USING MEMORY/ORGANIZATION INSTRUCTIONAL AIDS TO TEACH COMPUTER PROGRAMMING WITH LEGO MINDSTORMS

by

JASON TURMAN

(Under the Direction of Jay W. Rojewski)

ABSTRACT

This action research study examined the effect of custom memory/organization instructional aids to teach computer programming using iteration with LEGO Mindstorms. The strategies included KWL charts, flowcharts, and visual cues. This study focused on the strategies' influence on students' performances, students' perceptions, and observed patterns and frequency of use by students. Activities/lessons used LEGO Mindstorms EV3 robots and the *Introduction to Programming LEGO Mindstorms EV3 Curriculum* from Carnegie Mellon's Robotic Academy. Action research utilized three cycles with the curriculum. This study revealed a pattern in which students were hesitant to use instructional aids when programming their robots and preferred trial-and-error methods. However, once students used the instructional aids and understood how to use them, a majority of students stated that KWL charts, flowchart, and visual cues helped them with programming their robot. The three instructional aids had similar benefits: students stated that they (a) were good for note-taking and reference guide, (b) helped organize thoughts, and (c) helped track progress. Moreover, students' feedback led me to create a new version of the KWL chart to be used with programming, called the $K(WL)^n$ chart. Students' suggestions on the design process of flowcharts influenced the limitation of the number and types of symbols typically used in a flowchart to two. These changes to the flowchart design process helped students understand how to design and use a flowchart with programming. Through the study, I identified that the use of chunking or reducing the amount of information into more manageable steps helped students programming their robots. Chunking was a main component of the success of the custom instructional aids used in this study.

INDEX WORDS: Computer programming, education, LEGO Mindstorms, students

USING MEMORY/ORGANIZATION AIDS TO TEACH COMPUTER PROGRAMMING WITH LEGO MINDSTORMS

by

JASON TURMAN

BS, Jackson State University, 2002

JS, University of Phoenix, 2009

A Dissertation Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

DOCTOR OF EDUCATION

ATHENS, GEORGIA

2019

USING MEMORY/ORGANIZATION AIDS TO TEACH COMPUTER PROGRAMMING

WITH LEGO MINDSTORMS

by

JASON TURMAN

Major Professor: Jay W. Rojewski
Committee: Roger Hill
John Mativo

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
August 2019

DEDICATION

This dissertation is dedicated to my parents, the Dobbins (Tony and Pat) and Charles. They have always been a firm advocate of education since I was young. They were also very supportive of me during this journey of completing my doctorate. I am so fortunate for their blessings.

I also dedicate this dissertation to my wife, Brittany, and our daughter, Kennedy. Brittany has been encouraging on my journey towards this degree. I hope to be just as caring and supportive, as she works on her doctorate. I hope this dissertation serve as my testament to Kennedy of the importance of education.

ACKNOWLEDGEMENTS

I would like to thank God for allowing me the opportunity to attend the University of Georgia and enroll in this doctorate program. I also thank God for helping me through this seven-year journey to completing my doctoral degree.

I would like to thank my committee Dr. Rojewski, Dr. Mativo, Dr. Hill, Dr. Clinton, and Dr. Smith, for their guidance and their commitment to excellence. Thank you to all my professors (Dr. Tanner, Dr. Womble, Dr. Lee, and Dr. Wicklein) for their time, effort, and being a part of this process. Most of all, thank you, Dr. Rojewski for your undeniable support. Words cannot express my gratitude for your dedication in helping me complete my doctorate. Thanks to you, I have a greater appreciation for statistics, scholarly research, and the impact educators have on students' lives. Thank you!

Thank you to all the people that supported me on this seven-year journey. Thank you to my Sego family, who have been supportive since the beginning. Thank you to Dr. Brent Hollers for your initial advice. Thank you to UGA's Writing Center. Thank you to Dr. Bethany Johnson, on her continuing support and friendship. Thank you to Dr. Megan Morgan. You are more than just an editor to me. You were a truly a blessing. Your service was excellent, dependable, and informative. Thank you to everybody who ever helped me on my journey at UGA.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Scientific and technological innovations are imperative for the development of new industries, creation of job growth, and improvement in the quality of life (National Science Board, 2010). In order to remain at the cutting edge of knowledge, high schools should be encouraged to offer opportunities for learning engineering concepts. Pellegrino, Chudowsky, and Glaser (2001) indicated that more schools are implementing new technology-based curriculum, instruction, and assessments to (a) encourage a greater depth of understanding of content by making learning relevant, (b) making students' thinking visible for quicker, individualized, and more detailed feedback from teachers, and (c) develop students' thinking by designing lessons that are open-ended, complex, and problem-based.

High schools are having a difficult time attracting and maintaining students in new classes designed to teach students skills in high demand (Burbaite et al., 2013; Alvarez & Larranaga, 2014; National Science Board, 2010). In order to meet the need for new instructional strategies to improve students' knowledge of science, technology, engineering, and mathematics (STEM) principles, educational robots are being increasingly utilized in K-12 classrooms (Bos, Ortiz, & Smith, 2015). Educational robots provide several benefits for education in that they are (a) interdisciplinary, (b) engaging, (c) open-ended, (d) problem-based, (e) realistic, and (f) make students' learning more concrete (visible) (Alimisis, 2009; Petre & Price, 2004). Bransford, Brown, and Cocking (2000) noted that integrating proper instructional methods with technology, like robots, is required to improve student learning.

Trivodaliev et al. (2017) indicated that the advancement of technology has created an urgency to develop new computer science instructional designs that meets technology industries' demands for qualified graduates. Computer Science is the study of the theory, design, application, and impact of computers and computational systems (Code.org, ND). Geist (2016) suggested teaching young children programming because computers have become a ubiquitous part of everyday life (e.g., refrigerators, coffee pots, and even clothing). Programming exposes students to computational thinking (Lye & Koh, 2014). Wing (2006) coined the term computational thinking to describe programming. *Computational thinking* is a recursive mode of thinking utilized to solve problems, design systems, and understand human behavior by using computer science principles. Computer programming involves analyzing an issue, producing an algorithmic solution, and then translating the algorithm into program code (Saeli, Perrenet, Jochems, & Zwaneveld, 2011).

Sheard, Simon, Hamilton, and Lonnberg (2009) identified a wealth of research on the difficulty students experience when learning to program and the hardships teachers face when teaching programming. Computer science classes have a difficult time attracting and retaining students because of their complex concepts such as data structures, algorithms, and syntax (Alvarez & Larranaga, 2014; Burbaite, Damasevicius, & Stuikys, 2013). Also, despite national attention many teachers still have little understanding of computer science because of a lack of preparation (Israel, Pearson, Wherefel, Shehab, & Tapia, 2015).

Milne and Rowe (2002) studied the difficulties of learning and teaching computer programming using a web-based questionnaire administered to college students, teachers, and tutors. Results showed that students' main issue with programming was understanding the concept of program flow. Milne and Rowe (2002) suggested that visualization tools could support students in understanding the flow of a computer program. Visual indicators such as highlighters, color-coding, and graphic organizers can support students in visualizing smaller steps and organizing those steps into a sequence (Gierach, 2009).

To help prepare students for a possible job in cyber security, all high schools in the county in which I teach are offering the Web and Digital Design pathway. With cybersecurity on pace to be the largest employer in the Augusta area with an estimated 138% job increase within five years, local businesses want their current and future employees to have a bachelor's degree in information technology or computer science (Augusta University, 2017). The county in which I teach has begun teaching coding as early as kindergarten and offering computer science courses to middle schoolers (WRDW, 2017). The state of Georgia provides 17 career clusters that offer students opportunities to learn about specific careers (Georgia Department of Education, 2018). Each cluster offers multiple pathways that are composed of three sequential classes. As the instructor of the Web and Digital Design pathway in the Information Technology cluster, I have noticed greater enrollment and efforts to enroll students into my classes. The county I teach in is actively enrolling students into the Web and Digital Design classes as a precursor for students to enroll in AP Computer Science. The three classes of the Web and Digital Design pathway are:

1. *Introduction to Digital Technology* is the foundational course for the Web and Digital Design pathway. In this course, students learn, communicate, and apply basic knowledge of hardware, software, and other computer-related technologies.
2. *Digital Design* is the second class of the Web and Digital Design pathway. Students learn in this course to use various digital media applications to create text, graphics, animation, sound, video, and digital imaging for various formats.

3. *Web Design* is the third course in the Web and Digital Design pathway. In this course, students will learn to plan, design, and create professional looking websites.

AP Computer Science courses teach about problem-solving, hardware, algorithms, and computers' effects on society (College Board, 2014). AP Computer Science allows high school students to enroll in college-level classes and earn college credits (College Board, 2014).

Moreover, I am the only teacher with a class that has any reference to technology, so all up-coming ninth-graders are encouraged to enroll in this class. Because many students have no experience in computer programming and possess various learning strengths and styles, I am challenged to implement instructional strategies that aid students in learning computer programming. Teaching and learning computer programming is known to be difficult (Alvarez & Larranaga, 2014; Burbaite, Damasevicius, & Stuikys, 2013; Israel et. al., 2015; Sheard et. al., 2009). To aid and inspire students to enroll and stay in my computer programming classes I use LEGO Mindstorms educational robots. Panadero, Delgado-Kloos, and Villena-Roman (2010) suggested that LEGO Mindstorms attracts and motivates students because of its features of real-world application, peer-learning, versatility, simplicity, and competitive use. However, technology alone will not produce the desired educational results without proper instruction. An instructional strategy that acts as an interface between students and educational robots could reduce the complexity of learning programming. To be able to successfully design and implement an effective and new instructional strategy, I use a trial-and-error method in the past, making modifications based students' feedback and my observations. With repeated evaluation of various learning strategies, I hope to be able to modify instructional aids to complement my instruction on programming.

Need for the Study

Teaching computer sciences is a complex subject that requires teachers to focus on programming language's syntax, design skills, and creative thinking (Mohorovicic & Strcic, 2011). Mohorovicic and Strcic (2011) suggested teachers design and implement new and innovative teaching methods to help student in learning computer programming. Professional development opportunities for teachers are important to ensure that teachers develop new instructional strategies and stay current with new technologies (Adams & Hamm, 2011; Lawless & Pellegrino, 2007; Sithole et al., 2017). Educators must prepare high-school students for future jobs that will likely require the understanding and use of technology (Adams & Hamm, 2011). Studies by Thomas and Knezek (2008) and O'Sullivan and Dallas (2010) showed that even though high school teachers are trying to prepare students with skills for the 21st century, recent high-school graduates continue to lack technology-related skills needed for college or work. Teachers with limited knowledge of new technology will hinder learning experiences because of the difficulty of developing lessons, supporting students, and assessing students (Stein, McRobbie, & Ginns, 1999).

Lee (2015) noted that seniors in 2004 and their selected college majors in 2006 to predict the influence of computer science classes in high school on students' enrollment in STEM-related majors in postsecondary school. Using logistic regression, Lee (2015) showed that taking one computer science class in high school significantly increased the chances of enrolling in a STEM major in a four-year college.

Educators can use research to add to their professional knowledge and improve practice to become better professionals and better prepare students (Creswell, 2012). Mertler (2014) stated that education leaders often use educational research literature to influence school improvements plan. K-12 teachers may find traditional research in education ineffective because the methods uses to collect and analyze data in traditional research do not reflect the teachers'reality (Mertler, 2014). Schools often focus on the availability of computers rather than on the preparation of teachers Hixon & Buckenmeyer, 2009). Hixon and Buckenmeyer (2009) argued that technology levels in schools are often measured by the student-to-computer ratio. Sithole et al. (2017) declared that professional development for teachers teaching STEM lessons is crucial because students' success is tied directly to teachers' ability to receive ongoing training. Kopcha (2011) noted that teachers may avoid implementing lessons on new technology due to a lack of planning, teaching, or classroom management required to successfully integrate technology in the curriculum.

Teachers can use action research to make inquiries into their own practice as a way to improve instructional strategies and student learning (Craig, 2009). The teacher-researcher uses a systematic process of collecting data and analyzing it to gain a better understanding of a predetermined educational issue in order to develop a solution. Educators favor action research over other research methods because of its relevance to teachers' localized needs; thus, allowing teachers' professional growth to become part of the research process. Traditional educational research is usually conducted by researchers who are removed from the environment they are studying and who generalize findings that may not be applicable to teachers' specific individual situations (Mertler, 2014).

This study used action research to examine the specific memory/organization instructional aid of flowcharts, KWL charts, and color-coding. Flowcharts are graphic organizers that are used for presenting, describing, or examining processes (Ayverdi et al., 2014). Flowcharts are a popular type of chart used in introductory computer science courses to teach algorithms and programming (Xinogalos, 2013). Students learning how to program computers may have problems understanding algorithms when designing computer programs (Dol, 2015). Know-Want-to-learn-Learned (KWL) charts and their variations are graphic organizers utilized by teachers to aid students in linking prior knowledge on a specific topic to what they are learning and to organize their course of study (Hershberger, Zembal-Saul, & Starr, 2006; Ogle, 2009). KWL charts were developed by Ogle (1986) to encourage inquiry and student questioning (Mihardi, Harahap, & Sani, 2013; Ogle, 2009). When using KWL charts students record what they know under the *K* column, what they want to learn in the *W* column, and what they have learned in the *L* column. Visual cues such as color-coding can be used to enhance instruction and improve the learning process by reducing cognitive load through reducing the visual search time for content (Tabbers, Martens, & Merrienboer, 2004). Color-coding is commonly used with programming software to improve source code readability by highlighting source code based on a semantic category (Dimitri, 2015; Sarkar, 2015). Highlighted syntax allows programmers to complete tasks more quickly by reducing the time required to read and understand source code meaning (Sarkar, 2015).

Various studies have revealed how teachers used KWL charts, flowcharts, and color-coding to support students' achievement in class. Cassay et al. (2004) showed that KWL charts could be used in any classroom and grade to guide students' learning because they are an adaptive strategy. Mihardi, Harahap, and Sani (2013) demonstrated the benefit of using KWL charts to increase students' creativity when solving physics problems. Kalyuga, Chandler, and Sweller (1999) used color-coding to improve high school students' scores on multiple-choice questions. Conway and Brown (2013) and Ewoldt and Morgan (2017) used color-coded graphic organizers to help elementary students with learning disabilities in writing by using colors to show relationships between undeveloped ideas and complete sentences in the paragraph. Color-coding helped computer science graduate students identify programming errors and complete assignments faster than participants who did not use color coding (Sakar, 2015). Sakar (2015) used unique colors on correlating text to color elements of a diagram.

Purpose of the Study

The purpose of this action research study was to determine the effect of various, custom memory/organization instructional aids on the achievement of students in an urban high school in Augusta, GA students using LEGO Mindstorms robotics. McLeod, Fisher, and Hoover (2003) stated teachers can use knowledge of students' current abilities and their prior knowledge to develop appropriate instructional strategies. The instructional design process includes selecting and constructing resources to help learners meet their needs to accomplish learning objectives (Rothwell & Kazanas, 2008). Furthermore, various studies have indicated that students' perceptions are a major influence on what they learned, why they learned it, and how they learned it (Aghamolaei & Fazel, 2010; Bakhshialiabad, Bakhshi, & Hassanshahi, 2015; Marton & Saljo, 1976; Mayya & Roff, 2004). Noe (2008) described perception as the knowledge or understanding of one's surroundings that influences ensuing actions. Aghamolaei and Fazel (2010) and Bakhshialiabad et al. (2015) suggested that students' perceptions could be an important factor in implementing interventions because perception can have a notable effect on students' academic achievement. Gill, Timpane, Ross, and Brewer (2001) defined academic achievement as the satisfactory attainment of academic skill and knowledge that is measured through assessments.

The study was conducted in a web design class using LEGO Mindstorms robotics (2017) in Georgia. The instructional strategies used in this study allowed me to better understand, develop, and implement custom memory/organization instructional aids by improving my instruction, student achievement could be increased. The research questions for this study were:

1. Did my instruction improve when using KWL charts to teach computer programming with LEGO Mindstorms?
    a. What was the influence of using KWL charts on students' performance?
    b. What were students' perceptions of using KWL charts?
    c. What were the observed patterns and frequency use of KWL charts by students?
2. Did my instruction improve when using flowcharts to teach computer programming with LEGO Mindstorms?
    a. What was the influence of using flowcharts on students' performance?
    b. What were students' perceptions of using flowcharts?
    c. What were the observed patterns and frequency use of flowcharts by students?
3. Did my instruction improve when using visual cues to teach computer programming with LEGO Mindstorms?
    a. What was the influence of visual cues on students' performance?
    b. What were students' perceptions of using visual cues?
    c. What were the observed patterns and frequency use of visual cues by students?

Conceptual Framework

Ausubel's (1963) meaningful learning theory served as the conceptual framework for this study. *Meaningful learning* is a process in which learners understand and retain new information by creating relevant concepts between prior knowledge and new knowledge (Ausubel, 1963; Mayer, 2002). Ashburn and Floden (2006) stated that standard instruction, where learners recall and recite knowledge, is inadequate for meaningful learning. Without connecting new knowledge with prior knowledge, learners will likely not remember information or use it in new contexts. Ausubel (1968) insisted that to improve school learning, consideration must be given to different types of learning (e.g., rote learning, meaningful learning). *Rote learning* is a passive approach to learning characterized by repetition, focusing on memorization rather than comprehension, but it does not allow learners to use information in a meaningful way. Rote learning presents knowledge in its final form; no true learning is required, only reproducing information at a later time (Ausubel, 1963; Mayer, 2002).

Hand, Sanderson, and O'Neil (1996) suggested teachers design their instruction to promote meaningful learning. Jonassen, Howland, Marra, and Crismond (2008) suggested that for meaningful learning to occur, lessons should be: (a) *active*-utilize high-order thinking (e.g., observation, manipulation, and analyzing), (b)-*constructive*: reflect on what has been learned from the lesson to allow for connecting new experiences to prior knowledge and results in new understanding, (c) *intentional*-specific learning outcome/goal-oriented tasks, (d) *authentic*-real-world tasks are relevant to students, more memorable, and transferrable to other situations, and (e) *cooperative*- group learning for a natural and productive process of thinking.

This study used project-based learning (PBL) to promote meaningful learning by implementing lessons that focused on factual data but also encouraged students' involvement and investigation. *Project-based learning* is a student-centered instructional technique that employs complex, real-world situations that students can collaborate to solve. Meaningful learning and PBL were used together to try and enhance students' understanding of lessons to improve their grades. Alimisis (2009) suggested that teachers use PBL activities when using educational robotics as a learning tool. Alimisis (2009) noted that the perfect learning situation for PBL is with building and programming educational robots, because students can collaborate on solutions and then build their robot. Han and Bhattacharya's (2001) traits of PBL satisfy the criteria Jonassen et al. (2008) identified as necessary for meaningful learning to occur. They described PBL as having the following characteristics: (a) student-centered, (b) collaborative, (c) curricular content, (d) authentic tasks, (e) multiple means of representation, (f) time specific, and (g) various assessment types (e.g., teacher, peer, self).

Han (2017) advocated PBL as the instructional approach of choice for STEM classrooms because PBL provides students with skills needed in STEM jobs (authentic tasks, problem solving, complex problems). Bell (2010) indicated that PBL allowed students to gain a deeper learning and better understanding of the content because of its student-centered, inquiry-based approach to learning.

Importance of the Study

This action research study provided a systematic way to design and implement custom memory/organization instructional strategies to help students learn computer programming. Teachers have always created, revised, and implemented instructional strategies to teach curriculums' objectives and to enable desired student outcomes (Munyaradzi, 2013). The ability of teachers to evaluate the effectiveness of their strategies is a crucial principle in good teaching (McLeod et al., 2003). The benefits of this study included adding to the limited literature on (a) creating and implementing customized memory/ organization instructional aids, and using memory/organization instructional aids, and (b) improving my ability to create and implement new instructional aids.

This study allowed me to design pertinent instructional strategies specific to my content area. Craig (2009) acknowledged that teachers who conduct classroom research have an opportunity to improve instructional practices, classroom environments, and student learning.

