited implementation at two sites suggests that with future implementation, data collection, and material refinement, Creative Coding can serve as a flexible tool for increasing student engagement and as a basis for further research into contextualized computing.

key words: computer science, education, engagement, enrollment, art, design, contextualized, introductory, computing, computational thinking, creative coding, principle-based design.

ACKNOWLEDGMENTS

As an undergraduate student, I often felt lost in the endless maze of academia, searching for clarity in the topics which interested me. I found it difficult to develop continuity between the areas of study for which I had passion. It was only through my research that I was able to shine a light on the ambiguous region that lay between the creativity of Art and the logic of Computer Science and Mathematics.

I wish to express my sincerest gratitude to my advisor, Dr. Leen-Kiat Soh, for encouraging me to pursue this area of research and for lending his expertise and guidance along the way. This project would not have been possible without his unfailing support and commitment.

I would like to thank my co-adviser, Megan Elliot, for sharing her breadth of experience, fantastic humor, and network of talented individuals. She has been incredibly helpful in directing this project towards the frontline in interdisciplinary collaboration and endeavors.

I would also like to take this opportunity to thank Gretchen Larsen for lending support in the area of creative computational education. She has continually offered ideas, perspectives, advice, and experience – and always with a reminder of the importance of human empathy in education.

Finally, I would like to express my appreciation to everyone who assisted this project along the way, including Initialize UNL, CSTA Nebraska, Rich Powers, Dr. Ann Koopman, Colton Harper and countless others who have supported me in large and small ways.

I dedicate this thesis to my parents Fred and Donna Hoppe, for their love and support. They nurtured my curiosity and love for learning that allowed me to get to where I am now.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ENGAGEMENT AND COMPUTATIONAL THINKING THROUGH CREATIVE CODING

1 Introduction

1.1 Problem

There is without a doubt an increasing number of students entering the Computing field. Increased enrollment means that more diverse students in terms of background, motivation, and interests are enrolling in their first introductory computing courses including students from underrepresented groups and non-computing majors [*Assessing and Responding* 2018]. These new students, with their new and varied set of interests, talents, and perspectives pose a challenge and opportunity for educators to deliver on engagement, produce positive learning outcomes and retain high percentages of students. Unfortunately, student engagement experiences in Computer Science have been dismally low across the board with marked decline in the past four years [Morgan et al. 2019]. Given that there has been a rise in the proportion of students with non-computing majors enrolling in Computer Science courses across the board, there is a clear need for increasing engagement among individuals who are interested and skilled in domains beyond computing. Additionally, retention of student groups typically underrepresented in computing is especially damaged by decreased engagement. The trend in lack of engagement points to a deficiency in material inclusive enough to engage these students in computing.

1.2 Why is the Problem Important?

A more inclusive pipeline from primary to secondary to higher education needs to be created in which individuals from diverse perspectives, interests, and backgrounds will feel that there is a place for them in the field of computing. Increasing engagement is key to ensuring an effective education in computing because it increases learning performance among students [Chen 2017]. Increased learning performance in turn increases student retention and future performance in the field [Chen et al. 2008]. Allowing student engagement to decline puts students at risk of lower learning performance, decreased retention, and overall lower performance in the field. This could disproportionally affect students from underrepresented backgrounds and interests beyond computing because of outdated conceptions and stereotypes of who the average computing student is. Decreased engagement and learning performance among these groups puts them at a further disadvantage by leaving them with an inadequate understanding of the fundamental concepts of Computer Science and underdeveloped aptitude in Computational Thinking that will hinder them down the road. Computer Science and the underlying principles of Computational Thinking are both relevant and applicable to solving problems and understanding concepts from a broad range of domains [Wing 2006]. The idea of contextualizing computing within the context of various domains has been introduced to the educational realm as a promising method for engaging students [Cassel & Wolz 2013]. Specifically, computing contextualized within the context of interactive media has demonstrated effectiveness [Guzdial 2004]. The dire need for increased engagement engenders further research into the effectiveness of potential solutions such as contextualized computing.

1.3 What is Missing from the Literature?

Efforts have been successfully made to create introductory computing courses contextualized within visual arts with the motive of increasing student engagement [Guzdial 2004, Cassel & Wolz 2013]. The most recent effort developed a Creative Coding course based in Processing, an open source programming language designed for creating interactive visual media through code. This program has been implemented at the collegiate level [Greenburg et al. 2012] and at the high school level [Xu et al. 2018] with qualitatively positive results. While these efforts are significant and have demonstrated the possibility of contextualizing introductory concepts of Computer Science into an interdisciplinary program with Art and Design, the curriculum design and methodology does not provide a pedagogical background or follow a principle and theory-based approach to developing material. Additionally, to our knowledge, there is no data collection tool for measuring curriculum effectiveness in increasing student engagement or Computational Thinking (CT) aptitude. These are important next steps in determining whether implementation of contextualized computing on a wider scale is justified. Developing curriculum with a principle and theory-based approach would ensure that the material is directed towards achieving the pedagogical goals of the program. In order to measure the effectiveness in increasing CT aptitude and engagement, there must be a tool for gathering and analyzing data.

1.4 Overview of Proposed Solution

In order to further the development of contextualized computing as a tool for increasing engagement in computing and developing Computational Thinking aptitude in students, we have proposed the development and implementation of an introductory computing course contextualized with Art and Design with modules centered around the principles of Computational Thinking and aimed at middle school and high school students. This program would iterate on and implement elements from previous studies with four design aspects. First, in order to enable ***ease of use in program implementation*** for the teacher and help the students see programming syntax as less intimidating and more accessible, we use the Processing language. Second, to engage students, their teachers and parents, this program will ensure that the students are given creative outlets to make ***artwork as a tangible product*** that can be taken home or displayed. Third, as the goal of the program is to increase engagement and CT aptitude among students, and similar solutions have no quantifiable data relating to these features which can be utilized, we additionally propose developing ***measurement tools*** that would enable us to quantify these measures and compare changes over the implementation of the program. This would be most effective via a pre and post survey that measures student interest, self-efficacy, and CT aptitude. Fourth, learning module development will be centered on ***principled and theory-based design*** in which guiding principles are described and utilized in the design process including Introductory Computing, Computational Thinking, Engagement, and Art and Design. These principles provide a depth of background and theory that is not seen in similar attempts to develop a Creative Computing curriculum.

1.5 Contributions

Contributions of this project involve three aspects. First, we contribute to ***contextualized computing*** towards improving CS education through the development and implementation of this program and its measurement tools. By creating a middle school/high school introductory computing course built around CT principles and contextualized within Art/Design, we are giving students an introduction to the field of computing designed to engage them no matter what their interests and backgrounds are while simultaneously giving them a foundation in Computational Thinking. More specifically, the five-module program is ready for implementation and can be adapted to a wide range of student experiences. The measurement tools will benefit the growth of this area by enabling further research and development to evaluate the effectiveness of future implementations, iterations, and derivatives of the program. This allows for programs contextualized within different contexts such as physics, math, business, or biology to be created and evaluated based on a pre-existing format which includes pre-existing measurement instruments. Second, this ***research is IRB approved***, which clears the way for future work in the area. Finally, we have also written a ***SWOT analysis*** based on our experience in the design, implementation, and evaluation of the program.

1.6 Overview of Thesis

In Chapter 2 we will first cover the related work and background regarding research that supports or is related to our area of focus. This includes work relating to Computational Thinking, Creativity in Computing, student engagement, contextualized computing, and similar Creative Coding educational programs. We twill then outline and explain our methodology in Chapter 3 for designing our modules around CT principles and engagement. In Chapter 4, we will give an overview of the implementation and deployment of the program. Then, in Chapter 5 we will give an overview of the results including our SWOT analysis. Finally, in Chapter 6, we will summarize our progress and give suggestions for future work. References are listed in Chapter 7 and all supporting materials including student artwork are included in the Appendix Chapter 8.

2 Related Work

The areas of research drawn upon and incorporated into this paper can be divided into four main areas. The first area involves work regarding increased engagement and the resulting rise in students from diverse and non-computing backgrounds. The second area is related to student engagement and its relationship to learning outcomes and retention. Third, we provide background pertaining to Computational Thinking and its importance to education in computing and beyond. The fourth area reports on progress made in the area of contextualized computing and its relationship to engagements and CT. Finally, we discuss research regarding Art, Design, and Creativity in relation to contextualized computing education.

2.1 Increased Enrollment

Computer Science enrollment has always been of significant interest considering that Computer Science is the only major discipline where the rate of bachelor's degree production falls short of the projected number of job openings [Freeman, 2016]. The landscape of Computer Science enrollment has followed a cyclical pattern for the last five decades with typically a decade long interval between successive peaks and low points [Roberts 2011]. The current surge in enrollments is consistent with this trend, and therefore not guaranteed to last if it follows historical patterns. According to *Assessing and Responding to the Growth of Computer Science Undergraduate Enrollments* from the National Academies of Sciences, Engineering, and Medicine, the number of undergraduate students majoring in CS more than tripled between 2006 and 2015, surpassing the enrollment surge experienced in the dot-com peak in 2001 [*Assessing and Responding* 2018].

This increase in enrollment, when broken down into identifiable groups, reveals interesting new needs and opportunities. Non-majors in particular experienced enrollment increases at higher rates on average than majors, with a 184% increase for intro-level courses for majors, a 265% increase for mid-level courses, and a 146% for upper-level courses at doctoral-granting institutions [CRA, 2017], as shown in Figure 1 below. This represents a dramatic, but not surprising increase, as a greater number of jobs in all fields expect applicants to have a significant level of computing experience [*Assessing and Responding* 2018].

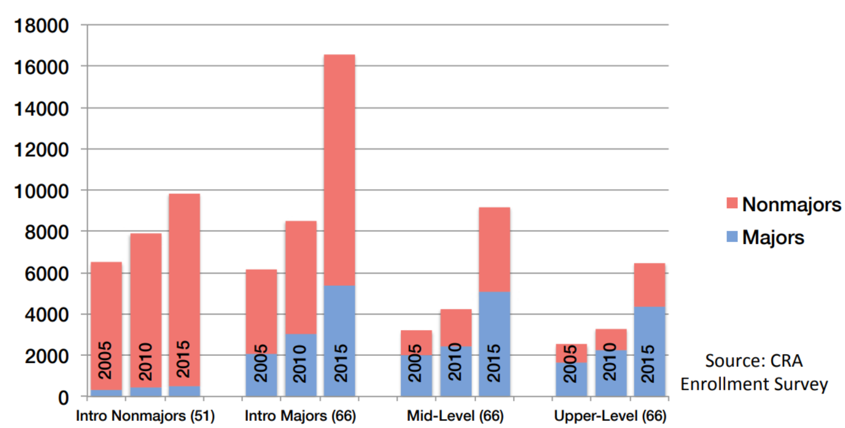


Figure 1. *Enrollment Rates for Majors and Nonmajors in CS Courses. The number in parenthesis represent the number of institutions participating. (from [CRA, 2017]).*

When it comes to students who are female or from underrepresented backgrounds (black, Hispanic/Latino, and American Indian/Alaska Native), the surge in enrollment has mixed interpretations. Females experienced a total increase in enrollment but experienced a smaller initial growth than males in comparison to previous enrollment surges historically [*Assessing and Responding* 2018]. Underrepresented enrollments have increased in total, as shown in Figure 2. This comes with an increase in degree shares for Hispanic/Latino but a decrease in degree shares among black, American Indian/Alaska Native students [*Assessing and Responding* 2018].

	2012-2013	2013-2014	2014-2015	2015-2016
Total enrollment	*43,391*	*72,447*	*90,604*	*104,634*
Women	13.9%	15.3%	16.5%	18.3%
American Indian or Alaska Native	0.4%	0.4%	0.4%	0.1%
Native Hawaiian or other Pacific Islander	0.2%	0.2%	0.3%	0.3%
Multiracial not Hispanic	2.1%	2.7%	2.8%	3.0%
Black or African American	4.9%	5.0%	5.4%	4.8%
Hispanic, any race	7.8%	8.8%	9.0%	9.5%
Asian	16.1%	19.5%	21.6%	23.8%
White	58.9%	54.4%	51.0%	47.4%
Nonresident alien	9.6%	8.9%	9.5%	11.0%

NOTE: Students with no or unknown race/ethnicity listed were excluded from the total in calculation of the corresponding percentages. SOURCE: CRA Taulbee Survey.

