EFFECTS OF ROBOTICS INSTRUCTIONAL METHODS ON COMPUTATIONAL THINKING SKILLS OF MIDDLE SCHOOL STUDENTS

by

Andrew Patrick Cook

A dissertation

submitted in partial fulfillment

of the requirements for the degree of

Doctor of Education in Educational Technology

Boise State University

May 2021

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the dissertation submitted by

Andrew Patrick Cook

Dissertation Title: Effects of Robotics Instructional Methods on Computational Thinking Skills of Middle School Students

Date of Final Oral Examination: 08 April 2021

The following individuals read and discussed the dissertation submitted by student Andrew Patrick Cook, and they evaluated his presentation and response to questions during the final oral examination. They found that the student passed the final oral examination.

Youngkyun Baek, Ph.D.	Chair, Supervisory Committee
Yu-Hui Ching, Ph.D.	Member, Supervisory Committee
Yu-Chang Hsu, Ph.D.	Member, Supervisory Committee

The final reading approval of the dissertation was granted by Youngkyun Baek, Ph.D., Chair of the Supervisory Committee. The dissertation was approved by the Graduate College.

DEDICATION

This body of work is dedicated to my parents, who instilled in me a love of learning as a child that has taken me on a life journey I could have never imagined. This is for you.

ACKNOWLEDGEMENTS

I would like to thank the entire faculty of the educational technology department at Boise State University, particularly the members of my committee, Dr. Youngkyun Baek, Dr. Yu-Hui Ching, and Dr. Yu-Chang Hsu. Their support, feedback, encouragement, and guidance were invaluable in completing this dissertation.

For the past 19 years, I have had the privilege of meeting and working with amazing educators and students from all over the world, many of whom have become friends. With 3 continents and 6 countries involved, there are simply too many of you to acknowledge here by name. In one way or another, you were all a part of this journey.

Finally, I need to thank my wife for her support, encouragement, and patience as I embarked on my doctoral journey. It was far from a smooth path to follow. Thank you for putting up with the lost weeknights and weekends spent writing and researching, and the grumpiness that sometimes resulted.

ABSTRACT

As a tangible and motivating medium for students to engage in computational thinking, robotics has drawn interest from educators and researchers as K-12 schools continue to integrate STEM into curriculum. Through this mixed methods study, the researcher sought to explore the effects of robotics instructional methods (task-based and project-based) on the computational thinking skills of middle school students, including the problem-solving strategies used and the role of peer collaboration. The quantitative results of this study indicated no significant difference in the computational thinking skills of students participating in task-based or project-based robotics instruction. Interviews consisted of open-ended questions in which problem-solving and collaboration in robotics were explored from the perspectives of the participants. In both groups, problem-solving strategies encompassed all aspects of computational thinking as students took an iterative approach to problem-solving in both tasks and projects. Peer collaboration was naturally occurring and frequent among both groups. In task-based robotics instruction, peer collaboration and problem-solving strategies were primarily focused on the programming of the robot. In project-based robotics, peer collaboration and problem-solving strategies were applied throughout the entire design process, including the building and the programming of the robot. Through this study, the researcher hoped to provide a roadmap for the implementation of robotics in schools for K-8 students. As schools are increasingly seeking ways to integrate robotics into school curriculum, further research in this area on a larger scale is recommended.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BCC	Bebras Computing Challenge
BSU	Boise State University
EV3	Third generation of Lego Mindstorms robotics kits
GC	Graduate College
IBL	Inquiry-based Learning
NXT	Second generation of Lego Mindstorms robotics kits
PBL	Project-based Learning
STEM	Science, Technology, Engineering, Mathematics
TDC	Thesis and Dissertation Coordinator
TIS	The International School (pseudonym for the research site)

CHAPTER ONE: INTRODUCTION

The attention paid to robotics programs by researchers, educators, and schools is part of a renewed interest in recent years about the role that computer programming can play in educational environments. This interest is not new, but has gradually reemerged in cycles over the past several decades (Kafai et al., 2014). Computer programming has traditionally been viewed as a screen-based activity. However, computer programming through robotics has captured the interest of researchers and educators in part due to the opportunity for users to build and manipulate tangible, real-world creations through programming. The tangible nature of robotics has been noted among researchers as promoting increased motivation, self-ownership over learning among students, and increased interest in STEM-related subject matter (Bers et al., 2014; Huang et al., 2013; Nugent et al., 2010; Park, 2015; Petre & Price, 2004; Ucgul & Cagiltay, 2014).

Part of the rise in interest in getting computer programming into schools is the promise and potential of computational thinking (Kafai et al., 2014; Lye & Koh, 2014; Wing, 2006). Computational thinking has become something of a buzzword in the field of educational technology over the past decade, but the phrase is still sometimes misunderstood and often wrongly assumed to mean thinking like a computer (Kafai et al., 2014; Wing, 2006). Computational thinking is much more complex than this, and is more of an umbrella term under which several strategies, problem-solving components and approaches, and dispositions are grouped together (Brennan & Resnick, 2012; ISTE & CSTA, 2011). Although associated with computer programming (coding) and its reemergence in schools, computational thinking is not limited to one subject discipline, nor is it solely in the domain of computer scientists and engineers. Rather, computational thinking is transdisciplinary and has broad implications for how students approach and solve problems regardless of the context (Wing, 2006).

Three main dimensions make up computational thinking: *concepts*, *practices*, and *perspectives* (Brennan & Resnick, 2012; Lye & Koh, 2014). However, Lye and Koh (2014) pointed out that most existing studies on computational thinking in schools focus heavily on the concepts dimension and ignore the dimensions of practices and perspectives. As the practices and perspectives dimensions are what can illustrate computational thinking as a powerful mindset for problem-solving across subject disciplines, this narrow focus represents a significant gap in existing literature. Robotics represents a natural fit for task-based and project-based learning engagements, which can provide ample opportunities to shed light on the practices and perspectives dimensions of computational thinking (Park, 2015; Petre & Price, 2004).

By design, students in a robotics class all participate in the experience of using programming with robots to problem solve, experiment, and create. However, the processes and strategies students use while living through this experience can vary. This is inevitable, as one of the hallmarks of programming and robotics is the ability to solve problems in multiple ways (Bers et al., 2014; Lye & Koh, 2014). Therefore, it is important to understand the common experiences among students participating in robotics instruction as well as take a closer look at the different ways students come to live this experience in the contexts of design thinking, problem-solving and peer collaboration.

Study Context and Setting

Middle school students at The International School (TIS) in southern Africa primarily participate in robotics through trimester-based elective courses. However, there are certain units in upper elementary classes and in middle school where robotics integration has been developed as an option for students to demonstrate their understanding of unit concepts. In these activities, the school's technology integrator typically works closely with teachers and students to facilitate these experiences. Students at TIS may also gain experience with robotics through afterschool activity programs and special events, but these experiences are not connected to school curriculum and are sometimes facilitated through outside providers. For most students, the middle school elective courses represent their initial exposure to robotics as well as the primary setting for problem-solving experiences and collaboration through robotics. This study focused on the experiences of students in these courses.

Research has suggested that robotics instruction needs to be concept-based and inquiry-driven to be able to deliver maximum benefits to teaching and student learning. The hands-on, exploratory nature of robotics meshes poorly with traditional teaching methodologies based on lecture (Kafai et al., 2014; Nugent et al., 2010; Park, 2015; Petre & Price, 2004; Slangen et al., 2011). High-level (abstract) concepts relevant to robotics include systems, function, and causation (Slangen et al., 2011). These concepts form the basis for students in robotics to explore artificial intelligence (AI), mechanical engineering, and communicating through programming (Petre & Price, 2004). Drawing on existing student background knowledge of robotics, the middle school robotics electives at TIS incorporate learning engagements as part of concept-based units that can facilitate meaningful connection to STEM and other disciplines. These learning engagements are rooted in inquiry through purposeful questioning and set up so that all necessary conditions are present for students to explore and construct knowledge.

There are two levels of robotics courses at TIS – beginning and advanced. Both courses were initially developed based on publicly available materials from the Carnegie Mellon University Robotics Academy. These materials emphasize a problem-solving approach based on computational thinking concepts, providing students practice with computational thinking strategies from the start of each course. The robotics courses have been gradually expanded and revised over a period of five years to incorporate greater opportunities for computational thinking, concept-based learning, project-based learning (PBL), and collaboration. Along with these changes and tweaks to the curriculum, another major change has been adapting the content originally developed for the older Lego NXT Mindstorms kits over to the newer Lego EV3 Mindstorms kits. Both courses now use the EV3 kits.

The Beginning Robotics course consists mostly of teacher-created leveled robotic tasks that students have to solve through programming. The tasks are based on real-world robotics problems that are modelled using the EV3 robots. For example, a challenge to navigate a robot along a simple route and stop when it detects an object incorporates the basics of moving, turning, and sensing. Apart from the simplest tasks at the beginning of the course, most tasks are designed to be able to be solved in multiple ways through programming. Programming as well as fundamental computational thinking strategies such as breaking down bigger problems into smaller parts are explicitly taught throughout the course. Students follow a design cycle as part of the problem-solving process.

Towards the end of the course, the tasks become more complex and involve programming concepts such as flow control, parallel processes, and data handling from multiple sensors. These terms are introduced to students in student-friendly language consistent with the *EV3-G* programming software. In one unit of this course, students also have the opportunity to design their own task and assign it to others. Beginning Robotics provides students the fundamentals of programming, robotics, and design that they need to approach the tasks and projects in the advanced course.

In Advanced Robotics, students use the skills they have learned in the beginning course to complete more advanced tasks as well as create their own projects. The tasks typically involve the use of multiple sensors and advanced programming techniques based on experience gained in Beginning Robotics, while the projects typically fulfill a need or solve a real-world problem. Personal passion projects are also common and encouraged in this course. For example, a challenge to emulate a self-driving car consists of the EV3 stopping, slowing down, or speeding up depending on what color is detected using a color sensor. More task examples, including pictures, can be found in Appendix B. Students still follow a design cycle and use computational thinking strategies, but this is more student-driven in this course than it is in the beginning course. Popular projects in this course have been variations of robotic arms, sumo robots that compete in sumo matches with others, and a robotic hoverboard that makes use of a gyroscope sensor. More project examples, including pictures, can be found in Appendix C. Students typically spend time investigating and researching their project before starting it, and usually have an opportunity to present it to other students or parents when they are finished. The type of project the students pursue often dictates the focus of any programming instruction the teacher needs to deliver, which is personalized and differentiated depending upon the needs of the project.

Statement of the Problem

As robotics continues to increase in popularity in schools, a more detailed look into what contexts and settings robotics is most effective in becomes vital. Nugent et al. (2010), for example, found differences in the effects and effectiveness of robotics instruction on student learning based on whether the instruction was delivered in an informal, camp-style setting or a formal classroom setting of varying lengths. Robotics in schools can include a mixture of contexts and settings such as afterschool activities, standalone electives, or direct integration into units of learning. The variations in the potential learning engagements and opportunities that exist within these contexts are important to explore if researchers and educators hope to build a framework that links theory and practice effectively.

Robotics has generated interest and enthusiasm among educators for its potential connections to inquiry-based and project-based approaches to learning, particularly among Science, Technology, Engineering, and Mathematics (STEM) subjects. However, implementations of robotics in schools have been inconsistent and there is no agreement among educators as to how to best integrate robotics into school curriculum (Park, 2015). Research has suggested that schools which have already implemented an inquiry-based curriculum stand to benefit the most from robotics in terms of learning gains, while potential benefits of robotics may be severely hampered in more traditional learning environments due to the focus on teacher-centered content delivery (Park, 2015; Ucgul & Cagiltay, 2014). However, even in schools which operate inquiry-based curriculum there are still many decisions that need to be made when considering robotics. These include instructional strategies for robotics as well as higher-level decisions from administrators that may include funding and teaching training for robotics (Park, 2015; Slangen et al., 2011).

Existing research into robotics at the K-12 level has focused more on informal instructional settings such as short-term camps or afterschool activities rather than robotics that is part of school curriculum or that has been integrated into existing units of learning (Park, 2015; Ucgul & Cagiltay, 2014). The benefits of robotics in terms of facilitating learning and connections with other subjects is limited in informal settings due to time limitations and other factors (Nugent et al., 2010). Disadvantages of robotics in informal settings include the lack of meaningful integration and connections to other subjects, an overemphasis on competition, and a reliance on shallow learning engagements (Park, 2015). For robotics that has been integrated into school curriculum through standalone courses or via unit integration, a variety of instructional approaches have been employed, including leveled challenges in various STEM contexts (Ucgul & Cagiltay, 2014).

While several existing studies on robotics have provided a window into robotics instruction and its characteristics, relatively few have focused on middle school students (Bers et al., 2014; Lye & Koh, 2014; Slangen et al., 2011). With middle school being an important time of transition as well as growth, the need exists for studies focused on the experience of robotics instruction among middle school students and the potential benefits it may bring as students begin to explore and direct their own learning in greater depth and engage in more meaningful collaboration with peers. This study explored the experiences of middle school students in robotics courses from the perspectives of the students. This included student performance and ability in the key dimensions of computational thinking discussed previously, as well as the roles that problem-solving and peer collaboration played in the robotics classroom. This performance was measured among 2 groups of students who participated in either task-based or project-based robotics.

Research Questions

1. Is there a significant difference between the change of computational thinking skills of middle school students participating in task-based robotics versus project-based robotics?
2. How do middle school students collaborate with peers in task-based robotics versus project-based robotics?
3. How do middle school students use various problem-solving strategies in robotics activities?

Definition of Terms

EV3 Mindstorms – Third generation of Lego robotics kits designed for children and young adults.

Computational Thinking - A mindset of inquiry for transdisciplinary problem solving that involves efficiently organizing a large problem into logical steps, developing a systematic, algorithmic solution, and adapting this solution for different contexts as necessary through meaningful collaboration and self-reflection (Brennan & Resnick, 2012; ITSE & CSTA, 2011; Wing, 2006, 2008).

Three Dimensions of Computational Thinking – A computational thinking framework developed by Brennan and Resnick (2012) consisting of concepts, practices, and perspectives. The concepts dimension focuses on the computer science concepts central to computational thinking, while the practices and perspectives dimensions center on problem-solving strategies and the experiences that result from them.

Robotics Activity Type – The instructional method of the robotics course. This is either task-based or project-based. The content difficulty level in the course curriculum scales with the student needs in both instructional methods (e.g., add-ons and tweaks to the tasks/projects can cater to all levels of difficulty as needed on a per student/group basis).

Task-based Robotics – A method of robotics instruction characterized by teacher-created tasks that students complete by programming the robot. The tasks can be adjusted to provide greater challenge when necessary.

Project-based Robotics – A method of robotics instruction in which students build and program robot to fill a need or solve a problem of their choice, with the teacher acting as a facilitator and guide.

Lived Experience – The unit of analysis in phenomenology. A focus of this study will be on the shared experience of participation in robotics instruction (with a focus on problem solving and computational thinking) from the perspective of the participants.

CHAPTER TWO: LITERATURE REVIEW

Computer programming (also known as coding) for young children is receiving renewed interest from researchers, educators, and administrators (Kafai et al., 2014). As one of the more tangible and authentic coding environments with appeal to young learners, robotics has emerged as a popular medium for teaching programming. However, the rise of robotics in schools is more than an educational fad. Recent research has shown that children learn programming more effectively within authentic contexts such as robotics (Lye & Koh, 2014; Slangen et al., 2011). Robotics has deep connections across the content areas of school curriculums, including mathematics, science, engineering, and computer science, and can facilitate authentic inquiry-based learning (IBL) through the manipulation of tangible creations (Slangen et al., 2011). Among schools with established robotics programs, it is not unusual to see robotics integrated into units of study across disciplines, and its social and collaborative nature aligns well with inquiry-based educational philosophies (Park, 2015).

Learning through Robotics

A logical starting point for a discussion about robotics and learning is with the late Seymour Papert, given his association with constructionism and robotics. In *The Children's Machine*, Papert (1993) specifically discussed his collaboration with Lego on an invention that would eventually help create the first line of robotics kits aimed specifically at facilitating self-directed learning for children – the *Lego Mindstorms* kits. The theory of constructionism, which emphasizes active learning and personal as well as social construction of knowledge through exploration and making, attributed to Papert was central to this invention (Kafai et al., 2014; Papert, 1993). Papert found the term ‘robotics’ too limiting for the kind of knowledge construction that his invention would enable, and instead referred to it as a “cybernetic construction set” which would serve “as a staging area for making connections with other intellectual areas, including biology, psychology, economics, history, and philosophy” (Papert, 1993, p. 181-182). Much of the recent research into robotics focuses on the potential for cross-curricular learning, including STEM, that Papert outlined over two decades ago (Bers et al., 2014; Nugent et al., 2010; Park, 2015; Petre & Price, 2004; Slangen et al., 2011; Ucgul & Cagiltay, 2014).

Research has outlined the connections between robotics and STEM, as well as how robotics can promote motivation, confidence, and enthusiasm across the curriculum. In a qualitative study about robotics in a camp setting, Ucgul and Cagiltay (2014) described the necessary shift needed in the traditional role of the teacher to a facilitator role in order to best facilitate constructionism in robotics instruction. This shift brings out the natural collaboration and enthusiasm characteristic of robotics, while also promoting social interaction and interest in applying new learning (Bers et al., 2014; Park, 2015, Slangen et al., 2011; Ucgul & Cagiltay, 2014). These characteristics are shared by both robotics and constructionism. With the teacher in a more traditional role of gatekeeper of knowledge, the opportunities for children to construct their own learning are limited, a consequence that Papert described in detail throughout his work (Kafai et al., 2014; Papert, 1993).

PBL and IBL are frequently mentioned in studies about robotics (Bers et al., 2014; Huang et al., 2013; Nugent et al., 2010; Park, 2015; Petre & Price, 2004; Ucgul &

Cagiltay, 2014). Robotics has deep potential connections across the content areas of school curriculums, including Mathematics, Science, Engineering, and Computer Science, and can facilitate and complement authentic IBL through the manipulation of tangible creations (Kafai et al., 2014; Nugent et al., 2010; Slangen et al., 2011). It is not unusual to see robotics integrated into units of study across disciplines, and its social and collaborative nature aligns well with inquiry-based educational philosophies (Nugent et al., 2010; Petre & Price, 2004). The cycle of critical thinking, analysis, reflection, and application that occurs in robotics instruction is consistent with and complementary to IBL (Gonzalez, 2013; Park, 2015). Educational experiences that promote critical thinking are vital for both IBL and PBL (Gonzalez, 2013). Research has also suggested that IBL is not only a compatible approach with robotics, but that an IBL environment that includes the teacher acting in a facilitator role may be crucial for effective robotics instruction that maximizes opportunities for student learning (Nugent et al., 2010; Park, 2015; Petre & Price, 2004).

PBL shares many characteristics of IBL, including roots in constructivism, but a few characteristics of PBL stand out as especially compatible with robotics. The focus on group collaboration in PBL makes it a desirable approach for robotics given the collaborative nature of robotics learning environments (Bers et al., 2014; Huang et al., 2013; Nugent et al., 2010; Park, 2015; Petre & Price, 2004; Slangen et al., 2011; Ucgul & Cagiltay, 2014). Robotics instruction in a PBL environment is highly engaging and can enhance essential practices in robotics such as risk-taking and the cyclic process of developing, testing, and revising in order to reach a solution (Nugent et al., 2010). As with IBL, in PBL it is vital that the teacher act as a facilitator or coach rather than a gatekeeper in order to maximize opportunities for authentic and engaging learning experiences.

Lye and Koh (2014) argued that PBL environments provide the greatest opportunities to maximize the potential for development of computational thinking skills in programming activities such as robotics, with the authentic nature of PBL being the key driver. PBL can inject a social element into robotics instruction, increasing engagement and motivation as well as provide authentic opportunities for collaboration. Slangen et al. (2011) observed that this collaboration occurred naturally in the context of robotics instruction as students sought help from classmates as well as an audience for their creations. Park (2015) came to similar conclusions, noting that the authentic contexts provided by PBL for robotics instruction can stimulate further interest and connections in related STEM fields. More research into how students and educators can leverage these connections from robotics instruction that is integrated within school curriculum is warranted (Park, 2015).

Robotics and Programming in Research and Practice

Robotics is emerging as the medium of choice for teaching students programming. Not only does robotics allow for tangible creations through coding, it is also a social and collaborative experience (Petre & Price, 2004). Slangen et al. (2011) observed that this collaboration occurred naturally in the context of the instruction as students sought help as well as an audience for their creations. As programming tasks with the robots become more complex with the introduction of sensors, students often begin to see the possibilities of imparting artificial intelligence (AI) into their robot by manipulating the data from the environment that sensors capture (Slangen et al., 2011).

Flow control (including looping and conditionals) is an example of a non-trivial aspect of programming that can be more easily demonstrated and practiced through robotics than an output on a computer screen. Students may create a grabbing contraption, for example, and then program it to pick up multiple objects in a line by using a loop. An object can then be changed (for example, it could be painted a different color than the other objects), and a conditional statement (if/then/else) can be added to the loop to avoid picking up that object. Experimenting with and tweaking a learning engagement in this way facilitates in-depth, authentic learning experiences that students can connect across the curriculum (Nugent et al., 2010).

A very important aspect of robotics instruction that has been the focus of research itself is the programming environment. For robotics studies involving children, the programming environment is usually a software application using a graphical, on-screen block-based language, although in some studies such as Bers et al. (2014), a visual language integrated with tangible blocks known as *CHERP* has been used. More commonly, studies involving the use of Lego Mindstorms robotics kits will focus on the participants using *NXT-G* or *EV3-G*, which are graphical drag-and-drop programming environments. These environments differ significantly from traditional text-based programming environments, which require a degree of expertise with programming syntax and technical knowledge to use effectively (Kafai et al., 2014).

Visual programming languages are most commonly made up of blocks represented on a screen. The blocks are placed in sequential order by the programmer and certain parameters for each block may be configured. When a program is run, the blocks are executed in the order represented on the screen. Although the program is still compiled, this phase is invisible to the programmer in visual programming environments and the programs execute immediately. The major difference between visual and text-based programming environments is the lack of syntactical requirements in visual languages. Blocks can be added, removed, and rearranged with a simple swipe or mouse click, enabling relatively simple debugging and experimentation (Kafai et al., 2014; Slangen et al., 2011; Petre & Price, 2004).

Visual programming languages have been shown to increase interest and motivation among young students by reducing cognitive load, while also eliminating the anxiety and frustration that can sometimes surface when learning how to program (Bers et al., 2014; Kafai et al., 2014). A study by Okita (2014) found that students who initially learned programming in text-based environments were more effective at transferring and applying their programming skills to new situations. However, the study was limited to tasks involving simple robotics programming such as basic movements. More advanced tasks such as using variables, conditionals, and flow control are more easily represented and accessible for young learners through the use of visual languages (Bers et al., 2014; Kafai et al., 2014; Petre & Price, 2004; Slangen et al., 2011).

A wide variety of research methodologies have been employed in recent research involving K-12 robotics instruction. These include qualitative, quantitative, mixed methods, and design-based research (DBR) (Kopcha et al., 2017; Park, 2015; Petre & Price, 2004; Slangen et al., 2011). An equally wide variety of settings and contexts have been explored in recent robotics research, including informal settings such as camps and after-school activities, formal standalone courses, and subject integration (Altin & Pedaste, 2013). While this review examined several studies across the age range of K-12 from a variety of contexts and methodological approaches, the main areas of focus were the types of robotics instruction implemented such as project and task-based instruction, the connections between robotics and inquiry-based (IBL) and project-based (PBL) learning, the role of collaboration in robotics, computational thinking, and the experience of robotics instruction from the perspective of the middle school students.