Inquiry into my own teaching has higher relevancy than the traditional mode in which all teachers, regardless of content area, are given and expected to follow the same instructional strategies. Often, I find school-wide professional development counterproductive because it is unrelatable to my curriculum and complicates my instruction. Inapplicable instructional strategies are commonly designed based on main content area subjects (e.g., mathematics, language arts, science, and history). The required use of these inapplicable instructional techniques creates hardship for my students and routinely confuses them and their learning. These suggested but inapplicable instructional techniques included using social media to provide instructions for lessons and grouping students by their lexile score.

Previously, without empirical data to justify an alternative instructional strategy, I was left with the only option of using common, generalized instructional methods in my classroom, regardless of appropriateness. This study provided a practical approach to investigating educational issues and working toward solutions that are relevant to my content area. Using the systematic approach of action research, I analyzed students' strengths and weaknesses and used the results to guide my instruction. I viewed enhancing my instruction as a catalyst to raising students' achievement in learning computer programming and to increasing their enrollment in AP Computer Science.

CHAPTER 2

LITERATURE REVIEW

This action research study determined the effect of various memory/organization instructional aids on the achievement of an urban high school in Augusta, GA students using LEGO Mindstorms robotics. This chapter examines the history of constructivism and constructionism and the shift from teacher-centered instruction to student-centered instruction with educational robotics. This chapter also defines and demonstrates how to assess for meaningful learning theory. This chapter concludes with reviews of three memory/organization instructional aids, KWL charts, flowcharts, and visual cues that were used to create meaningful learning in the classroom.

Teaching Strategies

Instructional strategies are a key element in a teacher's effectiveness (Marzano, 2003). Instructional strategies include a variety of teaching techniques, teaching tools and materials that are used by educators to meet teaching objectives (Ibrahim & Omwirhiren, 2016). Marzano (2003) asserted that successful teachers use efficient instructional strategies and are knowledgeable of various strategies. Ameh and Dantani (2012) stated teaching methods are very important to student success because they can either discourage engagement or promote learning by improving thinking in the classroom. A study by Sandholtz (2011) showed that 53% of preservice teachers acknowledged that ineffective instructional strategies impeded student engagement and understanding attributed this ineffectiveness to an abundance of direct teaching and a lack of hands-on activity (Sandholtz, 2011). Sandholtz (2011) noted that the inclusion of hands-on activity did not guarantee student engagement or student comprehension because students can find that certain activities are boring, fail to emphasize concepts, or are irrelevant. Dewey (1938) identified the importance of teachers developing engaging lessons for students because future experiences are dependent on student engagement. Each student's experience is a catalyst for or against learning, and this effect can only be evaluated by the direction students are moved toward and into (Dewey, 1938). Students' experiences can limit future learning efforts by causing opposition to and a lack of ability to perform in lessons that require effort and persistence. Alternatively, students' experiences can promote learning by encouraging curiosity, initiative, and desire for learning (Dewey, 1938). Sangoleye (2016) suggested that proper teaching strategies can assist students in organizing, remembering, monitoring, and assessing their own learning. Furthermore, VanTassel-Baska (2003) suggested that when selecting an instructional method, a teacher should consider allotted time, students' needs, various techniques, and the effectiveness of each concept for students.

The instructional design process includes selecting and constructing resources to help learners meet their needs to accomplish learning objectives (Rothwell & Kazanas, 2008). Various studies have indicated that students' perceptions are a major influence on what they learned, why they learned it, and how they learned it (Aghamolaei & Fazel, 2010; Bakhshialiabad, Bakhshi, & Hassanshahi, 2015; Marton & Saljo, 1976; Mayya & Roff, 2004). Noe (2008) described perception as the knowledge or understanding of one's surroundings that influences ensuing actions. Aghamolaei and Fazel (2010) and Bakhshialiabad et al. (2015) suggested that students' perceptions could be an important factor in implementing interventions because perception can have a notable effect on students' academic achievement. Gill, Timpane, Ross, and Brewer (2001) defined academic achievement as the satisfactory attainment of academic skill and knowledge that is measured through assessments.

Teacher-Centered

Instructional strategies can be classified into two categories: teacher-centered and learner/student-centered (Sangoleye, 2016). The oldest traditional teaching method used is the teacher-centered approach (Ganyaupfu, 2013; Holt & Kysilka, 2006). This format of teaching is also known as direct instruction, lecture format, didactic, or explicit teaching. It is characterized by lecturing, drilling of facts, and answering questions (Holt & Kysilka, 2006). Teachers often use direct instruction to teach specific skills and concepts that require precision (Rosenshine, 2007; Slavin, 2012). This strategy is often utilized when teaching mathematical calculations, reading fundamentals, and contextual facts (Rosenshine, 2007). A typical direct instruction lesson includes (a) introduction of learning objective, (b) teaching new information or skill by posing leading questions or guided practice, (c) independent practice, (d) checking for understanding through group discussion or an assessment, and (e) providing feedback (Rosenshine, 2007; Slavin, 2012). Students in this particular setting are passive listeners and will occupy their time sitting, copying, memorizing, and repeating the information from the instructor. Cognitive requirements for students are lowest in this situation, since the transfer of knowledge is directly from teacher to student and assignments are catered to average students. This teaching scenario does not stress learning how to do something, but rather the potential to gain new knowledge verbally, spatially, numerically, or interpersonally (Bruner, 1996). Bruner (1960) disagreed with teaching with the average students in mind because instruction should challenge the superior learner while not destroying the confidence of the struggling learner. Moreover, all students are required to start and finish at the same time during a traditional class, but more advanced students may finish early and remain idle in class (Ivic, 2016). Notable disadvantages to teacher-centered instruction include lecturing does not provide feedback to the instructor regarding student learning, requiring that the instructor instead make assumptions. Other disadvantages of teacher-centered instruction include that students can lose focus, information is readily forgotten when students are passive, and students may prefer different learning styles (Schwerdt & Wuppermann, 2011). Bruner (1996) described teacher-centered instruction as a method of presenting facts, principles, and rules of action by lecture or research that are to be learned, memorized, and then applied. Students are seen as “blank slates” vessels into which knowledge is to be poured by the instructor. The main appeal to this method of teaching is that students know exactly what is to be learned, assessments are plentiful, and any academic hardship is based on students lack of abilities (Bruner, 1996).

Student-Centered

Student-centered instruction is based on understanding how students think and how they arrive at their beliefs. Slavin (2009) suggested that teachers cannot simply give students knowledge. Rather, students must develop knowledge on their own, facilitated by the teacher and his or her knowledge, with the chance to discover or apply ideas of their own. The process of prioritizing students as active participants in their learning has become known as student-centered learning, an approach that focuses on students actively participating in their learning through inquiry, problem-solving, and critical thinking through real life experiences, collaboration, and discussion (Arseven et al., 2016; Bruner, 1996; Garrett, 2008).

Constructivism

Constructivism has not universal definition, but some view it as a learning theory, some as an epistemology, and others as a theory of pedagogy (Amineh & Asl, 2015). The learning theory suggest that learners create new knowledge by making connections between prior knowledge and new information, while he epistemological approach deals with how knowledge is created, and the pedagogical approach focuses on how instruction must take students' prior ideas, experiences and knowledge into account while offering opportunities for students to construct new understanding and connections (Anderman & Anderman, 2009). Constructivism involves student-centered concepts on how knowledge is created by linking prior knowledge with new information. Prior knowledge aids the learner in focusing on relevant information and creating links between new information with previous understanding (Gurlitt & Renkl, 2008). Organizing and linking new information allows for long-term memory storage by creating and using association cues for accessing information (Gurlitt & Renkl, 2008). Wetzels, Kester, Merrienboer, and Broern (2011) asserted that information is stored in memory by general ideas about an object (e.g., fruit, banana, green). Furthermore, Wetzels et al (2011) argued that how knowledge is stored in memory dictates how fluently and accurately it can be utilized. Concepts stored in memory are linked with other concepts based on associations. Learning is encouraged and strengthened by repeated use of the knowledge schemes (Wetzels et al., 2011). This approach favors students constructing their learning through activities for more meaningful and enduring knowledge instead of rote memorization (Arseven, Nakiboglu, & OzAydin 2016).

The constructivist learning theme has various methods but some of the most common are (a) *case-based learning*, an institution-based scenario that involves an issue that must be solved by students examining background information and considering possible solutions; (b) *discovery learning*, in which students work freely with little or no guidance on educational tasks; (c) *inquiry-based learning*, in which students use authentic scientific investigation (i.e., formulating scientific questions and hypotheses, conducting experiments and data analysis) to draw conclusions based on evidence; (d) *problem-based learning*, in which students work to solve complex, open-ended problems through collaboration, reflection, and demonstration of understanding by producing a final product; and (e) *project-based learning,* in which students create meaningful artifacts to exhibit their understanding as they try to answer a leading question. The leading questions help students understand the relevance of their task (Anderman & Anderman, 2009; Herreid, 1991). Instructors take a more active role in project-based learning than in problem-based learning. In project-based learning teachers use technology-based supports to aid students in planning, collaboration, and data collection (Anderman & Anderman, 2009). Student-centered instruction utilizes various instructional material, such as assistive technology, to guide or enhance learning (Cubukcu, 2012). For example, an educational robot could be used as a learning tool to create knowledge with robots or as a learning object, creating knowledge about the robot. Constructivism is greatly influenced by the work of Piaget and Vygotsky (Slavin, 2009). Kito (2009) indicated that all these instructional approaches are supported by the educational theory versions of cognitive and social constructivism. Piaget is the primary influence of cognitive constructivism, and his theory is that an individual's experience will determine learning (Fielder & Prince, 2006). Vygotsky's interpretation of social constructivism is based on learning as co-constructed by interactions with others (Fielder & Prince).

Jean Piaget

Jean Piaget was a well-known child psychologist revered for his research on children's cognitive development (Meece & Daniels, 2008). His studies have received much attention in the world of academics because Piaget theorized that children's thinking is developed by a continuous process and formed by interaction with their environment (Anderman & Anderman, 2009; Macmillan, 2003; Ojose, 2008).

Piaget believed that he could assess how children created their understanding of the world by studying their development of thoughts and actions (MacMillan, 2003). Even though children may have different experiences in constructing knowledge, the experience of generating ideas through environmental interaction is universal (Onchawari et al., 2008). Piaget described children's learning using the concept of organizing patterns, or mental schemes, into more sophisticated systems and adapting thinking to fit environmental needs (Meece & Daniels, 2008). He also identified how children adapted to their environment in terms of assimilation and accommodation (Meece & Daniels, 2008). *Assimilation* is constructing new meaning from existing knowledge to make sense of new information. If this new meaning is uniform with the child's current comprehension, then an equilibrium or balance is reached (Meece & Daniels, 2008; Piaget, 1963). However, *accommodation* occurs when children are unable to comprehend new meaning from their interaction with the environment and must alter their existing schemes. This disequilibrium causes new knowledge to be created by accommodation (Onchwari, 2008).

Piaget identified four stages of development that all children experience to acquire knowledge as they interact with their environment: sensorimotor, preoperational, concrete operational, and formal operational. The *sensorimotor stage* is from birth to two years old, during which time babies learn about their environment by interacting with it by using their senses and motor skills (Slavin, 2009). The *preoperational stage* is from two to seven years old, during which period the children's vocabulary grows, and they learn to recognize symbolic symbols, and understand logical operations in one direction (Woolfolk, 2005). The c*oncrete operational* is the third cognitive development stage, occurring from ages seven to 11, during which children begin to utilize logical reasoning with physical objects (Santrock, 2008). Piaget based this stage on concrete objects because he believed children could only harness operational schemes regarding objects, situations, or events that are tangible or imaginable (Shaffer, 1996). The ability to use relational logic with abstract identifiers (e.g., algebraic expressions) for children aged seven to 11 is not yet applicable (Shaffer, 1996). This stage employs operations on objects: (a) *seriation*, ordering objects along a continuum based on size; (b) *classification*, determining the relationship between objects' characteristics; (c) *space*, determining relationship based on proximity; and (d) *time and speed*, determining a relationship between the trajectory of an object and its distance (Piaget, 1969). Through differentiation of objects in the concrete operational stage, children's ability to base assumptions on facts is now possible (Piaget, 1969). This form of thinking is the beginning of hypothetico-deductive reasoning or formal thought. *Hypothetico-deductive reasoning* is a systematic problem-solving technique that explores all possible scenarios by evaluating all variables in a task (Shaffer, 1996). Even though hypothetico-deductive reasoning is available in the concreate operational stage children still lack the ability to systematically evaluate all variables or possibilities of a task to successfully identify the proper solution (Shaffer, 1996).

Piaget and Inhelder (1958) demonstrated the limitation of hypothetico-deductive reasoning, during the concreate operation stage, in their *Oscillation of a Pendulum and the Operations of Exclusion* experiment. In the experiment children were instructed to create a pendulum by suspending a weight using string. The objective was to identify which variable (e.g., length of string, weight of object, force of push, or height of drop) controlled the frequency of the oscillation (Meece, 2008). The experiment showed that concrete-operational thinkers can use problem-solving strategies and test hypotheses, but they fail to use a systematic approach to reach the proper solution. The concrete-operational thinkers lack a *combinatorial system*, or the ability to analyze and isolate all combination of variables in a task to find the correct answer (Piaget, 1969). This experiment had 16 different combinations to evaluate, and the answer was that the length of the string influenced oscillation regardless of the other three variables (Meece, 2008). The solution could have been reached by isolating one variable to test while keeping the other three constant (Meece, 2008). Concrete-operational thinkers fail at evaluating multiple variables because of the inability to isolate variables (Meece, 2008).

The *formal operational stage* is from 11 years old to adulthood. During this stage, children begin to acquire the ability of logical reasoning (e.g., problem solving, inductive and deductive reasoning). This stage highlights formal-operational thinkers' ability to routinely reason hypothetically and deduce consequences from their hypotheses (Piaget, 1971). Formal-operational thinkers are able to think abstractly due to four key traits: (a) *propositional logic*, or drawing conclusions based on two logical statements; (b) *hypothetico-deductive thinking*, or generating and testing hypotheses in a systematic manner; (c) *combinatorial reasoning*, or an organized ability to represent all possible combination of a task to discover the proper solution; (d) *probability*, is the calculated chance something might occur, and (e) *proportions*, which is the ability to work with a subset or part of a whole (Meece, 2008; Piaget, 1969). Piaget and Inhelder's (1958) pendulum experiment showed the difference between the concrete and formal stage. Formal operational thinkers would assess all 16 possible combinations of the pendulum's variables by modifying one variable at a time (Meece, 2008; Slavin, 2012). Piaget (1971) later theorized that the formal operational stage is not an actual stage, and not all children automatically attain formal-operational thinking. Only by specialized training or by personal aptitude does a person reach the formal operational stage (Piaget, 1971).

Piaget's learning theory is commonly known as cognitive constructivism or constructivism (Khan, 2013) and is characterized as students constructing and reconstructing their own interpretations of their environment instead of remaining passive and merely absorbing information from their teachers (Fielder & Prince, 2006). Thus, children use mental concepts or schemes to try comprehend the new information by incorporating prior knowledge with new information (Fielder & Prince, 2006).

Other contributions from Piaget came from his book *The Psychology of the Child* (1969). One assertation Piaget (1969) made in this book was that intelligence was a prerequisite for various other skills. For example, Piaget (1969) identified that intelligence made assimilation, play, drawing, memories, and language possible. Intelligence allows for development and organization of these skills (Piaget, 1969). Moreover, in his book, Piaget (1969) identified three forms of play:

(a) *exercise play*, a form of play that is performed at the sensorimotor level and is defined by the child's repetition as he or she is learning fine motor skills; (b) *symbolic play*, which is a child's way of playfully assimilating or imitating reality to the self; and (c) *games with rules*, which include social activities e.g., marbles, hopscotch, etc.

Lev Vygotsky

Lev Vygotsky was a social constructivist who studied child development and believed that knowledge is co-constructed by interaction with people and their environment (Anderman & Anderman, 2009; Meece & Daniels 2008). In his book *Thought and Language*, Vygotsky (1962) indicated how language influenced mental development. He suggested that the learning of words was a prerequisite for several intellectual functions e.g., deliberate attention, logical memory, abstraction, and comparison and contrast. During early childhood, understanding a new words is often imprecise because of the child's early mental state (Vygotsky, 1962). However, as the child's cognitive abilities improve, new words acquire realistic meaning. Children can then solve practical activities with the assistance of speech (and eyes and hands) because speech and actions coexist (Vygotsky, 1962). A complex task forces the child to reevaluate the situation and search verbally for a new plan (Vygotsky, 1962), whereas speech is seen as a tool for the child to employ (Vygotsky, 1962).

Vygotsky also theorized that collaborative-learning could intensify students through collaborative learning (Powerll & Kalina, 2009). Vygotsky (1978) disagreed with the traditional psychological notion that development preceded learning. Instead, he suggested, development followed learning because learning enabled internal development processes facilitated by working with others in one's environment. Thus, the development achievement occurs once these internalized processes are apparent.

In Vygotsky's (1978) *Mind in Society: The Development of Higher Psychological Process*, he theorized that the current or actual level of the learner's development and the potential level of development could be attainable by guidance from a more knowledgeable person and the use of semiotic tools (Shabani et. al., 2010). Vygotsky (1978) suggested that assessments of children's development should emphasize tasks that the child can complete without demonstrations, leading questions, and/or other forms of assistance. Vygotsky introduced the concept of *Zone of Proximal Development* (ZPD), which comprises the abilities that a child does not yet possess but which, with assistance, can be learned. He theorized that the tasks that the child can complete independently will ultimately determine his or her level of ability (Vygotsky, 1978). He further suggested that cooperative learning within the ZPD helped children increase their abilities (Meece & Daniels, 2008). Furthermore, Vygotsky (1978) stated the following assumptions regarding the role of play: (a) play derives initially from real situations; (b) as play develops it becomes more purposeful and should not be seen as an activity without cause; (c) at the final stage of development, rules and activities become more specific and demanding for children; and (d) the ZPD is utilized because children often assume roles that are above their given abilities. Vygotsky's works have in "early-literacy" programs such as *Reading Recovery* and *Guided Reading* (Blake & Pope, 2008).

Using Piaget's and Vygotsky's Theories in the Classroom

Dewey (1938) stated that a crucial responsibility of teachers is to understand what educational experiences are needed by students to induce learning. Teachers often customize lessons to students' strengths and weaknesses (Bruner, 1960). Students do not learn when content is trivial or too difficult, so the essential principle is to find a medium with which to guide students' learning (Bruner, 1960; Slavin, 2012). Children are viewed as having an unorganized and spontaneous understanding process, but with guidance from a more knowledgeable person, children can develop organized and systematized thinking (Santrock, 2008). Even though Piaget and Vygotsky had different ideologies, studying their works can improve teachers' instructions and students' achievement (Blake & Pope, 2008), and their principles can be implemented in classrooms in various ways. Table 2.1 identifies differences between the ideologies of Piaget and Vygotsky.

Table 2.1

Comparison of Piaget's and Vygotsky's Ideologies

	Piaget	Vygotsky
Constructivist style	Cognitive constructivism	Social constructivism
Learning style	Individually	Socially
Hands-on learning	Advocated	Advocated
Role of language	Thought proceeds language	Language proceeds thought
Learning and development	Development before learning	Learning before development
Play	Three forms of play Assimilation/make-believe	Leading factor in developing abstract thinking
Major contribution	Four stages of cognitive development	Scaffolding ZPD

An example of Vygotsky's ideology of social learning is shown in the study by Wood, Bruner, and Ross (1976). Wood et al. (1976) demonstrated the benefit of providing the appropriate amount of support to guide learners while accomplishing a task previously unattainable because it was just above the learners' abilities. They worked with three, four, and five-year-old children to try and teach them to construct a three-dimensional wooden pyramid structure using interconnecting blocks, which required a level of expertise initially beyond the children's abilities. Wood et al. (1976) studied 30 children in individual sessions ranging from 20 minutes to one hour. The task was designed to be entertaining, challenging, and comprehensive, but not too far above learners' abilities. A wooden pyramid was to be constructed from 21 interconnecting blocks to stand approximately nine inches tall, with a nine-inch square base. The wooden blocks connected via a system of holes and pegs on each block and could only be assembled by putting the correct pairs together with the right orientation.