Figure 2. *Enrollment Shares for Various Groups. (from [Assessing and Responding 2018])*

In summary, this data points to a dramatic shift in the volume of non-majors entering the field and uncertainty regarding the recruitment and retainment of women and underrepresented individuals in CS. This is information that k-12 educators should be aware as they introduce their students to the field of computing. Efforts at increasing engagement for the growing number of students interested in computing should be especially mindful that these students come from many different backgrounds and may not end up as CS majors. It is also important to keep in mind that, while they may not pursue computing as their future career, they will most likely need computing for the field they do pursue. In this thesis, we are therefore concerned with maximizing engagement in an introductory computing course for middle school and high school students such that students from a diverse range of backgrounds and interests will be retained.

2.2 Engagement

Student engagement is a critical aspect of the education process. Engagement increases are linked to increases in learning performance amongst students [Chen 2017]. Additionally, increased learning performance has been found to increase student retention and future performance in the field [Chen et al. 2008]. Consequently, increased engagement leads to increased learning performance and retention for students. Engagement among CS students, however, performs lower than all other STEM majors in engagement metrics [Morgan et al. 2019], as shown in

Figure 3. Metrics relating to engagement areas such as Reflective and Integrative Learning, Learning Strategies, and Student-Faculty Interaction were especially low [Morgan et al. 2019]. The data suggests that not only is CS engagement low, it is *steadily decreasing over time* and creating a broader gap between itself and other disciplines.

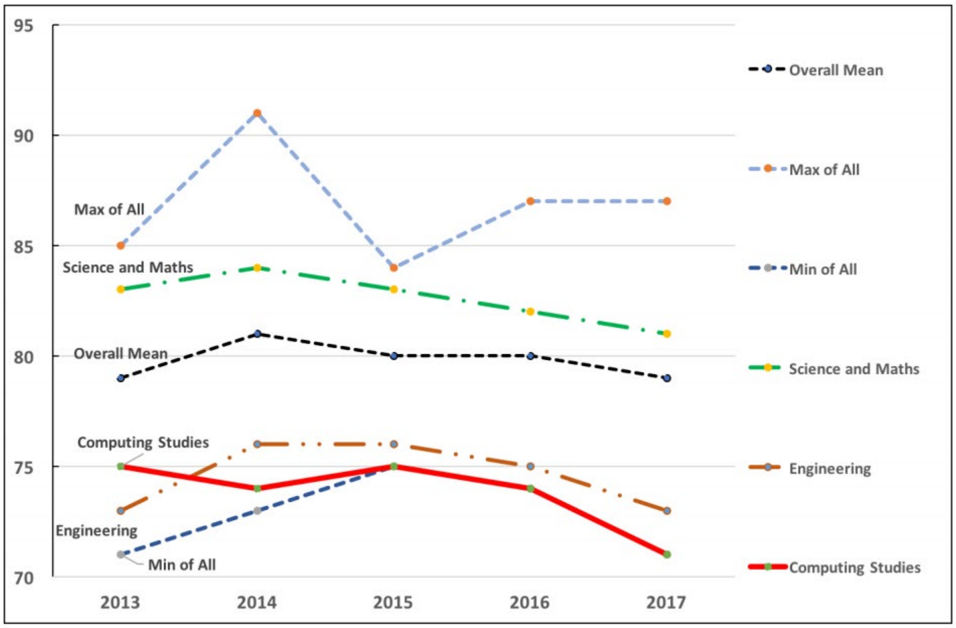


Figure 3. *Engagement Metrics for STEM Students (from [Morgan et al. 2019])*

A concerted effort must be made to improve student engagement, which this thesis aims to do. Fortunately, much work has been done in surveying teachers, students, and research to determine effective strategies for improving student engagement. [Parsons et al. 2011] points to six common elements found amongst the research. First, **Interaction** includes dialogue and conversation between students and with individuals in the community and or experts in the field. Second, **Exploration** gives students the opportunity to ask their own questions and find their own results and answers through risk taking and boundary breaking. Third, **Relevancy** is found in student work that relates to students' future goals, real world experience, or community. Fourth, **Multimedia** experiences integrate modern technology into the learning process. Fifth, **Instruction** must be challenging and include the student as part of the learning process. Finally, **Authentic Assessment** allows for student and instructor co-assessment of work based on criteria chosen to guide the learning process [Parsons et al. 2011]. Parsons analyzes the research on each of these common elements and synthesizes it with research regarding "epistemic culture" and "intellectual engagement" to determine that *embedded collaboration, integrated technology, inquiry-based learning, assessment for learning, and making learning interdisciplinary and relevant to real life are the key areas to improving student engagement* [Parsons et al. 2011]. In this thesis, we use a set of Principles of Engagement based upon these key areas for improving student engagement. These principles have been used in the design and development of course materials in order to maximize the potential for increasing student engagement.

While improving engagement generally in education makes sense in theory, it is another thing to apply it to computing curriculum effectively. Mark Guzdial created an introductory computing course contextualized in media with the goal of increasing retention among students [Guzdial 2003]. After years of implementation and drastically increased retention, Guzdial hypothesized that retention was driven by a sense of relevancy which was expressed as a reason for the success of the course by students [Guzdial 2013]. As seen in engagement research, relevancy is a key element of engagement, and it follows that if increased relevancy and thereby increased engagement was achieved, then increased retention would be an ultimate outcome [Chen et al., 2008, Chen 2017, Parsons et al. 2011]. The media computation course therefore suggests the *potential of contextualized computing in increasing engagement and consequently retention*. Thus, we have determined that contextualizing our introductory course within the context of a topic (discussed in Section 2.4) that supports a sense of relevancy will potentially increase student engagement.

2.3 Computational Thinking

Beginning with Jeanette Wing's 2006 article on Computational Thinking, CS education research has begun to focus on defining Computational Thinking and understanding its potential roles in the pedagogy. Computational Thinking is not Computer Science, it is a fundamental, rather than rote, skill which humans, rather than computers, use to conceptualize and solve problems via multiple levels of abstraction [Wing 2006]. It is a way of thinking for approaching problems that arise conceptually as well as in reality and "includes algorithmic thinking and parallel thinking, which in turn engage other kinds of thought processes, e.g., compositional reasoning, pattern matching, procedural thinking, and recursive thinking" [Wing 2011]. Google's Overview of CT includes a range of skills involved in CT such as logical data organization, analyzation, and representation; problem generalization, abstraction, and decomposition; and solution automation through algorithmic thinking [Google]. Most definitions agree that CT requires thinking about a problem through multiple levels of abstraction and taking an algorithmic approach in order to break it down into coherent sub-problems, recognize patterns, represent and analyze data, generalize solutions, and maximize efficiency.

Computational Thinking has beneficial applications that extend far beyond computing. It "represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use'' [Wing 2006]. While essential to developing solutions in Computer Science, CT can be applied across all disciplines including math, science, and the humanities [Google]. It is used in everyday life such as when we abstract our day into blocks of time in a schedule or recognize a pattern in the number of acquaintances we see at different coffee shops. Many CT elements are found in mathematical, engineering, and design thinking which demonstrates the ubiquitous nature of CT and serves to extend each of these areas into the information processing realm [Grover et al. 2013].

The idea that Computational Thinking is ubiquitous and offers high value to fields beyond computing naturally leads to and complements the concept of contextualized computing education. In this thesis, we recognize the importance of Computational Thinking to the field of computing and beyond and have determined that the development of Computational Thinking aptitude should be an intrinsic component in an introductory computing course for middle school and high school students. Furthermore, the ubiquitous nature of Computational Thinking suggests that the utilization of contextualized computing in this introductory course will contribute to student recognition of the universality and saliency of CT principles.

2.4 Contextualized Computing

Contextualized computing describes any form of computing within the context of a specific domain area, e.g. a physics simulation engine is computing within the context of physics, and financial guidance software is computing within the context of finance. As contextualized computing involves combining computing with another domain area, it is inherently interdisciplinary. Contextualizing computing within educational curriculum has been growing as a strategy for achieving interdisciplinary benefits and increased retainment [Guzdial 2008, Greenberg et al. 2012, and Xu et al. 2018]. It is important to know when to use contextualized computing in education, as it is harmful when it obscures essential information but is beneficial when it motivates students to learn the course material and retention is the primary goal [Guzdial 2010]. As this thesis is focused on the development of engaging (motivating) material and, in turn, retention of students, it is appropriate to use contextualized computing in our introductory course.

The most prominent efforts in contextualized computing begin with Guzdial (mentioned earlier in Section 2.2) and his Media Computation course. The course teaches introductory computing concepts through the manipulation of digital media such as sound, image, and video in Java and Python [Guzdial 2008]. Over the course of six years at Georgia Tech, the course successfully improved retention rates among students in general and among female students had a high enrollment rate along with a success rate as high or higher than the male students [Guzdial 2008]. This success is well documented and begs the question to what the key factors were in increasing student engagement. Guzdial found that students, especially female, were motivated by the perception that the course material provided an **opportunity to be creative** and conveyed a **sense of relevancy**. It was hypothesized that in addition to providing creative outlets, the material used a *consistent application context* which students could focus on, build upon, and most importantly, see themselves utilizing in the future [Guzdial 2008].

Building on this research, a Creative Coding course for CS1 was developed and implemented at Bryn Mawr College and Southern Methodist University (SMU) with the goal to 'present computing as a medium of creativity and nurture an accessible, engaging environment that attracts a modern, diverse student body that appreciates the excitement, creativity, and innovation that computing brings' [Greenburg et al. 2012]. The course 'draws on theory and methodology from computer science and engineering with aesthetic principles, creative practice and pedagogical approaches from the fine and graphic arts' by iterating on John Maeda's pioneering work in recontextualizing computing within the context of the visual arts [Xu et al. 2018]. This includes the use of *Processing*, an open source programming language initiated by Ben Fry and Casey Reas (processing.org) designed for creating interactive visual media through code, which provides a simplified syntax and supports accessible programming in Java, Python, and JavaScript.

In addition to Bryn Mawr and SMU, the course was adapted to the high school level and implemented at two high schools with promising feedback from educators and students including portfolios of student work [Xu et al. 2018]. Though no quantifiable data was produced from implementation, the successful implementation at the high school level and the developed course material, background, and theory in Creative Coding provides a strong basis for future work.

In this thesis, we have determined that contextualization of our introductory computing course will be beneficial to meeting the goal of increased engagement, especially for underrepresented demographics such as women. Additionally, contextualized computing contributes to student recognition of the universality and saliency of Computational Thinking principles. Course material will use a *consistent application context* which students can engage with beyond the course to effectively convey a **sense of relevancy**. In order to provide students with a **creative outlet**, course material has been contextualized within the context of Art and Design in the form of a Creative Coding course, thereby building upon the work of Xu et al. Finally, specific focus has been devoted to the development of underlying theory and pedagogy of contextualized material.

2.5 Art, Design, and Creativity

Creative Thinking includes *flexible, imaginative and innovative cognitive competencies* that (1) are universally applicable and valuable in any context and (2) *complement and expand* Computational Thinking competencies through new perspectives and novel approaches [Soh et al. 2014]. While all domains require creativity, it is most frequently understood, exercised, and taught in the domain of Art and Design. Note, however, the pedagogy of Art and Design has suffered from a lack of critical research into its underpinning theories [Oxman 2001]. Educators searching for pedagogical groundwork in Art and Design would find sparse guidelines or precedent [Lerner 2012].