Contexts and Settings for Robotics

Recent research about robotics has taken place in two main contexts – informal camp-style settings in which the robotics instruction and activities are not integrated into school curriculum, and classroom or classroom-like settings in which the robotics instruction is either integrated into units of learning or taught as standalone courses with its own curriculum within a school. Both settings have advantages and disadvantages in terms of the breadth of the overall educational experience, opportunities for connections to other disciplines, the application of new knowledge and skills, the role of competition, and several other factors (Bers et al., 2014; Nugent et al., 2010; Okita, 2014; Park, 2015; Petre & Price, 2004; Slangen et al., 2011; Ucgul & Cagiltay, 2014).

Regardless of the setting or context for the robotics instruction, there are some commonalities among the findings of many recent studies. These findings include the high levels of motivation and engagement among students, the tendency of robotics to promote natural and authentic collaboration between individuals and groups, and that robotics is a prime vehicle for learning programming even when students have no prior programming experience (Bers et al., 2014; Huang et al., 2013; Nugent et al., 2010; Okita, 2014; Park, 2015; Petre & Price, 2004; Slangen et al., 2011). Inquiry as a teaching and learning framework does not necessarily need to be specifically applied to robotics as might be necessary in other disciplines, as the nature of robotics itself is already aligned to inquiry and exploration (Bers et al., 2014; Kafai et al., 2014; Park, 2015; Slangen et al., 2011). Schools that are already implementing IBL in the classroom can expect robotics to be a natural fit for the curriculum, as it will aid students in making interdisciplinary connections as part of the learning process (Park, 2015; Slangen et al., 2011).

Robotics instruction in formal settings that is integrated into school curriculum is growing in popularity, though much of the research in robotics still focuses on informal instructional settings (Kafai et al., 2014; Park, 2015). This represents a significant gap in the current literature as robotics in formal school settings has a number of advantages over robotics instruction in informal camp-style settings, though many of these advantages may hinge on whether or not a school takes an inquiry or problem-based approach to learning (Nugent et al., 2010; Park, 2015). These advantages include the benefits of using robotics as vehicle for teaching programming, opportunities for in-depth exploration of robotics-related research and problem posing, and extended opportunities for students to make connections and integrate robotics into other subjects and disciplines, particularly STEM (Bers et al., 2014; Kafai et al., 2014; Nugent et al., 2010; Park, 2015).

Computational Thinking through Robotics

The power and promise of computational thinking as a mindset for problem solving has received much attention in recent years, but no recognized framework for facilitating and supporting it within school curricula exists (Kafai et al., 2014). Various definitions of computational thinking all recognize its roots in computer science principles (Brennan & Resnick, 2012; ISTE & CTSA, 2011; Kafai et al., 2014; Wing 2006). However, computational thinking has come to be recognized as much broader than just an aspect of computer science, and much of the current interest in it is due to the promise of its application to problem solving regardless of the context (Wing, 2006, 2008). Although computational thinking can be taught and modeled through sorting activities, puzzles, and games, computer programming is seen as a natural way to get students experimenting with and practicing computational thinking (Fessakis et al., 2013; Kafai et al., 2014). The tangible nature of robotics represents a logical medium for introducing and facilitating computational thinking skills via programming (Slangen et al., 2011).

Defining Computational Thinking

Wing (2006) essentially defines computational thinking as thinking like a computer scientist, while also taking great care to differentiate thinking like a computer scientist from thinking like a computer. Specifically, Wing (2008) defines it as “taking an approach to solving problems, designing systems and understanding human behavior that draws on concepts fundamental to computing” (p. 3717). Thinking like a computer scientist involves employing strategies and concepts from computer science such as abstraction, decomposition, and recursion. In computer science, abstraction is the process of encapsulating blocks or lines of code into a procedure that focuses on a specific part of a larger problem, and then often applying that procedure to other problems in other contexts (Wing, 2008). Abstraction can be described as “the essence of computational thinking” (Wing, 2008, p. 3717).

Brennan and Resnick (2012) define computational thinking by dividing it into the three dimensions mentioned above: concepts ("the concepts designers employ as they program"), practices ("the practices designers develop as they program"), and perspectives ("the perspectives designers form about the world around them and about themselves") (p. 3). This definition is more explicit than others regarding the potential role computational thinking can play as a problem-solving mindset regardless of the subject. In particular, some key components of computational practices such as testing, debugging, and iterative design are consistent with inquiry-based and problem-based approaches to teaching and learning (Gonzalez, 2013). However, it is through computational perspectives that computational thinking can be seen as more of a mindset. This mindset is a problem-solving approach not only dependent on abstraction, testing, and debugging, but also on collaboration and interaction with others (Brennan & Resnick, 2012).

ISTE and CSTA (2011) defined computational thinking as a process with similar characteristics emphasized by Wing (2006) and Brennan and Resnick (2012). However, ISTE and CSTA (2011) also identify five dispositions "that are essential dimensions of computational thinking" (p. 1). These dispositions are: confidence in dealing with complexity, persistence in working with difficult problems, tolerance for ambiguity, the ability to deal with open-ended problems, and the ability to communicate and work with others to achieve a common goal or solution (ISTE & CSTA, 2011). They have much in common with dispositions found in inquiry-based approaches and curriculum, which are normally a core set of non-subject specific attitudes which learners need to be successful and responsible for their own learning. These dispositions also help to position computational thinking as a mindset, much like the definition from Brennan and Resnick (2012).

In reviewing definitions of computational thinking from existing literature, three themes emerge: the idea of computational thinking as a mindset for problem-solving, the core characteristics of computational thinking that are rooted in computer science, and the idea that computational thinking is both multidimensional and transdisciplinary with a strong emphasis on the importance of collaboration and self-directed learning (Brennan & Resnick, 2012; ITSE & CTSA, 2011; Kafai et al., 2014; Wing, 2006, 2008). Looking at computational thinking through these themes avoids compartmentalizing computational thinking while emphasizing its potential to “be instrumental to new discovery and innovation in all fields of endeavor” (Wing, 2008, p. 3720). Papert (1993) had this goal in mind from the beginning of his collaboration with LEGO on early robotics kits.

Robotics as a Medium for Computational Thinking

Increased focus on the social component of computational thinking, known as *computational perspectives* or *computational participation*, has been prevalent in recent literature on the topic (Brennan & Resnick, 2012; Kafai et al., 2014). With the rise in inquiry-based educational approaches in schools, the prevalence of collaborative web-based applications, and the influence of the do-it-yourself (DIY) mindset of the maker movement, traditional contexts and settings for computer programming are being scrutinized and reevaluated by educators and researchers (Kafai et al., 2014). Wrestling with a compiler over syntax and function calls all in the name of getting a correct and functioning program was at the core of computer science instruction in schools in previous decades. Students still learned to think computationally, but the focus was not on the process of creation, interaction, or shared experiences, which are at the core of robotics instruction (Wing, 2008).

Robotics involves coding, but the results of the coding are in a tangible creation (the robot) rather than just on a screen. Particularly with younger learners, this has been shown to increase motivation and enables access to computational thinking strategies such as abstraction and decomposition (Bers et al., 2014, Kafai et al., 2014). Robotics also promotes a naturally collaborative approach to problem-solving, as students adopt an iterative approach while going through the cycle of planning, programming, testing, and debugging (Bers et al., 2014; Kafai et al., 2014; Lye & Koh, 2014; Park, 2015; Petre & Price, 2004, Slangen et al., 2011). Furthermore, robotics allows multiple methods of assessing the multiple dimensions of computational thinking as outlined by Brennan and Resnick (2012). These methods include the “artifact-based interviews” approach and the “design scenarios” approach (Brennan & Resnick, 2012, p. 22). These approaches allow more in-depth analysis and assessment of computational thinking, including the sharing and documentation of student experiences and perspectives (Brennan & Resnick, 2012). Robotics is particularly well-suited for the design scenario and artifact-based approaches for assessment of computational thinking.

Another reason robotics has been a focal point for research into facilitating computational thinking, particularly the robotics kits aimed at young learners such as *Lego Mindstorms*, are the visual, block-based drag-and-drop programming environments available. These programming environments scaffold the teaching of programming to a certain extent, eliminating potential frustrating experiences with syntax, compilers, or data types (Kafai et al., 2014; Petre & Price, 2004; Slangen et al., 2011). This results in an environment where students can easily access more advanced programming concepts such as variable usage, flow control, and procedure creation. This access allows students to experiment with and experience iterative design, decomposition, and abstraction in computer programming within a motivating and authentic context (Bers et al., 2014; Kafai et al., 2014; Petre & Price, 2004; Slangen et al., 2011).

The connections robotics has to various disciplines (especially STEM) means the types of learning engagements that can be created are numerous (Park, 2015; Slangen et al. 2011). Computer programming is a natural and authentic way to get students thinking computationally, which means robotics represents a medium for the integration of subject matter content with programming and computational thinking (Fessakis et al., 2013; Lye & Koh, 2014; Wing, 2006). The *EV3-G* programming environment, for example, has built in math libraries (in the form of blocks) to facilitate math operations, variables, and randomization, making math integration feasible and accessible. Robotics sensors such as the ultrasonic, temperature, and gyroscope sensors provide a framework for modeling scientific experiments. The motors and gearing mechanisms of robotics kits provide instant tangible examples of speed, torque, and gear ratio that can be manipulated and experimented with in many different contexts (Barak & Zadok, 2009).

The Bebras Computing Challenge for Measuring Computational Thinking

As a relatively recent worldwide initiative focused on bringing computational thinking opportunities to students of all ages, the Bebras Computing Challenge (BCC) has received attention from teachers and researchers, though it is still in the early stages of being used as an instrument for measuring computational thinking abilities (Dagiene &

Stupuriene, 2016; Román-González et al., 2017). The BCC consists of individual tasks that are not tied to specific computer languages or require background knowledge. This makes the BCC highly flexible and adaptable when integrating the tasks into classroom engagements or units of learning, including robotics courses (Dagiene & Stupuriene, 2016; Mannila et al., 2014). The BCC has shown the potential to contribute to both the assessment and development of students' computational thinking skills (Román-González et al., 2017).

A vital aspect of the BCC which has contributed to its promise and potential for addressing computational thinking skills is the organization and categorization of the tasks (Dagiene & Sentance, 2016; Lonati et al., 2017). BCC tasks are organized into sets intended for various age levels, with each set intended for an age range of about three years. The categorization of BCC tasks purposely addresses key concepts of computational thinking. These concepts are algorithmic thinking, pattern recognition, decomposition, and abstraction (Wing, 2006). For educators and researchers working in environments involving the BBC, awareness of and categorization of tasks with this organization in mind is crucial.

In a study about the content and usage of BCC tasks, Izu et al. (2017) found that the key concepts of computational thinking were not represented equally, with certain concepts being emphasized over others depending on the intended age levels for the tasks. Algorithmic thinking, for example, tends to be very well-represented in BCC task sets regardless of age range, while abstraction is emphasized more in the tasks aimed at younger age groups (Izu et al., 2017). If using the BCC for assessment of computational thinking, these factors can be vital for the reliability of the results that are intended to reflect computational thinking performance as a whole (Dagiene & Sentance, 2016). Elementary and middle school students may find tasks focused on abstraction and decomposition more challenging than tasks emphasizing pattern recognition and algorithmic thinking (Gujberova & Kalas, 2013).

Robotics Instruction

Task-based Robotics Instruction

Task-based instruction is a common instructional method for K-12 robotics and has been the subject of much of the recent research into robotics instruction involving children (Moorhead et al., 2015; Nugent et al., 2010; Okita, 2014; Rahman & Kapila, 2017; Sullivan & Bers, 2016). Some of the reasons for use of task-based instruction are logistical. Existing robotics curriculum is mostly task-based and aligns well to existing curriculum standards (particularly STEM), giving teachers and researchers a foundation upon which to build courses, unit integration, and experiments (Carnegie Mellon University, 2016).

Several other positive aspects of task-based robotics instruction related to student learning have been highlighted in research findings. Tasks that aimed at various levels of difficulty based on previously acquired skills can promote skills transfer and build confidence (Nugent et al., 2010; Okita, 2014). Task-based instruction provides opportunities for students to practice computational thinking strategies, and tasks can easily be aligned by researchers to test specific aspects of computational thinking (Rahman & Kapila, 2017; Sullivan & Bers, 2016). The iterative nature of task-based robotics instruction also supports the learning of programming in general, which is a necessary component of robotics instruction (Sullivan & Bers, 2016). Well-designed tasks can contribute to building student self-efficacy with robotics and programming, as well as promote collaboration on tasks with other students. According to Nemiro et al. (2017), well-designed tasks that can be solved through multiple approaches “develop and foster creative behavior in the students” (p. 85). Students or groups with different robot designs and different programs all working on a task is a sign of a well-designed task (Nemiro et al., 2017; Nugent et al., 2010). The peer observation that takes place as part of task-based robotics instruction is also key for learning, especially in environments in which robotics kits are shared between 2 or more students (Park, 2015; Yuen et al., 2014).

The nature of task-based robotics instruction also includes some potential drawbacks. Although student choice in approaching and creating solutions is a key part of robotics, for the most part the tasks are created by the teacher, which may limit student buy-in. The focus for both teachers and students in task-based robotics instruction tends to be on the end result or product that students create to solve a given task, rather than on the problem-solving process itself (Sullivan & Bers, 2016). Particularly in informal or short-term robotics contexts in which time for learning programming may be limited, this can result in learning experiences that lack depth and are limited in opportunities for practicing computational thinking strategies such as abstraction and decomposition (Nugent et al., 2010). It is important for robotics instructors to develop authentic and engaging tasks that motivate students (Barak & Assal, 2018; Park, 2015; Ucgul & Cagiltay, 2014). Just as teachers in an inquiry-based curriculum are constantly reevaluating the learning taking place and its direction, robotics instructors need to reassess tasks as the robotics instruction progresses, which may include task redesign to both maximize student motivation and provide an appropriate level of challenge (Yuen et al., 2014).

Project-based Robotics Instruction

Robotics instruction through a PBL-style approach has a long history, and many of the instructional interventions set up by researchers in recent studies about robotics are rooted in PBL (Ucgul & Cagiltay, 2014). Papert's theory of constructionism, which emphasizes active learning and personal as well as social construction of knowledge through exploration and making, was central to his early work on creating robotics kits specifically designed for children (Kafai et al., 2014; Papert, 1993). Given the characteristics of robotics instruction previously mentioned, PBL, with its student-centered philosophy and thematic, topical approach, is an appropriate research-based method for an instructional intervention in a robotics study. Robotics within a PBL-style environment helps give access to multiple perspectives and can facilitate critical questioning (Nugent et al., 2010). Through project-based approaches, opportunities exist to explore how using educational technology such as robotics in the classroom can enable more authentic learning experiences for students by facilitating collaboration, cooperation, and problem-solving within authentic contexts (Ching et al., 2018).

Opportunities for project-based instruction in robotics increase in more formal settings with adequate instructional time available. One advantage this approach of instruction has over task-based instruction is the potential for students to take over responsibility for their own learning, with the teacher playing the role of facilitator. Project-based approaches can increase motivation and enthusiasm for STEM fields by engaging students in authentic learning scenarios in which they play a role in selfdirecting their own learning (Bers et al., 2014; Park, 2015). This motivation can open up new pathways and opportunities for learning that students might not discover without robotics (Bers, 2007). In part because of its connection to other subject disciplines (especially STEM), project-based robotics has attracted students who would otherwise not be interested in robotics (Alimisis, 2013). This is an important factor in combating traditional biases in K-12 schools and among students that robotics is difficult, or only for boys.

Recent studies have shown that project-based robotics instruction aligns well with constructivist learning theory and inquiry-based learning. Kopcha et al. (2017) found that project-based instruction fostered both student independence and ownership of learning, which then put teachers in an ideal position to act as facilitators or guides. Bers et al. (2002) reported learning became more authentic for students working on robotics projects that were then shared with an audience, which is characteristic of PBL. High levels of student motivation have been observed in project-based robotics. Motivation, student ownership of learning, and student independence tend to be symbiotic characteristics that were observed in several studies (Altin & Pedaste, 2013; Ucgul & Cagiltay, 2014; Slangen et al., 2011). This can result in meaningful learning experiences for students participating in robotics activities that are student directed (Ucgul & Cagiltay, 2014). However, project-based robotics is often dependent on robotics and programming skills that inexperienced students may still be developing (Barak & Assal, 2018).

The relationship between project-based robotics and computational thinking has also been explored in recent research. The opportunity to create projects in robotics can facilitate skills transfer of previously learned programming and computational thinking strategies, potentially enabling application towards related problem-solving and projects in other disciplines such as science and mathematics (Atmatzidou & Demetriadis, 2016). These skills include decomposition, abstraction, and algorithmic thinking. Chen et al. (2017) noted a positive effect on computational thinking skills, particularly algorithm development and pattern recognition, among students participating in project-based robotics. Project-based robotics provides ample evidence for assessing computational thinking through the artifact-based approach as outlined by Brennan and Resnick (2012), and allows the instructor to facilitate and guide rather than lead (Carbonaro et al., 2004).

In a study involving project-based robotics instruction with middle school students, Barak and Zadok (2009) observed that the problem-solving process students engaged in was rooted in logical reasoning and of students' "instinctive understanding of the world" of how things work (p. 303). This aligns well with supporting the development of computational thinking in that it enables further learning and is part of what Brennan and Resnick (2012) refer to as "illuminating processes", or discussions with students about the process of problem-solving (p. 23). In the course of creating solutions for projects, students demonstrated the computational thinking skills of pattern recognition, algorithmic thinking, and abstraction (Barok & Zadok, 2009).

Collaborative Problem Solving in Robotics

The role that collaboration plays at the intersection of problem-solving, computational thinking, programming, and robotics has been the subject of research since Papert (1993) described his invention that came out of his early collaboration with LEGO. This invention, which led to the modern robotics kits of the present day, emphasized a social aspect towards knowledge construction as part of Papert's theory of constructionism (Papert, 1993). In recent years, this social aspect of problem-solving through programming and tinkering has been defined as computational participation (Kafai et al., 2014) and also categorized as a component of computational thinking called computational perspectives, which refers to expression and connection with others through computation as a medium for creation (Brennan & Resnick, 2012).

Computational participation is a phrase that refers to the injection of a social element into coding and computational thinking, and is particularly concerned with the perspectives dimension of computational thinking (Kafai et al., 2014). In addition to promoting and illustrating more authentic contexts in which to problem solve which include meaningful collaboration, computational participation could also describe the work environments of the software engineers of today (Denner et al., 2014). Defined by Kafai et al. (2014) as "the ability to solve problems with others, design systems for and with others, and draw on computer science concepts, practices, and perspectives to understand the cultural and social nature of human behavior" (p. 6), computational participation is embodied in the rise of computer programming environments and mediums with opportunities for authentic collaboration, sharing, and teamwork. The hands-on nature of robotics, with has been shown in studies to promote high levels of motivation, engagement, and natural collaboration, represents an authentic context for students to engage in computational participation (Leonard et al., 2016; Park, 2015; Petre & Price, 2004; Slangen et al., 2011).

Collaboration within robotics to some degree can be considered a natural effect of learning in an inquiry-based learning environment, and robotics instruction that takes place within established inquiry-based environments enhances opportunities for meaningful student collaboration (Lee et al., 2013; Park, 2015). In these environments, student collaboration is a significant factor in student motivation, student confidence, as well as knowledge construction through robotics (Leonard et al., 2016). However, robotics instructors need to carefully facilitate tasks or project opportunities if they want to encourage collaboration and maximize its benefits. Learning environments and tasks that are more structured and less open-ended can shift students to work independently rather than collaboratively. This can result from these types of tasks being seen as a competition as well as other factors (Lee et al., 2013).

Collaboration can be beneficial in both task-based and project-based robotics, but the goals of the collaboration and some of its effects can vary depending on the type of instruction. For both types, the tangible nature of robotics tends to promote student collaboration as well as peer feedback (Bers et al., 2014). In project-based robotics, collaboration "allowed students to observe how others learn and in turn helped individual students figure out new ways to expand their understanding" (Moorhead et al., 2015, p. 8). Authentic projects lead to greater opportunities for collaboration and help make the collaboration meaningful (Kopcha et al., 2017). This, in turn, enables instructors to observe and play the facilitator role, helping individuals and small groups as necessary.

Altin and Pedaste (2013) noted that collaboration in robotics has been defined "as actors sharing the same goal of task realization" (p. 369). Collaboration in task-based robotics often revolves specifically around the task, taking the form of initial brainstorming about the tasks to the steps involved in solving it (Altin & Pedaste, 2013; Rahman & Kapila, 2017). This collaboration is particularly important and beneficial in the planning stages of the task solving process. In this type of environment, students may collaborate and solve the task as a group, or collaborate in the planning process while still working on the actual tasks independently.

Although collaboration in some form will naturally arise from robotics, the role of the instructor can enhance student collaboration. Effective collaboration provides opportunities for peer observation and feedback, which in robotics is especially useful for programming and debugging (Kopcha et al., 2017). In collaborative programming activities, instructors should play the role of facilitator and deliver differentiated instruction between groups as necessary. Periodically checking-in with students and promoting multiple ways of problem-solving are effective ways of facilitating collaborative programming (Brennan & Resnick, 2012).

Conclusion

Robotics, as a tangible and multidisciplinary alternative to coding on a screen, represents an ideal medium for introducing programming to children (Bers et al., 2014; Kafai et al., 2014; Petre & Price, 2004; Slangen et al., 2011). Children are highly motivated when involved in robotics; robotics is naturally collaborative and inquiry-driven, and the skills and mindsets such as computational thinking that are developed while programming have potential applications to almost all other subject content areas (Kafai et al., 2014; Slangen et al., 2011). With the rise in interest in computational thinking, teachers and administrators are exploring ways to get robotics and coding integrated into schools at all levels (Kafai et al., 2014). However, middle school has not seen the same interest or scale of implementation from researchers and practitioners as other grade levels, and existing literature and implementations have not adequately explored the links and the common cognitive processes shared between problem-solving in robotics and in other subjects (Lye & Koh, 2014; Slangen et al., 2011). Furthermore, robotics studies still tend to focus on robotics instruction in informal camp-style contexts rather than as an integral part of school curricula (Nugent et al., 2010; Park, 2015).

Both task-based and project-based instruction are common approaches for robotics instructors. Ultimately, a skilled and experienced instructor will be able to facilitate meaningful learning experiences using either approach. However, both approaches have potential drawbacks and limitations. Task-based robotics usually aligns well with existing standards and benchmarks, and tasks can be designed to specifically address and enable practice of computational thinking concepts (Nugent et al., 2010; Rahman & Kapila, 2017; Sullivan & Bers, 2016). As tasks are usually designed by the teacher, the level of authenticity in the learning experience may fall short of what students can experience in project-based robotics. Project-based robotics facilitates deeper learning experiences in general, as well as authentic, meaningful student collaboration (Altin & Pedaste, 2013; Ucgul & Cagiltay, 2014; Slangen et al., 2011).

CHAPTER THREE: METHODOLOGY

Restatement of the Problem

As STEM programs proliferate in K-12 schools, interest in robotics among educators and researchers is on the rise. For educators, implementation of robotics in school settings presents multiple challenges, including choosing an instructional approach and how to integrate this approach effectively with existing school curriculum (Park, 2015). Robotics can often require a significant investment in both funding and teacher training for K-12 schools, but there is no agreement among educators on how to best integrate robotics into teaching and learning (Park, 2015; Slangen et al., 2011). Robotics represents a promising medium for students to practice computational thinking skills, however existing research has focused more on informal instructional settings for robotics rather than formal classroom settings in which robotics is part of the school curriculum (Park, 2015; Ucgul & Cagiltay, 2014).

A need exists for studies focused on deliberate instructional approaches to robotics in formal instructional settings in K-12 schools. Robotics provides a platform in which all three dimensions of computational thinking as defined by Brennan and Resnick (2012) can be practiced: concepts, practices, and perspectives. With existing research on computational thinking in schools primarily focused only on the concepts dimension, robotics studies from formal classroom settings as in this study provide opportunities for a broad exploration of computational thinking in K-12 schools that includes examinations of computational thinking from the practices and perspectives dimensions (Lye & Koh, 2014; Park, 2015; Petre & Price, 2004).