Interventions included (a) direct assistance, (b) verbal error prompt, and (c) straightforward verbal attempt (e.g., "Can you make more like this?"). Wood et al. (1976) concluded that the youngest children benefitted from being guided to learn through recognition of correct solutions. Moreover, they showed that all children in the study benefitted from (a) *reduction in degrees of freedom*, or lowering the level of complexity; (b) *marking critical features*, or accentuating critical parts of the task to allow for recognition; and (c) *direction maintenance*, keeping the learner engaged and monitoring progress.

Shooshtari and Mir (2014) explored the use of mentoring with peer learning. They studied 30 university students, divided into two groups with 15 members each, which they then subdivided into three groups of five members. Shooshtari and Mir (2014) used an argumentative writing assignment as the pretest. Pretest results between groups were measured with the t-test and showed that both groups were not significantly different before treatment. Students were both given an argumentative topic to write on as posttest. The G1 group received constant instructional support in the form of prompts and hints, while the G2 group was given random instructional support. A t-test was used to measure the probable difference in writing quality between the two groups. The results of the posttest showed that G1 made significant progress in writing quality and strategy use. Shooshtari and Mir (2014) suggested that teachers can guide students' learning by employing student-centered learning activities with teacher support using prompts and feedback.

Piaget's continue to influence education and education research even today (Anderman & Anderman, 2009; Meece & Daniels, 2008; Ojose, 2008). Bigge and Shermis (2004) suggested that Piaget's study of the stages of children's learning can benefit teachers greatly. For example, studying his works can help teachers understand children's thinking at different stages (Meece &

Daniels, 2008), thereby allowing a teacher to align the appropriate instructional strategies with students' cognitive level (Blake & Pope, 2008). Meece and Daniels (2008) stated that Piaget's works form the logical foundation for constructivist, discovery, inquiry and problem-oriented teaching in the classroom. Hoy (2007) agreed with Piaget's assumption that some students remain in the concrete operational stage throughout high school and adult life. Hoy (2007) and Santrock (2008) suggested strategies for working with concrete operational thinkers: (a) discovery learning, (b) group work, and (c) physical props and visual aids. Hoy (2007) stated that many adolescents will have difficulty thinking hypothetically and suggested using concrete-operational teaching strategies and materials (e.g., tangible props and visual aids) to aid students in formal-operational thinking.

Constructionism

Constructionism is a learning theory based on Piaget's constructivism principles that emphasizes learning by constructing meaningful and tangible items in a social environment (Papert, 1993; Stager, 2001, 2003). Stager (2001) asserted that constructivism could be viewed as "learning by doing", while constructionism could be seen as "learning by making". Thus, constructionism focuses on utilizing technology, and particularly computer technology in learning (Stager, 2001).

Papert (1980) originated constructionism and worked with Piaget for five years at Piaget's Center for Genetic Epistemology. Papert utilized the programming language LOGO to teach children to program and move a basketball-sized robotic turtle by given commands on a computer screen (Koh & Loh, 2014; Papert, 1980). For example, students could move the turtle 100 steps by typing the command FD 100 which stands for "forward 100" (Koh & Loh, 2014). LOGO was a programming language designed to aid teaching mathematical and logical strategies (Feurzeig & Papert, 2011). *Programming* is communicating with a computer in a language that it understands (Papert, 1980). Papert (1980) viewed a child programing a computer with LOGO as a child exploring his or her own thinking by teaching the computer how to act. Moreover, programming requires reflection on the children's part because, when learning programming, nobody ever gets the program correct the first time. As a result, becoming proficient in programming requires reflection on one's programming and on correcting programming errors. Thus, students must predict, plan, and consider sequencing in their program (Geist, 2016).

LEGO Mindstorms EV3

LEGO Mindstorms EV3 was the chosen educational robot for this study because of its cross-curricular opportunities, versatility, and ease of use. Arlegui et al. (2009) recommended that an educational robotic platform should have the following characteristics to maximize its suitability (a) programmable at various difficulty levels and supports different programming paradigms, (b) adaptable to provide rigorous learning experiences at diverse educational levels (i.e., different ages), and (c) capable of simple expansion to provide additional learning possibilities. Based on these recommendations, Arlegui et al. (2009) suggested using the LEGO MINDSTORMS kit because it satisfies these criteria and has other advantages. LEGO Mindstorms' most important benefits are its quick start-up time, lack of need for special tools or electrical wiring for assembly, easily recognizable since many students played with LEGO bricks growing up. This familiarity and ease can motivate students' engagement, since using LEGOs can be seen as recreational rather than a task. LEGO Mindstorms also supports constructionist learning and offers a plethora of free resources of learning activities, discussion groups, and technical support online (Arlegui et al., 2009)

The curriculum, *Introduction to Programming EV3*, created by the *Carnegie Mellon Robotics Academy,* was used with LEGO Mindstorms EV3. This curriculum was chosen based on its cross-curricular characteristics, comprehensive lessons, and assessments. It contains standards from *Common Core*, *Next Generation Science* (NGSS), and the *Computer Science Principles Framework*. Appendix A shows the correlation between educational standards and the Introduction to Programming EV3 curriculum by pairing educational standards with tasks from the Introduction to Programming EV3 curriculum.

The Common Core standards included are Mathematics Practice, Mathematics Content, and English Language Arts. Common Core standards provide guidelines on precise learning goals to help prepare students for college, career, and life (Common Core, 2018). These standards identify what students are expected to learn at each grade level from K-12. Moreover, the standards are research and evidence-based, aligned with college and job expectations, and influenced by top-performing countries to ensure students are prepared for success in a global economy and society (Common Core, 2018). The Mathematics Practice standards covered include problem-solving, abstract reasoning, modeling, and precision. Mathematics Content standards addressed include understanding ratios, ratio relationships, and problem-solving with ratios. The English Language Arts standards covered focuses on developing coherent writing appropriate for the task, purpose, and audience.

NGSS standards were created by various states to try and improve science education by implementing research-based and up-to-date K-12 standards (NGSS, 2018). Expectations were set for all students that detail what they should know and be able to do (NGSS, 2018). These standards are developed around three domains (a) *Crosscutting Concepts* – exploring the relationship between Physical Science, Life Science, Earth and Space Science, and Engineering

Design; (b) *Science and Engineering Practices* – describing the scientific inquiry process, its requirements, and engineering design process; (c) *Disciplinary Core Ideas* – principle concepts in science that are used to aid in creating instruction and assessments (NGSS, 2018).

The Computer Science Principles Framework identifies concepts and computational thinking practices that are the foundation of computer science principles. The computational thinking practices aid students' ability to understand and apply the concepts needed in analyzing data and developing computational artifacts (College Board, 2017). Each computer science standard correlates directly to one of the six computational thinking practices. For example, [P1] represents a correlation to Computational Thinking Practice 1: Connecting Computing. It is based on the *Understanding by Design* model by Wiggins and McTighe (2005). Wiggins and McTighe's (2005) model is based on six principles (a) *Can explain* – moving from general, true principles to a specific conclusion, (b) *Can interpret* – induction reasoning/moving from specific instances to a general conclusion, (c) *Can apply* – adapting what has been learned in context, (d) *Have perspective* – developing a viewpoint based on assessment, (e) *Can empathize* – perceiving others feelings, and (f) *Have self-knowledge* – understanding how knowledge or lack-of-knowledge reflects on the learning experience.

In the mid-1980s, the LOGO research group began to collaborate with the LEGO group to create tangible objects to be programmed by the LOGO language (Ucgul, 2013). Children could construct various objects, such as a Ferris wheel, elevator, and robot from an assortment of parts that included beams, gears, and motors. The first computerized LEGO products were released in 1986. In 1998, the LEGO MINDSTORMS RCX Intelligent Brick and Robotics Invention System were put on exhibit. The RCX brick was an 8-bit 16MHZ microcontroller with 32KB RAM. LEGO would release three more iteration of its computerized robotic system:

2006 LEGO MINDSTORMS NXT, 2009 LEGO MINDSTORMS NXT 2.0, AND 2013 LEGO MINDSTORMS EV3. The EV3 currently has a 300 MHz micro controller with 64 MB of RAM and supports various sensor attachments, input/output ports, and wireless connection.

Programming exposes students to computational thinking (Lye & Koh, 2014). Wing (2006) invented the term computational thinking to describe programming. *Computational thinking* is a recursive type of thinking utilized to solve problems, design systems, and understand human behavior using computer science principles (Wing, 2006). In addition to these benefits, Geist (2016) suggested teaching young children programming because computers are a ubiquitous part of everyday life, occurring in everything from refrigerators and coffee pots to clothing. Geist (2016) also indicated three keys to developmentally appropriate programming developmentally appropriate for young children: (a) *experimentation and exploration*: expose children to basic programming concepts with "drag and drop" and text-entering environments; (b) *construction and creation*: children begin to develop simple programs to demonstrate an action on screen or make a robot perform and action; and (c) *problem solving*: children exhibit their understanding of programming by creating a program for a specific task.

Robotics

Educational robots are utilized in various roles in education because of their unique ability to serve as learning tools/teaching aids (Murbin, Stevens, Shahid, Mahumd, & Dong, 2013). Cejka et al. (2006) suggested that students can learn about and become comfortable about technology through robotics. The implementation of educational robots will depend on the subject area, the instructor, the type of students, and the learning objective (Murbin et al., 2013). Advancements in technology, specifically in robotics, have created new opportunities for learning activities such as peer social interactions, creativity, and cognitive development (Bers &

Kazakoff, 2014). Since the subject of computer science is ubiquitous in today's society, it is considered the ideal subject, in both schools and universities, for learning fundamental concepts about computers, the internet, and modern technologies. However, computer science classes have a difficult time attracting and retaining students because of the subject's complex concepts, such as data structures, algorithms, and syntax (Burbaite et al., 2013; Alvarez & Larranaga, 2014). As a result, many researchers have advocated the use of visual programming environments and robots to teach computer science because of their ability to lessen the computational thinking required by students. Papert's popular book *Mindstorms: Children, Computers, and Powerful Ideas* led to the invention of the programmable LEGO Mindstorms robotic kit in 1998 (Beland et al., 2000). Alvarez and Larranaga (2014) suggested that LEGO robots might aid students learning basic programming concepts needed to be successful in a programming class.

Using educational robots and constructivist methods students can learn the complex subject of computer programming (Pasztor, Pap-Szigeti, & Torok, 2010). Equally important, using educational robots involves more than just assembling robots. It also requires students to be able to computer program (Bers, 2010). Through programming children must reflect on their own learning because in order to allow the robot to interact, explicit instructions must be constructed. This design process will allow teachers to observe what and how students learn (Cejka et al., 2006). With the skills in algorithm design, problem solving, and computer programming abilities they learn through such activities, students will have more opportunities to obtain technology related positions. Pasztor et al. (2010) advocated LEGO Mindstorms as a suitable tool for teaching programming because of its tangible, practical nature and versatility to a variety of students' skills and age. LEGO Mindstorms robots and their visual programming environment provide an efficient and practical way for students to learn programming with inquiry about motors, sensors, and automations. For example, students learn concepts of programming by constructing programming procedures, debugging programs, and solving problems. Deitrick, Sanfor, and Shapiro (2014) suggested that if teachers create links between introductory programming and more advanced programming, this will aid in students maintaining long-term interest programming.

Many teachers in K-12 have had difficulties teaching with robots because they have little or no experience and support with engineering and robotics. Participating in constructionist robotics engineering design challenges can frighten teachers and cause them to avoid teaching using educational robots before they have even tried to do so (Cejka et al., 2006). Snoeyink and Ertmer (2001) identified the lack of specific technology-based knowledge, skills, and pedagogy as significant reasons teachers are not implementing new technologies in the classroom. Teachers with limited understanding of new technology will hinder learning experiences because of the difficulty of developing lessons, supporting students, and assessing students (Stein, McRobbie, & Ginns, 1999). Incorporating new technology in the classroom requires a transition from teacher-centered instruction to student-centered introduction and new assessment practices (Snoeyink & Ertmer, 2001). Professional development for implementing new technology must extend beyond teachers' understanding of new technology to include comprehension of why the changes in their traditional practices are necessary and their potential benefits (Stein et al., 1999).

Robotics Curriculum

Papanikolaou and Frangou (2009) suggested that a robotics curriculum should implement the main principles of constructivism, constructionism, and problem-based learning. Doing so will provide lessons with (a) *authentic learning*, learning using real-life or occupational scenarios; (b) *social learning*, learning through collaboration with classmates; (c) *meaningful-active-reflective learning*, learning through experiments or problem solving utilizing resources of students' own interest; and (d) *problem-based-learning*, learning using critical and analytical thinking by investigating real world issues through a technology-based framework. For example, building and programming robots is the perfect task to harness project-based learning, social learning, and critical thinking skills. Mihardi et al. (2013) indicated that Piaget's and Vygotsky's theory on constructivism is directly in support of project-based learning. Han and Bhattacharya (2014) identified seven crucial parts of project-based learning: (a) *learner-centered environment*, in which students are active participants in their learning; (b) *collaboration*, which offers opportunities to learn social skills e.g., group decision making, interdependence, integration or peer and mentor feedback; (c) *curricular content*, which offers activities based on standards, clearly states goals, and supports and gives examples of content learning from the beginning to the end; (d) *authentic tasks*, i.e., real life or occupational scenario; (e) *multiple expression modes*, i.e., effective use of various technologies; (f) *emphasis on time management*, offering adequate time for iteration of planning, revision, and reflecting on learning; and (g) *innovative assessment*, in which assessment is an ongoing process.

The advantages of project-based learning include increased motivation, problem-solving ability, media research skills, collaboration, and resource management skills. Nevertheless, although project-based learning offers many advantages, there are a few disadvantages to the method as well, including potential difficulty for students to use robots as learning tools because they have difficulty in deciding what strategies to use, how to start, and how to proceed with the issue. Moreover, teachers may not be knowledgeable of the appropriate pedagogy, and assessing a student's individual contribution may be complicated (Han & Bhattacharya, 2014; Papanikolaou & Frangou, 2009).

This study utilized the Introduction to Programming LEGO Mindstorms EV3. This curriculum was chosen because it satisfies the four requirements of a robotics curriculum indicated by Papanikolaou and Frangou (2009) and the seven principles by Han and Bhattacharya (2014). Table 2.2 indicates the blend of instructional strategies (e.g., diagrams, videos, questions, and mini challenge) used in each cycle.

Table 2.2

Criteria for a Constructivist, Constructionist, and Project-Based Learning Curriculum

	Sensabot	Orchard	Arm Position
Authentic learning	Simulation of using a Sensabot	Simulation of a harvesting machine picking fruit	Simulation of sensors on machines
Social learning	Group activity	Group activity	Group activity
Meaningful-active-reflective learning	Experiment with programming a robot	Experiment with programming a robot	Experiment with programming a robot
Problem based learning	Students must program the robot to accomplish the given task	Students must program the robot to accomplish the given task	Students must program the robot to accomplish the given task
Learner-centered	Learner active participant	Learner active participant	Learner active participant
Collaboration	Group activity	Group activity	Group activity
Curricular content	STEM	STEM	STEM
Authentic task	Simulation of using a Sensabot	Simulation of a harvesting machine picking fruit	Simulation of sensors on machines
Expression modes	Internet, computer, and robot	Internet, computer, and robot	Internet, computer, and robot
Time	Time specific	Time specific	Time specific
Assessment	Observation, interviews, formal	Observation, interviews, formal and summative assessments	Observation, interviews, formal and summative assessments

and summative assessments

Meaningful Learning Theory

Rote Learning vs. Meaningful Learning

Ausubel (1968) insisted that to improve school learning, consideration must be given to different types of learning (e.g., rote learning and meaningful learning). *Rote learning* is a passive learning style by repetition that focuses on memorization rather than comprehension but does not allow the learner to use the information in a meaningful way (Ausubel, 1963; Daunoriene & Katiliute, 2011; Iqbal & Ahmad, 2015; Mayer, 2002). Ausubel (1963) indicated that in rote learning knowledge is given in its final form, and no learning is required, only reproducing the information at a later time. Thus, rote learned information is subject to short-term memory because it is stored arbitrarily and verbatim in memory (Ausubel, 1963; Daunoriene & Katiliute, 2011; Karpicke & Grimaldi, 2012). An example of rote learning is a student reading a textbook to memorize key facts and terms, but who is unable to analyze information to answer open-ended questions (Mayer, 2002). Nevertheless, Smelley (2013) stated that rote learning can be beneficial and may lead to improvements in verbal/episodic memory (Iqbal & Ahmad, 2015). In contrast, *meaningful learning* is a learning process which the learner understands and retains new information in memory by creating relevant connections between prior knowledge and new knowledge (Ausubel, 1963; Mayer, 2002). One example of meaningful learning is a student reading a textbook for memory and understanding and being able to interpret the information to provide solutions to open-ended problems (Mayer, 2002). Ausubel (1963) indicated that two important criteria are important in deciding if new learning will be meaningful:

1. New information must be logically relatable to prior information, and
2. The learner must understand the relationship between the new information and prior knowledge.

Meaningful learning is an important aspect of education because students are active participants in their learning (Mayer, 2002). However, Ausubel (1963) noted that rote learning can be meaningful without prior knowledge or problem solving. Equally important, Ausubel and Robinson (1969) identified four types of meaningful learning:

1. *representational learning* – learning meaning of symbols,
2. *concept learning* – grouping of similar items due to a common characteristic,
3. *proposition learning* – generalizations, and
4. *discovery learning* – the setting where the criteria to be learned is not readily available, so active engagement of problem solving and critical thinking is required by the learners.

Entwistle, McCune, and Walker (2001) noted that learners will naturally try to identify patterns.

Deep Learning vs. Surface Learning

Like Ausubel, Marton and Saljo studied students learning. Marton and Saljo (1976a, 1976b) studied the qualitative differences of students' content learning and steps taken to promote learning suggesting that understanding how students think would be useful in teaching Thirty college students were asked to read a newspaper article on school reform in Swedish universities, knowing follow-up questions would be asked. Students were asked questions individually, and responses were recorded and transcribed verbatim. The questions focused on students' ability to remember and comprehend the content of the article and the steps used to complete the task. Students were asked questions such as: (a) Could you describe how you went about reading the text? (b) Was there anything that you found difficult? (c) Did you find it interesting? and (d) While reading, was there anything that struck you as particularly important? Marton and Saljo (1976a, 1976b) identified two opposing techniques of learning based on students' answers: *surface-level processing* and *deep-level processing*. Their study (1976a) showed that student comments reflected an emphasis on memorization:

> *"Well, I just concentrated on trying to remember as much as possible."*
>
> *"I remembered … but, I'd sort of memorised everything I'd read…no, not everything, but more or less."*
>
> *"It would have been more interesting if I'd known that I wasn't going to be tested on it afterwards,' cos in that case I'd've more, you know, though about what it said instead of all the time trying to think now I must remember this and now I must remember that."*
>
> *"There were a lot of different lines of thought to follow and to try and memorise."* (p.9)

These comments reflect an emphasis on *surface-level* thinking, which was described as relying on rote learning instead of comprehension of information to answer questions. Marton and Saljo's (1976a) study also identified another level of processing, *deep-level* learning. Students' comments reflected that students focused on understanding the content:

> *". . . I tried to look for . . . you know, the principal ideas…"*
>
> *". . . and what you think about then, well it's you know, what was the point of the article, you know."*
>
> *Interviewer: "But when you were going to start, you tried to think: what came at the beginning of the article?"*
>
> *Subject: "No, I . . . I tried to think what it was all about . . ." ". . . I thought about how he had built up the whole thing."* (p.9)

This emphasis is looking for underlying patterns and attempting to discern "the point" or deeper meaning points to a "deep" approach to learning, in contrast to the surface level discussed above.

From Marton and Saljo's (1976a) earlier study, research on deep and surface learning has expanded. Entwistle and Ramsden (1982) suggested that a learner could adopt a deep approach and still fail to reach a deep understanding because a lack of prior knowledge or lack of effort. They asserted that a surface style of learning will never help students attain a deep understanding because it lacks the traits of deeper learning (e.g., relating prior knowledge to new knowledge, intention to understand the article, and reflection). Ramsden (1992) suggested that all students are capable of both deep and surface learning, but learning intention plays a major role in the approach. He also stated that surface learning appears differently in various subject areas, but the results of this learning will consistently be lower academic grades compared to deep learning. Studies by Wong and Watkins (1998), Cano (2007), and Hasnor (2013) show an inverse correlation of students' achievement and use of a surface approach. Students who used a surface learning technique had lower grades, while those who used a deep approach had higher grades. Table 2.3 identifies these three studies and their findings in more detail.