Looking back at the history of Art and Design pedagogy, the first structured pedagogical approach to teaching introductory design based upon ordered methods and principles was the Preliminary Course (known as *Vorkurs*) at the Bauhaus School (1919 – 1933) [Esen 2018]. This course is what the general introductory art course taught today is based upon, often called Foundations of Art and Design or Basic Art/Design Education [Esen 2018, Lerner 2012]. Designed and directed by Johannes Itten and supplemented by Vassily Kandinsky and Paul Klee, the Preliminary Course was focused on developing a firm grounding in form and color through practical exercise [Casciato 2020]. These practical exercises were intended to develop a universal visual-spatial literacy that could be applied creatively in any given domain [Lerner 2012]. Given this utility and universality, Lerner posits that the theory and practice of the Preliminary Course offers a basis for an Art and Design pedagogy where 'universal visual-spatial language mastery through studio laboratory exploration' is applied creatively in new and unspecified domains [Lerner 2012].

In the Preliminary Course, the concepts of form and color are further broken down into sub-categories and topics: (form) - point, line, and plane; and (color) - spectrum, contrasts, harmony, mixing, form and color, spatial effects, expression/impression theory, and composition [Itten 1964]. In this thesis, these topics (elaborated on in Section 3.1.4.) will be used as guiding principles of Art and Design in the development and contextualization of the introductory computing course. Furthermore, lesson activities draw on exercises, problem-solving processes and critique/presentation methods used in the Preliminary Course.

3 Methodology

To design and develop an introductory computing course which increases engagement as well as computational aptitude among middle school and high school students, as introduced in Chapter 1, we have used guiding principles and techniques relevant to each of our goals. These principles and techniques, as described and listed below, serve as the pedagogical framework which each lesson is built around.

In this Chapter, we first describe the design principles utilized in program development in Section 3.1. This includes principles from Introductory Computing, Computational Thinking, Engagement, and Art and Design. Then, we give a general outline of the lesson modules, including common elements, design principles implemented, and module progression in Section 3.2. Finally, we explain the motivation for the different sections of the survey developed and give a brief description of each portion in Section 3.3.

3.1 Design Principles

3.1.1 Introductory Computing Principles

First, as an introductory computing course, the fundamental concepts of Computer Science were set as the backbone of the lesson structure. These concepts are derived from the CSTA Standards for K-12 Computer Science education.

Basic concepts:

- Variables, data types, expression
    - Creating clearly named variables that represent different data types and performing operations on their values.
    - Creating programs that use variables to store and modify data.
- Arrays and Lists
    - Using lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
    - Manipulating two dimensional arrays.
- Mouse/keyboard interactions
- Control structures: loops and conditionals
    - Creating programs that include sequences, events, loops, and conditionals.
    - Writing functions with control structures
- Recursion, recursive functions

Advanced concepts:

- Simple objects, super and subclasses, object instantiation vs class definition
    - Abstracting to hide implementation details.
- OOP principles, inheritance, abstract classes and interfaces
    - Decomposing problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

- Data sets, data mining, data visualization
    - Creating interactive data visualizations to help others better understand real-world phenomena.
    - Creating computational models that represent different elements of data collected from a phenomenon or process.
    - Collecting data using computational tools and transform the data to make it more useful and reliable.

Applying CS Concepts:

- Math applications – polar coordinates, trigonometry, rotational geometry
- Randomization
- Computing artifacts
    - Creating artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
    - Creating artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
    - Designing and developing computational artifacts working in team roles using collaborative tools.
- Program and algorithm design
    - Using flowcharts and/or pseudocode to address complex problems as algorithms.
    - Modifying, remixing, or incorporating portions of an existing program into one's own work, to develop something new or add more advanced features.
- Impacts of computing
    - Evaluating how computing impacts personal, ethical, social, economic, and cultural practices.
    - Demonstrating how a given algorithm applies to problems across disciplines.

3.1.2 Computational Thinking Principles

We took into consideration the information provided from many different studies and sources regarding Computational Thinking and its implementation in order to determine the principles to focus on and the best methods for ensuring they were integrated effectively into our modules. There are six core Computational Thinking skills we have chosen to center our design around [Soh et al. 2014]:

1. **Problem Decomposition** – Breaking down a problem into more manageable subproblems
2. **Generalization** – Recognizing general methods of features that apply across multiple processes
3. **Abstraction** – Observing big picture details of a problem and ignoring details that get in the way of objectives in order to focus on only a small set of important details and features
4. **Evaluation** – Evaluating how well a solution meets problem requirements - speed, efficiency, space, accuracy, etc.

5. **Algorithmic Design/Thinking** – Creating a sequence of steps from input to output, i.e. instructions that work within parameters to accomplish task (e.g. recipe, assembly manuals, etc.)
6. **Pattern Recognition** – Looking for common patterns between different sets of data (e.g. traffic patterns, weather patterns, etc.)

3.1.3 Engagement Principles

As covered in Section 2.2, research indicates that embedded collaboration, integrated technology, inquiry-based learning, assessment for learning, and making learning interdisciplinary and relevant to real life are frequently identified as being key to improving student engagement [Parsons et al. 2011]. Parsons and Taylor point to five aspects which "successful, student-engaging classrooms combine" [Parsons et al. 2011]. We have integrated these five aspects with the key areas and techniques identified in the study to create a list of seven engagement principles for our modules:

1. **Interdisciplinary** – Informing students of the current state and growing knowledge bases of different subject disciplines for them to see conceptual connections across disciplines. Utilizing transfer thinking by asking students to look for wider relevance and application (explicitly discussing where current learning could be useful: What else could we do with this? Where else would this be useful knowledge?)
2. **Relevant** – Using problems or projects which are real, relevant, and make a positive difference in some way by presenting real life feedback for students and or a connection to their community. Students can see how what they are doing applies to 'real life'.
3. **Technology Integration** – Promoting digital literacy and exposing students to multi-media digital environments and tools that allow for new knowledge building opportunities; supporting a learning ecology which allows students to connect to and create broader learning communities via technology.
4. **Assessment Learning** – Using assessment criteria that is intended to guide and improve learning, is sensitive to emotional impact, develops student self-assessment, and recognizes all levels of student achievement.
5. **Embedded Collaboration** – Embedding opportunities and situations where students are working together to plan, research, develop, share, and implement new research, strategies, and materials.
6. **Inquiry-Based Learning** – Encouraging students to explore content, ask questions, take risks and share ideas about material rather than being told what they need to know by teachers. Student voice and autonomy is supported through a sense of ownership and responsibility for their own learning in a positive and open environment.
7. **Intellectually Challenging –** Facilitating students to develop a deep understanding of ideas, sort through misconceptions, and learn new ideas/create or improve upon ideas. Embedding metacognition into lesson plans to keep both content and process in mind so students understand how they just learned some content.

3.1.4 Art and Design Principles

Considering the continued influence and relevance of the Bauhaus Preliminary Course in Art and Design education pedagogy [Lerner 2012], we have elected to utilize its fundamental principles and methods in the contextualized aspect of the introductory course. Throughout all its changes and iterations, the Preliminary Course always considered that "… a firm grounding in the principles of form and color achieved through practical exercise was essential to the development of the new artist" [Casciato 2020]. These overarching principles of Form and Color are further broken down into component principles and areas of focus:

Form [Kandinsky 1966, Klee 2000]

- *Point* – beginning with the geometric point, the point represents a location in its most fundamental form. Once considered beyond this definition, the number, size, and relations of the point allow it to become a representational *element* of a composition.
- *Line* - composed of two or more points, a line can be expressed as an active, passive, or free *element* of the composition. Line types and orientations include medial, straight, curved, angular, warm, cold, horizontal, vertical, or diagonal.
- *Plane* – composed of groupings of pairs of lines, the plane allows the *concept of form to become actualized*. The distance, orientation, length, type, and angle count of the lines on the plane determine the forms created as well as their *dimensionality*. The fundamental forms are the circle, square and triangle, and express themselves in a composition via *repetition*, motion, and proportion.

Color [Itten 1964

- The color circle/wheel represents the fundamental representation of color theory, with the primary colors serving as a basis for any giving *abstraction* or detail of the color spectrum. Color relations are categorized through the seven color contrasts, color harmony, and color mixing. Color relates to form in through spatial effects, expression/impression theory, and composition.

3.2 General Outline of Lesson Modules

3.2.1 Common Elements of Each Lesson Module

To keep the program flexible for different implementation needs, the core concepts were divided into five modules that can be adapted to shorter durations of time, such as a week-long after-school course, to longer durations of time such as a quarter-long course that meets three times a week. Each module lists the guiding principles and techniques behind it.

The Engagement Principles have been used to create a general format that each module will follow. This format includes the following components related to specific Engagement Principles:

Component	Engagement Principles
Playing with Code/Changing Variables	Inquiry-Based Learning, Embedded Collaboration, Technology Integration
Guided Activity/Lesson	Inquiry-Based Learning, Intellectually Challenging, Technology Integration
Challenge/Task	Intellectually Challenging, Assessment Learning, Technology Integration
Sharing/Critique	Assessment Learning, Embedded Collaboration, Technology Integration
Creative Time/Exploration	Inquiry-Based Learning, Relevant, Interdisciplinary, Technology Integration
Contextualized Material	Interdisciplinary, Relevant
Community Showcase/Display	Relevant, Technology Integration

Table 1. *Module Elements and Engagement Principles*

3.2.2 Modules and Progression

The program is broken down into five modules with each having a base set of lessons that can be expanded upon if desired by the instructor. The module progression is based on the typical introductory computing course, with each module additionally synthesizing the Introductory Principles, CT Principles, and Art/Design Principles together in its lesson activities. Lesson topics and activites have been drawn from the Bauhaus Preliminary Course [Kandinsky 1966, Klee 2000, Itten 1964] and Daniel Shiffman's *Nature of Code* and *Learning Processing*.

1. Variables and Patterns - Learn about variables and abstract imagery.
2. Abstracting with Loops - Learn about loops, random numbers and transforming images.
3. Interactive Objects - Learn about vectors, parameters, and arrays with particle systems.
4. Simulated Systems - Learn about autonomous functions to simulate living systems
5. New Media - Learn about data visualization through interactive environments

Module	Principles		
	Introductory	**Computational Thinking**	**Art & Design**
1. Variables and Patterns	Variables, data types, expression, Control structures: conditionals	Abstraction, Problem Decomposition, and Algorithmic Design	Form – Point (position), Line (passive), Plane (repetition, relation) Color – color wheel
2. Abstracting with Loops	Conditional structures: loops, Randomization, Math application	Generalization, Abstraction, and Algorithmic Design	Form – Point (position, number), Line (horizontal, vertical, diagonal), Plane (circle, square, triangle, repetition) Color – color wheel, color contrasts
3. Interactive Objects	Mouse/keyboard interactions, Simple objects and inheritance, Arrays and Lists	Algorithmic Design, Problem Decomposition, Pattern Recognition, and Generalization	Form – Point (representation, size relations), Line (active, free), Plane (pairs of lines - orientation, motion) Color – spatial effects of color, color harmony, color mixing
4. Simulated Systems	Program and algorithm design, OOP principles, inheritance, abstract classes and interfaces	Abstraction, Algorithmic Design, Generalization, Problem Decomposition	Form – Point (representation, size relations), Line (active, free, warm, cold), Plane (pairs of lines - orientation, motion) Color – spatial effects of color, color harmony, color mixing
5. New Media	Data Visualization, Computing artifacts, Impacts of Computing	All themes and principles synthesized	Form and Color – expression, impression and composition

Table 2*. Lesson modules and their relationships with our design principles (note that we do not use the engagement principles as they are universal to every module).*

Module topics are based on a progression of Introductory Principles and Art/Design Principles from fundamental/straightforward to complex/comprehensive. This progression naturally begins with easier topics and increases in difficulty with each module, as each module builds upon the material in the previous. This progression is illustrated in Table 3.