Research Questions

1. Is there a significant difference between the change of computational thinking skills of middle school students participating in task-based robotics versus project-based robotics?
2. How do middle school students collaborate with peers in task-based robotics versus project-based robotics?
3. How do middle school students use various problem-solving strategies in robotics activities?

Research Method

Mixed methods research has been around in various forms for several decades, but has seen a surge of interest in recent years (Creswell & Plano Clark, 2017; Johnson et al., 2007). Mixed methods research always involves both qualitative and quantitative data and the mixing of these, but there are several types of mixed methods designs, as well as several techniques for mixing, that can be employed depending on the nature and the goals of the study (Creswell, 2014; Creswell & Plano Clark, 2017). The main appeal of mixed methods is generally the opportunity to add depth and breadth to quantitative and qualitative data, while increasing the credibility and validity of both within the study (Archibald et al., 2015; Creswell, 2014; Creswell & Plano Clark, 2017; Johnson et al., 2007). This study measured computational thinking skills quantitatively, but also incorporated participants' perspectives on the experiences of problem-solving and collaboration within robotics courses through qualitative methods.

This study collected quantitative data in the form of scores from separate pre and post tests of an instrument for measuring computational thinking skills. The pre-test was administered during the first week of the robotics courses and the post-test was administered during the last week of the robotics courses, for both groups. As discussed later in this chapter, both the total raw scores on the tests as well as student performance in four distinct areas of computational thinking were collected and analyzed to address research question 1. The pre-test can be found in Appendix I. A course timeline can be found in Appendix E.

This study used a qualitative phenomenological approach in the form of interviews focused on problem-solving strategies and peer collaboration that occurred during the robotics courses from the perspectives of the participants. Collected in the middle part of the robotics courses, this data was meant to complement the quantitative findings as they related to research question 1. The qualitative data added depth and breadth to the study as a whole through participants' descriptions of problem-solving and collaboration in robotics. The specifics of the data collection, instrumentation, and data analysis are discussed throughout the rest of this chapter. Interview questions and themes can be found in Appendix A.

Research Design

Several types of mixed methods research designs have emerged to suit a wide variety of studies, with Creswell and Plano Clark (2011) outlining six major types. The choice of design depends on several key factors (Creswell & Plano Clark, 2011). Central to this choice is whether the use of the quantitative and qualitative methods in a study are already determined, or if these methods are subject to change as the study progresses.

Creswell and Plano Clark (2011) refer to these designs as fixed and emergent, respectively. The choice to carry out the methods simultaneously or on their own, as well as the focus of the study in terms of which method, if any, is to be prioritized also merit consideration (Johnson et al., 2007). The decision as to which phase in the study the mixing primarily occurs will also help to determine the design choice (Creswell & Plano Clark, 2011).

This study reflected a quasi-experimental, mixed methods fixed convergent design, as participants were assigned into robotics courses based upon their elective choices for the academic school year and other scheduling needs. Figure 1 below illustrates the research design for this study. The convergent design allowed for both quantitative and qualitative data to be gathered together (during the classes), but analyzed separately, with the mixing occurring in the data interpretation phase (Archibald et al., 2015; Creswell & Plano Clark, 2017).

As there were two sections of the Advanced Robotics class with roughly even numbers of students, one class was selected at random and given task-based activities, while the other was given project-based activities. The qualitative data was meant to add depth and detail to the quantitative analysis on research question 1, particularly in the areas of problem-solving strategies and the role of peer collaboration in task-based and project-based robotics. Very basically, the quantitative data served as the *what* in this study, while the complementary qualitative data shed more light on the *why* and *how* through illustrating participant personal experiences with robotics instruction and computational thinking, as well as peer collaboration.

To address research question 1, the independent variable was robotics activity type which consisted of two levels – task-based and project-based. The dependent variable was computational thinking skills. Quantitative data consisted of performance results on the BCC task sets administered pre and post course. To address research questions 2 and 3, interviews were conducted with the participants. Interviews followed a phenomenological approach, consisting of broad and open-ended questions often arising in the moment as students worked through tasks and projects (Creswell & Poth, 2016). The open-ended aspect of these interviews aligned well with the hands-on and student-centered characteristics of a robotics class. As shown in Appendix A, the areas of focus for the interviews were problem-solving strategies and peer collaboration, which directly addressed research questions 2 and 3.

The computer programs students created during robotics instruction served as natural visible thinking opportunities for students to describe their experiences with robotics and problem-solving strategies and approaches, which were followed up by the researcher with questions incorporating the artifact-based approach described by Brennan and Resnick (2012) for assessing computational thinking strategies. Bracketing was employed in order to allow the study of the experience in an environment in which the perspectives of the participants, rather than the researcher, were emphasized (Creswell & Poth, 2016). This involved designing the questions, topics and themes, and format of the interviews in order to purposely facilitate open-ended and flexible conversations often centered around participant artifacts. The interview questions and topics can be found in Appendix A. The interview format and approach is discussed in further detail later in this chapter.

Data collection: Repeat for task-based robotics and project-based robotics

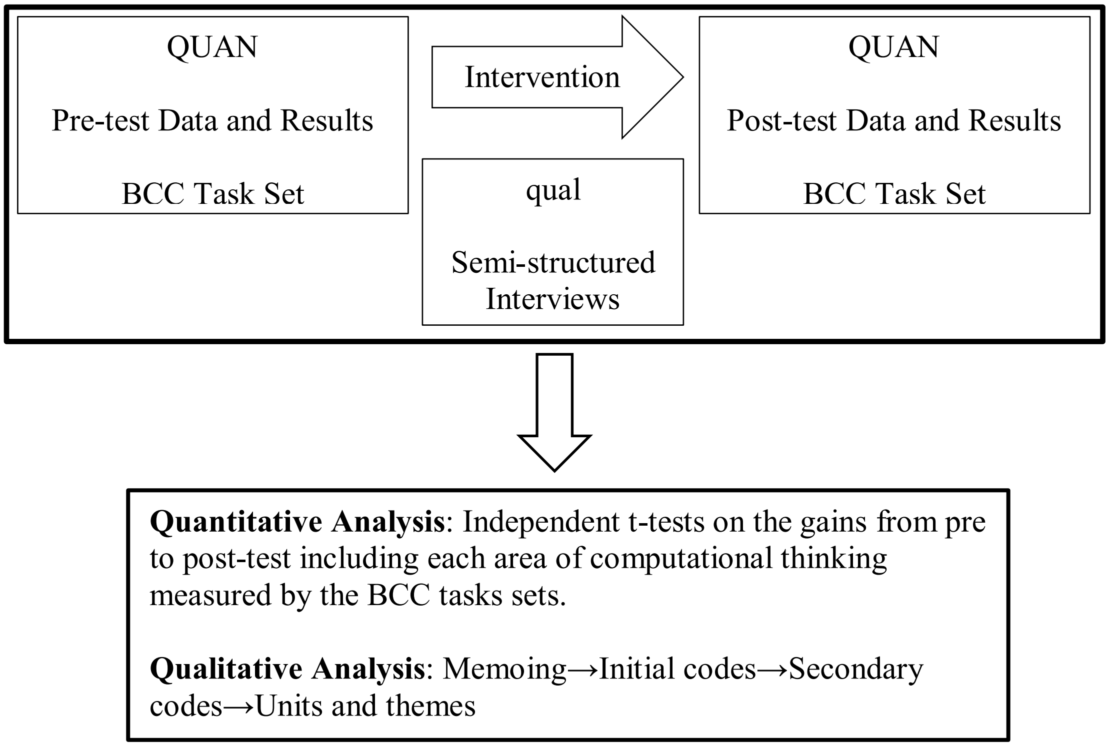


Figure 1 Research Approach

Research Context

As outlined in Chapter 1, there are two middle school robotics courses at the research site known as Beginning Robotics (with a task-based curriculum) and Advanced Robotics (with a primarily project-based curriculum along with some more advanced tasks). All students in Advanced Robotics had already taken and successfully completed Beginning Robotics, giving them a common baseline of robotics experience. Advanced Robotics was taught once in the second trimester of the school year and once in the third trimester, with different groups of students. One section was taught as project-based, while the other was taught as task-based. As electives, students chose to take these courses, though the specific class they were assigned to during the school year depended on scheduling and other logistics. Electives at the research site consist of mixed grade levels, so classes had a mix of students ranging from 11 – 14 years old. An example timeline of the robotics courses can be found in Appendix E.

Task-based Robotics

The task-based curriculum for this study built upon the programming and design fundamentals of robotics introduced in the Beginning Robotics course. As with the tasks in Beginning Robotics, the more advanced tasks for Advanced Robotics are loosely based on the curriculum from the Carnegie Mellon Robotics Academy, which focuses on problem-solving and computational thinking (Carnegie Mellon University, 2016). Before Advanced Robotics was shifted to a more project-based approach a few years ago, these advanced tasks constituted the majority of the course.

Students designed and built EV3 robots and programmed them to complete a variety of teacher-created tasks. These tasks included programming robotic attachments such as grippers and arms to pick up and move objects, combining multiple sensors to guide the robot through an obstacle course, and a series of challenges in which students created gearing mechanisms for speed and torque. Instruction in task-based robotics focused on the programming necessary for students to complete the tasks. The difficulty level for all tasks can scale with the needs of students. This is accomplished through extension options made for the tasks to make them more challenging (known as super challenges) as the students solve them. Through this process, the needs of all students are met and students are consistently challenged. Table 1 below shows the different tasks that were assigned, and more details including visuals can be found in Appendix B.

Table 1 Task-based Robotics Tasks

Task description	Extension option example
Navigate through "traffic lights" with the color sensor	Add in other behaviors when detecting certain colors
Program the robot to "mow the grass" by turning using the gyro sensor	Use the color sensor to detect the start and end of the "lawn"
Push objects into zones with the robot using the touch sensor	Push 4 boxes off a table without the robot falling off
Program a claw attachment to pick up a ball and move it to a location	Only pick up balls of a certain color
Make the robot move in a figure 8 pattern on the table	Make the robot speed up by 10% after completing each figure 8
Navigate through a maze using the gyro, ultrasonic, and color sensors	Navigate the maze in reverse after you complete it

Project-based Robotics

The project-based robotics curriculum draws on some elements from PBL. Students were introduced to several project ideas such as robotic arms, a robotic safe, and a moon rover robot, however, they were also encouraged to research other projects they had an interest in. Many students came into the course with their own ideas for a project (referred to at the research site as 'passion projects') and were encouraged to research and pursue these. As part of the course curriculum, students researched the needs and uses for their potential project and created a design plan. Students went through several design cycles of building, programming, and revising their plan, often seeking out feedback from both peers and the instructor. Towards the end of the course, students shared their creations with both their classmates and other students in the school.

As with the task-based robotics curriculum, the difficulty level of the projects can scale with student needs. Instructor and peer feedback was used to revise the projects, and the nature of the project drove the difficulty. Instruction in project-based robotics was necessarily differentiated to cater to the needs of the student or group. This included focused mini-lessons on a particular aspect of the EV3 programming language. Peer collaboration (particularly focused on programming) occurred among students working together on a project as well as those who were working on their own projects, with the testing phase of the projects often providing natural opportunities for collaboration. Table 2 contains a summary of the projects that the participants worked on during the study, and Appendix C contains more details and some visuals of the projects.

Table 2 Project-based Robotics Projects

Project	Description
Robot Sumo	Design, build, and program a “sumo” robot that tries to push other robots out of a ring.
Rubik’s cube solver	Program a robot to solve a Rubik’s cube
Robotic limb	Design, build, a program a robotic limb
Remote control robot	Program one EV3 brick to control another robot through Bluetooth

Participants

The research site for this study was a large preK-12 private international school in southern Africa. Research was conducted in the middle school (grades 6-8) robotics classes. These classes are electives, however they are still incorporated into the school curriculum as a whole, which is inquiry-based. The student community at the research site is primarily made up of children from diplomatic families who are working at embassies in the region, as well as families who have been posted abroad by multinational companies. Over 100 different nationalities are represented in the student community. The socio-economic status of the school community ranges from upper-middle class to wealthy.

Students participating in the Advanced Robotics elective classes, as well as the robotics teacher, were the participants. As the elective classes are mixed in grade level within the middle school, the students ranged in age from 11 – 14 years old. Of the total of 24 student participants, 19 were male and 5 were female. All participants had previously successfully completed the Beginning Robotics course, giving them a baseline of experience with robotics and programming. As class sizes were small, which is typical at the research site, every student in the robotics classes was recruited as a potential participant.

Participants were assigned into Advanced Robotics classes based on the scheduling needs and requirements of the school. One section of Advanced Robotics was randomly chosen for task-based robotics, while the other class participated in project-based robotics. There were 24 student participants, with a split of 11 students (10 males, 1 female) in the task-based robotics section and 13 students (9 males, 4 females) in project-based robotics. Normal procedures and safeguards concerning research with human subjects were implemented and followed. Half of the participants were interviewed (5 from task-based robotics and 7 from project-based) and asked questions from Appendix A. This included 1 female from the task-based section, and 2 females from the project-based section.

The researcher had worked with some of the participants in previous school years in other elective classes at TIS. However, this did not involve any possible bias, as all participants were interviewed based on the themes and questions outlined in Appendix A. Furthermore, the participants each drew from their own experiences with programming projects during the robotics courses when answering interview questions, with the researcher facilitating the discussion and taking notes. The researcher reminded all participants that although the interviews would be recorded, no personal identifying information would be used in the interview analysis or noted within the transcriptions. Individual transcriptions were filed under pseudonyms (e.g. “Student X”).

Instrumentation

To measure computational thinking skills, two hard copy BCC task sets of 15 tasks each for the 10 – 12 age group and 12 – 14 age group were administered to students. One set was administered during the first week of each robotics course and another during the final week of each robotics course. The task sets were organized in terms of difficulty and areas of computational thinking focus. The tasks addressed the key concepts of computational thinking and were categorized as easy (5 tasks), medium (5 tasks), or hard (5 tasks) and were organized by age group. The pre and post BCC task sets differed in that they consisted of variations of similar tasks that addressed the same objectives, however the format and difficulty levels remained the same. This is consistent with recent studies that have used the BCC tasks for measuring computational thinking skills (del Olmo-Muñoz et al., 2020; Delal & Oner, 2020). In some cases, these questions were slightly modified in order to be administered in paper and pencil format.

The BCC task sets were chosen primarily due to their relatively broad focus on computational thinking which included a focus on iteration and other practices, as well as the application of computational thinking skills in a variety of contexts (Román-González et al., 2017, 2019). Other considerations for the BCC task sets were logistical reasons involving ease of access and modification to meet the needs of this study. Other instruments for measuring computational thinking that were considered included *Dr. Scratch* and the *Computational Thinking Test*. While aimed at middle school students, the *Computational Thinking Test* had a narrow focus primarily limited to programming concepts (Román-González et al., 2017). *Dr. Scratch* was not compatible with the EV3-G robotics programming language at the time of this study.

The BCC task sets were purposely collated so that all key aspects of computational thinking, as well as difficulty levels, were as equally represented as possible. Each task was already categorized with the areas of computational thinking it addressed – algorithmic thinking, pattern recognition, decomposition, and abstraction – as well as its difficulty level. A breakdown of the BCC task sets (for both the 10-12 and 12-14 age groups) follows in Table 3. The BCC point counting system was used to score the tasks. This consisted of assigning 6, 9, or 12 points for each correctly answered task depending on the level of difficulty, subtracting a third of the possible points for incorrectly answered tasks, and assigning 0 points for tasks that were not attempted (Bebras, n.d.). The total of this represented the raw score of a student, with a maximum of 135 points possible.

Table 3 BCC Task Sets by Areas of Computational Thinking

Computational thinking area	Number of tasks present (out of 15)
Algorithmic Thinking	12
Pattern Recognition	8
Abstraction	4
Decomposition	4

Note. Several tasks addressed multiple areas of computational thinking.

Interviews questions and themes of focus were developed based on a phenomenological approach. Interviews were open-ended and semi-structured. The nature of this study, in which many types of projects and robotics tasks were attempted through many different problem-solving approaches, necessitated a flexible and adaptive approach to the interviews as students described and articulated their experiences with robotics instruction and computational thinking. However, a loose framework focused on problem-solving, peer collaboration and the experience of robotics from the participants' perspectives was employed for the interviews based on the suggestions for assessment of computational thinking developed by Brennan and Resnick (2012). Refer to Appendix A for a complete list of the interview questions. These strategies, which included guiding participants to discuss "illuminating experiences" and a focus on multiple perspectives and ways of problem-solving, aligned well with the phenomenological goal of describing the essence of an experience from the perspectives of the participants (Creswell & Poth, 2016).

Data Collection

In this design, quantitative data was collected in the form of BCC task sets (pre and post course) to address research question 1, while qualitative data in the form of interviews was collected from the participants to address research questions 2 and 3. Students received brief instructions before attempting the BCC task sets and were reminded that their performance on these tasks would not be connected to their grades or assessment in the robotics course. Students had 45 minutes to complete the task set in accordance with BCC guidelines (Bebras, n.d.). Raw scores were collected for both the pre- and post-tests according to the BCC scoring system described in the instrumentation section above. The differences in the scores between the pre- and post-tests were then computed and collected.

Qualitative data collection consisted of open-ended semi-structured interviews with students throughout each robotics course. Since problem-solving approaches and strategies in robotics can take many forms, no two interviews were exactly alike. However, Appendix A provided question starters and the areas of focus of which the interviews consisted. At times, the computer programs the students created during the robotics classes served as visible thinking opportunities for students to describe their experience and facilitated further questioning and discussion. Examples of these programs were collected and organized into the different aspects of computational thinking that they represent, and added into Appendix D. The interviews addressed the research questions by providing depth and descriptions for the quantitative findings.

A flexible rotating schedule of targeted students for interviews was developed in which all students would be asked the questions in Appendix A. The work that participants were doing in the classes drove much of the timing for interviews. Students were interviewed during class time within the classroom environment, however they had the option of being interviewed in more private location adjacent to the classroom.

Interviews took place when other students were engaged in independent work so as to minimize disruptions, and a teacher assistant was occasionally used to monitor the class during interviews. As all participants had completed Beginning Robotics previously, they were already somewhat accustomed to being asked about their thinking and problem-solving processes as part of the general assessment strategy for the course. However, interviews for this study were more in-depth than some participants were used to within a normal classroom environment. This was communicated to participants as part of the informed consent and assent processes.

All participants received and signed consent forms approved by the IRB. Four participants declined to be interviewed, while 8 others were not able to be interviewed due to scheduling and time constraints. Interviews were not formally scheduled for fixed dates and times; instead, the researcher identified opportune times to interview participants based on progress and milestones reached in their projects and tasks. This allowed the experiences to be fresh in the minds of participants as they described their problem-solving strategies and experiences. After each interview, the researcher made a short memo of initial reflections on the key points of the interview. All interviews were digitally recorded and automatically uploaded to password protected cloud storage. After completion of all the interviews, the researcher transcribed the interviews into a word processing document and removed any personal identifying information such as first names. Each participant's transcript was then put into separate word processing documents with the participant's pseudonym, in preparation for analysis using the ATLAS.ti Cloud analysis platform.

Data Analysis

For the quantitative data analysis, descriptive statistics were generated from the results (raw numerical scores) of the BCC task sets (pre and post course) as well as the growth between the pre and post results from each group. Descriptive statistics were also generated for each area of computational thinking from in Table 3 above. To address research question 1, independent t-tests were used to determine if there was a significant effect on the change (pre- to post-test) on computational thinking skills by each level of the independent variable. Additionally, independent t-tests were used to determine if there was a significant effect on the change in any area of computational thinking skills by each level of the independent variable. This data complemented the analysis of research question 1.

This study was carried out approximately 4 years after the researcher was first exposed to robotics, so personal experiences helped to frame and direct the qualitative analysis while also identifying personal perspectives that could be bracketed out during the research. Part of bracketing in this study was avoiding relating to the participants through personal experience during the qualitative data collection, and instead adopting an open and reflective mindset focused on the perspectives of students as they experienced robotics. This was important as the analysis of the qualitative data proceeded into the generation of textual and structural descriptions of participants' experiences with robotics, peer collaboration, problem-solving and computational thinking (Creswell & Poth, 2016).

The audio from the interviews was recorded digitally and uploaded to a cloud storage system as they were completed. Interviews were then transcribed into a word processing document for later analysis in ATLAS.ti. The audio files from the interviews were categorized by student name initially, but were converted to a number and letter code based on the robotics activity type (task-based or project-based). No names were used in the analysis or discussion of the results.

Initial analysis took place at the transcription stage by noticing key ideas and processes for problem-solving that occurred frequently in the transcripts as a whole, as well as the connections to the three dimensions of computational thinking (Brennan & Resnick, 2012). In further preparation for the initial coding, the transcripts were uploaded to the ATLAS.ti Cloud data analysis platform. Each transcript was reread, one by one, several times. Memoing was incorporated into the early stages of qualitative analysis, including the transcription phase, in order to "capture emerging thematic ideas" that assisted with the coding process (Creswell & Poth, 2016, p. 194). Initial codes were created based on concepts and patterns that emerged from the interview data, with a focus on problem-solving and collaboration in order to address research questions 2 and 3. Secondary codes resulting from analysis of the memoing from the interview data yielded "significant statements" related to the research questions which were then analyzed and grouped to form "units and themes" (Creswell & Poth, 2016, p. 200).

To address research question 2, the themes emerging from the interview data related to peer collaboration were grouped according to robotics activity type (task-based and project-based) and compared for similarities and differences. These included, but were not limited to, approaches and attitudes toward peer collaboration. For research question 3, the artifact-based interviews discussed above helped to facilitate the necessary rich descriptions of problem-solving strategies in robotics activities and computational thinking strategies, which provided depth and descriptions to the quantitative results.

Ethical Considerations

Ethical issues were addressed throughout the duration of this study. The researcher completed Human Subjects CITI training in 2016. An Institutional Review Board (IRB) application was filed upon successful defense of the research proposal. As all participants in this study were minors, the IRB application was filed with expedited status. Approval was also sought and granted from the school director since the research site was a private school. Relevant members of the school administration were made aware of the research and its timelines. Appendix H contains the IRB approval document.

The BCC task sets were labeled according to which group (task-based or project-based robotics) the students represented. No personally identifiable information was recorded or used in the data analysis. Interviews and transcripts of responses were assigned generic name identifiers (e.g., Student A) and another identifier indicating which group the students represented. Although the audio of the interviews was digitally recorded, no personally identifiable information was used or disseminated. In the results and discussion section of the final research report, pseudonyms were used as necessary when discussing or quoting the qualitative data.

Standard informed consent forms were distributed to parents of the participants upon IRB approval. Assent forms were distributed to students eligible to participate at the beginning of each robotics course. All forms were distributed in hard copy. These forms explained the nature of the study and the research, participants' rights, and confirmation that participants will remain anonymous when the research is disseminated to the professional community, researchers, and the university representatives involved. The forms included the option for participants to opt-out of any part of the research activity at any time.

Role of the Researcher and Addressing Biases

The researcher developed most of the curriculum for the robotics courses and taught robotics for five school years at the research site. However, with the qualitative portion of this study focused on the phenomenological approach, a balance was struck between positively utilizing the background of the researcher to drive the study and ensuring that the data collection reflected the essence of the experience from the perspective of the participants. The researcher's own experience with robotics instruction was described as part of the process of bracketing the perspective of the researcher in order to focus on participants' experiences. Although the experience and enthusiasm of the researcher for robotics may be viewed as potential bias, this bias was reflected in the facilitation of learning experiences for students that enabled them to take ownership of their learning and experiences in robotics, rather than influence or overshadow the students' own experiences. The interview format was semi-structured and the interview questions were open-ended, which allowed the participants' perspectives to drive the qualitative portion of this study.