Table 2.3

Correlation Studies on Deep and Surface Learning and Academic Achievement

Researcher	Purpose	Participants	Method	Statistical Analysis	Implications
Wong & Watkins (1998)	Add to the literature on students' learning.	356 ninth graders	3 Likert scales and 42 multiple choice questions	Correlation coefficients to math learning at the end of the term: Surface strategy = -.22 and deep strategy = .13	Deep learning associated with greater learning.
Cano (2007)	Study learning approaches.	572 high school students	Biggs (1987) *Learning Process Questionnaire*	Learning approach correlation to academic achievement: Surface: M = -.47, SD = .87. Deep: M = .62, SD = .96. One-way ANOVA	Deeper learning associated with higher student grades than those who used surface learning.
Hasnor, Ahmad, Nordin (2013)	Study learning approaches.	233 college students	*Approaches and Study Skills Inventory for Students* questionnaire	Learning approach correlation to academic achievement: Surface = -0.213 Deep = 0.388	The more students used surface learning, the lower their scores.

In contrast to surface learning, which Ramsden (1992) described as quantitative learning that emphasizes rote learning of random facts and does not allow for analysis of the information, deep learning permits qualitative learning in which students interpret and apply of knowledge to reach a logical conclusion. Entwistle (1997) indicated that the quality of teaching can influence students' learning approach. Claiming that when teachers' lessons were relevant to students' experience, they encouraged deep learning. Pellegrino and Hilton (2012) described deep learning as the process of transferring knowledge to other situations. *Transferable knowledge* is the ability to answer questions and solve problems, apply knowledge in other situations, and understand how, when, and why to use the knowledge (Pellegrino & Hilton, 2012). Martinez, McGrath, and Foster (2016) identified transferable knowledge as being important because it enables learners to be flexible, creative and innovative in future classrooms, jobs, and an ever-changing world. Transfer of knowledge is supported by instruction that promotes deep learning but is not supported by rote learning (Pellegrino & Hilton, 2012).

Entwistle, McCune, and Walker (2001) emphasized the similar correlations of deep and surface learning to Ausubel's rote and meaningful learning. They suggested the influence of teaching approaches: (a) surface learning is teacher-focused, content-oriented, and emphasizes the reproduction of facts, whereas (b) deep learning is student-centered, promotes student engagement, and encourages conceptual development. Table 2.4 compares deep learning and surface learning.

Table 2.4

Traits Comparison of Deep Learning and Surface Learning

Deep learning	Surface learning
Information stored in long-term memory	Information stored in short-term memory
Patterns created in memory by connecting new information with prior knowledge	Information stored randomly
High correlations to better grades compared to surface learning	Strong correlation to lower grades compared to deep learning.
Transfer of knowledge possible	Transfer of knowledge not possible
Motivated to learn (intrinsic)	Not interested in content subject
Understands information	Reproduces information without understanding
Goal is for comprehension	Goal is to meet minimum requirements

Teaching for Deeper Learning

Hand, Sanderson, and O'Neil (1996) proposed that instruction be designed to promote deeper learning because course design will influence the learning approach of students. A course that focuses on factual information will sway learners to use surface learning. Students will not understand what they are reading when they concentrate on information they assume will be on the test instead of searching for the meaning of the text (Rehm, 1995) To counter students' tendencies toward surface learning, Smith and Colby (2007) suggested that educators utilize instructional strategies that support students' opportunities to develop deeper learning.

Like Hand et al. (1996), Rehm (1995) noted the importance of task requirements and their influence students' perception of the learning approach needed. He offered four principles to encourage deep learning: (a) *motivational context* encourages students to want to learn, (b) *learner activity* engages students in the learning process instead of being passive, (c) *collaboration* allows enrichment the learning experience that cannot be duplicated individually, and (d) *well-structured knowledgebase* allows for associating new knowledge with prior knowledge to aid in understanding.

Carmean and Haefner (2002) studied body of research on deeper learning and identified five overlapping and significant concepts that can be utilized to create a meaningful learning experience. Although, they suggested that these principles were the basis for deeper learning, they suggested that it is not necessary for all characteristics to be present: (*a*) the *social* element encourages collaboration and useful, timely feedback between students; (*b*) the *active* element continuously engages students engaged in the learning process by offering prompt feedback to students' responses; (c) the *contextual* element involves relating students' prior knowledge to new knowledge by using real-world problem solving activities; (d) *engaging* students appeals to diverse learning styles, challenges students, and encourages authentic curiosity; and (e) *student-owned* learning occurs when learners take control of their learning process by demonstrating independence through initiating actions of planning, analysis, and reflection on a task.

Biggs and Tang (2011) described how surface learning is encouraged when teaching and assessment are not aligned. Table 2.5 compares elements that influences surface learning or deep learning.

Table 2.5

Influences on Learning According to Biggs and Tang (2011)

Surface Learning		Deep Learning	
Student	Teacher	Student	Teacher
Intention to meet minimal requirement to pass	Not relating lesson to students	Uses meaningful learning	Linking prior knowledge with new knowledge
Not understanding requirements	Assessing for independent facts; often result of short answers and multiple-choice	Appropriate background knowledge and well-structured knowledge base;	Correcting misconceptions
Unable to process content at a deeper level.	Promoting cynicism: "I hate teaching this section"	Intrinsic motivated	Assesses for understanding instead of independent facts
Not interested in the subject topic or of the teaching of the content High anxiety	Not spending enough time on content for understanding		Instruction that allows for reflection by students and clearly states observable outcomes.

Pellegrino and Hilton (2012) described ways teachers can teach for deeper learning by implementing: (a) multiple means of representations such as graphic organizers; (b) encouraging elaboration, questioning, and explanation; (c) engaging learners in challenging tasks; (d) teaching with examples and cases; (e) priming student motivation; and (f) using formative assessment.

Martinez et al. (2016) suggested ways teachers can implement deeper learning in their classrooms by (a) providing *student-centered lessons* with opportunities for students to work with their peers, collect feedback, make revisions, and reflect on the learning process aids in understanding that learning is a complex process; (b) *using instructional strategies to focus thinking* , such as guiding questions, common themes, and main ideas, to help students by putting information in context; (c) offering *relatable, real-world lessons* so students can experience workplace situations, allowing for real-world challenges; and making learning meaningful; (d) creating opportunities for *learning outside of school*, such as independent projects and internships, to provide relatable, real-world experiences; (e) using *open-ended lessons* to help students demonstrate creativity and pursue their own learning, and (f) *using technology as a learning aid* to support learning by allowing students to create knowledge by using technology such as using the internet to post a video to reflect, using Google Docs to collaborate and organize research, or downloading information for research. Of the five studies on teaching for deeper learning, common characteristics are evident such as collaboration, student-centered, motivational, real-world, knowledge structure and reflection. Table 2.6 shows a comparison of components that influence deeper learning techniques from various studies.

Table 2.6

Comparisons of Studies on Promoting Deeper Learning

Rehm (1995)	Carmen & Haefner (2002)	Biggs & Tang (2011)	Pellegrino & Hilton (2012)	Martinez et al. (2016)
Motivational content	Social	Linking prior knowledge with new knowledge	Multiple and varied representations	Student-centered
Learner activity	Active	Well-structured knowledge base	Collaboration	Instructional strategies
Collaboration	Contextual	Motivational	Challenging tasks	Relatable, real-world lessons

Well-structured knowledge	Engaging	Student reflection	Motivational	Learning outside the classroom
	Student-owned	Meaningful learning	Uses formative assessments	Open-ended lessons
			Teaching with examples and cases	Use of technology

Measuring Depth of Knowledge

Krathwohl (2002) suggested that taxonomies of educational objectives are frameworks used to identify intended outcomes for learners from the result of instruction. Researchers and educators have utilized a variety of educational taxonomies to assist in understanding students' thinking, academic needs, and assessment tools for programming tasks (Vrachnos & Jimoyiannis, 2017). The most widely used educational taxonomies now are Bloom's (1956), its revised form (Anderson et al., 2001) and the Structure of the Observed Learning Outcome (SOLO) taxonomy (Biggs & Collis, 1982; Vrachnos & Jimoyiannis, 2017).

Although Bloom's taxonomy and its modified version are widely used in education, various researchers have criticized the use of both Bloom's taxonomies with computer programming. Thompson et al. (2008) suggested that Bloom's taxonomy was difficult to use with introductory programming because categories from the taxonomy are often inconsistently matched to computer programming contexts. Key verbs in Bloom's taxonomies, such as *list, write, create, evaluate,* and other verbs could be confusing to students because (a) no examples are provided on how to use these actions with computer programming, and (b) the dual meanings of the verbs make assessing difficult. Using the word *write* or *create* in a programming question does not always identify which cognitive skill is involved. Thompson et al. (2008) gave this example of source code used in a class:

“*Write a statement that would create an object using…*”

Like Thompson et al. (2008), Shuhaida, Hamilton, and D’Souza (2009) found that aligning Bloom’s taxonomy with computer programming domains was difficult because (a) there are many subcategories in the taxonomy, (b) the taxonomy’s keywords are not easily relatable to computer programming questions because snippets of source code are often used, and (c) the original Bloom’s taxonomy is designed for core subjects.

Biggs and Collis (1982, 1989) developed the Structure of Observed Learning Outcome (SOLO) taxonomy to provide a systematic way to evaluate how a learner demonstrates mastery of academic tasks by showing increase of structural complexity. The increase in complexity is illustrated in two ways: the *quantitative* reflects the increase of the amount of detail in the learner’s response and the *qualitative* represents the combining of detail into a structural pattern (Biggs, 2011). Ramsden (1992) noted that the SOLO taxonomy is not content specific and can be applied to various subjects. Yassine, Chenouni, Berrada, and Tahiri (2017) apply the SOLO taxonomy to teaching introductory programming, describing several benefits of its use: (a) instruction is designed in a step-by-step approach; (b) the pre-structural stage accurately identifies learners as having no knowledge of programming or uni-structural as having some understanding, thus providing a starting point for learning; (c) a comprehensive approach to knowledge is utilized to move learners from novices to experts; and (d) a scale for learning objectives is provided. Table 2.7 shows the SOLO taxonomy’s five levels of depth of knowledge and students’ understanding (Biggs & Collis, 1982; Biggs & Tang 2011; Yassine et al., 2017). Biggs and Tang (2011) suggested that deeper learning is recommended over surface learning to help students proceed through the levels of the SOLO taxonomy.

Table 2.7

Levels of SOLO Taxonomy

	Depth of knowledge	Level of Understanding
1	Pre-structural	Lack of knowledge or understanding; Shows minimal evidence of relevant learning.
2	Uni-structural	Answer satisfies one aspect of the learning task, but misses other important factors.
3	Multi-structural	Response fulfills various features, but does not address key issues.
4	Relational	Answer has multiple parts correlated and the correlation between the various aspects can be qualitatively explained (e.g., when and why).
5	Extended abstract	Answers show learning beyond the relational stage by exhibiting high-order thinking skills and applying knowledge to other context (transferrable knowledge).

Hattie and Brown (2004) considered the unistructural and multistructural stages as reflective of surface learning because a student can easily encode the information and recall it later. However, the later stages, such as the relational and extended abstract stages, are considered deeper learning stages since students must have an understanding of the information and be able to differentiate it.

Hattie and Brown (2004) suggested using *testlets*, or a series of four questions that each measure an individual level of the SOLO taxonomy. This technique provides a way to assess students' knowledge and aids learners in comprehending the intention of the question and the criteria required to answer the questions correctly. Table 2.8 identifies guidelines for asking question for each stage of the SOLO taxonomy and the preferred answer.

Table 2.8

SOLO Taxonomy Guidelines for Asking Questions

	Stage	Answer requirement	Question
1	Pre-structural		
2	Unistructural	Only a single, relatable piece of information is needed for the learning objective	Which planet is the farthest from the sun?
3	Multistructural	Two separate pieces of information are needed	Which two planets are closest to earth?
4	Relational	Relationship between given information	Explain how the distance from the sun and temperature are related.
5	Extended abstract	Requires the learner to go beyond the information given and deduce a general principle	Given the Earth's position relative to the sun, in what ways does this affect the earth's climates and seasons?

Shuhidan et al. (2009) suggested using the SOLO taxonomy to assess student's understanding of computer programming because it provides a method to evaluate thinking. Table 2.9 provides information about various studies' that used the SOLO taxonomy in an introductory programming class.

Table 2.9

Studies Using SOLO Taxonomy with Introductory Programming Course

Researcher	Purpose	Participants	Method	Implication
Izu et al. (2016)	Study students' code design skills.	100 college engineering students.	Eight questions with two subquestions	SOLO good indicator to assess students' abilities at different levels.
Seiter (2015)	Analyze students' thinking.	4th graders	Scratch programming environment.	Scores consistent with SOLO level.
Jimoyiannis (2011)	Explore how students learn to program.	182 high school students	Survey with six programming tasks.	SOLO effectively showed students abilities.
Sheard et al. (2008)	Assess novice programmers	Mix of 120 under graduate and graduate students	Written response	Scores consistent with SOLO level.
Lister et al. (2006)	Evaluate students' design skills.	35 of 615	Oral response	SOLO good tool to assess students' abilities

Memory/Organization Instructional Strategies

KWL Charts

Know-Want-to-learn-Learned (KWL) charts and variations of them are graphic organizers utilized by teachers to aid students in linking prior knowledge on a specific topic and organizing what they are learning (Hershberger et al., 2006). KWL charts were developed to encourage inquiry and student questioning (Mihardi et al., 2013; Ogle, 2009). Ogle (1986) defined KWL charts as having three steps:

a. *Step K – What I know*. Students brainstorm regarding important information about the topic. The teacher may ask leading questions to help students ask questions and generate information.

b. *Step W- What do I want to learn*? Usually a group activity, students write down and categorize questions. This activity requires students to take an active part in learning by developing their own questions and seeking answers.

c. *Step L – What I learned*. Students write down what they have learned. A study by Mihardi et al. (2013) employed KWL charts with project-based learning to assess students' creative thinking in solved physics problems. KWL charts were used to help organize students' focus on their work and positively increase students' production. Cassady et al. (2004) indicated that KWL charts could be a reflective self-assessment tool to help students gauge their learning as they progress through an activity. They also suggested that adding a second assessment strategy, or an anchor, to evaluate students' learning reported on a

KWL chart could further increase KWL charts' effectiveness. Anchor tasks can be informal or formative assessment, a portfolio, or a simple to a complex task.

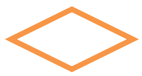
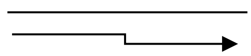
Flowcharts

Flowcharts are graphic organizers that are used for presenting, describing, or examining processes (Ayverdi et al., 2014). Flowcharts are a popular type of charts used in introductory computer science courses to teach beginners algorithms and programming (Xinogalos, 2013). John von Neumann developed the flowchart in the 1940s as a schematic representation of the logical steps a computer program followed. Computer analysts would develop an algorithmic solution to a problem in the form of a flowchart so computer programmers could create a computer program based on the flowchart (Ensmenger, 2016). Dol (2015) suggested that students need to understand algorithms in designing computer programming, but these can be difficult for students to understand. Charntaweekhun and Wangsiripitak (2006) indicated that flowcharts can used to teach beginners to write a program and even experts use them to discuss the algorithm of a program. Furthermore, using flowcharts with beginner programmers is the best way to construct programs because they are readable, ease-to-debug, effective, and user-friendly (Charntaweekhun & Wangsiripitak, 2006; Dol, 2015).

Flowcharts employ various shapes, lines/arrows, and Boolean statements to describe problems in terms of inputs, outputs, and processes. Some basic shapes include (a) rounded rectangle, (b) parallelogram, (c) rectangle, and (d) diamond. Lines/arrows are commonly used to present program flow and Boolean condition statements are differentiated by yes or *no* (Pasko & Bauer, 2002). Although, when developing a flowchart, the designer may use any shape with any meaning (Microsoft, 2017). Table 2.10 shows basic shapes and notations for a flowchart.

Table 2.10

Basic Flowchart Shapes and Functions

Name of shape	Image of shape	Function
Rounded rectangle		Start/end of a program
Parallelogram		Program input/output
Rectangle		Processing
Diamond		Decision
Line/arrow		Program flow
Boolean operator	Yes No	Yes/No

Visual Cues

Visual cues can be used to enhance instruction and improve the learning process by reducing cognitive load by reducing the visual search time for content (Martens et al., 2004). The *cognitive load theory* (CTL) states that instructional design can support or hinder learning depending on how the material is presented and the actions required by the learners to process it. Learners process all instructional material in their working (short-term) memory and the amount of space available is limited. However, the amount of information that can be handled can be increased if information is processed both visually and auditorily (Sweller et al., 1998). Sweller et al. (1998) identified two ways information is processed in working memory: the i*ntrinsic cognitive load* reflects the basic nature of how the material is presented, and the *extraneous cognitive load* reflects the actions required by the learner to understand poorly designed instruction. Designing instruction with a low extraneous load increases its effectiveness (Tabbers et al., 2004). Mayer (2009) suggested five ways to limit extraneous load: (a) *coherence*-omit extraneous words, sounds, or graphics, (b) *signaling*-emphasize essential words or graphics, (c) *redundancy*-remove redundant captions from narrated animation, (d) *spatial contiguity*-put key words next to complementary graphics on the screen or paper, and (e) *temporal contiguity*-show corresponding words and pictures simultaneously.

Equally important, Mayer's (2009) *cognitive theory of multimedia learning* (CTML) is a leaner-centered approach that infers that people process information through dual channels, visual/pictorial and auditory/verbal. Each channel has restriction on the amount of information that can be processed and restrictions should be implemented to ensure active learning (Mayer, 2009). It is important that teachers create multimedia instructions that utilize the three-thinking process of selecting, organizing, and integrating without a high extraneous cognitive load (Mayer, 2017). In addition, to limiting extraneous cognitive load, Mayer (2017) suggested that teachers manage *essential processing* and support *generative processing* to increase instruction effectiveness. Generative processing is the cognitive ability to have deeper understanding of content. Mayer (2017) indicated three ways in managing important information: (a) *segmenting*–making computer-based multimedia lessons into smaller, self-paced segments, (b) *pre-training*–introducing key concepts before the lesson, and (c) *modality*–showcasing words in spoken form. Teachers are advised to use social cues of conversational language, human voice, and human-like gestures to help motivate learners in learning (Mayer, 2017).

Mayer (2009) stated that signaling can assist learners in identifying key material by guiding them on how to proceed through cognition by circumventing extraneous cognition load caused by considering all information. The benefits of multimedia instruction (audio and visual) can be negated if the content is too complex because learners are unable to find the equivalent visual content (Jamet et al., 2008). However, a solution to this issue could be visually cueing the relevant information on screen to the images associated with the audio content. Signaling with visual cues has been substantiated by various studies (Jamet, 2013). For example, a study by Kalyuga et al. (1999) showed a positive effect of signaling using color. In the study, a push-button, on/off, light-switching circuit was used with various corresponding elements (e.g. circuit breaker, start and stop push-buttons, coil, light, and switch) in a split screen format. The study required students to search through text and for its complementary part. Corresponding text and elements in the diagram were in the same color. For example, the word *start* and the start-button were both blue in the diagram and the word *coil* and the coil in the diagram were both green. As a result, the study showed that the group using color-coded diagram and text was more effective that the group using the conventional format, and learners demonstrated a lower cognitive load and higher score on multiple-choice questions than the conventional group (Kalyuga, 1999). Ewoldt and Morgan (2017) stated that color-coded graphic organizers helped elementary students with learning disabilities in writing by using colors to show relationships between undeveloped ideas and complete sentences in the paragraph.