	Introductory Computer Science	Art and Design
Module 1	Variables and conditional statements are introduced through the representation of visual patterns.	The point as a geometric position, the line as a passive element, the plane as the relation and repetition of these elements, and the color wheel are introduced.
Module 2	Variables randomized and used in mathematical operations. Conditional statements are executed within loops to produce random abstract patterns.	The number of points relates to multiple positions and the line becomes dynamic in its orientation in order to develop the plane into more complex shapes and patterns. Intentional use of color relations introduces color contrasts.
Module 3	Simple objects are created to abstract methods and behaviors of previous module. Arrays are used to store objects and mouse interaction is introduced as a data input for object manipulation and alteration.	Points and their size become representative, lines become active and free, and the plane consists of the motion of elements. Spatial effects, mixing, and harmony are serve as representational information.
Module 4	Object Oriented Programming principles, inheritance, and abstract classes combine objects and methods through program design to create autonomous simulated systems reflective of systems in nature.	Active points and lines compose forms which are in motion and orientationally aware of other forms. Spatial effects and harmony of color become important to developing compositional unity and conceptual clarity.
Module 5	Program design is used to create computing artifacts that represent data in an interactive visual format with the intention of illustrating scientific, cultural, or ethical topics.	Forms and colors are intentionally designed to create a composition with the goal of expressing a concept and or making an intentional impression on the audience.

Table 3. *Module Progression of Introductory and Art/Design Topics, moving from introductory to more complex concepts and integrated use of concepts, culminating in synthesis in Module 5.*

For example, variables and conditional statements are introduced in Module 1. Then, these statements are used in loops in Module 2. Simple objects and arrays are introduced in Module 3, followed by more complex object-oriented programming (OOP) concepts in Module 4. Module 5 then synthesizes the concepts into creating a computing artifact. Similarly, the basic idea of a point, a line, and a plane, and the color wheel are introduced in Module 1. Then, multiple points, lines, and planes are considered to generate more complex shapes and patterns in Module 2, together with color contrasts. In Module 3, points and lines become active and planes involve motion. Module 4 further considers spatial effects—such as interactions among shapes and patterns—and harmony of color. Module 5, again, synthesizes these concepts into a composition.

3.3. Development of Survey

The survey is divided into three sections: (1) general information, (2) self-efficacy, interests & perception, and (3) CT aptitude test. As we are interested in the change in these metrics over the course of the module implementation, the survey is administered once at the beginning and once at the end of the course. Results and analysis are then based on changes in metrics. The full survey can be found in Appendix, Section 7.2.

First, the general information section is designed to gather information such as name, gender, age, grade, etc.

The second section is formatted as questions with an agree/disagree scale ranging from strongly disagree to strongly agree with three options in between. This section has three subsections. The first subsection includes questions regarding student self-efficacy. Self-efficacy questions were included because high self-efficacy has been shown to be an indicator of high engagement levels [Chen 2017]. This provides us with a quantifiable measurement proxy for effectiveness of engagement. The second subsection includes questions regarding interest in Art and Design, Computer Science, and STEM as domains and possible career paths. These questions provide an extra level of data relating to student attitudes and how they have been influenced by the course. The final subsection consists of a free response box that asks the students to describe what Computer Science is. This serves as a qualitative measurement of student perception of the field in order to determine whether contextualized computing has an impact.

The final section consists of puzzle-based questions designed to measure Computational Thinking aptitude are given [Román-González, Marcos, et al. 2019]. These questions help determine the base level of CT aptitude, and allow us to measure amount of CT learning.

4 Implementation and Deployment

4.1 IRB Approval

This research has been approved by the Institutional Review Board (IRB) for the University of Nebraska, Lincoln and Lincoln Public Schools (LPS). This approval represents compliance with highest ethical principles in the conduct of research with human subjects through the UNL Human Research Protection Program (HRPP). As the data collected is from human subjects under the age of eighteen, student consent and parental consent forms have been created with particular attention to making the student consent form friendly and comprehensible for younger students. The IRB approval letter and consent forms are available in Section 7.4. of the Appendix.

4.2 Basic Background

The program was implemented through two limited pilot opportunities.

The first implementation was during the STEM portion of the NCPA (Nebraska College Preparatory Academy) camp through the Computer Science and Engineering Department at the University of Nebraska, Lincoln. The camp offered students with a variety of hour-long workshops designed to engage them with computing which typically lasted around an hour. In total, 108 junior year high school students participated in the Creative Coding workshop, which offered an abridged version of Module 1 and Module 2 with the goal of introducing students to contextualized computing while allowing them to produce a tangible product.

The second implementation occurred at a Lincoln Public School middle school in Lincoln, Nebraska. The Creative Coding Modules 1-3 were taught over the course of two weeks during hour long sessions on Monday, Wednesday, and Friday to a class of 24 students.

4.3 Implementation Changes

Module design and delivery has changed between and after implementations to accommodate time parameters and respond to feedback and observations.

During the NCPA implementation, Modules 1 and 2 had to be synthesized into an hour-long workshop. This was done in order to expose the students to a broader range of applications, impart introductory concepts in Creative Coding, and give them the opportunity to produce more advanced tangible artwork. The drawback of this synthesis was that students were unable to do in-depth on every topic, and many activities only allowed for surface-level understanding. Table 4 shows the adaptation.

NCPA Camp Creative Coding	
Activity	**Description**
Playing with Variables (15 min)	Students experiment with starter code that includes several variables that are used in the instantiation of shapes and colors.
Random Walker (15 min)	Variables and conditionals are explained through a random walker program. Geometric point is introduced.
Random Pattern (15 min)	Students are given starter code for a program which uses a double-for loop and color palette array to generate random patterns. A color theory handout is given for them to use as reference for color selection.
Creative Time (15 min)	Students are asked to create their own version of one of these programs as a representation of their mood or personality.

***Table 4**. Activities included in the NCPA implementation based on Modules 1 and 2 (Table 2 above).*

LPS Creative Coding		
Activity	Description	Kept/Dropped/Added
Playing with Variables (Module 1)	Students experiment with starter code that includes several variables that are used in the instantiation of shapes and colors.	Dropped: Students found this activity boring and had a hard time paying attention.
Creating Kandinsky (Module 1)	*Playing with Variables* except students add new shapes and lines in the style of Kandinsky.	Added: The Kandinsky styled approach serves as a method for get students interested.
Random Walker (Module 1)	Variables and conditionals are explained through a random walker program. Geometric point is introduced.	Kept: Students find this engaging because of the highly customizable nature of the program.
Random Pattern (Module 2)	Students are given starter code for a program which uses a double-for loop and color palette array to generate random patterns. A color theory handout is given for them to use as reference for color selection.	Kept: This is an easy activity for students to iterate on and apply color schemes to. Additionally, it consistently results in a tangible art product.
Bouncing Balls (Module 3)	Students are given starter code with simple objects implemented for a bouncing ball and asked to experiment. They are then challenged with adding more balls via an array.	Added: This activity introduces students to object oriented programming and allows for highly customizable object interaction effects.

***Table 5.** Activities included in LPS implementation based on Modules 1-3 (Table 2 above).*

Between this NCPA implementation and the LPS implementation, Module delivery was altered to be less lecture- and handout-based and more hands-on and interactive with the facilitator. This was done to ensure students were staying on track and felt comfortable communicating with the facilitator and each other. Finally, topics that students had difficulty engaging with were dropped and new topics were added that included more dynamic content. Table 5 shows the changes for the LPS implementation.

5 Results

5.1 Data Collection

Due to four limiting factors, a significant amount of substantial data was not collected. First, there were limited implementation opportunities due to an already full or accounted for course schedule at most schools. Second, during implementation opportunities that did occur, there was a limited sample size of students both in numbers and groups represented. Third, not every module was completed, meaning that the entire effect of the program cannot be analyzed. Finally, a lack of follow through with faculty meant not all students were given the opportunity to fill out the post-survey. There are several ways to improve data collection in the future. To increase implementation opportunities and increase sample size, it may be necessary to promote to a larger network of teachers, administrators, and schools. Handing off the course to facilitators with thorough implementation guidelines only under an agreement that all modules are completed, and data is collected would ensure the third and fourth limiting factors are avoided. Another strategy for ensuring proper survey completion would be to have a researcher team member on site during the first and final days of implementation.

5.2 Analysis

In order to analysis our developed program despite a lack of data, we have conducted a Strengths, Weaknesses, Opportunities, and Threats (SWOT) analysis.

5.2.1 Strengths

We have found certain attributes and observations to be especially reflective of the strengths of this our Creative Coding program. These strengths can be categorized as (1) qualitative feedback, (2) program theory and design, and (3) ease of implementation and data collection.

Significant qualitative feedback includes observations of student interest, response, and understanding of lesson activities as well as course material in general. Students were generally interested in course material, which was expressed through their frequency of questions asked and excitement in sharing creative work with each other and facilitators. Students were able to grasp the majority of course material, with female students especially excelling by going above and beyond the lesson requirements in creative work (LPS middle school). Qualitative feedback from the Computer Science faculty for the class at LPS was positive, with indications that course material was effective in conveying information and engaging students, and wider adoption of material should be investigated.

Program theory and design is based upon thorough research into student enrollment, student engagement, contextualized computing, Computational Thinking, and Art and Design. This research supports the pedagogical choices made in developing the modules in our Creative Coding program. Additionally, the pedagogical approach in our design represents a unique and comprehensive synthesis of concepts of Computer Science, engagement theory, Computational Thinking, contextualized computing, and Art and Design. The principled approach and supporting background create a robust platform upon which to base future endeavors into this area of research.

Ease of implementation and data collection for future implementation represents the final set of strengths. The program material is ready for implementation at future sites with a full suite of modules that consists of student handouts, references, and a facilitator guideline. The course is modular and built to be flexible for implementation, making it easy to adopt it as a standalone or supplemental course that can be used for actual classroom teaching as well as for outreach activities. Data collection will be straightforward, as the survey is ready for use and available in the Appendix Section 8.2. or at request via Qualtrics (online academic data collection tool at www.qualtrics.com). Finally, this research has been IRB approved, both via the University of Nebraska as well as Lincoln Public Schools, making future approval more likely. All of these features support future scalability of the curriculum.

5.2.2 Weaknesses

Given the recent development of the program, there are still weaknesses that must be identified for future growth and improvement. There are three categories of weaknesses that have been identified: (1) material development, (2) implementation techniques, and (3) data collection.

The material development has been largely based on theory combined with previous research developing similar programs, and undoubtedly has issues that will only be worked out through future implementation and experience. The student handouts, for example, are in need of continued review and revision in order to ensure that they are conveying information appropriately and effectively. The same goes for the facilitator guideline, which is underdeveloped and will only be improved through future feedback from facilitators.

Implementation technique is a particularly weak point, as classroom management, lesson flow, and instruction style need improvement. For classes with students having different levels of computing or problem-solving experience and skills, some students find advanced concepts more difficult and lag behind while other experienced students finish too quickly. Some students additionally struggle with creativity and developing techniques to foster this ability will prove challenging.

As for data collection, the lack of substantive data poses many challenges for moving forward and improving the program. Additionally, the survey questions also lack feedback on their effectiveness in gathering appropriate data. From the survey implementation at LPS, the student group surveyed scored very high on the Computational Thinking (CT) aptitude test prior to the course, suggesting the survey used may not be rigorous enough. This could be due to the course being optional and mostly opted into by student already experienced an interested in computer science. Regardless, there is too little data to be sure what needs improvement.

5.2.3 Opportunities

The overall Creative Coding program presents many opportunities for students, communities, and the field of CS education research. The program's strengths for ease of implementation provide the potential for a wider adoption beyond the current implementation sites. This includes the opportunity of implementation via online and remote learning formats. If the program is successful in meeting its research goals, wider adoption will lead to increased engagement, retention, and learning outcome success within the field of Computer Science. This includes the potential widening of the middle-school/high school to college CS pipeline for students from diverse interests and underrepresented backgrounds. With this potential, also comes the opportunity of making changes to the program as more data is collected to meet students' needs that may be background specific.