CHAPTER FOUR: RESULTS

The purpose of this chapter is to present the results of the data analysis. This chapter is organized by the research questions and its contents reflect the mixed methodology used in this study. This chapter contains two main sections. The first section consists of a description of the quantitative analysis used to examine research question 1 and the corresponding results, including tables. The second section contains quotes from the interviews and a description of the qualitative analysis used to examine research questions 2 and 3. These research questions were discussed in the same section in this chapter because of the links between problem-solving and collaboration that arose during the qualitative data analysis for both groups of participants. This is discussed in more detail in the final chapter.

Research Question 1

To answer research question 1 "Is there a significant difference between the change of computational thinking skills of middle school students participating in task-based robotics versus project-based robotics?" data analysis of the quantitative data was performed in SPSS 25. Independent t-tests were performed on the post-treatment scores on the BCC task set as well as on the difference between pre- and post-treatment scores. The same procedures were performed on the sub scores for the four areas of computational thinking.

Table 4 indicates the means of the pre and post BCC task sets scores for both task-based robotics and project-based robotics. The maximum achievable score was 135.

Among all participants the mean of the pre BCC task set was 100.62 and the mean of post BCC task set was 98.63. Project-based participants scored higher than task-based participants on both pre and post BCC task sets. Overall the means of the post BCC task set were lower for both groups. The mean differences of the pre and post scores were -0.615 for project-based participants and -3.636 for task-based participants.

Table 4 Pre-post Means and Standard Deviations (SD)

	Group		Statistic	Std. Error
BCC PRE RAW	Project	Mean	103.85	3.733
		Std. Deviation	13.459	
	Task	Mean	96.82	6.458
		Std. Deviation	21.419	
BCC POST RAW	Project	Mean	103.23	3.774
		Std. Deviation	13.609	
	Task	Mean	93.18	5.922
		Std. Deviation	19.641	

When examining the means of the pre and post BCC sub scores for the four areas of computational thinking for both groups, only the mean score for algorithmic thinking increased. Mean scores for decomposition, abstraction, and pattern recognition decreased. There were no areas of computational thinking in which the mean scores increased for one group but decreased for the other. Both groups scored highest in decomposition in the pre BCC task set, and lowest in algorithmic thinking. In the post BCC task set, both groups scored highest in algorithmic thinking, with the lowest mean scores in abstraction for the task-based group and in pattern recognition for the project-based group. These results are in Table 5 and Figures 2 and 3 below.

Table 5 Sub Scores for Computational Thinking Areas

	Group	N	Mean	Std. Deviation	Std. Error Mean
Algorithms Pre %	Task	11	71.82	16.473	4.967
	Project	13	74.77	11.366	3.152
Algorithms Post %	Task	11	73.82	13.804	4.162
	Project	13	79.31	14.156	3.926
Abstraction Pre %	Task	11	73.45	15.958	4.812
	Project	13	79.69	12.385	3.435
Abstraction Post %	Task	11	68.18	12.600	3.799
	Project	13	76.77	9.816	2.723
Decomposition Pre %	Task	11	89.18	9.857	2.972
	Project	13	86.62	11.192	3.104
Decomposition Post %	Task	11	72.27	11.411	3.441
	Project	13	78.15	12.103	3.357
Patterns Pre %	Task	11	77.27	14.360	4.330
	Project	13	81.00	11.496	3.189
Patterns Post %	Task	11	68.73	15.994	4.822
	Project	13	74.77	10.918	3.028

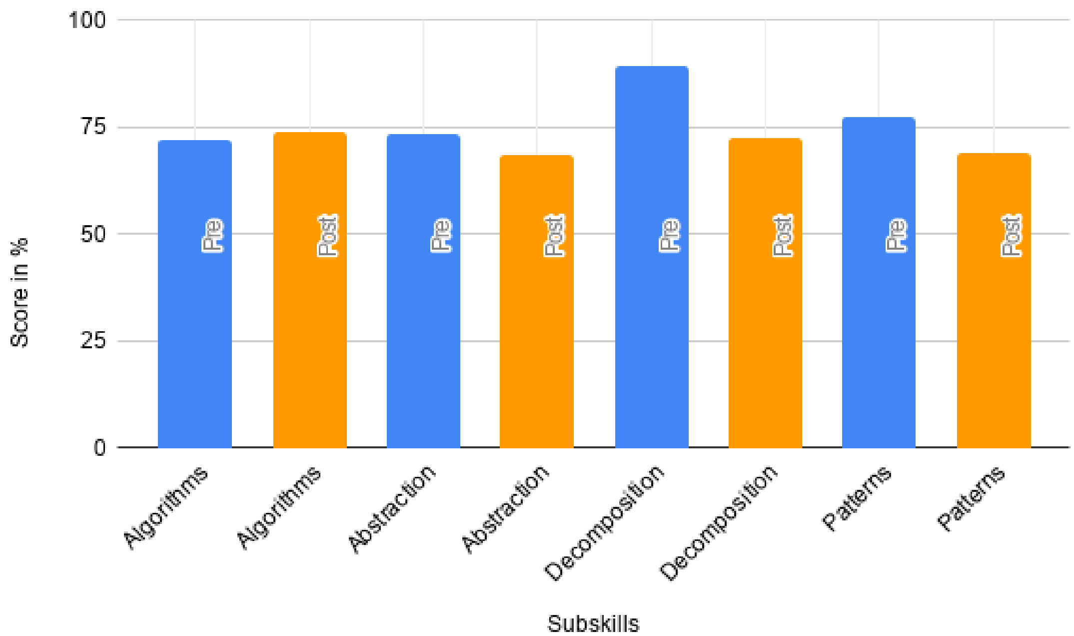


Figure 2 **Sub Scores on pre and post BCC Task Sets – Task-based group**

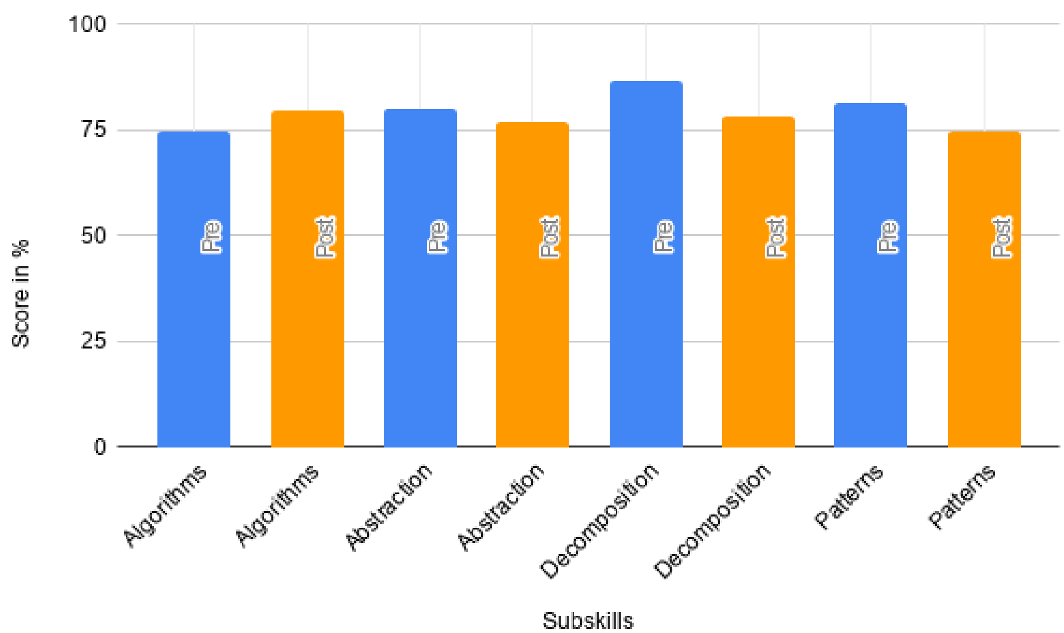


Figure 3 **Sub Scores on pre and post BCC Task Sets – Project-based group**

Histograms showed a reasonably normal distribution, with skewness screening revealing no significant skew. Kurtosis screening on the pre and post scores of both the project-based participants ($z = 0.887$ pre and $z = 1.832$ post) and task-based participants ($z = -1.477$ pre and $z = 0.518$) signified a normal distribution. The Shapiro-Wilk test showed $p > .05$ for both groups on both pre and post BCC task sets, indicating a normal distribution of scores. With a normal distribution confirmed, the next step was to conduct the independent t-tests.

The results of the independent t-tests are indicated in Table 6. The difference between pre and post BCC task set scores were calculated for each participant. No statistically significant difference between means were detected among task-based and project-based participants. Additionally, no significant differences were detected between the means of the sub scores. A further test on the mean scores of the pre- and post-tests also showed no significant difference.

Table 6 Independent Samples Tests for Computational Thinking Skills

		Levene's Test for Equality of Variances		t-test for Equality of Means		
		F	Sig.	t	Df	Sig. (2-tailed)
BCC Raw diff	Equal variances assumed	2.181	.154	-.454	22	.654
	Equal variances not assumed			-.435	15.465	.669
Algorithms diff	Equal variances assumed	3.087	.093	-.413	22	.684
	Equal variances not assumed			-.393	14.681	.700
Abstraction diff	Equal variances assumed	.669	.422	-.458	22	.651
	Equal variances not assumed			-.443	16.881	.663
Decomposition diff	Equal variances assumed	2.487	.129	-1.999	22	.058
	Equal variances not assumed			-1.939	17.383	.069
Patterns diff	Equal variances assumed	.363	.553	-.495	22	.626
	Equal variances not assumed			-.480	17.433	.637

Finally, one sample t-tests of the differences between pre and post BCC task set scores and sub scores were conducted to determine if any scores differed significantly from zero. The results of these tests can be found in Table 7. The results indicated that the differences in scores from pre to post in the computational thinking area of decomposition differed significantly from zero for both task-based and project-based participants. For project-based participants, the differences in scores from pre to post in the computational thinking area of pattern recognition also differed significantly from zero. In all other areas, scores were not significantly different from zero.

Table 7 One Sample Test Differing from 0 for Computational Thinking Skills

	Test Value = 0			
	T	df	Sig. (2-tailed)	Mean Difference
BCC Raw task diff	-.594	10	.566	-3.63636
BCC Raw project diff	-.188	12	.854	-.61538
Algorithms task diff	.345	10	.737	2.00000
Algorithms project diff	1.597	12	.136	4.53846
Abstraction task diff	-1.164	10	.271	-5.27273
Abstraction project diff	-1.058	12	.311	-2.92308
Decomposition task diff	-4.602	10	.001	-16.90909
Decomposition project diff	-3.614	12	.004	-8.46154
Patterns task diff	-2.104	10	.062	-8.54545
Patterns project diff	-2.397	12	.034	-6.23077

Research Questions 2 and 3

To answer research question 2 “How do middle school students collaborate with peers in task-based robotics versus project-based robotics?” and research question 3 “How do middle school students use various problem-solving strategies in robotics activities?”, qualitative data in the form of semi-structured interview responses were collected. These interviews took place around the midpoint of each robotics course and were predominately centered around a project (for project-based participants) or tasks (for task-based participants) that students had completed or nearly completed. Interviews took place with participants’ EV3-G program open on their laptop and their robot nearby.

Based on the results of the initial coding process, it became apparent that additional codes would need to be created, particularly for content involving computational thinking and programming. A single code for each of these areas was not sufficient for the breadth and depth of responses from the participants. Similarly, separate codes for collaboration within project and task-based groups were added to aid analysis due to the volume of responses in this area from participants in both groups. As these new codes were added, further rereads of the transcripts yielded additional examples from several participants. In several instances, different parts of participants' responses to a question were highlighted and assigned multiple codes involving collaboration, problem-solving, and computational thinking. In cases where participants described approaching problem-solving in general through the use of multiple computational thinking strategies, the code of CompThink.mindset was assigned. Through these iterations of reading and coding the transcripts, the themes discussed later in this chapter began to emerge. A list of codes and their frequencies can be found in Table 8.

Table 8 Codes and their Frequencies

Code	Frequency	
	Project-based	Task-based
ClassResources	10	4
Collaboration.project	33	0
Collaboration.task	0	19
CompThink.abstraction	6	8
CompThink.algorithmic	2	9
CompThink.decomposition	7	16
CompThink.mindset	14	16
ProblemSolving	25	27
Programming.conditionals	2	3
Programming.flow	5	6
Programming.functions	3	2

A few themes emerged from the coding process, which are listed in full in Table 9. Problem solving was at the heart of the discussions with participants during interviews. From a micro view of specific programming strategies to broader discussions of the phases of the design cycle, participants described the problem-solving processes and strategies they used in detail. Student K talked about a macro view of the problem-solving process in task-based robotics, describing how "*First of all, I do the initial challenge (task) and make sure that works. Then I do the super challenges (task extensions).*" More detailed descriptions of problem solving, including specific aspects of programming and computational thinking, were given by Students A, B, F, G, and K:

I used the different blocks, and these blocks (communication blocks) send messages to different parts of the program to be decoded. I had to put this in a loop and have a switch inside of it. (For example) If it gets in this part of the switch statement, the robot will move forward, and in this part it will go backwards. (Student A)

I'm always testing and double checking. Like here, I used a random speed, but that didn't work. So I had to change it. And I have to think about how the robot will function with this code. And if it's not working, I'll change it. (Student B)

First, I remember we had to calculate the number of rotations. Then we used this (points to sensor block) set to "angle", to check the turn. So 80 here means turns 80 degrees. When I tried 90 degrees it was a bit off, so I changed it to 80. Then I tested it, and it worked. (Student F)

Even if I changed something simple, like increasing the rotations, I would test that many times to make sure it was perfect. And the little tweaks that I couldn't do, I would just try that – make a tweak to get the ball into the cylinder. (Student G)

Well I knew it would have to turn one way and then the other. The loop thing – I knew that there would have to be a loop in the program, and maybe the switch – I thought maybe I wouldn't need to (have a switch) and just put one thing after the other, but it (would) have (to do) two things at the same time so I needed a switch. (Student K)

The descriptions provided by these students encompass the concepts and practices dimensions of computational thinking described by Brennan and Resnick (2012) and illustrate how decomposition and algorithmic thinking play a role in the approaches to problem-solving in both task-based and project-based robotics.

Descriptions of peer collaboration were present in some form in all of the interview transcripts. Collaborative problem-solving was evident in all interviews of both task-based and project-based robotics participants. Student D described collaborating in the initial planning stages of a project, commenting "*I like working with others during planning. It's interesting to know about the different designs people can come up with for their build and program.*" Student L talked about how collaboration was present in all phases of solving a task, pointing out that "*We would all kind of work together to come up with ideas. Then we would work together on the program – we would all be on our different computers and make our own program. (For example), we all test our programs and watch the tests, to see what each robot would do.*" Students A, I, K, F and G described other benefits of collaboration in robotics, including just having another set of eyes on the problem and getting a different perspective:

> *In the programming, it (working with others) just makes it a little bit easier because (you can both look at) one laptop.* (Student K)
>
> *Well, <student> and I, we both worked on this, she worked on one half and I worked on the other. Sometimes, I'd work on the program while she would work on building the robot, and then we would switch.* (Student I)
>
> *We kind of use each other for help – if I find this part and see it's working, but it's not working for another person, I'll help them with that. And they will do*

the same thing for you. We also collaborate and use each other's ideas. (Student F)

Especially in the planning, like if we have two different ideas, we can put them together to make something better. (Student L)

I wanted to try it, and I saw he built gears, so I did that too. Then I got the idea for a ramp from when people – sometimes my robot was going so fast, I needed to get under people to push them off. (Student A)

If I just can't figure something out, if I can't refer to anything, or I'm just not getting it, I ask for their help, I ask them if they have any ideas. (Student G)

The responses from Student L were examples of the natural and authentic collaboration that took place during the robotics courses. This was evident with both the task-based group and project-based group, with further examples below in Table 9. Collaboration within the task-based group often centered around the cycles of programming, testing, and debugging. While collaboration within the project-based group also involved programming, it extended to all aspects of the project including the planning stage and the building of the robot. The response from Student I above was one illustration of this, with others also in Table 9.

With the detailed approaches to problem-solving present within the interview transcripts, taken into account with the surrounding descriptions of collaboration, the three dimensions of computational thinking were well represented: concepts, practices, and perspectives (Brennan & Resnick, 2012). Given the frequency and the detail of the descriptions in these transcripts, these three dimensions were also used as themes. These themes and samples of their indicative quotes can be found in Table 9 below.

Table 9 Theme, Participants Contributing, and Indicative Quotes

Theme	Participants Contributing	Indicative Quotes
Course design to promote collaboration, student choice, and ownership of learning	Student B, Student C, Student D, Student E, Student G, Student I, Student J, Student K, Student L	"*First of all, I do the initial challenge (task) and make sure that works. Then I do the super challenges (task extensions).*" Student K, task-based "*When there is more than one option (of tasks to solve), that's a fun time to work together. Like for the gearing tasks, you can gear for power or for speed. So one person might work on speed and the other for power.*" Student L, task-based
Natural and authentic collaboration facilitating problem solving approaches	Student A, Student B, Student C, Student D, Student E, Student F, Student G, Student H, Student I, Student J, Student K	"*Also, I don't know if I intentionally did it, but when I first got the ramp on my robot, <another student> came up with an idea of an 'anti-ramp', which I think helped <student> with his project.*" Student A, project-based "*We would all kind of work together to come up with ideas. Then we would work together on the program – we would all be on our different computers and make our own program. (For example), we all test our programs and watch the tests, to see what each robot would do.*" Student L, task-based
Collaboration in task-based robotics focused on programming and	Student B, Student F, Student G, Student K	"*First, I remember we had to calculate the number of rotations. Then we used this (points to sensor block) set to*

computational thinking concepts		*"angle", to check the turn. So 80 here means turns 80 degrees. When I tried 90 degrees it was a bit off, so I changed it to 80. Then I tested it, and it worked.*" Student F, task-based "*In the programming, it (working with others) just makes it a little bit easier because (you can both look at) one laptop.*" Student K, task-based
Collaboration in project-based robotics focused on the project as whole – the design, the build, the program.	Student A, Student C, Student D, Student E, Student H, Student I, Student J	"*Well, <student> and I, we both worked on this, she worked on one half and I worked on the other. Sometimes, I'd work on the program while she would work on building the robot, and then we would switch.*" Student I, project-based "*I like working with others during planning. It's interesting to know about the different designs people can come up with for their build and program.*" Student D, project-based
Computational thinking concepts	Student A, Student B, Student C, Student D, Student E, Student F, Student G, Student H, Student I, Student J, Student K, Student L	"*Well I knew it would have to turn one way and then the other. The loop thing – I knew that there would have to be a loop in the program, and maybe the switch – I thought maybe I wouldn't need to (have a switch) and just put one thing after the other, but it (would) have (to do) two things at the same time so I needed a switch.*" Student K, task-based

		"*I used the different blocks, and these blocks (communication blocks) send messages to different parts of the program to be decoded. I had to put this in a loop and have a switch inside of it. (For example) If it gets in this part of the switch statement, the robot will move forward, and in this part it will go backwards.*" Student A, project-based
Computational thinking practices	Student A, Student B, Student C, Student D, Student E, Student F, Student G, Student H, Student I, Student J, Student K, Student L	"*Sometimes I go back and forth between the objective and the program. I'll go to the robotics table and take a look at what I need to, then I come back to the program and make sure my ideas are correct. Then I'll start programming.*" Student G, task-based "*We kind of use each other for help – if I find this part and see it's working, but it's not working for another person, I'll help them with that. And they will do the same thing for you. We also collaborate and use each other's ideas.*" Student F, task-based
Computational thinking perspectives	Student A, Student D, Student E, Student G, Student H, Student J	*When we were testing with other robots, we would see how their robot would react and try to program our robot to adapt to that.*" Student D, project-based "*You could build a delivery robot since the messages are*

	sent really fast. Like maybe you could have a Bluetooth delivery system, for a robot to bring packages somewhere. I heard that drones do this." Student A, project-based "*My classmates obviously have ideas as well, and I like to take feedback from them to improve my ideas as well.*" Student G, task-based

CHAPTER FIVE: DISCUSSION & CONCLUSION

The purpose of this study was to examine the effects that task-based and project-based robotics instruction has on computational thinking, collaboration, and problem-solving strategies in a robotics elective course. This first part of this chapter is organized by the research questions. The first research question was examined based on the quantitative analysis of the data from the BCC task sets given to participants at the beginning and end of the robotics courses. Descriptions of problem-solving and collaboration from the qualitative analysis of the interview data were also discussed and examined in the context of participant performance on the BCC task sets. The second and third research questions were examined and discussed based on the qualitative analysis of the interview data. The remaining parts of this chapter discuss the limitations of this study, as well as the potential impacts of this study and possible next steps.

Research Question 1

Research question 1 was "Is there a significant difference between the change of computational thinking skills of middle school students participating in task-based robotics versus project-based robotics?" Overall raw mean scores decreased between the pre and post course BCC task sets for both task-based and project-based participants. Out of a total of 135 points, the mean decrease was less than 1 point (0.615) for the project-based participants and less than 4 points (3.636) for the task-based participants. The change was not found to be significant from 0 for either group. A between groups t-test to compare the difference in the change of scores showed no significant difference between the task-based and project-based groups.

When examining specific computational thinking skill areas within the BCC task sets for both groups, mean scores for algorithmic thinking increased between pre and post BCC task sets and decreased for abstraction, pattern recognition, and decomposition. Between groups t-tests to compare the difference in the change of scores showed no significant difference between the task-based and project-based groups for all skill areas. The negative change was found to be significant from 0 for both groups in the area of decomposition, and for the project-based group in the area of pattern recognition. The change was not found to be significant in algorithmic thinking and abstraction for both groups.

The lack of significance in the results of the quantitative analysis needs to be discussed within the context of the instrument used for measuring computational thinking skills. The paper-and-pencil, timed nature of the BCC task sets may not have been connected well enough to the open-ended, collaborative nature of project-based and task-based robotics instruction where, as revealed in the qualitative analysis, the problem-solving focus is on careful planning and multiple iterations while learning from mistakes (Chiazzese et al., 2019; Román-González et al., 2019). The scoring system of the BCC task sets may also skew the results in a study involving pre and post administration. This system includes tasks with three levels of difficulty, with more points at stake for higher difficulty tasks, and maximum penalties (no points) applied for questions left blank due to running out of time (Bebras, n.d.). BCC tasks do not equally represent the four areas of computational thinking (Izu et al., 2017). As a result of this, a task that is not attempted can adversely affect scores in a particular area. A high difficulty task in the BCC post set, for example, that involved decomposition and pattern recognition was left blank by 7 participants. As this task was the second to last task in the task packet, it can be inferred that the reason it was left blank was due to running out of time.

Computational thinking strategies were discussed and described by participants in every interview conducted. Decomposition, or breaking down problems into smaller chunks, was a key strategy for both groups of participants. 100% of participants interviewed in the task-based group described decomposition in the context of how this was used to solve tasks, often giving specific examples from their robotics programming. Similarly, all but 1 of the participants interviewed in the project-based group described instances of decomposition. One key difference was that task-based participants talked about decomposition exclusively in the context of programming, while project-based participants described using decomposition in both the build and the programming of their projects. This application of computational thinking outside the context of programming is indicative of viewing computational thinking as dispositional and as a mindset for problem-solving (Brennan & Resnick, 2012; ISTE & CTSA, 2011). Task-based and project-based participants both described several aspects of the concepts and practices dimensions of computational thinking measured by the BCC task sets (Brennan & Resnick, 2012).

Algorithmic thinking and iterative design for problem-solving was evident within the interview data of both task-based and project-based participants. Student A, when discussing a situation where they got stuck in a project, said that "*I tried to look at the problem in a different way and make my program in a different way*." Student G, when describing how they approached solving a task through multiple iterative cycles, said "*For this challenge, I went to the robotics table and planned it out – like a mental mind map. I tested my program many times*." These examples illustrate the practices of decomposition, algorithmic thinking, pattern recognition, and abstraction within problem-solving approaches in a robotics classroom. Student K (task-based), describing their approach for breaking down a problem and developing an algorithm said "*I thought maybe I wouldn't need to (have a switch) and just put one thing after the other, but it (would) have (to do) two things at the same time so I needed a switch.*" While Student A (project-based), describing a similar approach in much different context, said "*I had to put this in a loop and have a switch inside of it. (For example) If it gets in this part of the switch statement, the robot will move forward, and in this part it will go backwards.*" Overall, the descriptions of the use and application of computational thinking strategies measured by the BCC tasks sets given by the participants were consistent across both groups despite the different contexts, which may connect to the lack of significant difference in the quantitative analysis.