Jamet (2013) utilized a computer-based learning environment to observe students' reaction to cueing (changing words red when mentioned) and no-cueing (no change in color). Four areas of interest on the computer screen were of interest: the relevant area (content panel) and the irrelevant areas (progress bar, list of contents, and blank areas of the screen). Results showed that during no-cueing, 86% of students fixated on the content panel compared to 93% of students with cueing. Also, during cueing time, students' focus on irrelevant areas of the screen was cut in half. Jamet (2013) demonstrated that cueing by color-coding helped reduced search and process time. All mean fixation cueing time was lower than no-cueing. Color-coding is often used in specialized software known, Integrated Development Environments (IDE), for reading and writing computer code (Beelders & Plessis, 2016). Popular IDEs are Visual Studios and Eclipse. Color-coding with programming software is known as *syntax highlighting* source code and this feature allows for source to be designated a specific color based on some sem antic category (Dimitri, 2015). Sarkar (2015) stated this feature has made reading coding easier by allowing programmers to identify syntax errors based on syntax color. For example, if a certain source code syntax is designated a certain color than it will be easier to find subsequent source code that is the same color. Sarkar (2015) studied two groups of students using syntax highlighting with one group and nonsyntax highlighting with the second group. The study showed that the group that used syntax highlighting completed tasks significantly faster than the group not receiving syntax highlighting. Sarkar (2015) theorized that the syntax highlight reduced the time required to read and process source code because of its additional meaning and easily identifiable feature. The median completion time of the task favored the group that used syntax highlighting by 8.4 seconds.

Action Research

Definition

Action research is a qualitative research method used to enhance conditions and practices in classrooms and other practitioner-based environments (Craig, 2009). It is a recursive process of repeating and revising procedures and interpretations, which can enhance the quality of instruction (Charles & Mertler, 2011; Creswell, 2012). Lewin is credited with developing *action research* (Hine, 2013; Mertler, 2014; Mills, 2011) to examine social issues, but educators have also found his concepts useful for teachers in class-based research (Keegan, 2016). Lewin used a four-stage process consisting of planning, acting, observing, and reflecting. This four-stage process is generally maintained in modern action research, through four basic steps: problem or issue identification, data collection, reflection, and analysis/ action (Adleman, 1993; Creswell, 2012; Mills, 2011).

The design of action research depends on the researcher, reasons for the study, and participants (Sigler, 2009). The first step, *planning stage*, defines and describes the problem or issue. The *acting stage* is used to design and implement a plan, as well as to collect and analyze data. The *developing stage*, consists of revisions, changes, or improvements to the initial plan based on data analysis. Future actions are also developed. The fourth and final step, *reflecting stage*, allows researchers to review the entire process (Mertler, 2014). Figure 2.1 shows Mertler's (2014) depiction of the action research process.

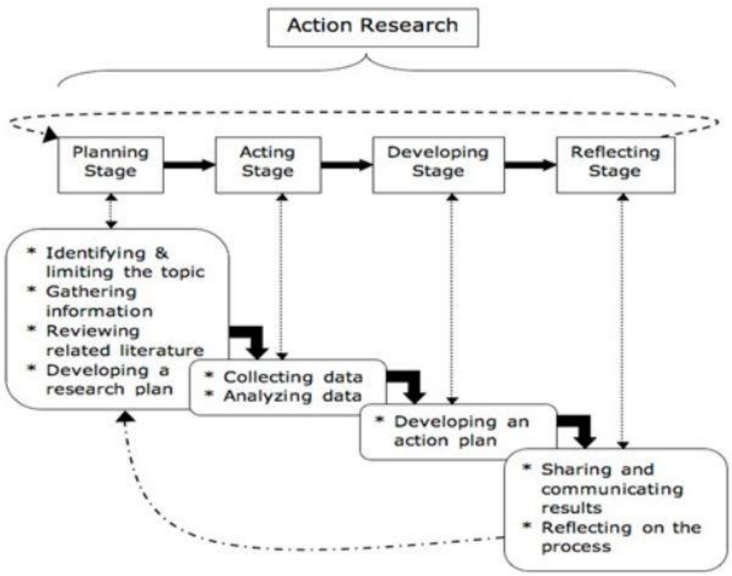


Figure 2.1. Mertler's (2014) example of action research. From *Action research: Improving schools and empowering educators* (p. 37), by C. Mertler, 2014, Thousand Oaks, CA: Sage. Copyright 2014 Sage. Reprinted with permission.

How Action Research Differs from Other Research Methods

Traditional research can be identified by the methods employed; qualitative (verbal descriptions and opinions), quantitative (numerical scores and measurements), and mixed methods (both qualitative and quantitative) (Charles & Mertler, 2011). Traditional experimental/scientific research seeks out generalizable explanations to be applied to all circumstances (Stringer, 2007). Traditional researchers are viewed as outsiders because the setting being study is not their normal place of occupancy (Mertler, 2014).

Teacher researchers are different from traditional educational researchers because they are studying their own practice and are committed to taking action and effecting positive educational change in their own classroom and school based on their findings (Mertler, 2014).

Action research focuses on a particular situation and localized solution (Stringer, 2007). Action research is often used by school teachers and principals who actively participate in their own inquiries as both teacher and researcher (Mills, 2011).

Why Action Research?

Action research provides systematic steps to (a) investigate problematic events or phenomena, (b) analyze and interpret happenings, (c) plan a solution, and (d) implement the solution (Stringer, 2007). Action research can benefit teachers because it improves teaching because it is tailored to the need of teachers (Keegan, 2016; Mills, 2011). Teachers are empowered by examining their own teaching practices instead of receiving training from an outside source that may or may not be of value to them (Creswell, 2012; Miller, Rosendale & Shanks, 2012). Mertler (2014) suggested that the goal of all classroom teachers should be to improve their professional practice and student outcomes, a goal that can be attained through action research.

Action research allows teachers to improve their effectiveness by studying their own classroom (i.e., instructional methods, students and assessments). Moreover, it is research conducted by teachers for themselves (Mills, 2011). Siegler (2009) stated teachers face the issue of developing and designing lessons and modifying their instructional strategies to meet the educational needs of their students. Teachers must continue to learn from their experiences to be able to align instructional strategies with changing classroom conditions (Siegler, 2009).

Advantages of Action Research

Action research has important advantages in education settings, including relevance, professional growth and active participants (Charles & Mertler, 2011).

Often, there is a gap between results reported by researchers on educational topics and the practice of classroom teachers (Mertler, 2014). Kennedy (1997) stated that researchers don't always address educators' concerns about teaching and that research is not usually conducted in settings comparable to most classrooms. Various studies have suggested explanations for the disconnect between educational research and practice. For example, teachers may seek *pedagogical knowledge* or understanding of the curriculum, students' learning and thinking style, and other contextual aspects, while researchers' knowledge goal is to find ways to interpret or explain phenomena by identifying patterns and devising theories that detail these patterns. Furthermore educational research is produced with generalized results that have little or no pertinence to particular contexts, and practitioners may not know how to access and use educational research (Broekkamp & Hout-Wolters, 2007; Kennedy, 1997; McIntyre, 2005; Mertler, 2014; Stevens, 2004; Stringer, 2007).

However, action research is conducted by teachers in their actual classroom and results are directly applicable to those settings (Charles & Mertler, 2011). In addition, putting research knowledge directly in the hands of educators may encourage teachers to reconsider their prior assumptions and initiate change (Kenney, 1997). Mills (2011) suggested that simply informing teachers about research is unlikely to bring about change, but findings from one's own action research are meaningful. McIntyre (2005) suggested that action research allowed teachers to interface between research and practice by testing and verifying claims.

Improvement of educational practice and knowledge is another benefit of action research (Charles & Mertler, 2011; Keegan, 2016). The most successful teachers are those who constantly and systematically reflect on their actions and the consequences of those actions. This constant reflection thereby results in the acquisition of new knowledge about teaching and learning. Furthermore, when teachers are actively engaged in using data-driven decision making in their own classroom, with their own students, teachers become empowered (Keegan, 2016; Mertler, 2014).

Disadvantages of Action Research

Action research also has drawbacks. For example, time constraints often prevent teachers from conducting this type of research (Creswell, 2012). Furthermore, action research is less precise than more controlled research designs and most subject to errors of bias, reliability, and validity (Charles & Mertler, 2011). Charles and Mertler (2011) noted that if collected data is not accurate, or the researcher has collected data that does not measure the intended scenario, then results will be inaccurate. Mills (2011) suggested that it may be challenging for researchers to remain objective in their research because it is easy to collect data that validates existing practices and to ignore discrepant data.

Conrad and Serlin (2006) insisted that researchers be aware of their biases because beliefs and values can easily influence elementary reasoning tasks and produce errors. These biases are possible because researchers are actively trying to identify plausible relationships between new findings and prior ideas during the research process. The issue of researcher’s bias lies trying to create a conclusion without basing it off prior knowledge. Conrad and Serlin (2006) suggested comparing and contrasting prior knowledge and new findings instead of justifying prior knowledge to explain away new knowledge as false.

Mills (2011) identified action research as a personal activity and it is important that researchers be mindful of their biases. Personal biases can pose a dilemma because researchers may have a difficult time remaining objective when the data collected does not validate their practices or research results (Mills, 2011). Conrad and Serlin (2006) and Mills (2011) suggested that researchers can offset their biases by using a list of hypotheses they expect to find in their research. By recognizing partiality, researchers pinpoint their beliefs and origins of theories about teaching. As a result, researchers can identify their assumptions before the start of their research and can faithfully consider a plan of action to ensure integrity.

To attempt to prevent biases in this study, I decided to make my hypothesis of this study known. I predicted that use of flowcharts and visual cues will positively influence their academic achievement and their perception of intervention. I also expect to observe a consistent and frequent use of the flowchart and visual cue interventions. My students often benefit when they are able to use prompts in a lesson. I believe that prompts allow students to check their progress in intervals, and which aids in students' learning. I did not believe that the KWL chart would be a positive influence on students' academics and that, instead, students would express negative perceptions of it. Students were likely to not like this intervention because it does not provide the same guidance throughout a lesson as a flowchart or visual cues. I expect students to use the KWL at the initial start of a lesson and not to look at it again.

Quality of Action Research

To establish quality research, researchers must be concerned with the validity and reliability of the study. Validity is how well the data collected actually measures what it is intended to measure. Reliability refers to the consistency of test scores. In action research *rigor* is used to refer to the quality of the research (Charles & Mertler, 2014; Mertler, 2014). Guba (1981) suggested that the *trustworthiness* of research inquiries can be judged by four concepts;

(a) *credibility*-the accuracy of findings; *transferability*-outcomes of the study applicability to other contexts; *dependability*-research procedures that were used to collect and analyze data are clearly and accurately documented; and *confirmability*-the researcher's objectivity of collected data, with credibility being the most important of these criteria. Action research is not generalizable, since it is context-specific and unique to the specific participants, environment, and objective (Charles & Mertler, 2014).

Mertler (2014) stated that the level of quality in action research is known as *rigor*. Stringer (2007) stated that rigor in action research is represented by the steps taken to ensure that results are not biased or influenced by the researcher. Four techniques can be used to ensure validity and consistency of findings: (a) triangulation, (b) member checking, (c) multiple data sources, and (d) prolonged engagement and persistent observation (Mertler, 2014; Patton, 2015). If multiple data sources support the justification of a phenomenon, the event is viewed as accurate, and substantiates data accuracy (Creswell, 2014). Studies that rely on only one method of data collection are susceptible to errors and biases specific to that method (Gall et al., 2003; Patton, 2015).

Guba (1981) indicated the criteria for trustworthiness of research included triangulation, member checks, prolonged participation, and persistent observations. *Triangulation* is collecting data from multiple sources and checking the findings between these sources to verify the accuracy of results. Triangulation can be attained in a qualitative study by using interviews and observations or by employing qualitative and quantitative methods (Patton, 2015). *Member checking* involves reviewing notes from observations and interviews with participants to clarify that the researcher captured the participants' ideas correctly and in full (Gall et al., 2003).

Prolonged engagement is the extended period spent at the research site to allow participants to adjust to the presence of the researcher and the researcher accurately observe and learn the environment (Guba, 1981). *Persistent observation* is the extended time spent in an environment to accurately learn and document essential characteristics of the participants. Prolonged engagement and persistent observation enhance reliability by allowing the researcher to recognize more consistent trends due to extended time spent by the participants in the setting (Creswell, 2014; Gall et al., 2003). A follow-up interview can be utilized to corroborate findings (Creswell, 2014). The repetition of cycles allows the researcher more time to examine and understand improvements (Melrose, 2001).

CHAPTER 3

METHOD

This action research study determined the effect of various memory/organization instructional aids on my teaching and on the achievement of high school students using LEGO Mindstorms robotics curriculum. This chapter details the research process, identifies participants, and explains data collection and analysis procedures used for three action research cycles.

Design

This study used an action research design consisting of three cycles, with four phases in each cycle. Each cycle was used to inquire about a specific instructional strategy. Research question was answered at the end of each cycle to help me better understand the influence of the memory/organization instructional strategy on student achievement. This study focused on the following research questions:

1. Did my instruction improve when using KWL charts to teach computer programming with LEGO Mindstorms?
    d. What was the influence of using KWL charts on students' performance?
    e. What were students' perceptions of using KWL charts?
    f. What were the observed patterns and frequency use of KWL charts by students?
2. Did my instruction improve when using flowcharts to teach computer programming with LEGO Mindstorms?

a. What was the influence of using flowcharts on students' performance?
b. What were students' perceptions of using flowcharts?
c. What were the observed patterns and frequency use of flowcharts by students?

3. Did my instruction improve when using visual cues to teach computer programming with LEGO Mindstorms?
   d. What was the influence of visual cues on students' performance?
   e. What were students' perceptions of using visual cues?
   f. What were the observed patterns and frequency use of visual cues by students?

Action research is a repetitive process of revising procedures and interpretations, which can enhance the quality of instruction. (Charles & Mertler, 2011; Creswell, 2012). Modern versions of action research consist of four phases: planning, acting, reflecting, and developing (Adleman, 1993; Creswell, 2012; Mills, 2011). Action research is not necessarily a linear process, as some steps may be repeated more than once (Mertler, 2014). The design of action research depends on the researcher, reasons for the study, and participants (Sigler, 2009). Completion of all four phases constitutes a cycle. Typically, an action research design consists of three or more cycles.

The *planning* stage was used to prepare myself and students for the upcoming cycle. Each morning, the LEGO Mindstorms EV3 (LEGO, 2017) robots were charged and ready to use when students arrived to class at 11:20 a.m. Copies of handouts and forms used to record data were made in advance. Any materials needed for mini-challenges or summative challenges were also available. Attendance was taken at the beginning of each class period. Then, students received instructions regarding class assignments, e.g., description, expectations, and responsibilities.

The *acting* stage was actual classroom activities including implementation of the custom memory/organization instructional strategy, completion of the Introduction to Programming LEGO Mindstorms (LEGO, 2017) curriculum, and collection and analysis of quantitative and qualitative data. For each cycle, students were introduced to the purpose of the targeted custom memory/organization instructional strategy. Students completed lessons using the custom memory/organization instructional strategy. Multiple methods used in data collection included several observations, interviews, and assessments. At the end of the cycle, data was analyzed to determine students' level of understanding after using the custom memory/organization instructional strategy.

The *developing* stage included a review of the collected data, and adjustments to instructional strategies for future cycles. Close attention was paid to observations and interviews to identify patterns of students' preferences for instructional strategies. Observations were used to collect verbal and nonverbal data on students' interactions and behaviors (Mills, 2011). Interviews were used to collect data through questioning. Interviews provided an understanding of elements that I could not directly observe and provided further comprehension of what I had observed (Patton, 2015). Collected data was searched for reoccurring words and phrases to reduce the amount of data and identify emergent patterns. Newly identified patterns helped determine where further investigation was needed and helped me further understand my topic and answer my research questions.

The *reflection* stage occurred when I summarized the process of the current cycle. Results from data collection, interpretations of results, and final conclusions from the interpretations were used to gain insight into the effectiveness of my lesson, the influence of the selected teaching aid, and possible revisions needed (Mertler, 2014). Reflection was performed daily and as data was collected to make time changes. Constant and consistent reflection allowed for persistent examination, which led to new discoveries and more accurate designs (Schon, 1987).

Participants

The site for this study was an urban high school in Augusta, GA. Where I teach students in grades 9-12. The school has a population of approximately 1,300 students and offers several career pathways. Georgia offers in high school course that represent 17 different career areas (Georgia Department of Education, 2017). Each area offers various instruction in career pathways, in a three, year-long, comprehensive classes designed to help students explore career areas and prepare them for employment (Scott & Sarkees-Wircenski, 2004).

I teach a three-course sequence in the Web and Digital Design pathway. This pathway is designed to teach students how to create, design, and produce interactive multimedia products and services used in business, training, entertainment, and marketing (Georgia Department of Education, 2017). The first class is Introduction to Digital Technology, the second is Digital Design, and the final class is Web Design. Each semester, I teach approximately 150 students in six classes: three Introduction to Digital Technology classes, two Digital Design classes, and one Web Design class. For the 2018-2019 academic year, enrollment in each Introduction to Digital Technology class was 27, 25, and 28 students, respectively. Digital Design had enrollments of

20 in each class section, and Website Design had 16 students enrolled. Students were scheduled to attend class daily, five days a week, for 55 minutes each day.

A convenience sampling of my fifth-period class was used in this study. I chose this class because these students would enroll in the AP Computer Science class in the following school year after successfully passing the Web Design class. Teachers usually use convenience samples, e.g., their intact class, when conducting action research because students are easily accessible (Charles & Mertler, 2011). Students in my Web design class were of various socioeconomic backgrounds, genders, and races/ethnicities. Participants included 16 students, 11 boys and 5 girls, with ages ranging from 15 to 18. The breakdown of students included two 10th-graders, ten 11th-graders, and four 12th-graders. The race/ethnicity of participants included 11 African American students, two White students, and three Hispanic students. Two students were 15 years old, seven were 16 years old, five were 17 years old, and two were 18 years old.

Data Collection

During the *acting* stage of each cycle, I delivered the treatment (my instruction and selected instructional aid) and collected data. When collecting data, I utilized four procedures: (a) participating in the setting, (b) direct observation, (c) in-depth interviewing, and (d) document analysis (Marshall & Rossman, 2011). Table 3.1 identifies the type of data collection methods I used and their purpose.

Table 3.1

Data Collection Methods Used in the Study

Data collection method	Purpose of data

Observations	Gathered information on the setting, participants, and activities in the setting (see Appendix B for collected data).
Informal interviews	Probed, clarified things I observed and not directly observed.
Formal interviews	Gathered descriptive data to gain insight into the interviewee's thinking (see Appendix C for collected data).
Formal assessments	Checked that students understand the content (see Appendix D for collected data).
Summative assessments	Overview of students' achievement (see Appendix E for collected data).

This schedule was followed for the entire week, constituting one cycle, one session per day. The first 20 minutes were dedicated to taking attendance and giving instructions to students for the planned class activities. Students were also introduced to the selected custom memory/organization instructional strategy. I explained the purpose of the instructional strategy and provided guided practice on its use. To check students' understanding, two tasks that required them to use the new instructional strategy were completed. I checked students' progress and asked for questions and concerns before moving on. Next, 15 minutes were allotted for non-participant observations. Non-participant observation is an observational role of recording notes without interacting with the participants during activities (Creswell, 2012). The three 5-minute sessions allowed for extended time to observe a group of students working. In addition, 15 minutes was allocated for participant observation, split into three 5-minute sessions. Participant observation is the observational role of participating in the setting or group as an active member (Mertler, 2014). The last 5 minutes of the class period were reserved for informal and formal interviewing of students. In the event that the last 5 minutes of class was not enough time for interviews, they were continued in the next scheduled class period. Transition between interviews was managed by an online timer from the website https://www.timeanddate.com/timer/. The timer was shown on the promethean board, a digital board on the wall, in front of the class to allow for all participants to easily be aware and prepared for the predetermined scheduled tasks. This website allowed me to setup simultaneous timers that would each make a beeping noise at the end of its time to indicate when a session ended and the next one began.