Finally, the exposure of students to computing contextualized within Art and Design contributes to the development of the field of Creative Computing and can foster the development of Creative Computing communities and culture, which in turn contribute to the Arts, Tech, and New Media culture of the broader community. The pedagogical theory developed in this thesis lies at the crossroad of various domains which cross frequently in industry and research, but hardly at all in education. The groundwork laid out in this thesis presents the opportunity for future development of additional interdisciplinary pedagogical research and endeavors.

5.2.4 Threats

The threats for future implementation and development must also be identified in order to prepare and adapt successfully.

The first threat is regarding teacher and administrative buy-in for cooperation in implementation and data collection. It may be difficult to pitch this program given the development level and amount of data currently collected combined with the typically jammed packed schedule and plans that are already in place or offered to students. It will be important to convey the program goals and background effectively in order to compete with other options administrators and faculty may have in addition to the already busy schedule planned.

Second, there are many unforeseen or unrecognized bugs and issues that may pop up as the program is implemented more extensively. It will be vital to build in mechanisms for receiving and responding to feedback over time.

Finally, a prominent possible issue is in the effectiveness of the survey in collecting data and the possibility of failure in data collection and interpretation.

6 Conclusion and Future Work

6.1 Summary

The increasing enrollment rates for students in Computer Science pose a number of challenges and opportunities. More students are entering the field with diverse backgrounds and interests, with growing percentage of these students from fields outside of Computer Science. Educators are challenged to deliver on positive learning outcomes and adapt their curriculum to their new audience. This also presents an opportunity to increase engagement and retention of students from underrepresented backgrounds and diverse interests.

Student engagement in Computer Science is unfortunately not only at a low point but is also decreasing over time. This is troubling for the field of Computer Science, as engagement leads to increased student learning performance which results in increased retention. Considering that this low level of engagement is coming at a time when it is vital to have high levels of engagement further strengthens the cause for focusing on increasing student engagement. In recent years, researchers have turned to contextualized computing as a method for increasing student engagement, with courses such as Mark Guzdial's Computation Media course demonstrating positive results over a six-year period. This course and similar contextualized courses were hypothesized to be successful due to their consistent application context and opportunity for creative expression. In order to respond to increasing enrollments and decreasing engagement, we have created an introductory computing course for middle-school and high-school students with the goal of increasing student engagement and Computational Thinking (CT) aptitude. As a consistent context with a creative outlet is suggested as an effective strategy, we have contextualized our course within the domain of Art and Design.

Computational Thinking (CT) serves as the cognitive basis for approaching problem solving in Computer Science and overlaps with cognitive skills valuable for many different domains. Acquiring CT aptitude is desirable for not only any student planning on learning computing, but any student in general. The universality of CT compliments the interdisciplinary nature of contextualized computing and serves as a strong starting point for an introductory computing course. CT is complimented by Creative Thinking, which serve to extend CT skills.

In order to develop an effective curriculum, we have taken a principle-based approach to designing our curriculum. We have researched pedagogical theory behind Computational Thinking, Engagement, Art and Design, and Introductory Computing to find guiding principles which we have used in module development.

Our five-module introductory Creative Computing course for middle-school and high-school students is designed for flexibility and ease of implementation. Implementation opportunities thus far have provided our team with insight into the curriculum's strengths, weakness, opportunities, and threats. Promising observations and outcomes include high levels of student excited and a large body of tangible student digital artwork.

6.2 Future Work

Moving forward, there are three critical next steps that can be taken to further progress in this area.

First, increasing implementation opportunities will be vital to ensuring that enough data is collected to assess the effectiveness of the program in accomplishing its objectives. Furthermore, with increased implementation opportunities, it will be important to increase the robustness of the survey and ensure that enough feedback data is available for weak points to be identified and improvements to be made. Additions to the survey could also include metrics for gauging Creative Thinking in order to expand the scope of the research to be even more interdisciplinary in its value.

Second, materials will need to be continually developed, as new research emerges, and feedback is reviewed. This will consist of identifying which aspects of the modules are successful and which are not, and whether new additions, alterations, or removals need to be made. Part of this process will be continued research into the pedagogical theory and background which has been compiled and synthesized to create this program. Additionally, steps should be taken to develop and improve scalability. This could include the creation of online resources which host and deliver material in a format that is accessible for both educators and students.

Finally, research into contextualized computing beyond Art and Design is needed to help future educators determine when contextualized computing is needed, and what domain contextualization to utilize. This could lead to the creation of contextualized computing courses in the context of many new domains. A wide array of contexts would be useful in offering options to students of diverse interests and provide educators from disciplines outside of computing with supplemental material tailored to their domain. In this changing landscape of increasing interconnectivity among domains, future steps in this research area will help expand the state of the art in contextualized computing and help improve student engagement and retention in Computer Science.

7 References

Assessing and Responding to the Growth of Computer Science Undergraduate Enrollments. National Academies Press, 2018.

Casciato, Maristella; Fox, Gary; Rochester, Katherine, *Bauhaus: Building the New Artist*, accessed March 17, 2020, https://www.getty.edu/research/exhibitions_events/exhibitions/bauhaus/new_artist/

Cassel, Lillian N., and Ursula Wolz. "Interdisciplinary Computing, Successes and Challenges." *Proceeding of the 44th ACM Technical Symposium on Computer Science Education - SIGCSE '13*, 2013, doi:10.1145/2445196.2445438.

Chen, Helen L., et al. "Conceptualizing Engagement: Contributions of Faculty to Student Engagement in Engineering." *Journal of Engineering Education*, vol. 97, no. 3, 2008, pp. 339–353., doi:10.1002/j.2168-9830.2008.tb00983.x.

Chen, I-Shuo. "Computer Self-Efficacy, Learning Performance, and the Mediating Role of Learning Engagement." *Computers in Human Behavior*, vol. 72, 2017, pp. 362–370., doi:10.1016/j.chb.2017.02.059.

CRA. 2017a. Generation CS: Computer Science Undergraduate Enrollments Surge Since 2006. Online. Available at http://cra.org/data/Generation-CS/.

Esen, Ece, et al. "Basic Design Education and Bauhaus." *The Turkish Online Journal of Design Art And Communication*, vol. 8, no. 1, 2018, pp. 37–44., doi:10.7456/10801100/004.

Freeman, R. 2016. Workforce and Degree Trends Panel. Public Workshop of the National Academies Committee on the Growth of Computer Science Undergraduate Enrollments, National Academy of Sciences. Washington, DC. August 15.

Google for Education: Computational Thinking, Google, edu.google.com/resources/programs/exploring-computational-thinking.

Grover, Shuchi, and Roy Pea. "Computational Thinking in K–12." Educational Researcher, vol. 42, no. 1, 2013, pp. 38–43., doi:10.3102/0013189x12463051.s

Guzdial, Mark. "A Media Computation Course for Non-Majors." *ACM SIGCSE Bulletin*, vol. 35, no. 3, 2003, p. 104., doi:10.1145/961290.961542.

Hershkovitz, Arnon, et al. "Creativity in the Acquisition of Computational Thinking." *Interactive Learning Environments*, vol. 27, no. 5-6, 2019, pp. 628–644., doi:10.1080/10494820.2019.1610451.Lerner, Fern. "Liberating Foundations of Art and Design." *International Journal of Art & Design Education*, vol. 31, no. 2, 2012, pp. 140–152., doi:10.1111/j.1476-8070.2012.01703.x.

Kandinsky, Wassily. *Point and Line to Plane; Contribution to the Analysis of the Pictorial Elements*. Solomon R. Guggenheim Foundation for the Museum of Non-Objective Painting, 1966.

Klee, Paul, and Sibyl Moholy-Nagy. *Pedagogical Sketchbook.* Faber and Faber, 2000.

Morgan, Michael, et al. "Contrasting CS Student and Academic Perspectives and Experiences of Student Engagement." *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE 2018*, 2018, doi:10.1145/3197091.3205839.

Roberts, Eric S. "Meeting the Challenges of Rising Enrollments." *ACM Inroads*, vol. 2, no. 3, 2011, p. 4., doi:10.1145/2003616.2003617.

Román-González, Marcos, et al. "Combining Assessment Tools for a Comprehensive Evaluation of Computational Thinking Interventions." *Computational Thinking Education*, 2019, pp. 79–98., doi:10.1007/978-981-13-6528-7_6.

Romero, Margarida, et al. "Computational Thinking Development through Creative Programming in Higher Education." *International Journal of Educational Technology in Higher Education*, vol. 14, no. 1, 2017, doi:10.1186/s41239-017-0080-z.

Ruddley, John, and Johannes Itten. "Design and Form: The Basic Course at the Bauhaus." *Art Education*, vol. 17, no. 7, 1964, p. 44., doi:10.2307/3190622.

Soh, Leen-Kiat, et al. "Integrating Computational and Creative Thinking to Improve Learning and Performance in CS1." *Proceedings of the 45th ACM Technical Symposium on Computer Science Education - SIGCSE '14*, 2014, doi:10.1145/2538862.2538940.

Parsons, Jim, and Leah Taylor. "Improving student engagement." *Current issues in education* 14.1 (2011).Weintrop, David, et al. "Defining Computational Thinking for Mathematics and Science Classrooms." *Journal of Science Education and Technology*, vol. 25, no. 1, 2015, pp. 127–147., doi:10.1007/s10956-015-9581-5.

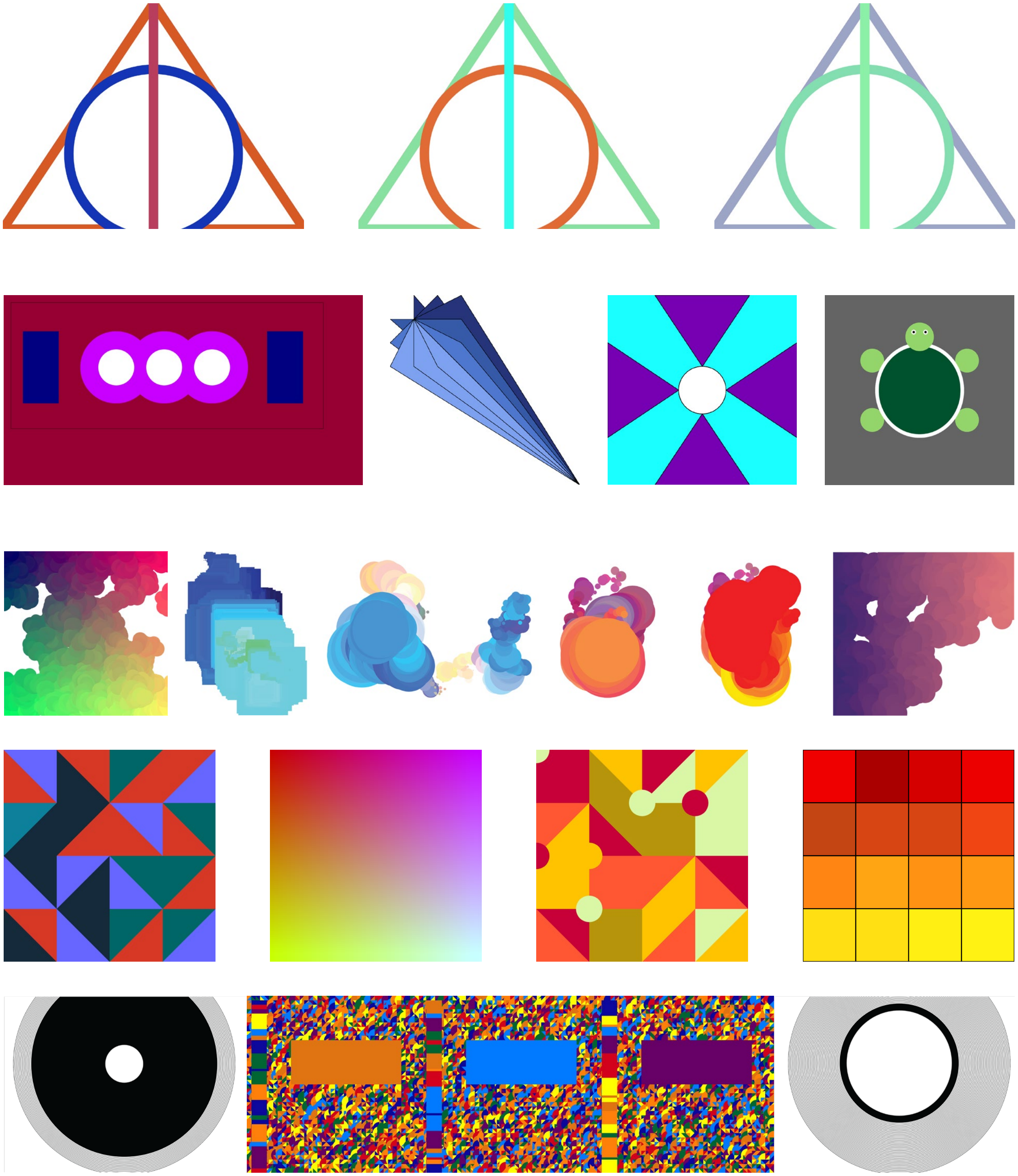
Wing, Jeannette M. "Computational Thinking." *Communications of the ACM*, vol. 49, no. 3, 2006, p. 33., doi:10.1145/1118178.1118215.Wing, J. (2011). Research notebook: Computational thinking—What and why? *The Link Magazine*, Spring. Carnegie Mellon University, Pittsburgh. Retrieved from http://link.cs.cmu.edu/article.php?a=600