Research Question 2

Research question 2 was "How do middle school students collaborate with peers in task-based robotics versus project-based robotics?" Collaboration in the robotics courses occurred throughout the study among both the project-based and task-based participants. Based on the interview data, collaboration was natural and authentic among both groups of participants. This was the case not only for the programming, but also for the planning, testing, and building phases of particular tasks and projects. Some differences in collaboration between task-based and project-based participants arose during the qualitative analysis. These are discussed in more detail below.

Collaboration was often spontaneous, arising as problems and obstacles arose. In the project-based group, collaboration would sometimes occur based on a common interest in a particular idea. Often these ideas were connected to a broad range of STEM domains, such as engineering specialized vehicles or robotic limbs, and collaboration during the research phase of the projects generated ideas on how to prototype and design using the robotics kit. This is consistent with findings that robotics is a driver of interest and motivation in STEM topics (Barak & Assal, 2018; Bers et al., 2014; Park, 2015). This motivation around a project idea would then carry over when problem-solving collaboratively during the programming and testing phases of the projects, as evidenced in the interview quotations shared in Chapter 4.

Recent literature has underscored the importance that the social aspect of robotics, including opportunities for collaborative problem-solving, have on learning motivation and creativity (Anwar et al., 2019; Barak & Assal, 2018; Ucgul & Cagiltay, 2014). In a review of educational robotics studies, Anwar et al. (2019) noted several studies that showed a connection between student creativity and the social aspect of robotics. A study involving both task-based and project-based robotics described how student motivation increased as students collaborated and tested solutions together (Barak & Assal, 2018). Ucgul and Cagiltay (2014) noted the opportunity robotics provides as a medium for students to practice and develop social skills, and observed collaboration occurring naturally within small groups and across project groups. These characteristics were also evident in this study, with several students describing in interviews how the programming and testing phase of tasks and projects led to collaboration.

While collaboration among the project-based participants occurred in the planning, building, programming, and testing phases, collaboration among task-based participants was primarily focused on developing and debugging their programs. The interview quotes presented in Chapter 4 indicated that collaboration in task-based robotics took the form of gathering feedback from peers on an existing program, or just talking through a particular algorithm while programming and testing. Participants described being able to incorporate this feedback into the iterative design process of their programs, carefully testing and revising at each step in the task. This is characteristic of high levels of motivation, as well as the social aspect of robotics facilitating creative problem-solving mentioned in recent literature (Anwar et al., 2019; Barak & Assal, 2018; Park, 2015).

Overall qualitative analysis indicated a strong connection between collaboration and computational thinking. Interview data indicated that collaboration in this study spanned all three dimensions of computational thinking (Brennan & Resnick, 2012). The data under the theme of computational perspectives showed a natural tendency to collaborate and explore with others as part of the experience of robotics. While task-based participants primarily collaborated on the development, debugging, and iterating their solutions to a task, project-based participants collaborated in all phases of project-based robotics, including the planning, design, and building stages. As shown in Table 9 above, in both groups of participants collaboration took place both in the moment as design problems arose, and as a planned activity due to shared interest in a particular project or task option. This supports findings that robotics is a naturally and authentically collaborative experience (Bers et al., 2014; Kafai et al., 2014; Nugent et al., 2010; Park, 2015). Research has indicated that the social aspect of robotics plays a role in student approaches to problem-solving and is an important part of robotics instruction for students (Anwar et al., 2019; Barak & Assal, 2018). This was evident within the work of both groups of participants in this study. Barak and Assal (2018) also found that task-based robotics focused on and facilitated robotics and computational thinking skill development including programming, while project-based robotics allowed students to use these skills in more authentic and open-ended context. This was also a characteristic of this study and is further discussed in the sections at the end of this chapter.

Research Question 3

Research question 3 was "How do middle school students use various problem-solving strategies in robotics activities?" All interview participants discussed problem-solving strategies in their interviews, with nearly all having their program open and their robot next to them to demonstrate specific examples. This helped ensure that the data collected from interviews represented the experience of problem-solving in robotics from the student participants' perspectives. Problem-solving in the robotics courses during the study was iterative, involved natural collaboration, and included several examples of solving problems through a mindset of computational thinking. The importance of the social aspect of robotics has been well documented (Anwar et al., 2019; Barak & Assal, 2018; Bers et al., 2014; Kafai et al., 2014; Park, 2015). As the interview excerpts from Chapter 4 show, collaborative problem-solving took the form of exchanging ideas about a project, getting another perspective on a program, talking through an algorithm, seeking out feedback, or adapting a design or program based on observations of others.

Iteration was a key strategy for students when programming. The practice of iteration in the robotics courses for both groups encompassed planning, programming, testing, and debugging. Students in task-based robotics described programming steps of a solution based on their plans, testing it at the robotics table, and making specific changes to an individual programming block or loop sequence before testing again. This narrow, incremental approach which focused on specific areas of the program enabled many students to solve initial tasks quickly, with occasional changes made to their plan. Several interview quotes described instances of very focused decomposition centered on a specific part of a task, which provided opportunities for participants to practice computational thinking concepts and practices in clear and straightforward contexts. This is something that task-based robotics can facilitate for students (Barak & Assal, 2018; Rahman & Kapila, 2017; Sullivan & Bers, 2016).

However, task-based participants also experienced a broader iterative approach to problem-solving through the task extensions described in Table 1. These extensions often required students to iteratively cycle through multiple potential solutions requiring testing and revise their plans in addition to their program. Interview data along with informal observations throughout the task-based robotics courses indicated high levels of motivation and student self-ownership of learning while problem-solving, which is consistent with existing literature (Anwar et al., 2019; Barak & Assal, 2018; Bers et al., 2014; Kafai et al., 2014; Park, 2015). In terms of the three dimensions of computational thinking described by Brennan and Resnick (2012), students in task-based robotics were heavily engaged in the concepts and practices dimensions.

For students in the project-based group, iterative problem-solving often extended further into the design and build of the robot, with changes in the program sometimes resulting in changes to the build and the plan as a whole. The open-ended and student driven nature of project-based robotics was reflected in the cycles of problem-solving that students worked through. Interview data indicated a focus on the process of project design and development as a whole, rather than incrementally working through a program to solve a task. This included descriptions of decomposition and abstraction in more complex contexts that were applied to problem-solving in general as opposed to a specific part of a given task as in task-based robotics. Other studies incorporating project-based robotics have reported related findings (Atmatzidou & Demetriadis, 2016; Barak & Assal, 2018; Bers, 2007; Chen et al., 2017).

Although problem-solving specific to programming was a key part of the experience for project-based participants, the programming was driven by the project design and the build of the robot rather than focused on solving a given task. This broader learning context in which students expressed their own ideas and perspectives into the project design is reflective of the perspectives dimension of computational thinking (Brennan & Resnick, 2012). Existing literature has mentioned how project-based robotics drives motivation through its authenticity and connections to other subject areas and student passions – providing opportune settings for practicing computational thinking skills in authentic contexts (Barak & Assal, 2018; Barak & Zadok, 2009; Lye & Koh, 2014; Ucgul & Cagiltay, 2014; Yuen et al., 2014).

Limitations

Sample size was limited due to the circumstances and logistics of the study context. Quantitative data was collected from 24 participants and 12 participants were interviewed. The overall population size of the middle school of TIS means that all classes are small, including elective classes. TIS electives run on a trimester system, with new groups of students each trimester. The study took place from November 2018 to May 2019. IRB approval was received in September 2018, by which time it was too late to collect data on the students in robotics courses in the first trimester. The researcher had no access to the site after May 2019 due to starting a new job in another country, which eliminated the opportunity to recruit more participants for the following school year.

The BCC task sets as an instrument for measuring computational thinking skills may not align well with the type of learning that takes place in a robotics classroom. As shown in the discussion of the results, robotics is highly collaborative and problem-solving within robotics is iterative and open-ended in nature. The BCC task sets were completed individually with a set time limit, without the possibility to iteratively develop solutions through learning from mistakes. Robotics is a hands-on and tangible medium for computational thinking practice, while the paper-and-pencil BCC task sets are not. This may have affected both motivation and confidence among the participants when completing the BCC task sets.

For the qualitative interviews, bracketing was another challenge in the context of this study. A balance needed to be struck between the experience of the researcher as a robotics teacher and the perspectives of the students in a robotics course. The fact that the researcher was the robotics instructor for all of the participants may have contributed to what Lincoln and Guba (1985) described as "prolonged engagement" by establishing trust and immediate familiarity with the context of the study (p. 301). Although pure bracketing was not realistic or desired in this study, the researcher was conscious of interjecting personal experience and bias into the study (Creswell & Poth, 2016).

Another concern and possible limitation of this study was the gender diversity of the participants. In past years at the research site, the participants in the robotics electives courses (and in STEM-related courses and clubs in general) have been overwhelmingly male, despite efforts to promote the courses among female students. Of the 5 female participants in this study, 3 were interviewed. A plan was put in place to interview females at the research site who had taken robotics in prior years in hopes of getting a more gender-balanced perspective, but timing and other logistical constraints prevented this from happening.

Middle school students, due to normal developmental factors, represented a concern for this study. Middle school students are at various stages of personal and social development and are prone to extreme swings in emotions and mood. The nature and aims of this study necessitated that participants often be asked questions that were broad and open in order to generate the breadth and depth of responses necessary to support the goals of the study. Therefore it was important to establish rapport and an environment of open and honest discussion in order for middle school students to articulate responses to the degree that would benefit this study.

Potential Impacts

The quantitative data analysis showed no significant differences in the change of computational thinking skills between participants in task-based robotics instruction and participants in project-based robotics instruction. In addition, the qualitative data analysis indicated connections between peer collaboration and computational thinking strategies among both groups of participants. Task-based participants collaborated primarily in the programming phase of solving tasks, while collaboration in project-based robotics extended to all phases of a project, including building and planning. Problem-solving through collaboration and the use of computational thinking strategies was evident among both groups of participants.

Qualitative data analysis indicated that the student self-accessible class resources incorporating student choice (a wide range of topics and examples for projects, and multiple options for solving tasks) provided by the teacher were important for student planning, idea generation, and as a starting point for problem-solving. Particularly in the project-based group, participants discovered common interests upon investigating the class resources, which promoted authentic collaboration. This allowed participants to explore the perspectives dimension of computational thinking in which connecting with others, expressing themselves through design, and questioning where and how technology integrates with the world are key components (Brennan & Resnick, 2012). Examples of the collation and organization of class resources can be found in Appendix C, and starting points for gathering resources can be found in Appendix F. Collaboration in the task-based group often developed authentically when testing programs at the robotics tables, with students sharing ideas and giving feedback for next steps. The qualitative data indicated that these experiences fueled student ownership of their learning, with the teacher facilitating as necessary. This information could aid schools in building a robotics program into the school curriculum.

This study could be a resource for researchers interested in the role robotics could play as a subject within the larger school curriculum. As the scope of this study was limited to two out of three trimesters in a single school year, a next step could be gathering data from a larger pool of participants over a longer period of time and comparing the results. Additionally, more time would allow more interviews to be conducted for a wider range of perspectives on peer collaboration and problem-solving within robotics. The development of a hands-on and tangible instrument for measuring computational thinking skills that is more connected to robotics than the BCC task sets could also assist further research in the area of robotics and computational thinking. The data suggests that, provided robotics courses are purposefully planned and facilitated carefully, robotics classrooms can be places where students engage with and practice computational thinking skills through solving tasks or working through projects.

For practitioners who are developing robotics curriculum, teaching introductory programming through a task-based approach may keep the focus on the programming while developing computational thinking skills in what Brennan and Resnick (2012) categorize as the concepts and practices dimensions. A project-based approach may be recommended when students already have a grasp on programming fundamentals and robot building. This context has the potential to allow students to apply their programming skills in broader and more authentic contexts within the perspectives dimension, with the building and the programming phases of robotics driving each other. In both task-based and project-based approaches, the social aspect of robotics is an integral part of the robotics experience, and collaborative problem-solving will often occur naturally within the programming and testing phases of robotics (Anwar et al., 2019; Barak & Assal, 2018; Bers et al., 2014; Kafai et al., 2014; Nugent et al., 2010; Park, 2015).

Finally, a closer look at the roles and practices of collaboration and problem-solving within a robotics classroom may be warranted. At the TIS, robotics has also been incorporated into middle school Science and Math courses as assessment options for students to demonstrate their learning. No formal research has been conducted by the researcher on robotics in this context, but the collaboration and approaches to problem-solving in the qualitative data in this study have also been observed in these contexts by the researcher. Future research might focus on whether skills developed in robotics courses, such as approaching problem-solving through computational thinking concepts and practices, could transfer over to STEM and other subjects across the curriculum.

Conclusion

TIS has the resources to offer robotics electives courses as part of a middle school electives program encompassing many areas of STEM. These elective courses are based on inquiry approaches that align with the school curriculum. This study examined two instructional approaches for robotics in this context – task-based and project-based. As robotics kits for K-8 become more widespread and cost effective, and as STEM initiatives continue to expand across schools, robotics may gradually shift from informal settings such as camps to formal instructional settings within the school environment (Kafai et al., 2014; Park, 2015). In this situation, it is important to have a plan for facilitating robotics courses that incorporates a long term approach that facilitates connections to STEM contexts and students' interests and passions. The robotics courses described in this study along with their resources contained in the appendices can provide a starting point for schools wishing to move from informal to formal robotics learning environments within K-8 schools.

This study can potentially serve as a window into how robotics courses can be facilitated and the role that collaboration plays in problem-solving within the three dimensions of computational thinking (Brennan & Resnick, 2012). Both the task-based and project-based robotics courses in this study facilitated participant practice in core computational thinking skills within the concepts and practices dimensions through naturally collaborative problem-solving (Brennan & Resnick, 2012). The project-based courses offered participants a broader and more creative context to apply and practice computational thinking skills, including those in the perspectives dimension (Brennan & Resnick, 2012). By adopting a facilitator role in the classroom while offering relevant class resources for student self-access, robotics instructors can maximize opportunities for collaborative problem-solving to occur. The appendices below offer examples of class resources, tasks and projects, course outlines, and other resources for instructors.

REFERENCES

Alimisis, D. (2013). Educational Robotics: new challenges and trends. *Themes in Science and Technology Education*, 6(1), 63-71.

Altin, H., & Pedaste, M. (2013). Learning approaches to applying robotics in science education. *Journal of Baltic Science Education*, *12*(3), 365–377.

Anwar, S., Bascou, N. A., Menekse, M., & Kardgar, A. (2019). A systematic review of studies on educational robotics. *Journal of Pre-College Engineering Education Research*. Purdue University Press. https://doi.org/10.7771/2157-9288.1223

Archibald, M. M., Radil, A. I., Zhang, X., & Hanson, W. E. (2015). Current mixed methods practices in qualitative research: A content analysis of leading journals. *International Journal of Qualitative Methods*, *14*(2), 5–33. http://ejournals.library.ualberta.ca/index.php/IJQM/article/view/23006

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, *75*, 661–670. https://doi.org/10.1016/j.robot.2015.10.008

Barak, M., & Assal, M. (2018). Robotics and STEM learning: students' achievements in assignments according to the P3 Task Taxonomy—practice, problem solving, and projects. *International Journal of Technology & Design Education*, *28*(1), 121–144. https://doi-org.libproxy.boisestate.edu/10.1007/s10798-016-9385-9

Barak, M., & Zadok, Y. (2009). Robotics projects and learning concepts in science, technology and problem solving. *International Journal of Technology and Design Education*, *19*(3), 289-307.

Bebras. (n.d.). *Structure of a challenge*. https://www.bebras.org/?q=structure

Bers, M. U. (2007). Project InterActions: A multigenerational robotic learning environment. *Journal of Science Education and Technology*, *16*(6), 537–552. https://doi.org/10.1007/s10956-007-9074-2

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145-157.

Bers, M. U., Ponte, I., Juelich, K., Viera, A., & Schenker, J. (2002). Teachers as designers: Integrating robotics in early childhood education. *Information Technology in Childhood Education*, 123–145.

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American Educational Research Association (pp. 1-25). Vancouver, Canada.

Carbonaro, M., Rex, M., & Chambers, J. (2004). Using LEGO robotics in a project-based learning environment. *The Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, *6*(1).

Carnegie Mellon University. (2016). Carnegie Mellon Robotics Academy. https://www.cmu.edu/roboticsacademy/index.html

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers and Education*, *109*. https://doi.org/10.1016/j.compedu.2017.03.001

Ching, Y. -H., Hsu, Y. -C., & Baldwin, S. (2018). Developing computational thinking with educational technologies for young learners. *TechTrends, 62*(6), 563-573. doi.org/10.1007/s11528-018-0292-7

Chiazzese, G., Arrigo, M., Chifari, A., Lonati, V., & Tosto, C. (2019, December). Educational Robotics in Primary School: Measuring the Development of Computational Thinking Skills with the Bebras Tasks. In *Informatics* (Vol. 6, No. 4, p. 43). Multidisciplinary Digital Publishing Institute.

Creswell, J. W. (2014). *Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research* (5th ed.). Boston, MA: Pearson.

Creswell, J. W., & Plano Clark, V. L. (2011). Choosing a mixed methods design. In *Designing and Conducting Mixed Methods Research* (2nd ed., pp. 53-106). Thousand Oaks, CA: SAGE Publications, Inc.

Creswell, J. W., & Plano Clark, V. L. (2017). *Designing and Conducting Mixed Methods Research* (3rd ed.). Thousand Oaks, CA: SAGE Publications, Inc.

Creswell, J. W., & Poth, C. N. (2016). *Qualitative inquiry and research design: Choosing among five approaches* (4th ed.). Thousand Oaks, CA: SAGE Publications, Inc.

Dagiene, V., & Sentance, S. (2016). Informatics in Schools: Improvement of Informatics Knowledge and Perception, *9973*(October 2017). https://doi.org/10.1007/978-3-319-46747-4

Dagiene, V., & Stupuriene, G. (2016). Bebras - A sustainable community building model for the concept based learning of informatics and computational thinking. *Informatics in Education*, *15*(3), 25–44. https://doi.org/10.15388/infedu.2016.02

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, *150*, 103832.

Delal, H., & Oner, D. (2020). Developing middle school students' computational thinking skills using unplugged computing activities. *Informatics in Education*, *19*(1), 1-13.

Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education (Routledge)*, *46*(3), 277-296.

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education, 63*, 87-97.

Gonzalez, J. J. (2013). My journey with inquiry-based learning. *Journal on Excellence in College Teaching*, *24*(2), 33-50.

Gujberova, M., & Kalas, I. (2013). Designing productive gradations of tasks in primary programming education. *Proceedings of the 8th Workshop in Primary and Secondary Computing Education on - WiPSE '13*, 108–117. https://doi.org/10.1145/2532748.2532750

Huang, K. H., Yang, T. M., & Cheng, C. C. (2013). Engineering to see and move: Teaching computer programming with flowcharts vs. LEGO robots. *International Journal of Emerging Technologies in Learning*, *8*(4), 23-26.

ISTE & CSTA (2011). Operational definition of computational thinking for K-12 education. http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf

Izu, C., Mirolo, C., Settle, A., Mannila, L., & Stupuriene, G. (2017). Exploring bebras tasks content and performance: A multinational study. *Informatics in Education*, *16*(1), 39–59. https://doi.org/10.15388/infedu.2017.03

Johnson, R. B., Onwuegbuzie, A. J., & Turner, L. A. (2007). Toward a definition of mixed methods research. *Journal of Mixed Methods Research, 1*(2), 112-133. doi: 10.1177/1558689806298224

Kafai, Y. B., Burke, Q., & Resnick, M. (2014). *Connected code: Why children need to learn programming*. Cambridge, MA: MIT Press.

Kopcha, T. J., McGregor, J., Shin, S., Qian, Y., Choi, J., Hill, R., Mativo, J., & Choi, I. (2017). Developing an integrative STEM curriculum for robotics education through educational design research. *Journal of Formative Design in Learning*, 31–44. https://doi.org/10.1007/s41686-017-0005-1

Lee, K. T. H., Sullivan, A., & Bers, M. U. (2013). Collaboration by design: Using robotics to foster social interaction in kindergarten. *Computers in the Schools*, *30*(3), 271–281. https://doi.org/10.1080/07380569.2013.805676

Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T., & Almughyirah, S. (2016). Using robotics and game design to enhance children's self-efficacy, STEM attitudes, and computational thinking skills. *Journal of Science Education and Technology*, *25*(6). https://doi.org/10.1007/s10956-016-9628-2

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. Beverly Hills, CA: Sage Publications.

Lonati, V., Malchiodi, D., Monga, M., & Morpurgo, A. (2017). Bebras as a Teaching Resource. *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE '17*, 366–366. https://doi.org/10.1145/3059009.3072987

Lye, S. Y., & Koh, J. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41,* 51-61.

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational Thinking in K-9 Education. *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference - ITiCSE-WGR '14*, 1–29. https://doi.org/10.1145/2713609.2713610

Moorhead, M., Listman, J. B., & Kapila, V. (2015). A robotics-focused instructional framework for design-based research in middle school classrooms. *ASEE Annual Conference and Exposition, Conference Proceedings*, *122nd ASEE*(122nd ASEE Annual Conference and Exposition: Making Value for Society). https://doi.org/10.18260/p.23444

Nemiro, J., Larriva, C., & Jawaharlal, M. (2017). Developing creative behavior in elementary school students with robotics. *Journal of Creative Behavior*, *51*(1), 70–90. https://doi.org/10.1002/jocb.87

Nugent, G., Barker, B., Grandgenett, N., & Adamchuk, V. I. (2010). Impact of robotics and geospatial technology interventions on youth STEM learning and attitudes. *Journal of Research on Technology in Education*, *42*(4), 391-408.

Okita, S. Y. (2014). The relative merits of transparency: Investigating situations that support the use of robotics in developing student learning adaptability across virtual and physical computing platforms. *British Journal of Educational Technology*, *45*(5), 844-862. doi:10.1111/bjet.12101

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. New York, NY: Basic Books, Inc.

Park, J. (2015). Effect of robotics-enhanced inquiry-based learning in elementary science education. *Journal of Computers in Mathematics & Science Teaching*, *34*(1), 71-95.

Petre, M., & Price, B. (2004). Using robotics to motivate ‘back door’ learning. *Education and Information Technologies*, *9*(2), 147-158.

Rahman, S. M. M., & Kapila, V. (2017). A systems approach to analyzing design-based research in robotics-focused middle school STEM lessons through cognitive apprenticeship. *ASEE Annual Conference and Exposition, Conference Proceedings*, *2017–June*.

Román-González, M., Moreno-León, J., & Robles, G. (2017, July). Complementary tools for computational thinking assessment. In *Proceedings of International Conference on Computational Thinking Education (CTE 2017), S. C Kong, J Sheldon, and K. Y Li (Eds.). The Education University of Hong Kong* (pp. 154-159).

Román-González, M., Moreno-León, J., & Robles, G. (2019). Combining assessment tools for a comprehensive evaluation of computational thinking interventions. In *Computational thinking education* (pp. 79-98). Springer, Singapore.

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior*, *72*, 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Slangen, L., van Keulen, H., & Gravemeijer, K. (2011). What pupils can learn from working with robotic direct manipulation environments. *International Journal of Technology and Design Education*, *21*(4), 449-469.