Observations

I used observations to collect verbal and nonverbal data on students' interactions and behaviors (Mills, 2011). I ensured that systematic observations were conducted by predefining procedural steps and identifying target behaviors that could be easily watched and recorded (Fry, Curtis, Considine, & Shaban, 2017; Lewis, Scott, Wehby, & Wills, 2014; Manolov & Losada, 2017). For each cycle, the target behavior was the use of the custom memory/organization instructional strategy. I notated KWL chart target behavior whenever I witnessed students writing on or looking at their KWL chart. Flowchart target behavior was notated when I observed students looking at their flowchart or walking through its steps of it. Color-coding target behavior was notated whenever I observed students looking at the color-coded instructions. Manolov and Losada (2017) indicated that systematizing an observation decreases the presence of bias.

Event sampling was an observational method used to efficiently measure the frequency, duration, latency, and intensity of target behaviors (Ostrov & Hort, 2013). I classified the recording of quantitative data during observations into four categories: (a) *duration*, the elapsed time for which each target behavior occurred; (b) *frequency-count*, each time a target behavior occurred; (c) *interval*, a participant's behavior at a given time; and (d) *continuous*, recording all activity for a specific amount of time (Gall, Gall, & Borg, 2003).

The first type of observation I implemented was non-participant observation. I conducted three intervals of 5-minute observations as a non-participant observer daily. Because students can repeatedly start and stop their use of memory/organizational instructional aids, these short intervals were used to obtain more accurate measurements of students use. As a non-participant observer, I checked for (a) nonverbal expressions, (b) how well students worked together, (c) how well students communicated, and (d) how much time students spent on a particular task (Schmuck, 1997). Additional information recorded during observations included (a) who was observed, (b) what was observed, (c) when it took place, and (d) how long the observation proceeded (Creswell, 2012). I identified patterns that aided me in detecting when students were using the instructional aids. When students were using their KWL charts, they consistently and candidly looked down at their charts. I was able to observe flowchart use when students were looking down at their flowcharts and when students held their charts and walked out its steps. The use of color-coded instructions was easily determined because many students placed the color-coded instructions on one side of the computer screen and had the programming screen on the other side.

I also spent time as a *participant observer* by engaging in tasks with students (Creswell, 2012). When I conducted participant observations, students were aware of being watched. I asked students questions to obtain insight into their thinking and to document students' own words in describing their actions. I used observation data to search for key issues and reoccurring events, and categorized data to help me make sense of it during reflection (Mertler, 2014).

Appendix B provides a copy of the form that was used to record event-sampling data. Lessons from the Introduction to Programming LEGO Mindstorms EV3 were designed so that students could work at their own pace and did not require direct instruction. This allowed me the opportunity to attach copies of the event-sampling form to my clipboard to use with observations. The first data recorded pertained to the timeframe of the observation (i.e., the date of the observation, cycle identifier, time of day, and documenting the number of the observation). Next, participants being observed were identified by a predefined unique identifier to maintain their confidentiality. The task was described by its objective, resources needed/used, allowed time, and time students spent engaged. The memory/organization instructional aid being used was circled from a list of three choices (i.e., KWL chart, flowchart, and visual cues). The 5 minutes allotted for each observation session, were divided into ten 30-second intervals. These intervals were used to obtain a more accurate view of the students' use of custom memory/organization instructional aids, since students could repeatedly start and stop their use. The frequency count was used to identify the number of times that students demonstrated the target behavior. Observations aided in collecting information relating to research questions about observed patterns and frequencies of the target behavior.

Field Notes

Field notes were used to record descriptions of what has been observed. I implemented specific procedures to guide my documentation of field notes to enhance the study's objectivity and precision, as well as to identify components relevant to the research questions (Parsons &

Brown, 2002). *Descriptive field* notes were used to record explanations of research participants, dialogue, physical settings of the environment; offer accounts of events; and report on observed behavior. *Reflective field* notes were used to record my thoughts or themes that developed during observations. Reflective field notes included (a) things learned from data collection procedures, (b) results of analysis, (c) conflicts, (d) my personal thoughts, and (e) clarification of previous notes (Bogdan & Biklen, 2003). The following guidelines were used for documenting field notes: (a) heading with time and date of observation, (b) location of observation, (c) the chronology number of observations, and (d) participants being observed.

I used the form in Appendix B to collect descriptive and reflective field notes. This form was divided into two sides. The left-side heading was labeled *Descriptive notes*, and the right-side heading was labeled *Reflective notes*. Under each heading several blank lines provided space to add notes. Field notes helped in collecting information about observed patterns in and frequency of target behaviors.

Interviews

I used interviews to collect data from students through questioning. Interviewing allowed me to gather descriptive data in students' own words to gain insight into their thinking (Bogdan & Biklen, 2003). Interviews provided me an awareness of elements that I could not directly observe and provide further comprehension of what I had observed (Patton, 2015). I established procedure guidelines to ensure that the interviewing process was consistent each time. Not every student in the class was interviewed; instead, I selected students based on various factors, such as knowledge, expertise in the area of focus, verbal skills, and desire to be interviewed (Mills, 2011). I employed three different types of questions, each of which served a different purpose.

Open-ended questions were one method of questioning used to invite thoughtful and in-depth responses to whatever was important to the interviewee. *Informal* or *semistructured interviews* consisted of basic questions and implemented follow-up questions where necessary based on the responses (Mertler, 2014; Patton, 2015). S*tructured interviews* used a specific set of predetermined questions, and only those questions were asked of each interviewee (Mertler, 2014; Mills, 2011). The technique of *probing* involved asking additional questions to acquire more information based on a previously asked question (Creswell, 2012). Probing allowed me to obtain more in-depth answers by asking questions in the format of "*Tell me more about that." "Can you give me an example of that?" "What, in particular, are you thinking about there?" "How were you feeling at that time?"* (Schmuck, 1997).

Appendix C provides a copy of the form used during interviews with participants. The first data recorded pertained to the timeframe of the interview (i.e., the date, current research cycle, time of day, and documenting the number of interview). Participant(s) being interviewed were noted, along with the specific memory/organization memory aid being used. Predetermined questions about a specific memory/organization memory aid were asked of students pertaining to (a) purpose, (b) the participant's perception of using the aid, (c) suggested changes, and (d) whether they would recommend the aid to other students. Interview questions helped me collect essential information relating to students' perceptions (the subject of several of my research questions).

Formal and Summative Assessments

Formal assessments were administered *during* instruction to determine if modifications were needed to enhance instruction (Mertler, 2003). I used open-ended questions and mini-design challenges as formal assessments. Open-ended questions were administered during lessons 1, 3, and 4 for each cycle. The *Introduction to Programming LEGO Mindstorms EV3* (LEGO, 2017) curriculum originally used multiple-choice question, but these were converted to open-ended questions to better evaluate students' understanding of the content area. A matrix was used to document students' understanding of open-ended questions (see Appendix H). The SOLO taxonomy is provided with the copy of the form in Appendix D to identify students' understanding of questions following lessons. This form had space to provide each student's identification number and individual score to open-ended questions. The SOLO taxonomy scale was used to evaluate students' responses. Biggs and Collis (1982, 1989) developed the Structure of Observed Learning Outcome (SOLO) taxonomy to provide a systematic way to evaluate how a learner demonstrates mastery of academic tasks by showing increase of structural complexity. Table 3.2 shows the SOLO taxonomy, which was used to assess students' level of understanding.

Table 3.2

Levels of Understanding Based on SOLO Taxonomy

	Depth of knowledge	Level of understanding
1	Prestructural	Lack of knowledge or understanding; Shows minimal evidence of relevant learning.
2	Unistructural	Answer satisfies one aspect of the learning task, but misses other important factors.
3	Multistructural	Response fulfills various features, but does not address key issues.
4	Relational	Answer has multiple correlated parts and can qualitatively explain (e.g., when and why) the correlation between the various aspects.
5	Extended abstract	Answer shows learning beyond the relational stage by exhibiting higher-order thinking skills and applying knowledge to other context (transferrable knowledge).

Multiple-choice questions from *the Introduction to Programming LEGO Mindstorms EV3 Curriculum* were converted to open-ended questions. Stankous (2016) indicated that teachers prefer multiple choice questions because they are easy to grade and produce higher grades than open-ended questions. However, multiple-choice questions do not always indicate students' true level of understanding because a correct answer can be chosen by chance, even if a student does not comprehend the question (Stankous, 2016). Multiple-choice questions are composed of a *stem*, a question or incomplete sentence free of complex language and alternatives, and *distracters,* wrong answers that divert attention away from the correct answer (Bakhtiar, Habib, & Khan, 2014). I removed the distracters to make the multiple-choice questions open-ended. One question in the Turning 3 lesson that was originally a "TRUE or FALSE" question. was converted to an open-ended question.

Original question

TRUE or FALSE: With "Rotations" on the Move Steering Block to 1, the whole robot rotates 1 time. True: the robot will turn around 1 time or False: the wheels will turn 1 time, not the body

Open-ended question

What happens when "Rotations" on the Move Steering Block is set to 1?

I administered summative assessments at the end of each cycle to evaluate students' academic achievement. Students received the grade *pass* if the summative assessment was successfully completed or *fail* if the summative assessment was not successfully completed. The three summative assessments were the Sensabot challenge in Cycle 1, the Orchard challenge in Cycle 2, and the Arm Position challenge in Cycle 3. Challenges helped to assess how well students responded to my custom memory/organization instructional strategy (Mills, 2011; Mertler, 2014). I used the form in Appendix D was used to record participants' results for all three of the summative challenges by indicating pass or fail. The formal and summative assessments helped me collect essential information relating to the research question about students' performance using custom memory/organization instruction aids.

Reflection

I took the results from data collection, interpretations of results, and final conclusions from the interpretations into consideration when creating a plan of action for the next cycle (Mertler, 2014). A reflection journal was used daily. I engaged in planned reflection at three intervals during the study: (a) daily after the completion of class to review the day's activities; (b) at the end of the week of the entire Cycle to review the process and results; and (c) at the end of the study to summarize my thoughts on the entire study (Farrell, 2015). By reviewing each step of the Cycle. I was able to closely monitor the progress of my study and make improvements, as needed (Mertler, 2014). Appendix G provides a copy of the form used to record my daily thoughts. The first data recorded pertained to the timeframe of the interview (i.e., the date, current research cycle, time of day, and documenting the number of interview). Predetermined questions that focused around my research questions were used to guide my reflection; these questions pertained to a) the day's activities, b) challenges, c) success, and d) failures to help me discover new insights about my research topic (Hendrick, 2009).

Data Analysis

Quantitative Data

Quantitative data retrieved during event-based observations were sorted into four categories: (a) frequency, (b) duration, (c) interval, and (d) continuous (Gall et al., 2003). This data was used with descriptive statistics to calculate mean, median, frequency, and standard deviation (Mertler, 2014). Standard deviation showed the dispersion or how spread out the scores were around the mean (Creswell, 2012). I used this information to check for patterns of behavior and the dispersion of scores. Information collected from calculating the measures of central tendency helped me understand how often the entire class utilized my custom memory/organization strategy (Mertler, 2014). I cross-referenced observations and interview data to help me answer my research questions.

Qualitative Data

I used inductive reasoning, to make sense of massive amounts of data by distinguishing significant information from trivial information, pinpointing significant patterns, and developing a framework for communicating the main idea of the results (Creswell, 2014; Patton, 2015). A matrix was used to organize data by type: interviews or observation. During data collection, I searched for reoccurring words or themes to assign codes or symbolic meaning in an effort to reduce the amount of data. These descriptive or inferential codes were used to categorize similar data for further analysis and drawing conclusions. Codes were grouped and reduced into one of the four common patterns that emerged: (a) categories or themes, (b) causes/explanations, (c) relationships among people, or (d) theoretical constructs (Creswell, 2014; Miles, Huberman, & Saldana, 2014). These classifications were further examined for the possibility of additional grouping. Consideration was given to emerging classes that enlightened me about my topic and helped me answer my research questions (Mertler, 2014). Table 3.3 shows data analysis completed for each research question of this study.

Table 3.3

Data Analysis for Research Questions

Research questions	Data type	Data analysis
1. Does my instruction improve when using the memory/organization instructional aid of KWL charts to teach computer programming with LEGO Mindstorms?		
A. What is the influence of KWL on students' performance?	Academic achievement	Mean, median, frequency Standard deviation
B. What are students' perceptions of using KWL charts?	Student feedback	Inductive analysis
C. What are the observed patterns and frequency of use of KWL charts by students?	Researcher observation	Inductive analysis
2. Does my instruction improve when using the memory/organization instructional aid of flowcharts to teach computer programming using iteration with LEGO Mindstorms?		
A. What is the influence of flowcharts on students' performance?	Academic achievement	Mean, median, frequency Standard deviation
B. What are students' perceptions of using flowcharts?	Student feedback	Inductive analysis
C. What are the observed patterns and frequency of use of flowcharts by students?	Researcher observation	Inductive analysis
3. Does my instruction improve when using the memory/organization instructional aid of visual cues to		

teach computer programming using iteration with LEGO Mindstorms?		
A. What is the influence of visual cues on students' performance?	Academic achievement	Mean, median, frequency Standard deviation
B. What are students' perceptions of using visual cues?	Student feedback	Inductive analysis
C. What are the observed patterns and frequency of use of visual cues by students?	Researcher observation	Inductive analysis

Display of Data

I used a checklist matrix to display data because it allowed for verification and comparability and even enabled basic quantification (Miles et al., 2014). Checklist matrices have a predefined format of rows and columns to easily enter and compare variables of interest. This study's variables of interest were the KWL chart, flowchart, and color-coded instructions. These variables were cross-referenced with data collected from observations, interviews, fieldnotes, and assessment results. A copy of this matrix is found in Appendix H. I was easily able to identify and compare participants' responses to interviews, assessment results, and observational notes. The research questions were placed at the top of the checklist matrix in separate rows. Each row was divided into three smaller rows because each research question was divided into three parts. Columns were divided into fourteen spaces, one for each student.

Triangulation of Data

Various steps were taken to ensure the level of quality or rigor in this action research study. I implemented four techniques to ensure the validity and consistency of findings: (a) triangulation, (b) member checking, (c) prolonged engagement, and (d) persistent observation (Mertler, 2014). *Triangulation* is collecting data from multiple sources and checking the findings between these sources to verify the accuracy of results. Triangulation was carried out in the study by using multiple data sources (i.e., observations, interviews, assessments). Multiple data sources with converging data added to the accuracy of the data (Creswell, 2014; Patton). I cross-referenced data for consistency by comparing observations, interviews, and assessments with one another and comparing quantitative and qualitative data. *Member checking* involves reviewing notes from observations and interviews with participants to ensure that the researcher has captured participants' ideas correctly and in full (Gall et al., 2003). I used member checking by asking participants to verify my findings from observations and to clarify interview responses to reconcile any inconsistencies or gaps in the data collected (Creswell, 2014; Patton, 2014). *Prolonged engagement* is the extended period spent at the research site to allow participants to adjust to the presence of the researcher and the researcher accurately observe and learn the environment (Guba, 1981). Prolonged engagement and persistent observation included additional observations to check the consistency and accuracy of interviews with students. Cross-referencing aided in confirming data collected because discrepancies are known to arise between what students say or do (Conrad & Serlin, 2006). Acting as a nonparticipant observer allowed me to view students in their natural role because students may have acted differently if they were aware, that they were being watched (Patton, 2015). The follow-up interviews were used to corroborate my findings (Creswell, 2014).

Ethical Considerations

Since this action research study involved human participants, I took steps to protect participants' rights. Institutional Review Board (IRB) approval was applied for and granted from the University of Georgia. Guidelines were followed to protect students' confidentiality and safety. I provided school district administrators and parents with: (a) a clear explanation of data collection procedures to show that coercion was not used; (b) a cover letter explaining the study and the my dual role as teacher and researcher; (c) parental consent forms stating that students were guaranteed no harm and would not be penalized for not participating; and (d) a statement that students had the option to opt-out of any or all data collection.

Since some participants were under the age of 18, steps were taken to obtain parental consent. Parental and student consent forms were provided to all students. Participants at least 18 years old did not require parental consent. (see Appendix K and L). These forms required parents' and students' signatures and the date for consent.

Procedure

This section outlines the Introduction to Programming LEGO Mindstorms EV3 lessons (LEGO, 2017) used during each cycle. Each lesson was composed of mini-lessons that included various instructional guides (e.g., diagrams and videos) and formal assessments (e.g., questions and mini-challenges; see Table 3.4). A checklist was created to help guide and track progress through the lessons (see Appendix L).

Table 3.4

Instructional Strategies and Data Collected Per Lesson

Cycle	Lesson	Overview	Diagram	Video	Question	Mini-Challenge
1	Moving Straight 1	Sensabot		X	X	
1	Moving Straight 2	Robot configuration	X			
1	Moving Straight 3	Steering forward		X	X	X
1	Moving Straight 4	Arm control		X	X	X
1	Moving Straight 5	Moving forward review	X			
1	Sensabot Challenge	Design challenge				
2	Turning 1	Crop tractor		X	X	
2	Turning 2	Robot configuration	X			
2	Turning 3	Turning in place		X	X	X
2	Turning 4	Other turns		X	X	X
2	Turning 5	Turning review	X			
2	Orchard Challenge	Design challenge				
3	Touch Sensor 1	Touch sensors		X	X	
3	Touch Sensor 2	Robot configuration	X			
3	Touch Sensor 3	Wait for touch	X	X	X	
3	Touch Sensor 4	Forward until touch		X	X	X
3	Touch Sensor 5	Touch sensor review	X			
3	Arm Position Challenge	Design challenge				

Learning exercises were implemented in each cycle to check for students' understanding of memory/organization instructional aid before summative assessments. For Cycle 1, KWL charts were used with *Moving Straight 3* and *Moving Straight 4*. For Cycle 2, the *Introduction to Programming LEGO Mindstorms EV3* curriculum suggested various lessons to teach students to use a flowchart. The initial lessons chosen for practice on the use of flowcharts were *Putting your shoes on* and *Getting dressed* because they were deemed universally applicable to students. Students then used a flowchart for the Turning 3 mini-challenge *Turning in Place*.

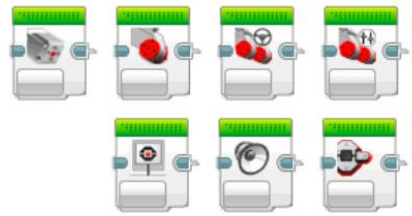
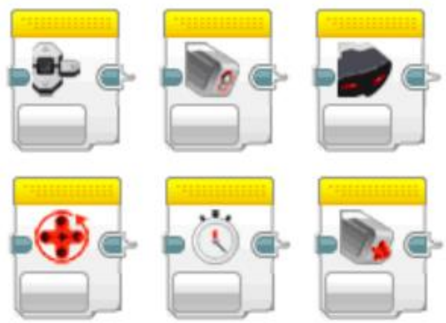
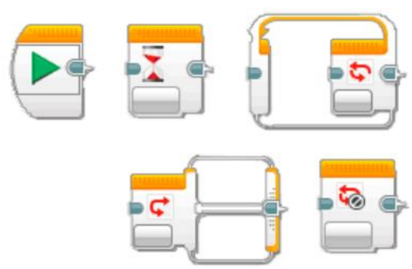
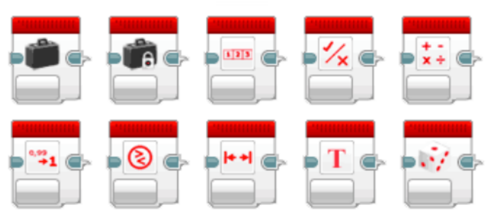
For Cycle 3, visual cues were used with the Touch Sensor 4 mini-challenge: *Four Walls*. The steps to complete this challenge were written out in pseudocode, a hybrid language between English and code that some programmers use to simplify behaviors. For example, to for a robot to stop when it approaches an object and move forward when there is no object the pseudocode might look like this:

1. Move forward
2. If (sonar sensor detects object) stop
3. When the sonar sensor no longer sees object move forward
4. Do this forever

Visuals cues for Cycle 3 utilized the same colors from LEGO Mindstorms EV3 programming blocks for corresponding actions in pseudocode (LEGO, 2017). Table 3.5 shows the color-coding scheme used with LEGO Mindstorms from the LEGO website.