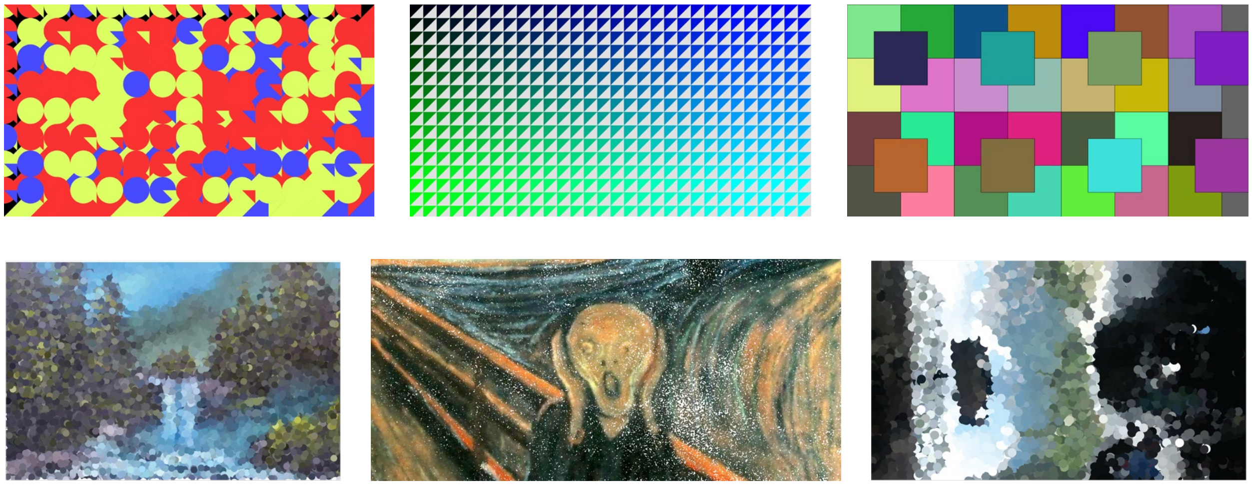
Xu, Dianna, et al. "Creative Computation in High School." *Proceedings of the 47th ACM Technical Symposium on Computing Science Education - SIGCSE '16*, 2016, doi:10.1145/2839509.2844611.

Xu, Dianna, et al. "Updating Introductory Computer Science with Creative Computation." *Proceedings of the 49th ACM Technical Symposium on Computer Science Education - SIGCSE '18*, 2018, doi:10.1145/3159450.3159539.

Appendices

A.1. Student Work





A.2. Survey

Survey Testing:

Computer Science/Art Self-Efficacy
STEM/Computer Science/Art Interest
Computational Thinking Aptitude

---

1. What is your age?

- [ ] 12
- [ ] 13
- [ ] 14
- [ ] 15
- [ ] 16
- [ ] 17
- [ ] 18

2. What grade are you in school?

- [ ] 6
- [ ] 7
- [ ] 8
- [ ] 9
- [ ] 10
- [ ] 11
- [ ] 12

3. Are you male or female?

- [ ] Male
- [ ] Female

	STRONGLY DISAGREE	DISAGREE	NOT SURE	AGREE	STRONGLY AGREE
4. I know how to code					
5. I am skilled at computer programming					
6. I am skilled at creating art					
7. I am interested in Computer Science					
8. I am interested in Science & Technology					
9. I am interested in Art					
10. I am interested in Math					
11. I am interested in a career in Science & Technology					
12. I am interested in a career in Computer Science					

13. What do you think Computer Science is? (please answer below)

‘Pac-Man’

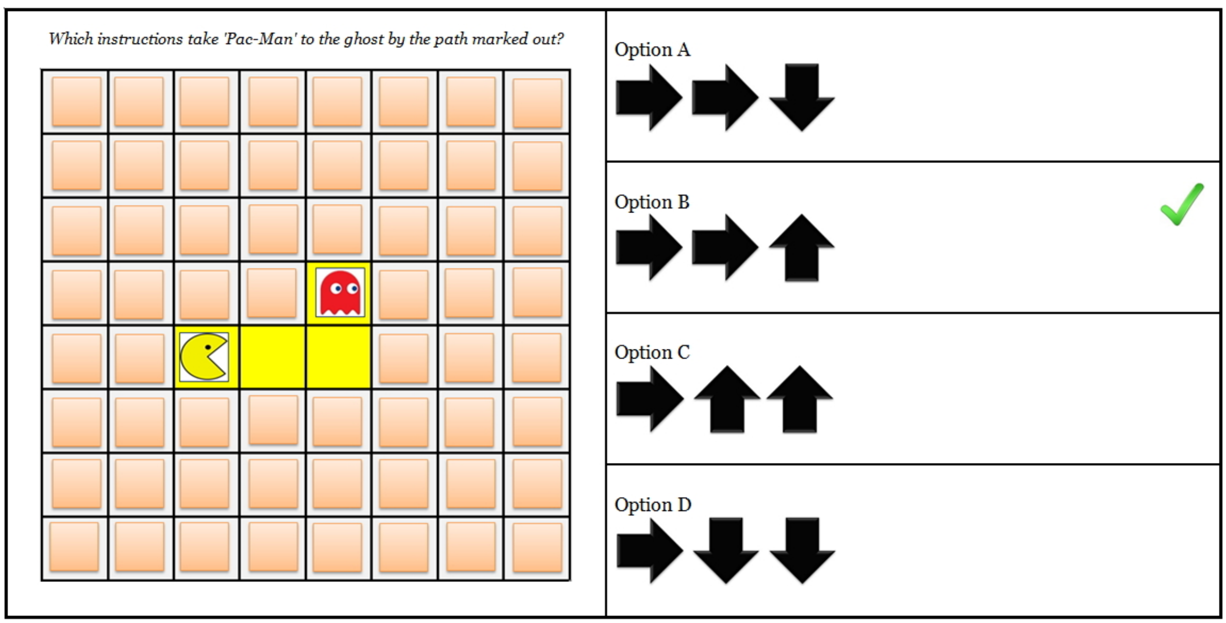
Ghost

EXAMPLE

In the first example you are asked which instructions take 'Pac-Man' to the ghost by the path marked out.

That is, to take 'Pac-Man' EXACTLY to the square where the ghost is (without going over or stopping short), and strictly following the path marked in yellow (without touching the walls, represented by the orange squares).

The correct answer in this example is B. Click the “B” button below the question.



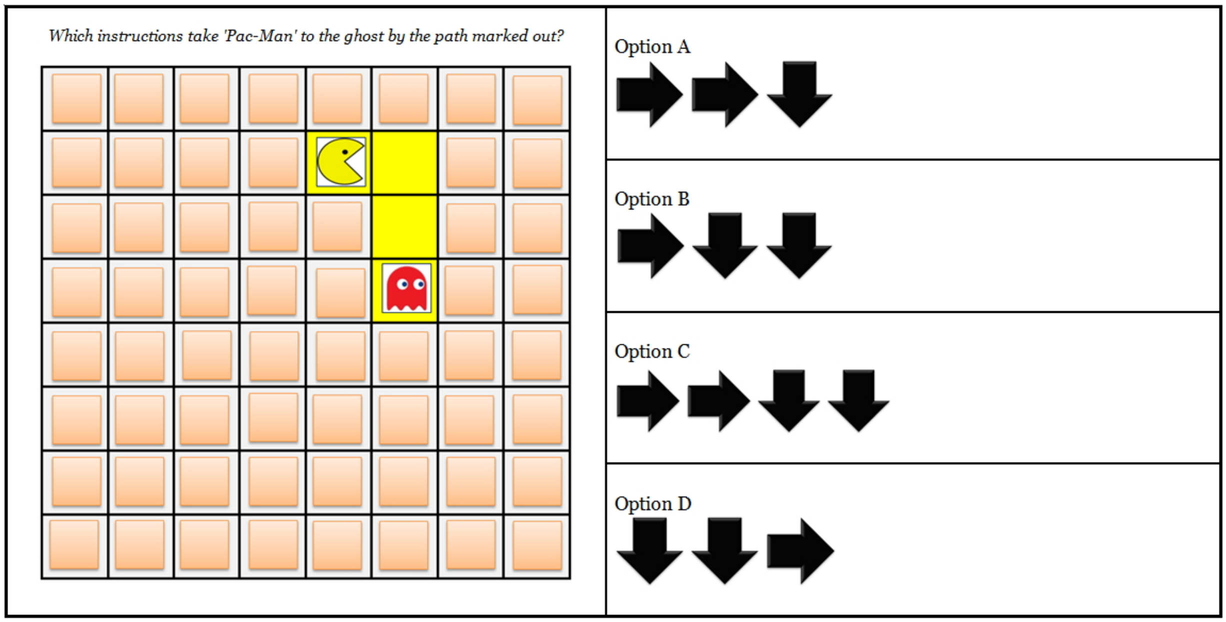
Example

Select the correct answer (in this example, the correct answer is B)

Mark only one oval.

- ◯ A
- ◯ B
- ◯ C
- ◯ D

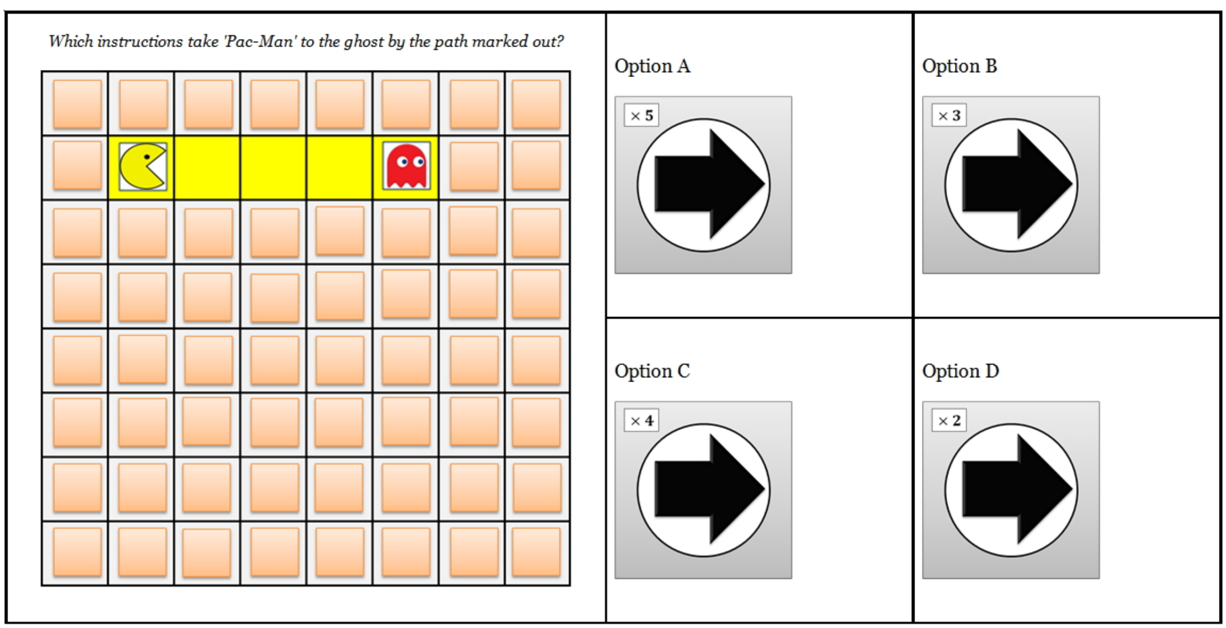
14. Which instructions take 'Pac-Man' to the ghost by the path marked out?