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, *26*(1), 3–20. https://doi.org/10.1007/s10798-015-9304-5

Ucgul, M., & Cagiltay, K. (2014). Design and development issues for educational robotics training camps. *International Journal of Technology & Design Education*, *24*(2), 203-222. doi:10.1007/s10798-013-9253-9

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A – Mathematical Physical and Engineering Sciences, 366*(1881), 3717–3725.

Yuen, T. T., Boecking, M., Tiger, E. P., Gomez, A., Guillen, A., Arreguin, A., & Stone, J. (2014). Group tasks, activities, dynamics, and interactions in collaborative robotics projects with elementary and middle school children. *Journal of STEM Education: Innovations & Research*, *15*(1), 39-45.

APPENDIX A

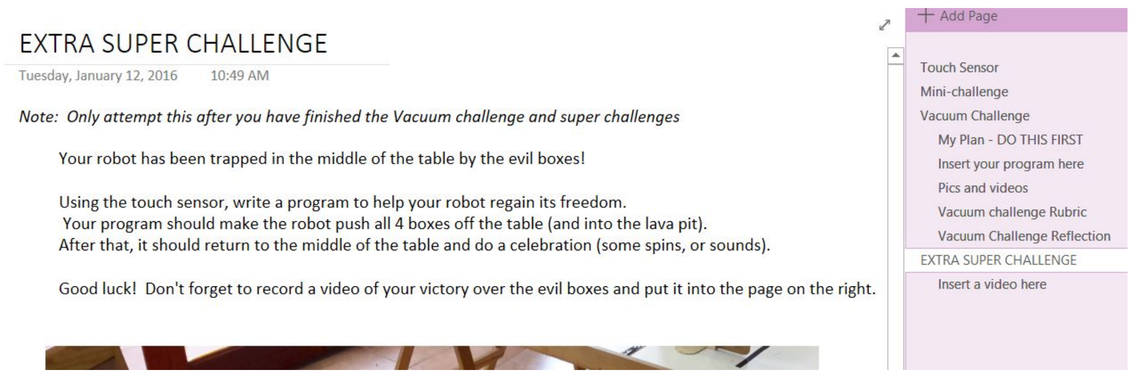
Interview Starter Questions and Themes of Focus

Some questions were adapted from Brennan and Resnick (2012)

- Project/task approaches:
  - How did you get your idea for the project/task solution?
  - How do you make a start on your work?
  - What is your strategy when you get stuck?
  - How do you know you are on the right track?
  - How do you know when you are finished?
- Collaboration:
  - How have you utilized classmates in this course?
  - How do classmates help you learn? Do you help other classmates?
  - When do you prefer to work with others in robotics? (All the time? When planning? When programming? Debugging? Building?)
- Programming & Computational Thinking (concepts, practices, and perspectives)
  - What was important for you to know when making this program?
  - Describe how you developed this program.
  - What issues did you encounter? How did you solve them?
  - How did you get your program working perfectly?
  - In what ways did your ideas for the program change as you developed it?
  - How could this program (or parts of it) be used to make the robot do something else? What other kinds of robotics projects could use programs like this one?
  - What was surprising about developing this program?
  - What ideas for this program did you get from working with others?

APPENDIX B

Examples of Robotics Tasks



EXTRA SUPER CHALLENGE

Tuesday, January 12, 2016 10:49 AM

Note: Only attempt this after you have finished the Vacuum challenge and super challenges

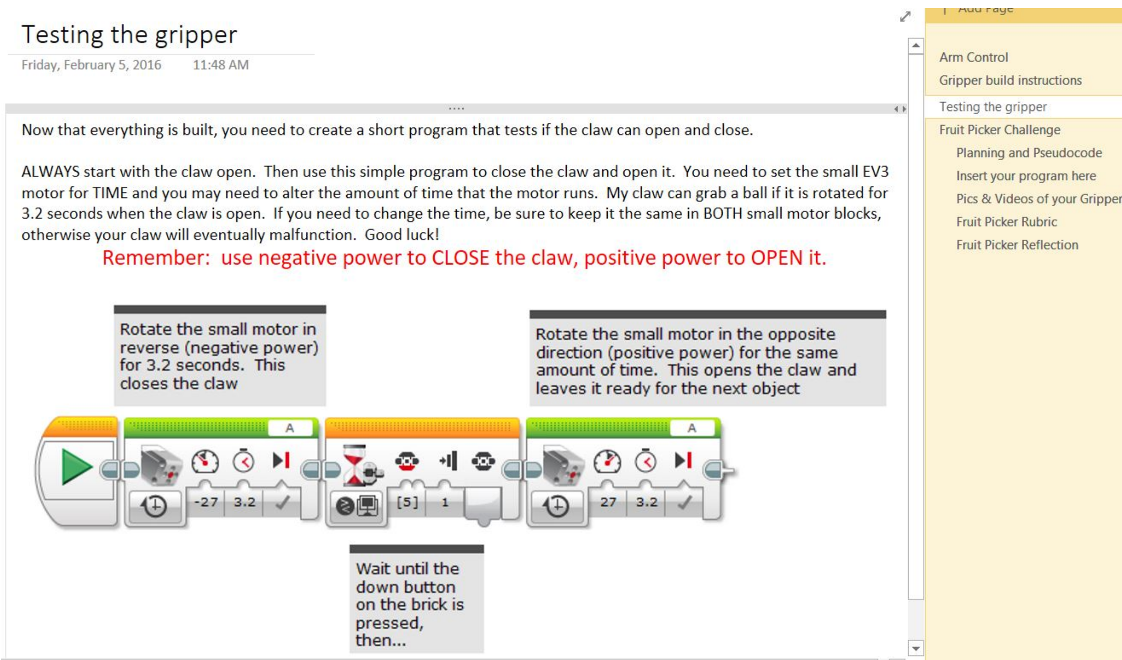
Your robot has been trapped in the middle of the table by the evil boxes!

Using the touch sensor, write a program to help your robot regain its freedom.
Your program should make the robot push all 4 boxes off the table (and into the lava pit).
After that, it should return to the middle of the table and do a celebration (some spins, or sounds).

Good luck! Don't forget to record a video of your victory over the evil boxes and put it into the page on the right.

+ Add Page
Touch Sensor
Mini-challenge
Vacuum Challenge
My Plan - DO THIS FIRST
Insert your program here
Pics and videos
Vacuum challenge Rubric
Vacuum Challenge Reflection
EXTRA SUPER CHALLENGE
Insert a video here

Task: Use the touch sensor to push four boxes off the table.



Testing the gripper

Friday, February 5, 2016 11:48 AM

Now that everything is built, you need to create a short program that tests if the claw can open and close.

ALWAYS start with the claw open. Then use this simple program to close the claw and open it. You need to set the small EV3 motor for TIME and you may need to alter the amount of time that the motor runs. My claw can grab a ball if it is rotated for 3.2 seconds when the claw is open. If you need to change the time, be sure to keep it the same in BOTH small motor blocks, otherwise your claw will eventually malfunction. Good luck!

Remember: use negative power to CLOSE the claw, positive power to OPEN it.

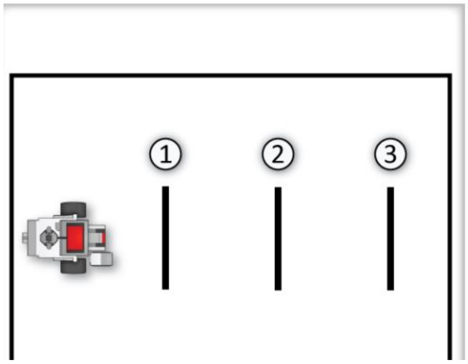
Arm Control
Gripper build instructions
Testing the gripper
Fruit Picker Challenge
Planning and Pseudocode
Insert your program here
Pics & Videos of your Gripper
Fruit Picker Rubric
Fruit Picker Reflection

Task: Program a claw attachment to pick up and move a ball

CHAPTER 6: Traffic Signal Challenge

In this challenge, you will program your EV3 robot to through three different intersection, each of which has a traffic signal. The traffic signal, which can be either the colored block or the red/green card, is held by hand at a set height. Unlike a camera, the detection range of the Color Sensor is short, so you will need to modify its placement on the robot so that it can see the traffic signal and react appropiately.

Rules and Procedures:



- Traffic signals are represented by holding the colored block with the signal side downward and toward the robot at a 45-degree angle.
- You can also use red and green colored sheets of paper, which may be easier to hold at the correct height.
- The initial signal color for each intersection is determined by flipping three coins before the run. Heads = Green, Tails = Red.
- The robot MUST stop if the light is red, and MUST NOT stop if the light is green.
- After the robot successfully gotten through all three intersections, the robot can be stopped by hand.

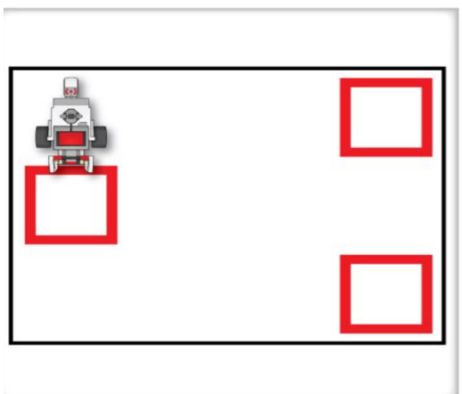
+ Add Page

The EV3 Color Sensor
Building
Mini-challenges
Programming Tutorial Video
Traffic challenge
Insert your program here
Pics & Videos of the challeng
Rubric
Reflection
Superchallenges!
Untitled page

Task: Using the color sensor, navigate through "traffic lights" on the table

In this challenge, you will program your EV3 robot to erase or clear the entire gameboard of either markings or parts. The robot is able to move freely in straight lines, using any method you want. However, there are three mud zones marked on the game board. When turning in one of these areas, the robot must be picked up by hand, and placed back down.

Rules and Procedures:



- The robot must clear the board of marks or parts to complete its mission.
- If any part of the robot is in the mud zone (red squares) at any point during a turn, it must be picked up and placed back down, as close to the same spot and facing as possible.
- Use overlapping paths to compensate for sen-sor inaccuracies

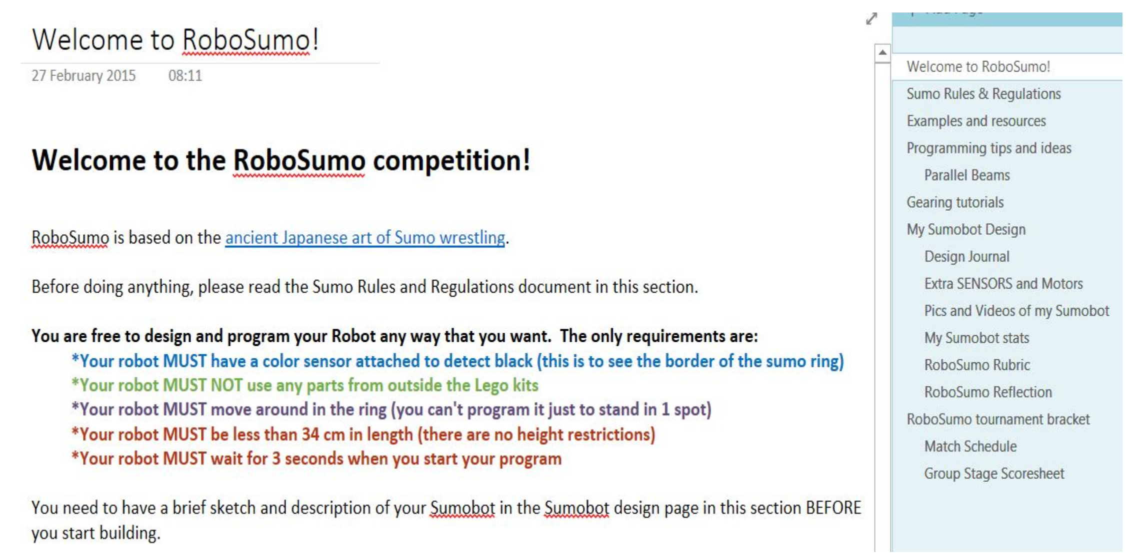
+ Add Page

Gyro Sensor
Mower Challenge
Insert program here
Pics & Videos of the challenge
Rubric
Reflection
Superchallenges

Task: Program the gyro sensor to "mow" the grass by precision turning.

APPENDIX C

Examples of Robotics Projects



Welcome to RoboSumo!

27 February 2015 08:11

Welcome to the RoboSumo competition!

RoboSumo is based on the ancient Japanese art of Sumo wrestling.

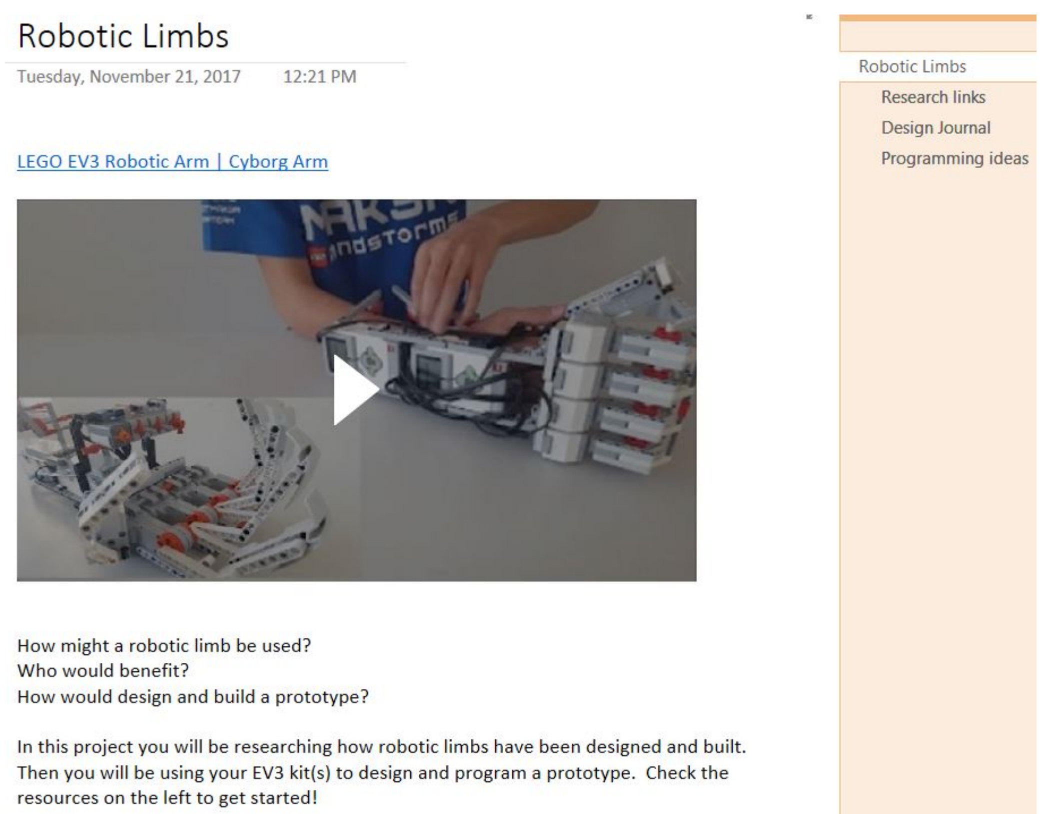
Before doing anything, please read the Sumo Rules and Regulations document in this section.

You are free to design and program your Robot any way that you want. The only requirements are:

***Your robot MUST have a color sensor attached to detect black (this is to see the border of the sumo ring)**
***Your robot MUST NOT use any parts from outside the Lego kits**
***Your robot MUST move around in the ring (you can't program it just to stand in 1 spot)**
***Your robot MUST be less than 34 cm in length (there are no height restrictions)**
***Your robot MUST wait for 3 seconds when you start your program**

You need to have a brief sketch and description of your Sumobot in the Sumobot design page in this section BEFORE you start building.

Project: Students create and program a "sumo" robot to push other robots out of a "ring"



Robotic Limbs

Tuesday, November 21, 2017 12:21 PM

LEGO EV3 Robotic Arm | Cyborg Arm

How might a robotic limb be used?
Who would benefit?
How would design and build a prototype?

In this project you will be researching how robotic limbs have been designed and built. Then you will be using your EV3 kit(s) to design and program a prototype. Check the resources on the left to get started!

Project: Research, plan, design, build, and program a robotic limb prototype.

Bluetooth controller

Tuesday, July 21, 2020 2:05 PM

Did you ever have a remote control car? How do you think it worked?
What other robots are controlled by remotes?

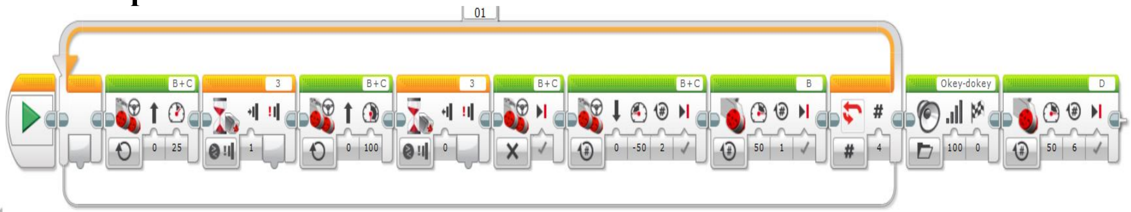
In this project you can create a robot that is controlled by another EV3 brick.

Project: Remote-controlled robot using Bluetooth

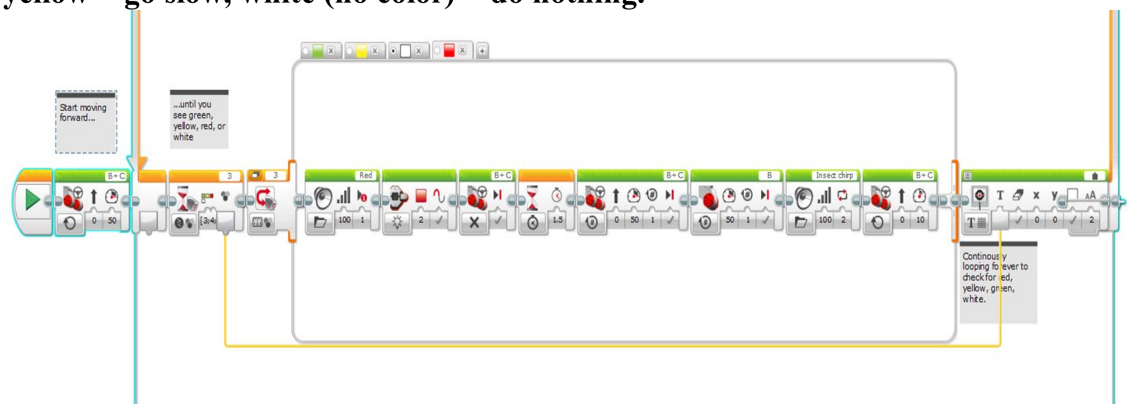
APPENDIX D

Programs

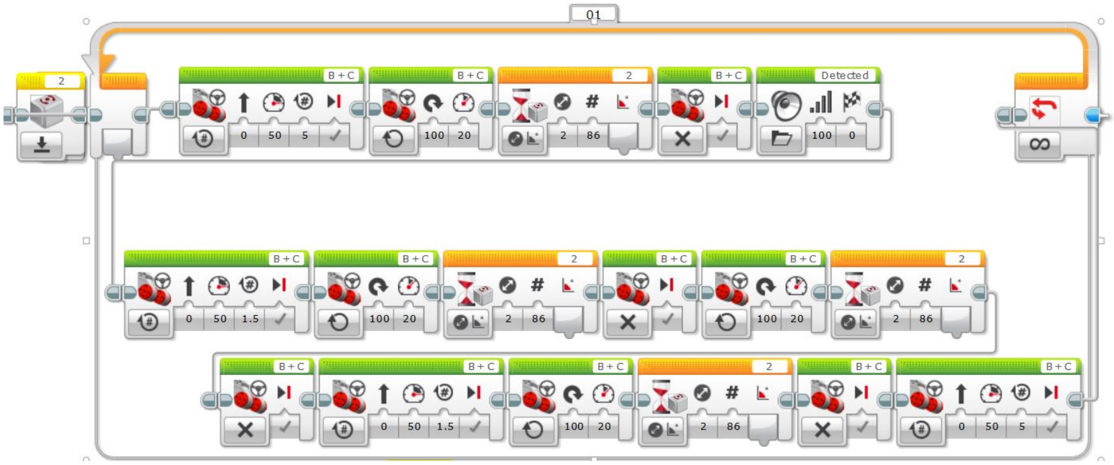
This program uses an algorithm within a loop that checks the state of the touch sensor to push boxes off of a table.



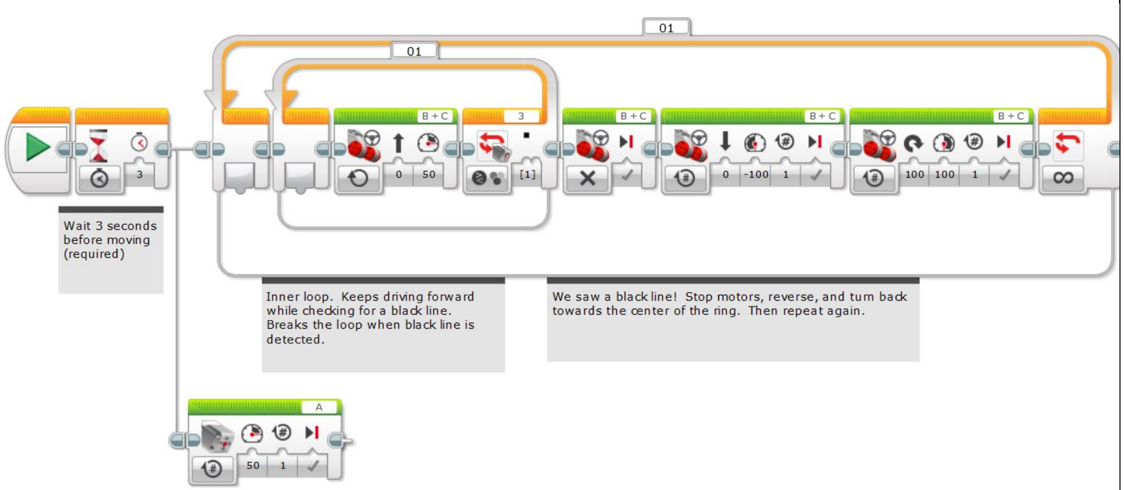
This program uses an algorithm that checks what color the sensor reads and performs different actions based on that color. Red = stop, green = go full speed, yellow = go slow, white (no color) = do nothing.