Table 3.5

Visual Cueing Strategy of Block Color-Coding

Name	Color	Function	Blocks
Action blocks	Green	Controls the actions of the program, such as motor rotations, image, sound, and light.	
Sensor blocks	Yellow	Reads inputs from the color sensor, IR sensor, touch sensor, and more.	
Flow blocks	Orange	Controls the flow of the program. All programs started with a start block.	
Data operation blocks	Red	Allows for reading and writing variables and comparing them.	

An example of color-coded pseudocode to move a robot and check for an object in front of it could follow this type of order:

1. Move forward,

2. If (sonar sensor detects an object) stop,

3. When the sonar sensor no longer sees an object, move forward,

4. Do this forever.

Green highlighted text indicates movement and stoppage of the robot. Yellow highlighted text signifies the use of sensors. Finally, the orange text identifies the program's flow control.

Cycle 1. Week 1 (Cycle 1) activities included five formative lessons and a final summative assessment. The purpose of Cycle 1 was to teach students how to program a robot to move straight using KWL charts. Students received instruction through videos and diagrams, and were assessed by three sets of open-ended questions (see Appendix N), three mini-challenges (see Appendix N), and a final challenge. Open-ended questions were used to check students' understanding of the purpose of a Sensabot, steering the robot forward, and moving the robot's arm. Open-ended questions were chosen because of their ability to gauge students' level-of-understanding (LOU) by rating answers according to the SOLO taxonomy. Table 3.6 shows the SOLO taxonomy depth of learning. Post-interviews were performed after the summative challenge. Students were purposely sampled based on their LOU. Three categories were created for this sampling: (a) Pre-structural and Unistructural, (b) Multi-structural, and (c) Relational and Extended Abstract. These categories helped me identify students' thinking and instructional needs by identifying and further exploring any patterns that emerge. All six student groups were interviewed after the summative challenge. The assessments allowed me to collect data on students' perception of KWL charts, the KWL charts' influence on students' performance, and observed patterns and frequency of use of KWL charts by students.

In Week 2 (Cycle 2), I reflected on Week 1 data to decide if any changes needed to be made. During this reflection process, I considered what worked and did not work during Cycle 1. I used these findings to enhance my actions for the next cycle. Follow-up interviews were conducted depending on students' responses to post-interviews. This time was used to probe students' answers for patterns. Results from interviews allowed me to clarify my observations and provided insight into students' thinking about using a KWL chart. This process also allowed me to anticipate the time needed to complete data collection and possible pitfalls to be aware of for future cycles. Open-ended questions enabled me to categorize students by LOU. Students' LOU was examined for patterns to help planning and designing for Cycle 2.

Table 3.6

Levels of Understanding Based on SOLO Taxonomy

	Depth of Knowledge	Level of Understanding
1	Prestructural	Lack of knowledge or understanding; Shows minimal evidence of relevant learning.
2	Unistructural	Answer satisfies one aspect of the learning task, but misses other important factors.
3	Multistructural	Response fulfills various features, but does not address key issues.
4	Relational	Answer has multiple correlated parts and can qualitatively explain (e.g., when and why) the correlation between the various aspects.
5	Extended abstract	Answer shows learning beyond the relational stage by exhibiting higher-order thinking skills and applying knowledge to other context (transferrable knowledge).

Cycle 2. Week 3 (Cycle 3) activities were similar to those in Cycle 1, with five formative lessons and a final summative assessment. The purpose of Cycle 3 was to teach students how to program a robot to turn using flowcharts. Students received instruction through

videos and diagrams and were assessed by three sets of open-ended questions, three mini-challenges, and a final challenge. Open-ended questions were used to check students' understanding of the purpose of an autonomous crop tractor. Students were chosen for post-interviews following the same format from Cycle 1. The assessments allowed me to collect data on students' perception of flowcharts, flowcharts' influence on students' performance, and observed patterns and frequency of use of flowcharts by students.

In Week 4, I reflected on Week 3 data to decide if any changes needed to be made. During this reflection process, I considered what worked and did not work during Cycle 2 and used these findings to enhance my actions for the next cycle. Results from interviews were used to verify observations and gain insight into students' thinking about using flowcharts. This information aided me in evaluating changes implemented based on the reflection from Week 2. Students' answers to open-ended questions were used to categorize LOU from Week 3.

Cycle 3. During Week 5, students learned how to use touch sensors with the robot. There were five formative lessons and a final summative assessment. The purpose of Cycle 5 was to teach students how to program the robot to respond with the touch sensor. Open-ended questions were used to check students' understanding of the purpose of an autonomous crop tractor. The assessments allowed me to collect data on students' perception of visual cues, visual cues' influence on students' performance, and observed patterns and frequency of use of visual cues by students.

Week 6 was used to reflect on Week 5 activities, as well as the entire study. Results from interviews were used to verify observations and gain insight into students' thinking about using visual cues. This process aided me in evaluating changes implemented based on the reflection from Week 4. Students' answers to open-ended questions were used to categorize LOU from Week 5. Post-interviews were conducted in the same format as previous cycles.

CHAPTER 4

RESULTS

This chapter describes the results of an action research study conducted in a high school web design classroom using LEGO Mindstorms EV3 (2017) and the *Introduction to Programming LEGO Mindstorms EV3 Curriculum* from Carnegie Mellon's Robotic Academy (2019). Specifically, I examined the effect of using three custom memory/organization instructional aids when teaching computer programming, including KWL charts, flowcharts, and color-coded visual cues on students' performance, perceptions, and frequency and observed patterns of their use.

Participants represented a mixed-class of high school students. While the class had 16 students enrolled, only 14 students participated (one student was present for only two days; another student refused to participate). The 14 participating students were divided into six groups for the series of planned lessons and based on students' choice. I interviewed students within their groups during participant-observations periods. Group 1 included three women students in Grades 10 (n=1), 11 (n=1), and 12 (n=1). Group 2 included three juniors, all young men. Group 3 was composed of two junior men. Group 4 included two junior men. Group 5 include two seniors. Group 6 included one senior and one junior man.

This action research study consisted of three cycles. Each cycle used a four-stage process consisting of planning (studying and preparing for upcoming cycle), acting (collecting data), developing (analyzing data), and reflecting (reviewing entire process for a better understanding). Table 4.1 shows the instructional strategies and data collected for each cycle.

Table 4.1

Instructional Strategies and Data Collected Per Lesson

Cycle	Lesson	Overview	Diagram	Video	Questions	Mini-Challenge
1	Moving Straight 1	Introduction to a Sensabot		X	X	
1	Moving Straight 2	Robot base configuration	X			
1	Moving Straight 3	Steering forward		X	X	X
1	Moving Straight 4	Arm control		X	X	X
1	Moving Straight 5	Moving forward review	X			
1	Sensabot Challenge	Design challenge				
2	Turning 1	Introduction to crop tractor		X	X	
2	Turning 2	Robot base configuration	X			
2	Turning 3	Turning in place		X	X	X
2	Turning 4	Other turns		X	X	X
2	Turning 5	Turning review	X			
2	Orchard Challenge	Design challenge				
3	Touch Sensor 1	Introduction to touch sensors		X	X	
3	Touch Sensor 2	Robot base configuration	X			
3	Touch Sensor 3	Wait for touch	X	X	X	
3	Touch Sensor 4	Forward until touch		X	X	X
3	Touch Sensor 5	Touch sensor review	X			
3	Arm Position Challenge	Design challenge				

Action Research Cycle 1: Moving Straight

Planning

The first action research cycle consisted of five 55 minutes lessons taught using the *Introduction to Programming LEGO Mindstorms EV3 Curriculum* and an online learning management system (LMS). The Move Straight lesson contained information about the purpose of a sensabot, how to program a robot to move forward, how to adjust the distance of travel, and how to operate the robotic arm to move up and down. Instructional material from the *Introduction to Programming LEGO Mindstorms EV3 Curriculum* was posted online for students to access. The content was posted online the night before, but it was not made available to students until the scheduled class time. Lessons were posted to the LMS daily with corresponding instructional material (e.g., diagrams, videos, mini-challenge instructions).

Students routinely logged-on to the LMS to view daily assignments. At the beginning of the class period, I read each day's activities aloud, while students followed along by viewing on their computer screen. Open-ended questions were printed and administered to students at the designated time. Hardcopies were used to ensure I received all students' work. With students being in different grades, I did not want to make assumptions about students' prior experiences and knowledge. Therefore, I developed interview questions to inquire about students' previous knowledge and usage of KWL charts, computer programming, and LEGO Mindstorms robots. A copy of these questions can be seen in Appendix Q.

Acting

Before implementing the Introduction *to Programming LEGO Mindstorms EV3 Curriculum*, I used guided instruction to teach students to use a KWL chart, a type of graphic organizer. Guided practice is an instructional technique in which students work under the direct guidance of a teacher (Hunter, 1989). During guided practice, students created a KWL chart based on the LEGO Mindstorms EV3 robot. Students wrote what they knew about the robot in column "K" and what they wanted to learn in column "W". Column "L" was left blank until after the lesson so students could indicate what they had learned. Students completed KWL charts for four lessons in the first cycle: *Move Straight 1*, *Move Straight 3*, *Move Straight 4*, and *Sensabot Challenge*.

Lesson *Moving Straight 1* included an instructional video and three open-ended questions on the purpose and advantages of a sensabot and the required skills needed to program the *Sensabot Challenge*. *Move Straight 2* was a diagram showing how to assemble the robot. However, I assembled each robot beforehand to focus students' attention on the KWL chart and programming activities. Lesson *Moving Straight 2* required no action from students and was included on the same day as *Moving Straight 1*. Students could view the diagram from *Moving Straight 2* to make sure their robot was designed correctly.

Lesson *Move Straight 3* taught students how to program the robot to move forward. Students watched an instructional video and answered three open-ended questions about programming the robot to move forward and running the program. Then, students were introduced to the mini-challenge, *50cm Challenge*, programming the robot to move 50 cm.

Lesson *Move Straight 4* demonstrated how to move the robot's arm up and down. The lesson included an instruction video and three open-ended questions. Lesson *Move Straight 4* included the *Cargo Retrieval* mini-challenge, which required students to program the robot to raise its arm, move forward 50cm, lower its arm around a block, and reverse direction back to the starting point while dragging the block.

Lesson *Move Straight 5* was a review of the mini-challenge from *Move Straight 4*. It showed the programming blocks needed to solve *Move Straight 4*. There was no video to watch, questions to answer, or programming challenge. I used the available time during this lesson to ask students if they had any questions from previous lessons and to interview students to confirm my notes from previous observations.

The final lesson of Cycle 1was also the summative assessment for the unit, the *Sensabot Challenge*. Students watched an instructional video that explained the challenge. There were no open-ended questions to answer. Students had to program a robot to move in a linear path from a starting point to three marked lines. At each line the robot must stop, lower then raise its arm and move to the next line. After the robot made it to the last line and lowered and raised its arm, it was to return to the starting line.

Observations. Observational data was collected in the form of field notes, including duration and frequency of using KWL charts. Field notes included my descriptions of viewed behavior, conversations, and contexts (Patton, 2015). Duration was used to represent the length of time that students used the KWL chart. Students' use of the KWL charts was recorded on the observation form. A copy of this form can be found I Appendix B. I observed students in three 5-minute intervals during each class as a non-participant, not interacting with students, only watching. Next, I proceeded with three 5-minute participant observations, both observing progress and asking questions.

Students were open in using their cell phones and socializing with other groups during periods when they should have been focusing on the assignment. I frequently had to prompt students to return to task. Some students only watched related videos once, whereas others watched them several times. I did notice some students answering the questions before watching the video. Most students were reluctant to complete the KWL chart before or after the lessons. On several occasions, I noticed that some students would complete all three sections of the KWL chart as soon as they received the form. On other occasions some students would not even attempt to fill out the KWL chart until the end of the period, when I started to collect them. Reviewing students' completion of KWL charts showed that some students focused on irrelevant aspects of lessons. On several occasions, students wrote in the *What I Want to Know* section about concepts not pertaining to the lesson (e.g., sensors, turning, and sounds), even after being instructed on the day's activities. More information about students' reactions to completing the KWL chart was explored during interviews. The fact that some students were noncompliant made collecting accurate data for duration and frequency difficult.

I compiled statistics for the open-ended questions, challenges, and use of the KWL chart. Students watched instructional videos and completed open-ended questions for lessons *Move Straight 1*, *Move Straight 3*, and *Move Straight 4*. Students' answers to open-ended questions were measured based on their composition and identified on the level of thinking used to answer the questions (Brown, 2014). Students' answers were assessed by Biggs and Tang's (2011) SOLO taxonomy. Answer content was measured on four levels (1) *Unistructural* – one aspect of a lesson is understood, (2) *Multistructural* – two or more concepts are understood, but are not interrelated, (3) *Relational* – many relatable concepts; as a whole have meaning, (4) *Extended Abstract* – masters abstract concepts and relationships that allows the students to apply their understanding in other tasks and situations ; Brown, 2014). Unistructural and Multistructural are considered surface learning and relational and extended abstract are considered deep learning (Brown, 2014). I was hoping students' scores would show that students were using deep learning. Each question for a lesson received an individual score.

The class' average for each question was calculated. *Move Straight 1*, *Move Straight 3*, and *Move Straight 4* all had three questions, so there were three averages for each lesson.

The average student rating for *Move Straight 1's* questions was 2.92; for Move Straight 3, it was 2.10; and for *Move Straight 4*, it was 1.92. I hoped students would take their time and answer the questions in detail, since the answers to most open-ended questions were available in the instructional videos, and all the instructional videos were shorter than two minutes. The open-ended questions were scored on a scale from 0 to 5, based on Biggs and Tang's (2011) SOLO taxonomy. The mean score for *Move Straight 1* open-ended questions was 2.92 (SD = .97), and mean time to complete the questions was 1.29 minutes (SD = 0.51). The mean score for *Move Straight 3* was 3.10 (SD = .95), and the mean time to complete the questions was 1.71 minutes (SD = 0.72). The mean score for *Move Straight 4* was 3.00 (SD = 1.00), and the mean time to complete the questions was 1.517minutes (SD = 0.68). All students completed the 50 cm mini-challenge with a mean completion time of 10.29 minutes (SD = 2.93).

None of the students completed the Cargo Retrieval mini-challenge, so no time statistics were calculated. All students completed the Sensabot challenge, taking an average time of 10.20 minutes (SD = 2.93). The mean for the time observed using the KWL chart was 1.31 minutes (SD = 0.49). I considered the scores for the open-ended questions to be subpar because the instructional video discussed the answers to many of the questions verbatim. A score of at least of 4 would have been sufficient. I based the low score for the open-ended questions on students' lack of interest in answering the questions. Moreover, the time spent to complete the KWL charts seemed low, which I also attributed to a lack of interest. I believed that three to five minutes to complete the KWL charts would allow students enough time to give reasonable consideration to their answers.

I attributed the high standard deviations present in Cycle 1 answers to open-ended questions to some students rushing through the assignments, while other students took longer to answer the questions. The length of the instructional video for the open-ended questions for *Move Straight 1* was 1:16 minutes. Two groups completed the questions in 1:15 and 0:45 minutes, respectively. The *Move Straight 3* video was 1:59 minutes long, and three groups completed the open-ended questions in less time (0:57, 1:35, and 1:45 minutes, respectively). The *Move Straight 4* video was 1:44 minutes long, and three groups completed the open-ended questions in less time (0:45, 1:30, and 1:33 minutes, respectively).

Interviews. I interviewed the 14 participating students within their work groups. Interview answers were checked with my observation's field notes to make sure I accurately documented students' actions and to probe for a better understanding of students' thinking. Five minutes were allotted for participant observation during each class. I asked open-ended questions to help me understand students' actions and thoughts and conducted follow-up questions during the last five minutes of class. Follow-up interviews lasted up to two minutes and were conducted in groups. While reviewing data during the reflection period, I discovered several emergent themes. These emergent themes were

- Students were not completing the KWL chart because they believed it to be a waste of time.
- Students stated that the only disadvantage of using the KWL chart was time consuming.

- Advantages of using a KWL chart included that it was useful as a reference guide and for note taking, helps organize thoughts, and helps track progress.
- Students preferred to use the trial-and-error method instead of using rulers to measure, when the distance was unknown.
- Students suggested improvements for the KWL chart. The *What I Know* section is moved to under the topic but above the *What I Want to Know* and *What I Learned*. *What I Want to Know* and *What I Learned* sections are redesigned with numbered spaces to allow students to use the form multiple times.

As themes were discovered, I interviewed students on an individual basis to obtain insight into their individual thinking. This inquiry process took approximately two minutes and occurred during the reflection week. These individual questions often focused on feedback from Cycle 1 class activities. I asked students if the KWL chart had benefits or disadvantages, what improvements would they suggest, and how often they referred to the KWL chart. I conducted follow-up questions with all the students. Depending on answers to the follow-up questions, I interviewed some groups a second time. Group 6 students were questioned a second time because they consistently rushed through the KWL chart design process and open-ended questions.

One student had used a KWL chart before, two students had used LEGO Mindstorms before, and two students had prior programming experience. This information was important in determining students' prior experience levels since students were in different grades and not on the same academic level. When I asked several students why they had not completed the KWL chart, responses varied, including "*I find it a waste of time,*" "*I don't need it,*",and "*It takes too* *long to complete."* Students who regularly completed the KWL charts had positive opinions about the technique. Common themes from students' positive responses were that KWL charts were a) good for note-taking and as a reference guide, (b) helped organize thoughts, and (c) helped track progress.

I asked students why they did not use rulers in the programming challenges, even though they did not know the distance the robot was required to travel. I found it interesting that students were not using this tool to complete the challenge. Students gave consistent answers on why they had chosen not to use a ruler during the lessons. The common themes were that they preferred to (a) use trial-and-error or estimate distance, or (b) believed the ruler was not necessary.

When I asked students about suggestions for improving the KWL chart the most common answer was more space for the *L - What You Learned* section. During the Cargo Retrieval mini-challenge, one group asked to see their KWL chart from the previous mini-challenge to see information previously completed. In a follow-up question, they suggested that using the same KWL chart repeatedly rather than a new one for each lesson would be beneficial because it would contain all the information, they had collected in one place.

Reflecting. At the end of each day's activities, I journaled my thoughts on the data collected from observations, interviews, and field notes. Data from observations and interviews were searched for reoccurring events and phrases from students' feedback and behavior. Identified patterns from data suggested areas to investigate for better understanding of the phenomena. Reviewing my journal helped me focus on what I learned for the day and identify where further investigations were needed to answer my research questions.

At the beginning of the *Move Straight 1* lesson, I grew frustrated because students were not focused on completing the KWL chart, but were instead more focused on what they wanted to do with the robot or on using their cell phones. The next day, *Move Straight 3* produced similar results, and students focused on racing the robots and making them turn around. Students were still not following the outline of the activities given to them. When I interviewed students about the day's activities, students told me they were excited about using the robots and overlooked completing the KWL chart. My first impression observing students with the robots was they were not following instructions. Feedback from the students corrected my initial belief that they were being disobedient and highlighted to me that the importance of verifying data collected by observation. Students were very excited about using robots in class. They spent the majority of their time inspecting the robot and telling me about the actions they wanted the robot to perform. Students' curiosity about the robot was taking precedence over the established classroom procedures. The daily routine of me introducing the day's activity and students proceeding to the work period was abandoned. The systematic way of beginning a new task in class was replaced with students' random actions. Reflecting on the actions of students not following instructions encouraged me to inquire about the reasons for this behavior. This inquiry showed me the importance of cross-referencing my observational data with students' interviews and reflecting on my instructional practices.

After reflecting on *Move Straight 1* lesson, I reviewed literature on action research to look for additional ideas to aid me in conducting my research and engaging in the reflection process. I identified two ideas to incorporate into my study. Based on Mertler's (2014) suggestion of repeating some steps more than others, I decided to reflect after each instructional day and as I collected data on my findings instead of waiting until the end of the cycle or instructional day. This approach was a way for me to make timely changes and immediately assess the effects of instructional modifications. I was further encouraged to scrutinize the data collected frequently by Schon's (1987) suggestion that repeated analysis can create new discoveries and more accurate designs.