Select the correct answer
Mark only one oval.

- ◯ A
- ◯ B
- ◯ C
- ◯ D

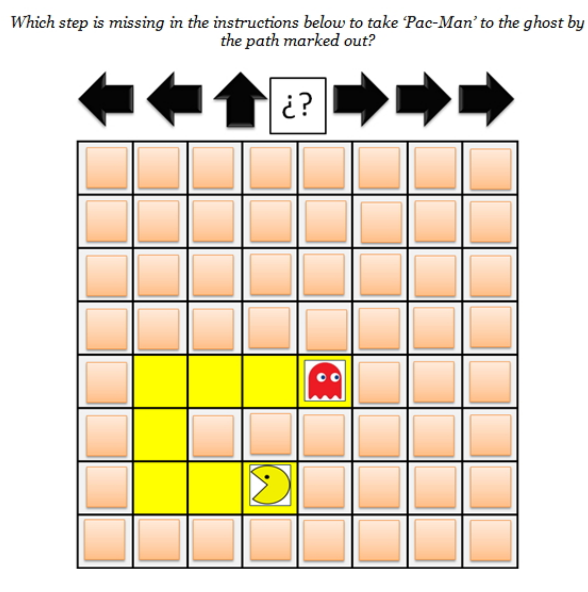
15. Which instructions take 'Pac-Man' to the ghost by the path marked out?



Select the correct answer
Mark only one oval.

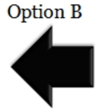
- ◯ A
- ◯ B
- ◯ C
- ◯ D

16. Which step is missing in the instructions below take 'Pac-Man' to the ghost by the path marked out?

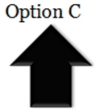


Option A

Option B



Option C

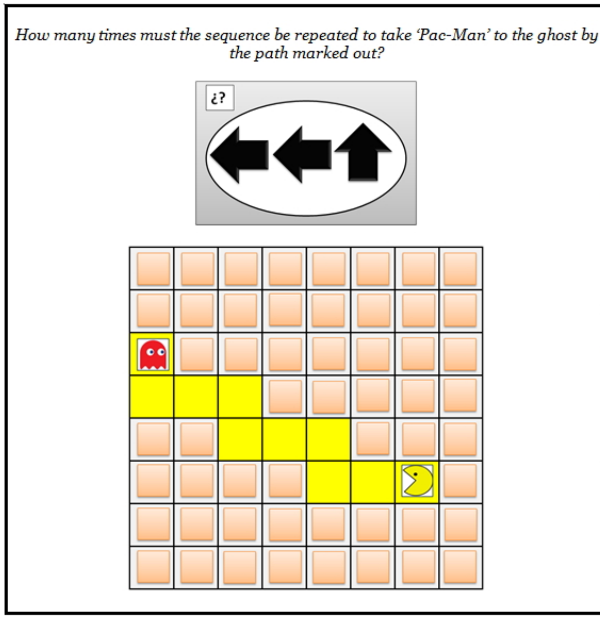


Option D

Mark only one oval.

- A
- B
- C
- D

17. How many times must the sequence be repeated to take 'Pac-Man' to the ghost by the path marked out?



Option A
× **2**

Option B
× **1**

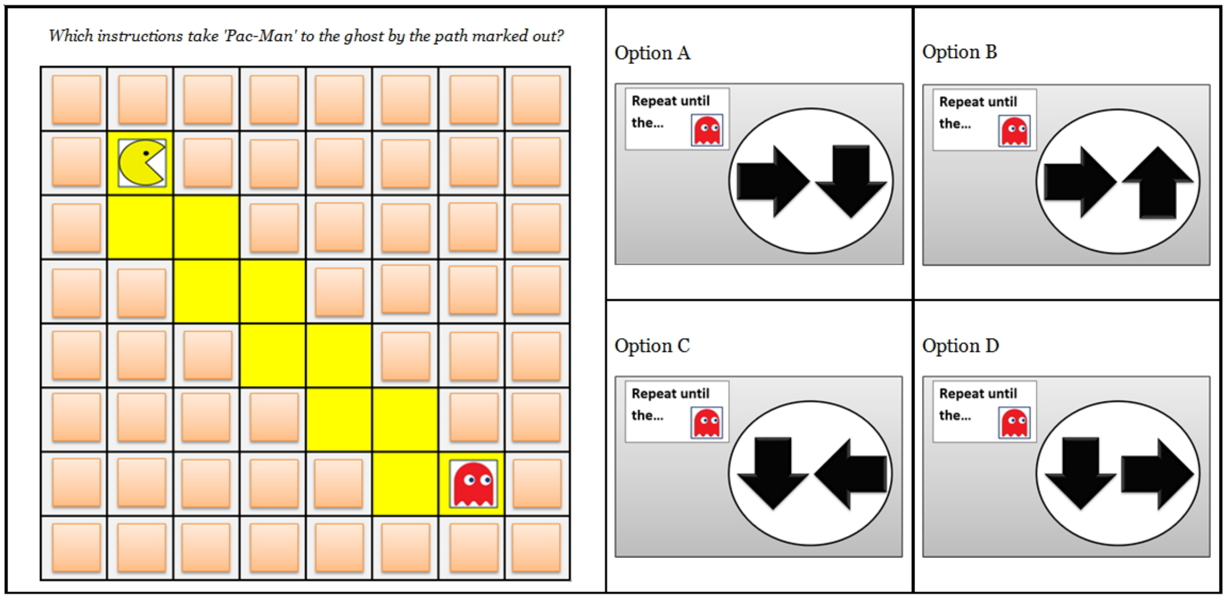
Option C
× **4**

Option D
× **3**

Mark only one oval.

- A
- B
- C
- D

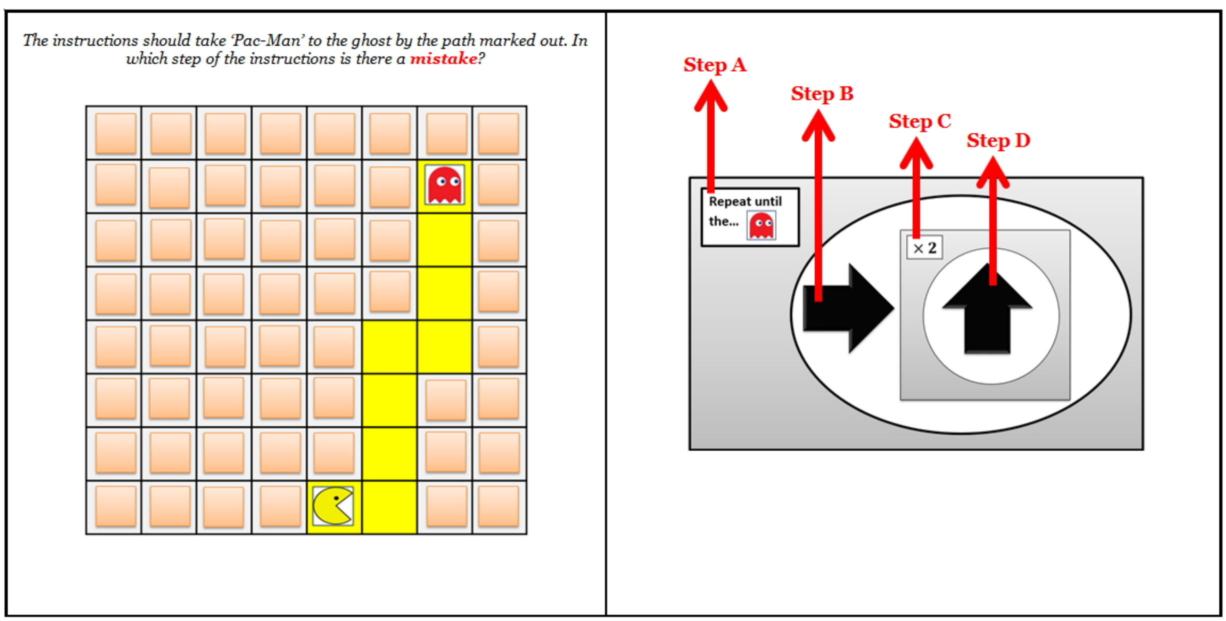
18. Which instructions take 'Pac-Man' to the ghost by the path marked out?



Mark only one oval.

- A
- B
- C
- D

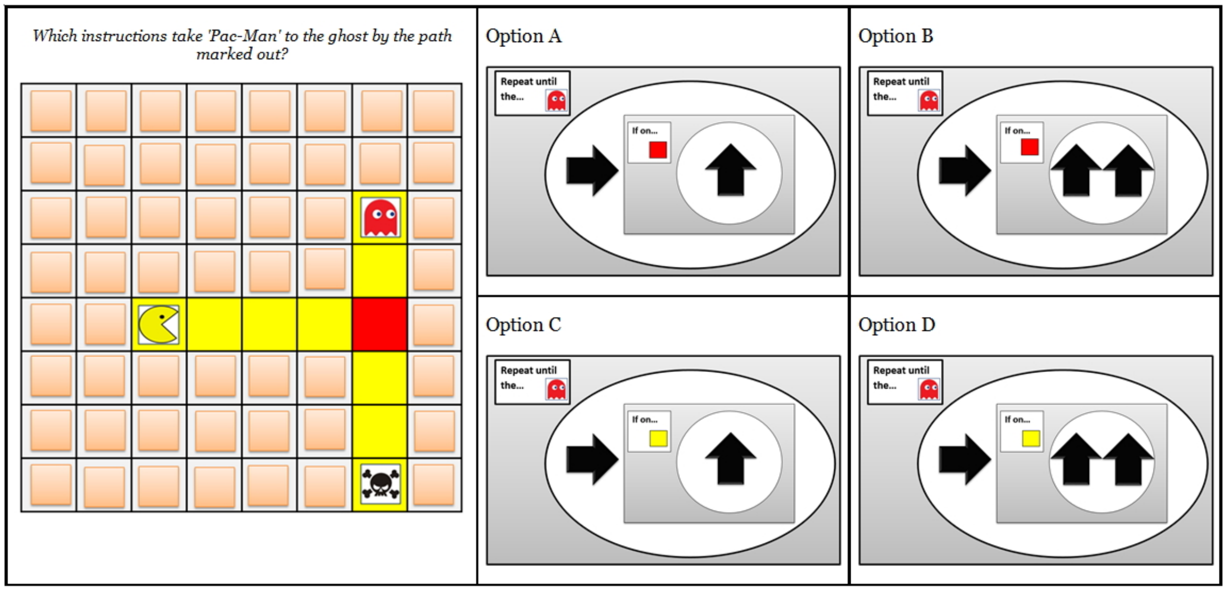
19. The instructions should take 'Pac-Man' to the ghost by the path marked out.
In which step of the instructions is there a mistake?



Mark only one oval.