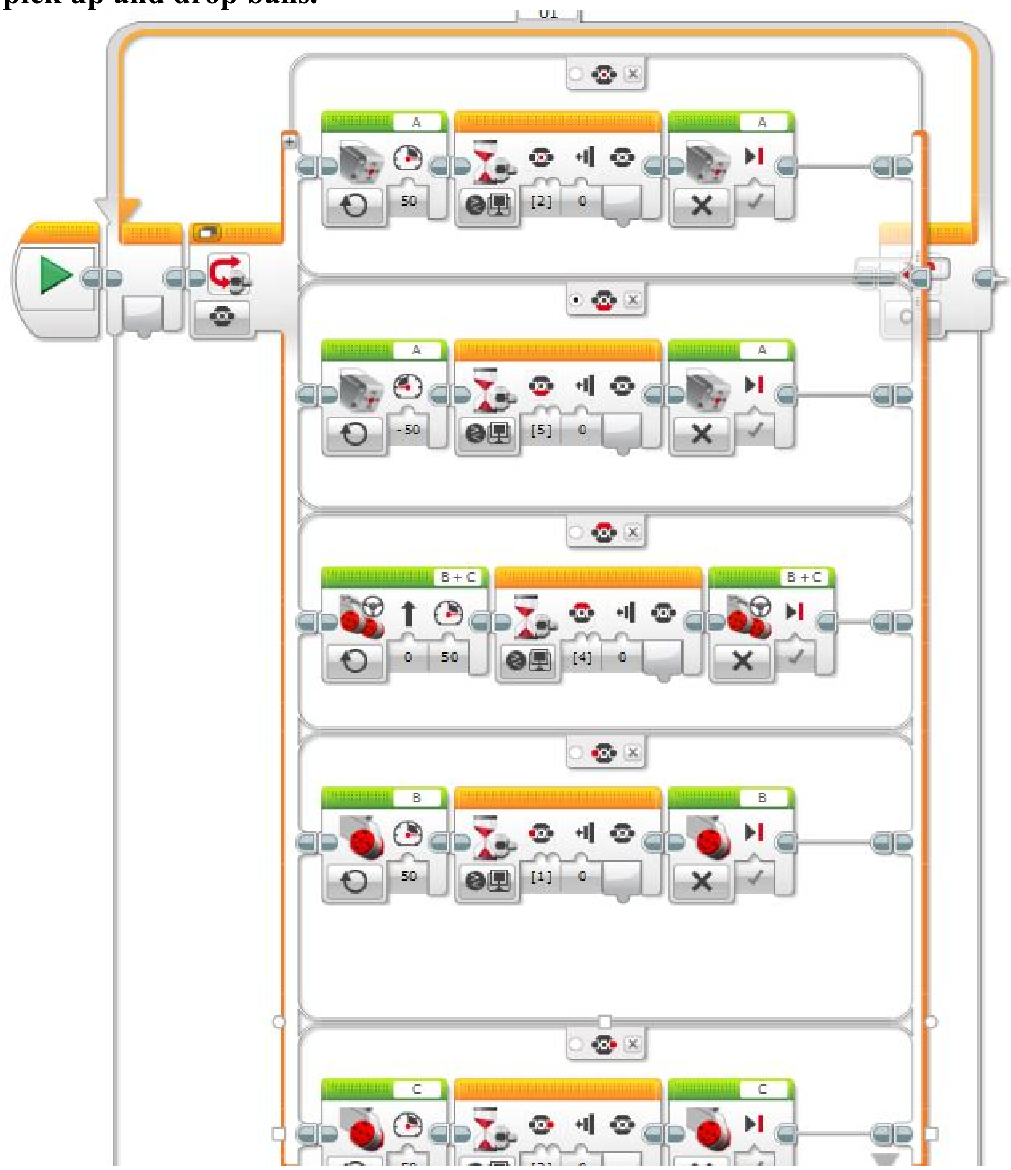
This program uses the gyro sensor to perform 90 degree turns in designated spots on the robotics table.



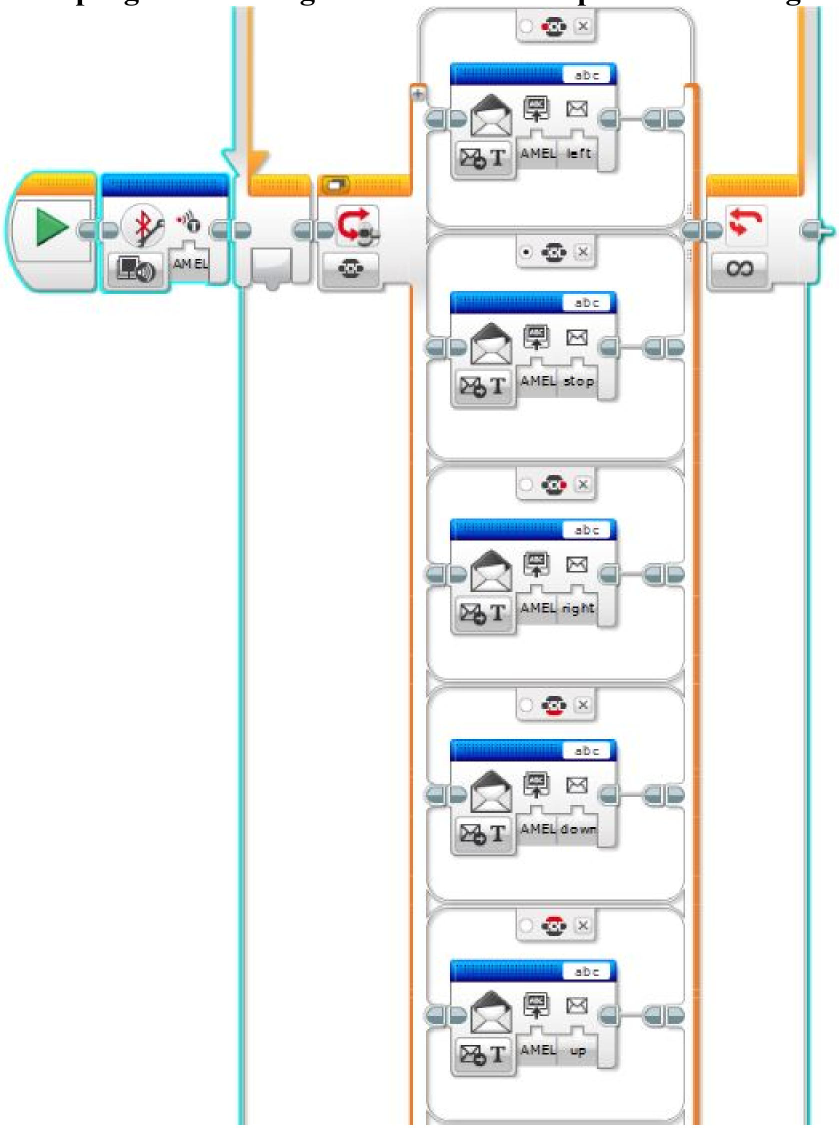
This program for the sumo robot project uses two algorithms at the same time (parallelism). The top row of blocks moves the robot forward while checking for a black line that marks the edge of the sumo ring. The bottom row swings an attachment using the medium motor.



This program uses a loop and a conditional (switch) statement to respond based on input. The middle and down robot buttons manipulate the claw attachment to pick up and drop balls.



This program establishes a Bluetooth connection with another EV3 brick and sends signals to this brick based on user input using a conditional within a loop. A receiver program running on the other brick processes the signals.



APPENDIX E

Timelines of Robotics Courses

Courses ran on an A-B day rotation. For students, this worked out to five 75-minute classes every 2 weeks. Some classes lost due to various events and holidays.

Task-based

Week 1: Introduction, kit inventory, BCC pre test administered.

Week 2: Building the robot with required sensors attached, begin planning for initial task options.

Week 3: Planning, programming, testing for the "EV3 mini challenges" task options.

Week 4: Planning, programming, testing for the "fruit picker" task options.

Week 5: Planning, programming, testing for fruit picker task options.

Week 6: Planning programming, testing for the fruit picker super challenge options.

Week 7: Planning, programming, testing for the "traffic light" challenges using the color sensor. Interviews begin.

Week 8: Planning, programming, testing for the color sensor super challenge options. Interviews ongoing.

Week 9: Planning, programming, testing for the gyro sensor "mower" tasks. Interviews ongoing.

Week 10: Planning for the "obstacle course" multi-sensor task options.

Week 11: Planning, programming, testing for the obstacle course tasks.

Week 12: Wrap up, BCC post test administered.

Week 13: Clean up and kit inventory.

Project-based

Week 1: Introduction, kit inventory, BCC pre test administered.

Week 2: Exploring project options in the class resources. Discussing and identifying interests. Begin research.

Week 3: Research and planning. Begin filling out design journals.

Week 4: Research, planning, and building. Filling in design journals.

Week 5: Planning and building.

Week 6: Building, programming, and testing.

Week 7: Building, programming, and testing. Begin revisions to build and program based on testing. Interviews begin.

Week 8: Revising and testing. Interviews ongoing.

Week 9: Revising and testing. Interviews ongoing.

Week 10: Testing and demonstrations. Interviews ongoing.

Week 11: Demonstrations and participation in events (for sumo robots and remote-controlled robots)

Week 12: Wrap up, BCC post test administered.

Week 13: Clean up and kit inventory.

APPENDIX F

Robotics Resources

The following resources have been used by teachers and students (or both) in the robotics courses at TIS. Most of the resources are specific to Lego Mindstorms robotics kits, but could be applied to other types of robotics kits as well.

https://www.cmu.edu/roboticsacademy/roboticscurriculum/Lego%20Curriculum/index.html - The Carnegie Mellon Robotics Academy website contains several strands of robotics curriculum. The Lego EV3 curriculum focuses on the fundamentals of programming and also contains a few project ideas.

https://stemrobotics.cs.pdx.edu/ - Robotics resources and curriculum examples and ideas from Portland State University.

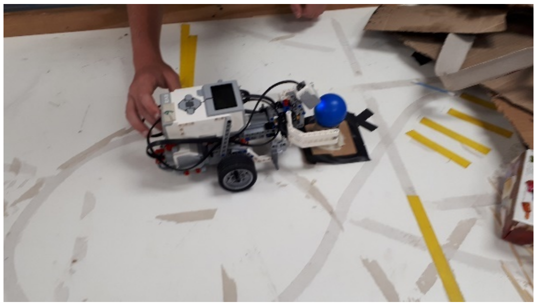
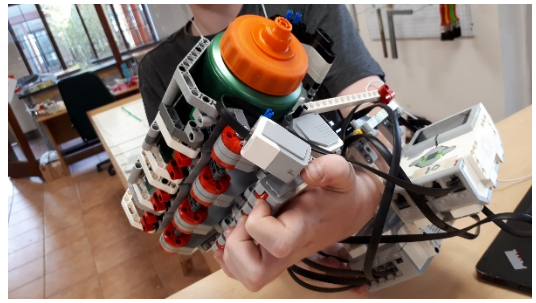
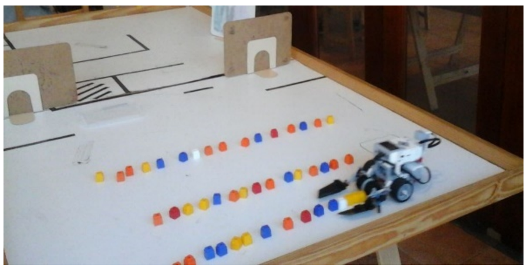
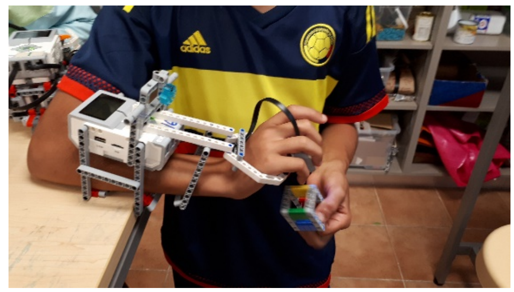
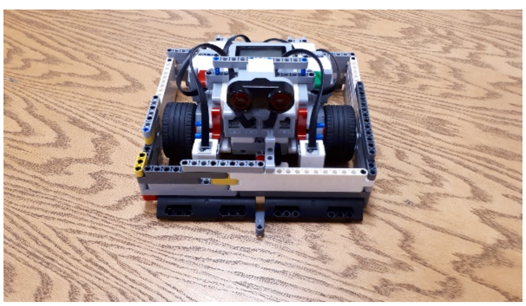
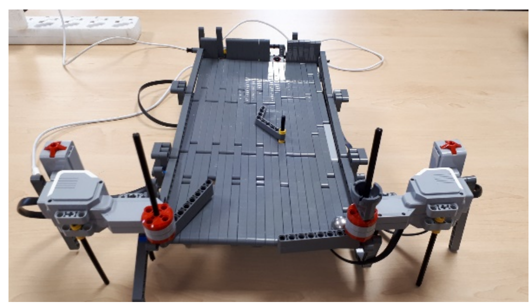
https://builderdude35.com/ - EV3 Mindstorms tips and tutorials on everything related to building and programming the EV3 from an MIT student. Contains a link to his YouTube channel with more tutorials and resources.

https://education.lego.com/en-us/support/mindstorms-ev3/building-instructions - Official Lego Education website with building instructions and programming resources for the Lego EV3 Mindstorms Education kits.

http://www.legoengineering.com/ - Website with ideas for a variety of tasks and projects involving Lego robotics kits.

APPENDIX G

Pictures from Projects and Tasks

“Fruit Picker” tasks / claw attachment 	**Robotic limb prototype with fingers** 
“Mower” tasks with gyro sensor 	**Robotic limb with sensor & voice** 
Sumo robot 	**Pinball project** 

APPENDIX H

IRB Approval

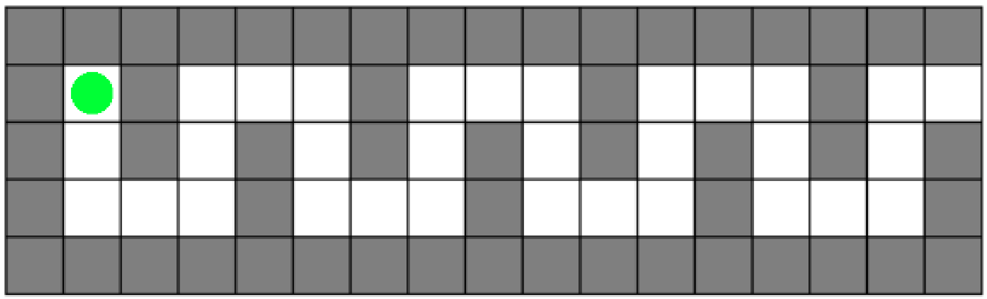
This research was conducted with the permission of the Institutional Review Board (IRB) at Boise State University, protocol number 101-SB18-192.

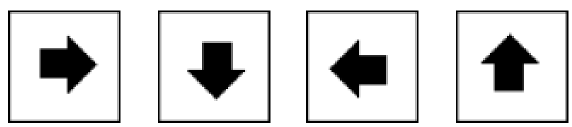
APPENDIX I

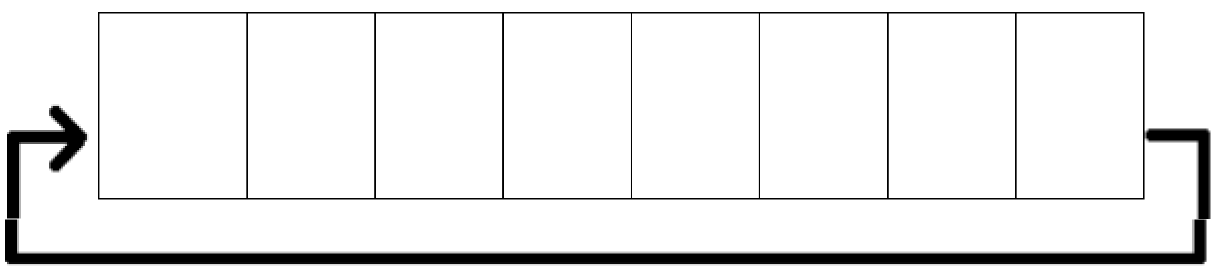
BCC Task Set Example

Robot Exit

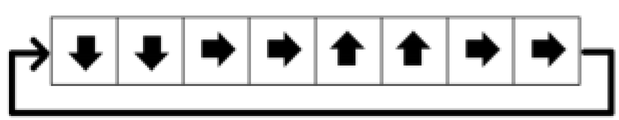
Help the green robot to exit the maze.
Draw the arrows in the boxes to form a set of instructions. You can use each arrow as many times as you want. The robot will repeat these instructions 4 times.







Answer:



Explanation:

In mobile robotics, maze problem solving is one of the most common problems. To solve this problem, an autonomous robot is used. Mazes can be of different kinds; having loops, without any loops, grid systems or without a grid system. In this short loop maze algorithm, the robot is instructed to follow a preference of directions.

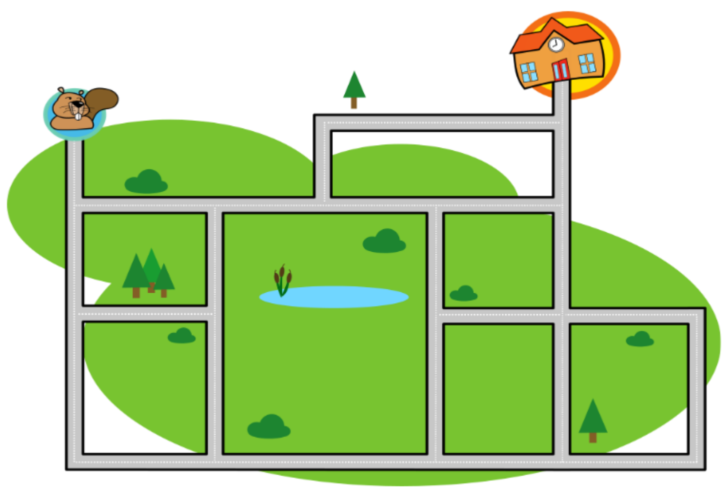
Car Trip

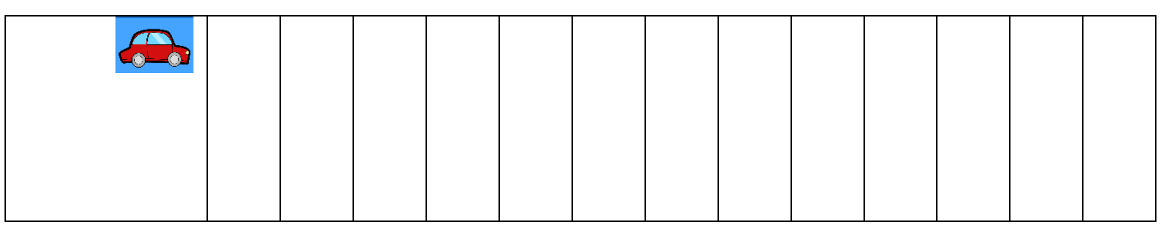
A self-driving car needs to take a student to school.

The car is programmed so that it only use these 3 instructions:

Forward: go forward until you cannot go forward anymore
Left: turn 90° left
Right: turn 90° right

Question:
Write a set of instructions (a program) that will get the beaver to his school.
You can do this by drawing the three instruction blocks in the boxes next to the car. You can use each block as many times as you want.





Answer: 

Explanation:

The important thing for participants to remember is that there is no forward movement when turning 90 degrees, so the 'straight' command has to be entered between every turn

Party Banner

Beaver Bert has a long strip of coloured paper for a party.

The strip has three different colours (yellow, red, blue) in a regularly repeating pattern.

Bert's friend, James, has cut out a section of the paper, as shown in the diagram below.

Y	R	R	B	Y	R	R	B	Y	R	R	. . .	R	B	Y	R	R	B

James says that he will give back the missing piece of paper if Bert can correctly guess the size of the piece cut out.

Question:
How many coloured squares can the missing piece of paper have?

Circle the answer below:

31 32 33 34

Answer: 31

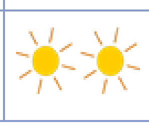
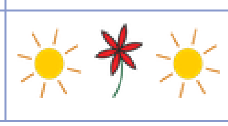
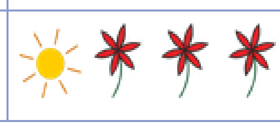
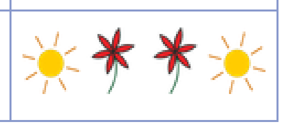
Explanation:
We know the pattern ended with YRR, meaning that the James has cut out at least one B. After that, he cuts out some number of sequences of 4 (i.e., YRRB). After that, the right side of his piece of paper must have YR, since the second piece begins with RB. So, the length of his piece of paper is 1 (for B) + 4*X (where X is the number of repeated patterns YRRB) + 2 (for the YR). So, the length of her paper is 4X+3.

Looking at the possible answers, we see that 31/4 has remainder 3: that is, 31 = 4*7 + 3. So, our equation is solved when X=7. None of the other answers can be written as 4X+3.

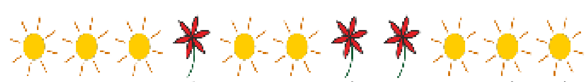
Beaver Code

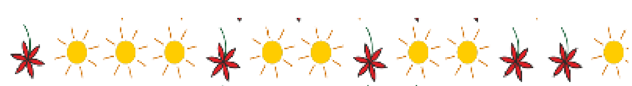
Barbara has been given two stamps. With one she can produce a little flower, with the other a little sun.

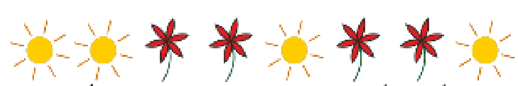
Being a clever girl, she thinks of a way to write her own name by using the code below:

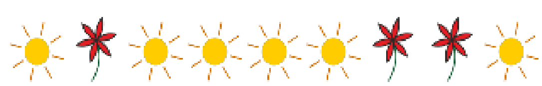
Letter	B	A	R	E	Y
Code					

Question:
Match the 4 sun-flower-codes below to the names of her four friends. (Draw a line from the name to the correct sun-flower-code).

Abby 

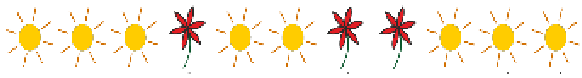
Arya 

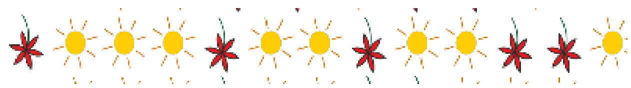
Barry 

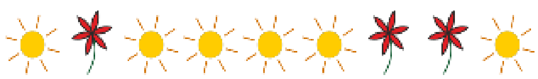
Ray 

Answer:

Abby = 

Arya = 

Barry = 

Ray = 

Explanation:

This problem is most easily solved by noting that Abby starts with an A and a B and so we look for a code with two suns and a flower at the start. There is only one of these so this is assigned. Next it is noted that Arya's code begins with three suns and a flower. Again there is only one of these so this is assigned. By continuing in this way, all the codes are quickly assigned to the correct names.

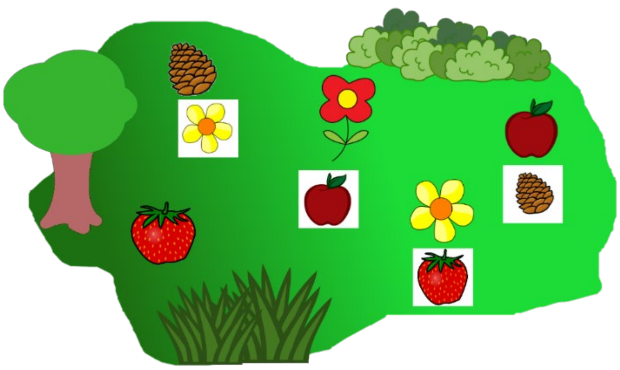
Secret Recipe

Eszter has asked István to cook a special cake made of five ingredients.

She has put labels with white backgrounds next to the ingredients in the garden. One ingredient has no label.

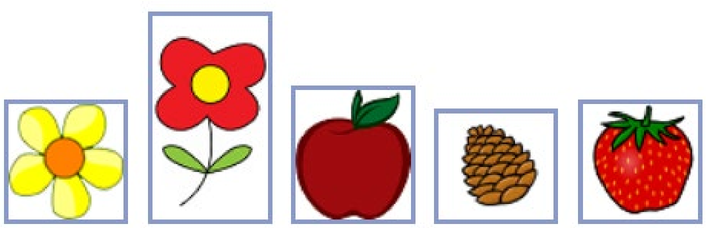
The labels next to each ingredient tell István the next ingredient that needs to be added.

The garden looks like this:



Question:
Which ingredient should be added first?

Circle the answer:



Answer: 

Explanation:

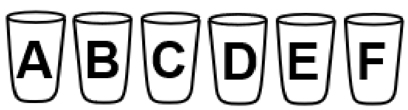
If Eszter starts with the flower, she can add all five ingredients in the right order.
The first added ingredient must be the one with no referring image.

Magic Potions

Betaro Beaver has discovered five new magic potions:

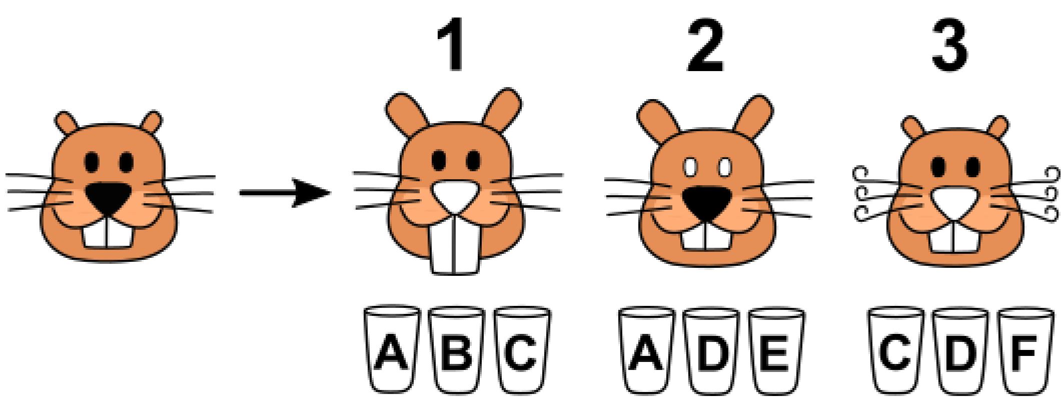
- one makes ears longer
- another makes teeth longer
- another makes whiskers curly
- another turns the nose white
- the last one turns eyes white.

Betaro put each magic potion into a separate beaker. He put pure water into another beaker, so there are six beakers in total. The beakers are labeled A to F. The problem is, he forgot to record which beaker contains which magic potion!



To find out which potion is in each beaker, Betaro set up the following experiments:

Experiment 1: A beaver drinks from beakers A, B and C together - the effects are shown in Figure 1.
Experiment 2: A beaver drinks from beakers A, D and E together - the effects are shown in Figure 2.
Experiment 3: A beaver drinks from beakers C, D and F together - the effects are shown in Figure 3.



Question:
Which beaker contains pure water?

Circle the answer:

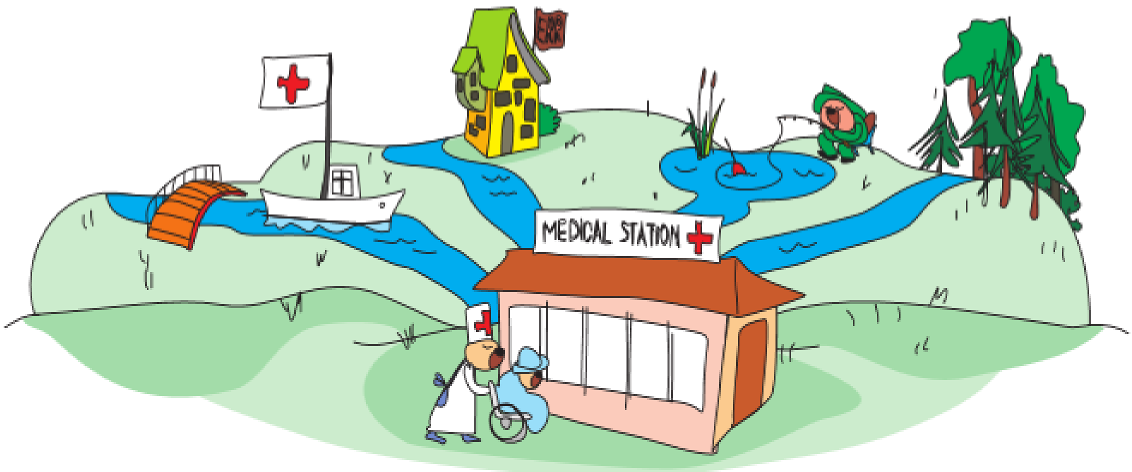
A B C D E F

Answer: D

Explanation:

By Experiment 1, none of A, B and C is pure water, since there are three changes that happen to the beaver.
By Experiment 2, either D or E is pure water or the magic potion making his nose white since A is not pure water, from Experiment 1.
By Experiment 3, D and F are pure water or the magic potion making his whiskers curly, since C is not pure water, again from Experiment 1.
Therefore, D is pure water.

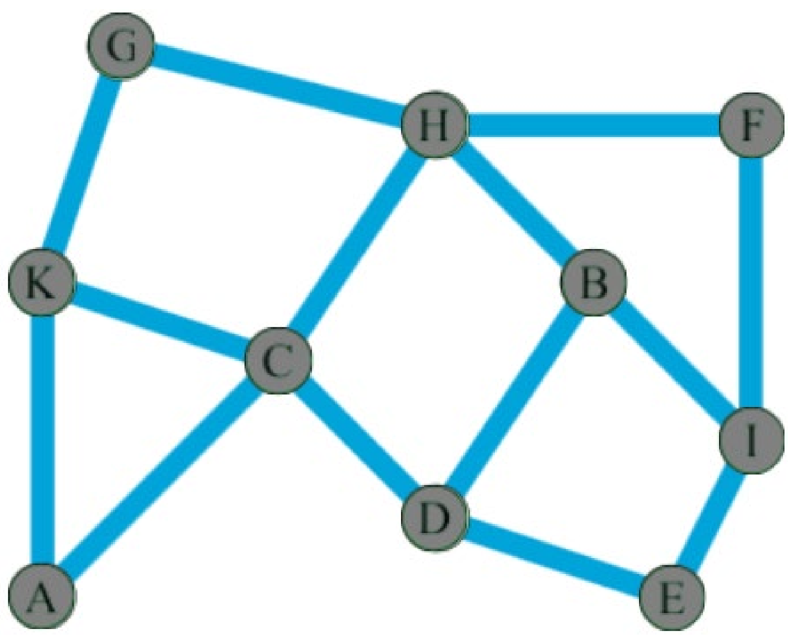
Primary Health Care



Doctor Hamid wants to build three hospitals for the beavers.
The hospitals can only be built on the places shown on the map below.
To get to a hospital, the **beavers should not have to swim through more than one stream from any of these places.**

Question:
Choose three places to build the hospitals for Doctor Hamid.

Circle the 3 letters where the hospitals should be built on the map below:



Answer:

There are several correct solutions, one for instance uses the places E, H and K:
• For the places D, E and I the beavers can swim to E.
• For the places B, C, F, G and H the beavers can swim to H.
• For the places A, C, G and K the beavers can swim to K.
The other solutions are: A E H, C G I, C H I, C I K, D F K, B I K and C E H.

Explanations:

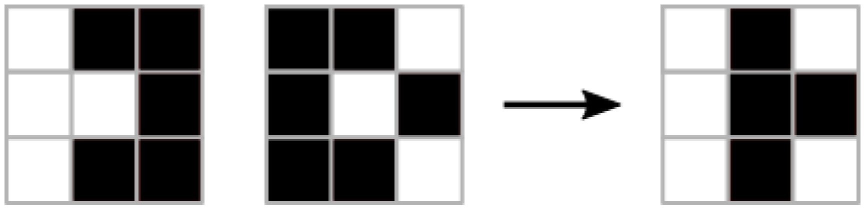
The solutions can be found by placing a station at a random position and marking all stations that are reachable within one step. Then you can position the next station and so on.

Once all three stations are placed there are two possibilities: either it's a solution or there are one or more places that are not marked. If it's not a solution, you can remove the last station you've placed and place it in another place and check again.

If you are still not lucky to find a solution with 3 stations you have to "backtrack" and place the last station on another place. By doing this systematically one can find all possible solutions.

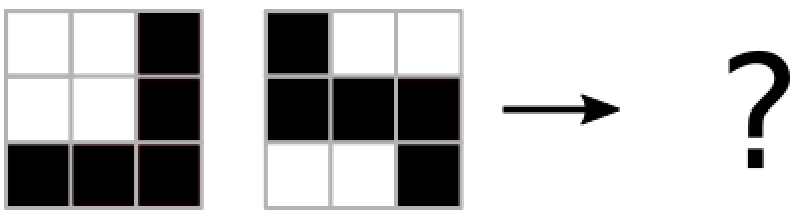
Paint It Black

Combining Card A and Card B, you get Card C:



Card A Card B Card C

Question:
How many black cells will Card F have after combining Card D and Card E if the same pattern is followed?



Card D Card E Card F

Write your answer here: ________

Answer: 3

Explanation:

Combining the cards obeys the following rule. When the colour of the corresponding cells is the same the resulting colour is black. Otherwise the resulting colour is white.

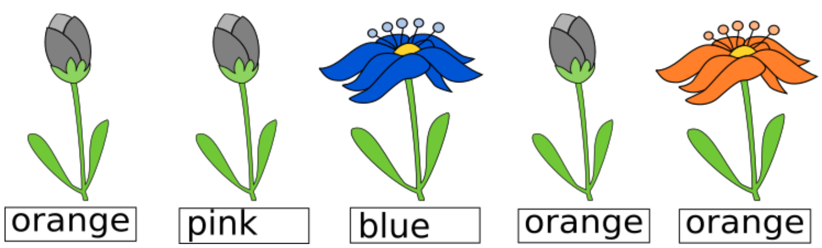
Blossom

Jane is playing a computer game.

First the computer secretly chooses colours for five buds.
The available colours for each flower are **blue**, **orange**, and **pink**.

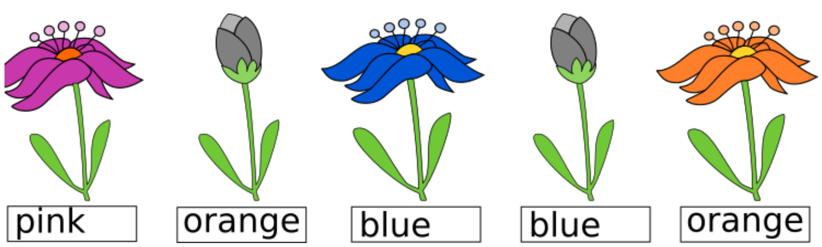
Jane has to guess which flower has which colour. She makes her first five guesses and presses the *Blossom* button.
The buds, whose colours she guessed correctly, break into flowers. The others remain as buds.

Jane's first go:



Jane then has another go at guessing and presses the *Blossom* button again.

Jane's second go:



Question:
What colours did the computer choose for the five flowers?

Circle the correct row below:

Blue, pink, blue, orange, orange

Pink, blue, blue, blue, orange

Pink, blue, blue, pink, orange

Pink, pink, blue, pink, orange

Answer: pink, blue, blue, pink, orange

Explanation:

After two guesses there are three blossomed flowers. So we can already see the colour chosen by the computer for the first, third and fifth flower. The colour of the first flower is pink, so answer **blue, pink, blue, orange, orange**, cannot be correct.

For the second flower Jane guessed pink in the first guess and it did not blossom, then she guessed orange and it did not blossom either. As there are only three colours available, the second flower must be blue. This rules out answer **pink, pink, blue, pink, orange**.

Similarly, Jane chose orange and blue for the fourth flower and it still has not blossomed, so it must be pink. And this rules out answer **pink, blue, blue, blue, orange**.

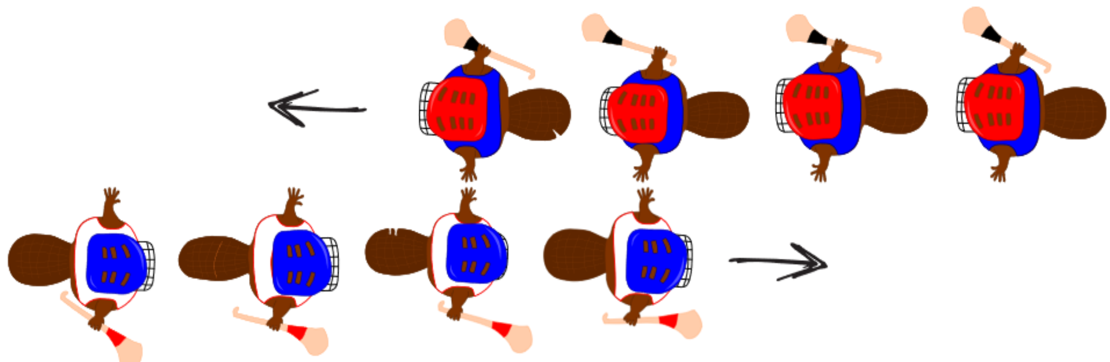
Hurlers Shake Hands

Beavers enjoy playing hurling.

After the game ends, the beavers in each of the two teams line up in a row and walk past the other team.

As they pass each other, they shake hands.

At the beginning, only the first player on each team shakes hands.
Next, the first two players shake hands (see picture below).
This continues until each player has shaken hands with every player on the other team.



Question:
There are 15 players on each team.
If each player takes one second to shake hands and move to the next player, how many seconds of shaking hands will there be?

Write your answer here: ____________

Answer: 29

Explanation:

The amount of handshaking is exactly the length of one line plus the length of the other line, minus one.

Let us imagine that there is only 1 player on each team. After 1 second, all handshaking has finished. Let us imagine that there are only 2 players on each team.

During the first second, the first player on each team shakes hands. During the second second, the first player on each team is shaking hands with the second player on the other team, and during the third second, the second two players are shaking hands with each other. So, that’s three seconds.

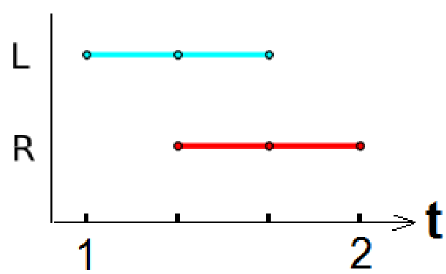
With 15 players in each team, the number of seconds required is $15 + 15 - 1 = 29$.

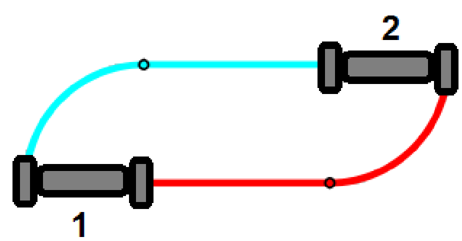
Segway

Jan has a special vehicle that looks like a Segway. He moves it by pressing two buttons: a blue (light) button on the left, and a red (dark) button on the right.

When he presses a button, the wheel on that side of the vehicle rotates:
If both buttons are pushed at the same time, both wheels rotate and the vehicle moves forward.
If he pushes a single button, only one wheel rotates and the vehicle turns.

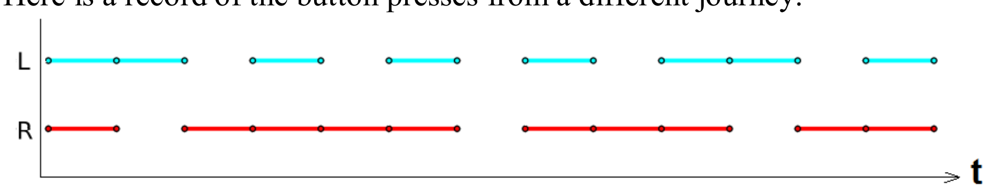
Example:
The follow tables shows which button was pushed when, and how the vehicle moved from location 1 to location 2.





First, the blue button was pressed and the vehicle turned to the right. Then both buttons were pressed, and the vehicle moved forward. Finally the red button was pressed, and the vehicle turned left. The orientation of the vehicle is now the same as in the beginning: facing towards the upper wall.

Question:
Here is a record of the button presses from a different journey:



The vehicle kept going until it hit one of the walls. At the start the vehicle was facing towards the upper wall.

Towards which wall was the vehicle facing in the end? Circle the answer below:

Upper Lower Left Right

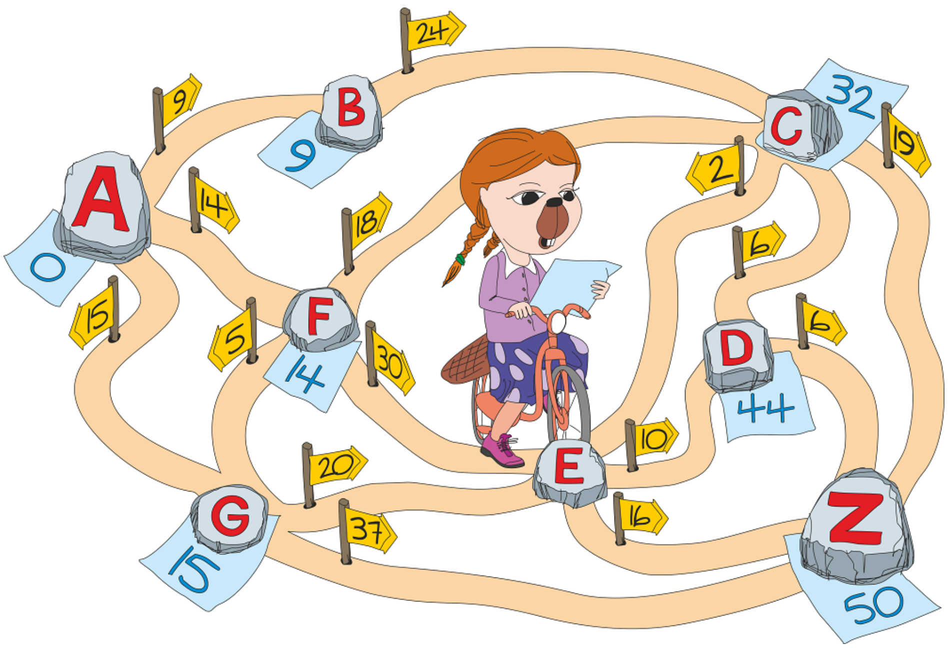
Answer: Lower

Explanation:

The left button was pressed 8 times during the ride, while the right button was pressed 10 times. That means the right button was pressed two times more and the vehicle turned left twice, so will face the opposite direction from where it started - it must hit the lower wall.

Bike Paths

Cleveria is a beaver biker. She explores the one-way paths that pass through the villages in her district. Each village has a village stone labeled with a single letter. All the paths have a distance and a direction. The distance and direction are given by the yellow flags.



Over the course of many different trips Cleveria leaves blue notes with a number on under a stone in each village. The notes are about the distance from village A to the village stone with the note under.

Question:
What is the meaning of the numbers she has left under the stones?

Circle the correct answer:

The shortest distance going through the least number of villages

The shortest distance to this village

The shortest distance to this village by taking a left turn at crossings if possible

The shortest distance to this village by taking a right turn at crossings if possible

Answer: The shortest distance to this village

Explanation:

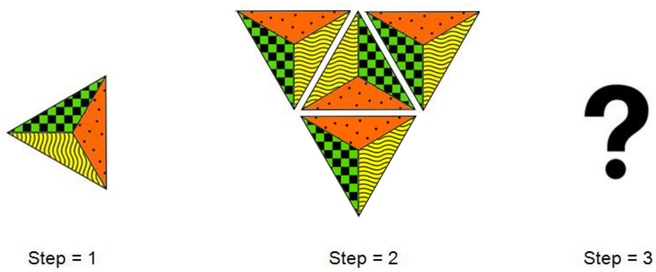
In order to find the correct answer, the distances for each village according to the different specifications have to be computed:
Shortest distance going through least number of villages is wrong because otherwise D = 45, Z = 52;
Shortest distance to the village by taking left turns is wrong because otherwise C = 33, D = 45, Z = 52;
Shortest distance to the village by taking right turns is wrong because otherwise C = 51, D = 45, Z = 52.
So the blue number shows the length of the shortest route from A to a particular village.

Triangles

A beaver wants to create a mosaic with identical, triangle-shaped tiles.

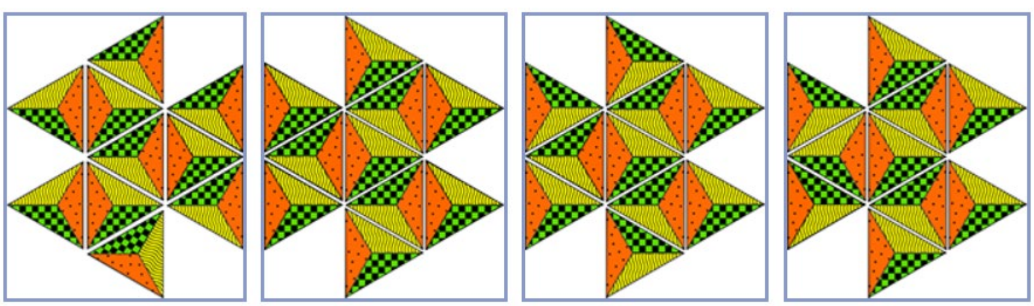
He starts with one tile. He rotates it 90 degrees clockwise and then adds tiles on each side of the triangle-shaped tile, as shown in the picture below.

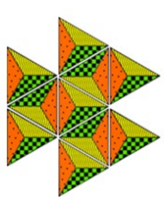
Then he rotates the whole shape 90 degrees clockwise again and adds tiles to the sides as before.



Question:
What will be the final shape of the triangles after step 3?

Circle the correct figure:



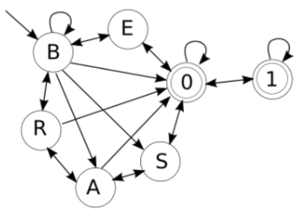
Answer: 

Explanation:

Answer a is incorrect because the tiles are not rotated 90 degrees clockwise.
Answers c and d are incorrect because the tiles do not match on their adjacent sides.

Rafting

Beavers build rafts. For river traffic control, all rafts should be registered.
This means that each raft should have a license plate with unique text.
The text is made up of letters and digits as shown in the diagram below.
The license must start with the letter B and end with the digit 0 or 1.



Question:
Which **two** of the license plates **cannot** be registered?

Circle two answers below:

BB0001 BBB100

BBB011 BB0100

BR00A0 BSA001

BE0S01

Answer: BBB100 & BR00A0

Explanation:

The best way to solve this is simply follow the diagram and check the solutions one by one.
BBB100 is incorrect, because the digit part starts with 1 (you can't get from the B to the 1) and BR00A0 is incorrect because you can't get from 0 to A as it is a one way arrow.

Find the Thief

OH NO! The famous Blue Diamond was stolen from the museum today: a thief has swapped it for a cheap imitation with a green color.

Facts:
There were 2000 people who visited the diamond room today. They entered one by one.
Inspector Bebro must find the thief by interrogating some of these visitors.
He has a list of all 2000 visitors in the order they entered the room.
He will ask each person the same question: *Was the diamond green or blue when you saw it?*
Each person will answer truthfully, except for the thief, who will say that the diamond was already green.

Question:
Inspector Bebro is very clever and will use a strategy where the number of people interviewed is as small as possible.
Which of the following statements can he make without lying?

Circle the correct statement:

"I can guarantee that I will find the thief by interviewing fewer than 20 people"

"I can find the thief but I need to interview between 20 and 200 people"

"I can find the thief but I need to interview between 200 and 1999 people"

"I have to interview every single visitor in order to find the thief"

Answer: "I can guarantee that I will find the thief by interviewing fewer than 20 people"

Explanation:

Since the order of the visitors is known, we can use this to devise an algorithm. Asking the 1000th visitor, for example, will tell us if the diamond was taken from someone in the first 1000 visitors or second 1000 visitors, reducing the number of suspects by half. Repeating this strategy will uncover the thief by interviewing less than 20 people.

Notes about Modifications of Questions for Paper and Pencil Format

Robot Exit

Added empty boxes so students could draw the arrows by hand. Clarified the directions so students would know they can use each arrow as many times as they want.

Car Trip

Added empty boxes so students could draw the arrows by hand. Clarified the directions so students would know they can use each arrow as many times as they want.

Party Banner

Changed the answer format so students could circle the correct answer.

Beaver Code

Changed the answer format so students could match the name with the code by drawing a line between them.

Secret Recipe

Clarified the directions so students would know the labels in the picture have white backgrounds. Changed the answer format so students could circle the answer.

Magic Potions

Changed the answer format so students could circle the correct answer.

Primary Health Care

Changed the answer format so students could circle the correct answer.

Paint It Black

No modifications necessary.

Blossom

Changed the answer format so students could circle the correct answer.

Hurlers Shake Hands

No modifications necessary.

Segway

Changed the answer format so students could circle the correct answer.

Bike Paths

Changed the answer format so students could circle the correct answer.

Triangles

Changed the answer format so students could circle the correct answer.

Rafting

Changed the answer format so students could circle the correct answers.

Find the Thief

Changed the answer format so students could circle the correct answer.