- A
- B
- C
- D

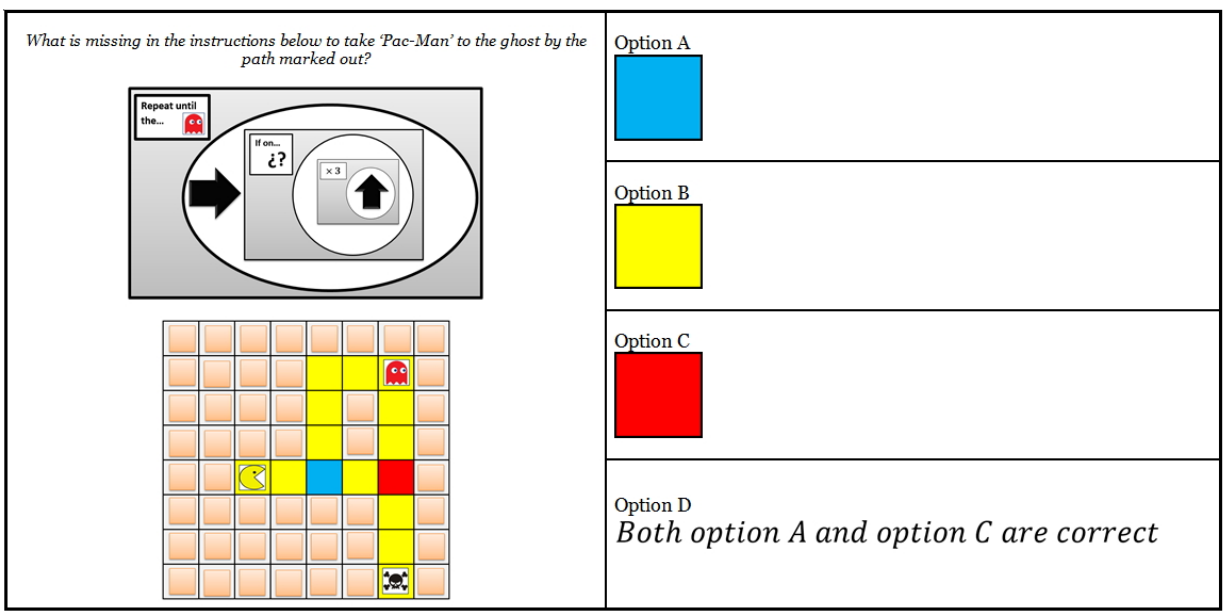
20. Which instructions take 'Pac-Man' to the ghost by the path marked out?



Mark only one oval.

- A
- B
- C
- D

21. What is missing in the instructions below to take 'Pac-Man' to the ghost by the path marked out?



Mark only one oval.

- A
- B
- C
- D

A.3. IRB Documents

Official Approval Letter for IRB project #19419 - New Project Form

September 23, 2019

Dana Hoppe
Department of Computer Science and Engineering
AVH 256 UNL NE 685880115

Leen-Kiat Soh
Department of Computer Science and Engineering
AVH 122E UNL NE 685880115

IRB Number: 20190919419EX
Project ID: 19419
Project Title: Effectiveness of Creative Coding Education for Improving Computational Thinking

Dear Dana:

This letter is to officially notify you of the certification of exemption of your project for the Protection of Human Subjects. Your proposal is in compliance with this institution's Federal Wide Assurance 00002258 and the DHHS Regulations for the Protection of Human Subjects at 45 CFR 46 2018 Requirements and has been classified as exempt. Exempt categories are listed within HRPP Policy #4.001: Exempt Research available at: http://research.unl.edu/researchcompliance/policies-procedures/.

o Date of Final Exemption: 09/23/2019
o Review conducted using exempt category 1 at 45 CFR 46.104
o Funding (Grant congruency, OSP Project/Form ID and Funding Sponsor Award Number, if applicable): UCARE

We wish to remind you that the principal investigator is responsible for reporting to this Board any of the following events within 48 hours of the event:
* Any serious event (including on-site and off-site adverse events, injuries, side effects, deaths, or other problems) which in the opinion of the local investigator was unanticipated, involved risk to subjects or others, and was possibly related to the research procedures;
* Any serious accidental or unintentional change to the IRB-approved protocol that involves risk or has the potential to recur;
* Any protocol violation or protocol deviation
* An incarceration of a research participant in a protocol that was not approved to include prisoners
* Any knowledge of adverse audits or enforcement actions required by Sponsors
* Any publication in the literature, safety monitoring report, interim result or other finding that indicates an unexpected change to the risk/benefit ratio of the research;
* Any breach in confidentiality or compromise in data privacy related to the subject or others; or
* Any complaint of a subject that indicates an unanticipated risk or that cannot be resolved by the research staff.

This project should be conducted in full accordance with all applicable sections of the IRB Guidelines and you should notify the IRB immediately of any proposed changes that may affect the exempt status of your research project. You should report any unanticipated problems involving risks to the participants or others to the Board.

If you have any questions, please contact the IRB office at 402-472-6965.

Sincerely,

Becky R. Freeman

Becky R. Freeman, CIP
for the IRB

University of Nebraska-Lincoln Office of Research and Economic Development
nugrant.unl.edu

NUgrant

INFORMED ASSENT FORM

What is a research study?

Research studies help us learn new things. We can test new ideas. First, we ask a question. Then we try to find the answer.

This paper talks about our research and the choice that you have to take part in it. We want you to ask us any questions that you have. You can ask questions any time.

Important things to know…

- You get to decide if you want to take part.
- You can say ‘No’ or you can say ‘Yes’.
- No one will be upset if you say ‘No’.
- If you say ‘Yes’, you can always say ‘No’ later.
- You can say ‘No’ at anytime.
- We would still take good care of you no matter what you decide.

Why are we doing this research?

We are doing this research to find out more about the possible benefits of teaching code in a creative setting. Computer Science can be a complicated topic, and we hope to make learning about it through coding less intimidating. We also hope that we can show students that Computer Science can be used for many different things including creative projects. We want to show students that Computer Science is simply a tool you can use to do a lot of cool things with.

What would happen if I join this research?

If you decide to be in the research, you can expect the following:

- Complete two surveys, one at the beginning and one at end of the educational program which will ask about:
    - Your age, year in school, and gender
    - Your interest in Science, Computer Science, and Art
    - Different picture-based puzzles
- You do not have to complete the surveys to participate in the lessons and may opt out at any time.
- These surveys will take about 15-20 minutes

Could bad things happen if I join this research?

Some of the tests might make you uncomfortable or the questions might be hard to answer. We will try to make sure that no bad things happen. You do not have to complete survey questions if they cause you any distress.

Could the research help me?

This research will not help you. We do hope to learn something from this research though. And someday we hope it will help kids to learn about Computer Science.

What else should I know about this research?

If you don't want to be in the study, you don't have to be.

It is also OK to say yes and change your mind later. You can stop being in the research at any time. If you want to stop, please tell the research doctors.

You would not be paid to be in the study.

You can ask questions any time. You can talk to Dr. Leen-Kiat Soh or Dana Hoppe anytime. Ask us any questions you have. Take the time you need to make your choice.

Dana Hoppe, UGRD, Principal Investigator Office (402) 942-4201
Leen-Kiat Soh, PhD, Secondary Investigator Office (402) 472-6738.

Is there anything else?

If you want to be in the research after we talk, please write your name below. We will write our name too. This shows we talked about the research and that you want to take part.

Name of Participant ____
(To be written by child/adolescent)

Printed Name of Researcher ____

Signature of Researcher ____

______________ ______________
Date ***Time***

INFORMED CONSENT FORM

Identification of Project : Effectiveness of Creative Coding Education for Computational Thinking

Purpose of the Research:

This study will explore the educational outcomes of creative coding and its potential for integrating concepts from Computer Science and Art/Design. The educational program will focus on using visual learning to connect the complex syntax of the code itself to the task it is accomplishes in order to make understanding code more intuitive and less intimidating. The ability to interact with a program can potentially increase interest and engagement. Our premise is that putting Computational Thinking within the setting of Art/Design can catalyze a synthesis of Computational Thinking abstraction with creative projects.

The main questions of the research are: Can the fundamentals of Computer Science and Art/Design be synthesized into an interdisciplinary educational program? Does programming within a visual/interactive context significantly increase Computational Thinking aptitude?

Procedures:

The participants will be asked to complete a pre- and post- survey at the beginning and end of the educational program. The survey will be divided into four types of questions: (1) Background Information such as age, year in school, and gender, (2) Self-Efficacy in Computer Science/Art in terms of a participant's perception of his or her ability in the related fields,
(3) Interest in STEM/Computer Science/Art, and (4) Computational Aptitude Evaluation that include pictures of different puzzles and related questions based on other Computational Thinking Aptitude tests. The students do not have to complete the surveys to participate in the lessons and may opt out at any time.

This survey will take approximately 15-20 minutes and will be given at the beginning of the first lesson and at the end of the final lesson. There will be five individual lessons taught in the program.
Lessons will be conducted as follows:
Playing with Code/Changing Variables – 15 min
Guided Activity/Lesson – 15 min
Challenge/Task – 15 min
Sharing/Creative Time – 15 min
.

Risks and/or Discomforts:

There are no known risks or discomforts associated with this research. Student answers to surveys will not affect participation in the program.

Benefits:

There are no direct benefits to a participant from participating in the evaluation study. The information we gain will help us improve the further development of Creative Coding Education for future use within after-school clubs and other secondary education institutions.

Confidentiality:

Any information obtained during this study, which could identify you, will be kept strictly confidential. We will keep a record of names connected to the surveys only to ensure the pre- and post-surveys are linked for the same individual. We will not retain any names in the data files after all data have been obtained and linked. All files of survey responses and other data will be maintained on secure computers and will only be seen by the investigator and co-investigator and other members of their research teams certified to conduct research. Original surveys collected will be destroyed after the data has been downloaded to the investigator's computer. The information obtained in this study may be published in scientific journals or presented at scientific meetings. The data will be reported as aggregate data.
.

Compensation:

There is no compensation for participating in this research.

Opportunity to Ask Questions:
You may ask any questions concerning this research and have those questions answered before agreeing to participate in or during the study. Or you may contact any of the Investigators: Dana Hoppe, UGRD, Principal Investigator Office (402) 942-4201 or Leen-Kiat Soh, PhD, Secondary Investigator Office (402) 472-6738.

Sometimes participants have questions or concerns about their rights. In this case, you may contact the University of Nebraska- Lincoln Institutional Review Board at (402) 472-6965 for the following:
- if you want to voice concerns or complaints about the research
- in the event of a research related injury.
- you wish to talk to someone other than the research staff to obtain answers to questions about your rights as a research participant
- to voice concerns or complaints about the research
- to provide input concerning the research process
- in the event the study staff could not be reached.

Freedom to Withdraw:

Participation in this study is voluntary. Students are free to decide whether or not to participate in this study or to withdraw at any time without adversely affecting your child's relationship with the program, LPS, the University of Nebraska-Lincoln, or participation in the program. Your decision will not result in any loss or benefits to which your child is otherwise entitled. Students are free to not answer any specific survey questions if they feel uncomfortable providing an honest answer.

Consent, Right to Receive a Copy:
YOU ARE VOLUNTARILY MAKING A DECISION WHETHER OR NOT FOR YOUR CHILD TO PARTICIPATE IN THIS STUDY. BY SIGNING BELOW, YOU CERTIFY THAT YOU HAVE DECIDED FOR YOUR CHILD TO PARTICIPATE. YOU MAY COPY THIS CONSENT FORM TO KEEP FOR YOUR OWN REFERENCE.

"Having read and understood the information presented here,

I,______________________________(Parental/Guardian Name Printed)

agree to allow_________________________(Student Name Printed) to participate in the study."

Parental/Guardian Signature_________________________________.

Name and Phone number of investigator(s)
Dana Hoppe*, UGRD, Principal Investigator Office (402) 942-4201*
Leen-Kiat Soh*, PhD, Secondary Investigator Office (402) 472-6738*