After reviewing the difficulties from the previous two lessons of *Move Straight 1* and *Move Straight 3*, which involved students rushing through KWL chart completion, I noticed that I had been too focused on recording data about students' use of the KWL chart and not sufficiently focused on students transitioning from filling out the KWL chart to answering the open-ended questions and, finally, to programming the robot. Revisions were made for *Move Straight 4* lessons to include check points to confirm students' progress. These checkpoints allowed me to examine students' understanding, reteach if needed, and transition students to the next activity. Reviewing students' KWL chart daily allowed me to better understand how they approached each lesson. The KWL charts repeatedly showed that students focused on unrelated concepts during the lesson. This lack of focus caused students to have difficulties in completing programming assignments. After I reviewed the use of KWL charts and emphasized to students the importance of concentrating on the topic of the KWL chart, students were less inclined to engage in irrelevant programming aspects. I observed less random robot movement than seen during *Move Straight 1* and *Move Straight 3*. Furthermore, when I interviewed students during and after *Move Straight 4*, none of the students mentioned trying to make the robot turn around or other inappropriate actions.

At the conclusion of the *Move Straight 4* lesson, I created post-interview questions that were administered after the *Sensabot Challenge*. With one scheduled day left in Cycle 1, I wanted to make sure to obtain feedback from all students. Interviewing students during Cycle 1 was very valuable to my study because it allowed me to continually check students' progress and perceptions and to obtain a better understanding of their learning needs. I used the post-interview questions to learn about students' perceptions and understanding of a KWL chart. After analyzing students' feedback from post-interview questions, I conducted follow-up interviews with students. I interviewed students individually, for approximately two minutes each, to obtain a better understanding of students' answers to the post-interview. A copy of these questions can be seen in Appendix R.

During the *Sensabot Challenge*, I noticed that all students did not use a ruler, which would have been very helpful in successful completion of the challenge. This lesson required students to program the robot to move in a linear path, stopping at three predetermined points, and then returning to the starting point. Written instructions suggested that students use a ruler to measure distances. I made plans to interview students about why they did not use the ruler. This is information further explored in the interview section.

At the conclusion of Cycle 1, I obtained a better understanding of how students approached learning a new activity, as well as instructional needs to be successful in learning to computer program. Students were very excited to learn about using LEGO Mindstorms EV3 robots, and this excitement was one reason behind their impatience and tendency not to use best practices when learning to program the robot using a KWL chart. Instead, students wanted to proceed to what they believed to be exciting. So, while students were required to create the

KWL chart and answering the open-ended questions I saw a substantial decrease in their enthusiasm and focus. Students preferred to socialize and use their cell phone when the robots were not involved. The results of this lack of interest were easily seen in the low scores on the open-ended questions and short amount of tine designing KWL charts.

Students' instructional needs were identified from reoccurring themes discovered from interviews and observation. When students were engaged in the programming challenges, they needed help organizing their thoughts and needed guidance in how to proceed to solve the programming tasks. The KWL chart allowed students to identify their prior knowledge and the new knowledge needed. The KWL chart provided students with the focus needed to utilize their time wisely while working on programming challenges.

Statements that students wrote in the *What I Learned* section did not always correlate to their mastery of the concept or ability to apply it in other situations. Some students would accurately explain the minor aspects of the lesson that aided them with the challenge, while other students would describe unrelated concepts (e.g., turning and honking the horn). The *What I Learned* section was not a consistent predictor for identifying whether students would successfully complete the *Sensabot Challenge*. Students' statements in the *What I Learned* section was accordant with the mini-challenges' objective of Moving Straight 3: Steering Forward and Moving Straight 4: Arm-Control. Students' wrote they learned to program their robot to move forward and move the robot's arm up and down. This information was helpful to me because it allowed me to track students' learning. The main issue I observed during the Sensabot Challenge was students not using a ruler to measure the unknown distance between the predetermined stopping points. Students preference of trial-and-error method of estimated distance during the *Sensabot Challenge* is most likely the reason none of the groups completed the challenge. Students had no problem programming their robot to move forward and raise the robot's arm up and down.

Developing. Based on data collected from observations, interviews, and field notes, I created an alternative KWL chart to help students learn how to computer program. This new instructional strategy was called $K(WL)^n$. An example of this design can be seen in Appendix Q. The name loosely correlates to how this new chart will be used. The *What I Know* section was moved to under the topic but above the *What I Want to Know* and *What I Learned* sections. The *What I Want to Know* and *What I Learned* sections were redesigned with numbered spaces to allow students to use the form multiple times. Feedback from students' interviews indicated that they wanted to track their learning by using the KWL chart multiple times or with multiple lessons.

After completing Cycle 1, I was amazed at how the literature had accurately guided my study and helped me obtain a better understanding of how to design a custom KWL chart based on students' needs. The professional development I have received as a teacher is usually designed for core subject teachers (e.g., mathematics, language arts, science, and social studies), and literature on educational strategies shared was minimal and not placed in context. I have a new appreciation for using literature to influence my instructional practices. Using multiple data sources allowed me to check my findings with other sources and confirm my findings. I learned that making assumptions about students' actions and learning can lead me to make unreasonable decisions in my instruction and reactions to students' behavior. KWL charts have always been the most popular graphic organizer in professional development at schools where I have worked.

It was surprising to me when students indicated their lack of use of the KWL charts. Furthermore, the literature showed me how to strategically reflect on my educational practices. Reflecting on teaching is not new to me, but making such reflection routine is. Making the reflection process methodical allowed me to reflect consistently and with a purpose. Predefining what, when, and how to reflect on my educational practices allowed me to reflect more efficiently. This improved and constant reflection allowed me to refine my instruction practice. Instead of waiting until after an assessment to consider changes in my instruction, I now understand the need to think about classroom's activities daily.

Action Research Cycle 2: Turning

Planning

Preparation for Cycle 2 was similar to the planning I completed in Cycle 1. Instructional material (e.g., robots, rulers, and computer paper) were gathered, prepared, and made available for later use. In addition, I used notes from Cycle 1 observations, interviews, field notes, and reflection in my planning for Cycle 2. These notes encouraged me to focus on students' prior knowledge and experiences of flowcharts, their use of rulers when distance was unknown in programming challenges, and observed patterns of student use of flowcharts. Flowchart pre-assessment and post-assessment interview questions were developed similar to those in Cycle 1. The flowchart pre-assessment interview questions were designed to collect data about students' prior knowledge and experience using flowcharts. A copy of the pre-assessment interview questions can be seen in Appendix T. The post-interview questions were developed to be administered immediately after the *Orchard Challenge* and were used to collect data on students' perceptions, understanding, and suggestions for using flowcharts to program robots. A copy of the post-interview questions can be seen in Appendix U. The objective for Cycle 2 was to teach students how to program the robot to turn. Cycle 2 consisted of five 55-minutes lessons. I introduced the design process of creating flowcharts to help with programming, prior to *Turning 1* lesson. The emphasis in my notes indicated a need to implement checkpoints for students after they created flowcharts, answered questions from the video, and completed challenges. I also reminded myself to make sure that rulers were clearly placed and referenced whenever students used the robots, which would allow me to observe if students used the rulers or if they focused on the robot rather than on using the learning aids provided.

Cycle 2 designed to teach students how to program the robot to turn. Guided practice was used to teach students the design process of a flowchart. Guided practice is an instructional technique in which students work under the direct guidance of a teacher (Hunter, 1989). Students created separate flowcharts before beginning the programming challenges: *90-Degree Turn* in Turning 3 and *Dizzy Drill* in *Turning 4*, and culminating in the *Orchard Challenge*.

The *Turning 1* lesson introduced students to the purpose of the autonomous tractor with an instructional video and three open-ended questions. The *Turning* 3 lesson taught students how to program the robot to turn in place and required answers to two open-ended questions. After answering the questions, students attempted *90-Degree Turn* mini-challenge. The *Turning 4* lesson used an instructional video to show students how to program the robot to turn while moving, with two open-ended questions to check understanding. The final task, *Dizzy Drill*, required students to program the robot to travel around an object. *Turning 5* reviewed *Dizzy Drill* and *90-Degree Turn* and showed the program solution for the mini-challenge. The last activity for Cycle 2 was the *Orchard Challenge*.

Acting

Students were introduced to the design process of creating flowcharts using guided-practice. Students first attempted to create a flowchart that described putting on their shoes. This lesson was very vague because the instructions were not specific and left students to design their flowchart based on their own interpretation. The instructions did not specify whether students already had their shoes on or if they had to find them. I instructed students that looking for the shoes and tying the shoe laces could be considered one step. *Turning 1* lesson included an instructional video on the purpose of autonomous tractors and three open-ended questions on the video. The Turning 2 activity was a diagram for construction of the robot. First, students watched an instructional video to learn about the autonomous tracker and the week's activities. After watching the video students would answer three open-ended questions. *Turning 1* lesson did not have a programming challenge.

The objective for the *Turning 3* lesson was to program the robot to turn. Before starting this lesson, I reviewed flowcharts with students by showing an example of the sequence of steps included in adding two numbers together and printing the sum. Students were given two assignments to create flowcharts to check their understanding. The first task was to design a flowchart adding three numbers and printing out the sum. I used guided-practice again to check students' understanding of the design process of the flowchart. Some students were still having difficulty with designing a flowchart. Instead of having one action for the rectangle symbol block, some students were writing more than one action for a single block. Some students would write *Enter first number* and *Enter second number* in the same block. The solution for this task is shown in Figure 4.1.

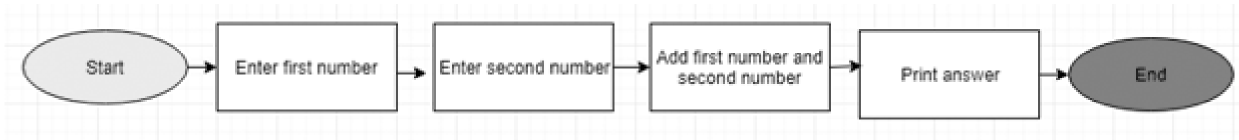


Figure 4.1. Solution to flowchart design process.

The second activity was to create a flowchart for the *Sensabot Challenge* from Cycle 1. After guided practice with the first flowchart, students' understanding greatly improved. Now, students showed an accurate design process for the flowchart representing the *Sensabot Challenge*. Figure 4.2 shows the flowchart solution for the *Sensabot Challenge*.

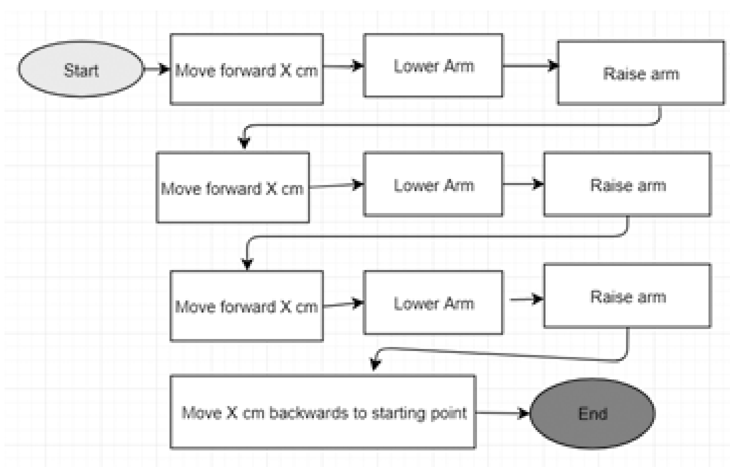


Figure 4.2. Flowchart solution for the *Sensabot* challenge.

After the review session, students proceeded with the *Turning 3* activity. *Turning 3* lesson included an instructional video with two open-ended questions, based on the video, on programming the robot to turn in place. The lesson included a mini-challenge, *90-Degree Turn*, which required students to program the robot to turn exactly 90 degrees. Before students were allowed to work with robots, they were given a blank sheet of computer paper with which to construct a flowchart showing the program flow, step-by-step, for the challenge. After all flowcharts were completed students were allowed to proceed to the mini-challenge.

Activity *Turning 4* explored additional ways to make the robot turn. It included an instructional video and two open-ended questions to check understanding. Lesson *Turning 4* included a mini-challenge, *Dizzy Drill*, which required programming the robot to start from a designated point, drive around an obstacle, and return to the starting point. Before starting the mini-challenge, students created a flowchart for the challenge on a blank sheet of computer paper. Students were allowed to work with their robot and begin the mini-challenge after showing me their completed flowchart.

The *Turning 5* lesson involved showing the programming solution for the *Dizzy Drill* and *90-Degree Turn* mini-challenges. There was no video to watch, no open-ended questions to answer, or and no programming challenge. A flowchart was not used for the *Turning 5* lesson.

The *Orchard Challenge* was the final lesson in Cycle 2. Students watched a video on the challenge and then created a flowchart on computer paper to resemble a program solution. Students were required to program their robot to move on both sides of the three designated lines. Three rows of lines were created with tape on the floor to resemble trees. The exact locations of the lines were not important.

I compiled statistics for Cycle 2 in the same manner as completed in Cycle 1. Eight students were absent for the *Turning 1* lesson. Two students were absent for the entire cycle. Another six students were absent due to testing for students interested in joining the military after graduating from high school. The score range for Cycle 2 was 0 to 5, the same as Cycle 1. The mean score for the *Turning 1* lesson was 2.11 (SD = .62) and the mean time used to complete the questions was 1.25 minutes (SD = 0.59). The mean scores of lessons *Turning 3* and *Turning 4* were 2.32 (SD = .72) and 2.36 (SD = .45). The mean times to complete open-ended questions *Turning 3* and *Turning 4* was 1.44 minutes (SD = 1.17) and 1.68 minutes (SD = 0.68), respectively. All six groups completed the *Dizzy Drill* mini-challenge. The mean time to complete the *Dizzy Drill* challenge was 5.32 minutes (SD = 1. 94). No time was calculated for the *Orchard Challenge*, since none of the groups completed the challenge. The mean time for students designing and using a flowchart was 3.41 minutes (SD = 1.79). The scores for the open-ended questions and the mean time for flowchart use were lower than I desired. I was hoping for students to take their time answering the open-ended questions and designing their individual flowcharts. Some students continued to put little effort into the open-ended questions and use of the flowchart. I even returned flowcharts to some students to redo because they were poorly designed. Furthermore, some students spent little time answering the open-ended questions, and the answers they did provide were insubstantial.

Similar to Cycle 1, I associated Cycle 2's high standard deviation scores for answering open-ended questions to some students rushing through answering the questions, while others took much longer to answer the questions, creating a discrepancy in scores. The length of the *Turning 1* instructional video was 0:45 minutes and three groups answered the open-ended questions quickly (1:00, 1:11, and 1:15 minutes, respectively). The *Turning 3* instructional video was 1:18 minutes long, and three groups answered the open-ended question in less time than the video (0:43, 0:50, 1:01 minutes, respectively). The *Turning 4* instructional video was 1:31

minutes long, and two groups answered the open-ended questions in quick succession (0:57 and 1:45 minutes, respectively).

Based on Biggs and Tang's SOLO taxonomy (2011), a majority of Cycle 2 scores were a 2. A score of 2 indicated that one piece of information has been identified in an answer that can have multiple answers. The average scores for Cycle 2 were as follows: *Turning* 1, 2.1; *Turning* 3, 2.3; and *Turning 4*, 2.25. A score of 4 or 5 was desired because they were considered as representing deep learning, whereas levels 1 through 3 were considered surface learning (Biggs & Tang, 2011).

Observations. Observation data for Cycle 2 was collected in the form of field notes and notes on duration and frequency use of flowchart. Duration and frequency were calculated using the same method as in Cycle 1. The flowchart lesson, included with the *Introduction to Programming LEGO Mindstorms EV3*, contained only one diagram and was not helpful in explaining how to use a flowchart. Students had difficulties understanding how to use the diamond (decision) block symbol in their flowchart. Even after guided practice in creating a flowchart, students were unable to correctly design a flowchart.

The most common issue students demonstrated was understanding whether the decision block symbol represented a question or decision point in their flowchart; importantly depending on the choice or answer, this would determine at least two different actions. For this type of program, a diamond block symbol was used in a "yes" or "no" question. An answer of *yes* would allow the program flow to continue, while a *no* answer required the program flow to move back to the block symbol prior to the diamond block symbol. Students' early flowchart designs showed the diamond block symbol being used, but the implementation of the symbol only showed the program flow proceeding forward, regardless of the decision.

The second most common difficulty was using the rectangle (statement) block symbol correctly. The rectangle block symbol executed one action or behavior at a time. Students commonly used a single rectangle block symbol to perform multiple tasks. Students' instructions designated for the rectangle block symbol varied and included "*go in room, look in closet, look under bed,*" "*find shoe, put on shoe, tie shoe,*" and "*find shoe, put on sock and shoe.*" I eventually told students to think of the statement block as one verb. It can only do one thing.

Another difficulty I noticed was that some students rushed through the design process and turned in a poorly constructed flowchart. These bad designs were returned to students to redo. Additionally, the same issues that occurred during Cycle 1 were a problem in Cycle 2. Cell phone use, socializing, and a lack of effort in completing the questions continued to be a daily issue in class. Some students were more engaged when using the robots than at any other time during the class period. When it came time to answer the open-ended questions and create the flowchart, some students would wander off task. I believe that some students put more effort into activities that they found enjoyable and/or important.

The *Turning 3* mini-challenge, *90-Degree Turn*, required only one programming block for the solution. This mini-challenge was not a good assessment to check students understanding of flowcharts because the complementary flowchart would require only one shape in addition to the stop and end blocks. I also noticed that students relied on the trial-and-error method instead of using the ruler for precision for the *Dizzy Drill* mini-challenge and the *Orchard Challenge*. For the *Orchard Challenge*, I did notice two groups using the flowchart and walking the course to verify its accuracy. The group that preferred to rush through making its flowcharts had a lot of difficulty with this challenge. Their robot went substantially off the designated course several times.

Interviews. Interviews were used to collect information about students' views of using flowcharts with computer programming and to clarify observation data. Students were interviewed about their prior experience using flowcharts in school, the classes they were used in, and their level of expertise using them. Interviews revealed that 8 of 14 students had never used flowcharts before; 2 of those eight students said they had used one in high school. Moreover, 7 of 8 students who said they had used flowcharts before felt their understanding of flowcharts was at a beginner level.

Several students mentioned that they did not understand how to use the diamond block symbol. After I instructed students to avoid using the diamond block symbol they indicated that it was easier for them to construct a flowchart. Three students mentioned that flowcharts took too much time and were a waste of time. They preferred to get straight to programming the robot instead. Two students were in the same group and rushed through their flowchart design process to interact with the robot more quickly. The remaining nine students stated that using flowcharts was beneficial to them and offered the following insight:

(a) A flowchart made programming easier or more manageable by breaking it down into smaller steps. The smaller steps allowed students to focus on one thing at a time;

(b) They suggested avoiding using the diamond block symbol because it confused them and was difficult to use;

(c) They suggested that a flowchart would be helpful for first-time learners of programming and flowcharts;

(d) Flowcharts can be used as a reference; and

(e) Flowcharts help organize thoughts.

Students identified the disadvantages of flowcharts as not giving the distance or degrees needed and being time consuming. When I asked why they did not use the ruler, some students responded that they preferred the trial-and-error method.

Reflecting. Simplifying the flowchart by removing the diamond block symbol improved students' understanding and creation of flowcharts. The flowchart helped students break down the programming assignment into smaller and more manageable parts. Whenever the students' robot did not perform as expected, it was easy for them to identify exactly where in the computer program source code to make changes. Even if a student in a particular group had problems programming the robot, flowcharts were very easy to read.

Students, nonetheless, struggled to understand the design process of creating two possible solutions for the diamond block symbol. No other lesson in my class this year had required a similar syntax so far, so this may have been the first-time students ever had to consider multiple outputs for one input.

The *90-Degree Turn* and *Dizzy Drill* mini-challenges were very short for the students to program. I was not sure if these tasks were long enough to accurately measure students' understanding of the flowchart or to warrant a need for a flowchart. My observation that students not wanting to use a KWL chart, flowchart, or a ruler was interesting. Students preferred to rely on a time-consuming trial-and-error method. The added checkpoints allowed me to assess students learning and provide mediation in a timely manner. The checkpoints also prevented students from rushing through lessons that they felt were boring or irrelevant and encouraged them to take their time to turn in quality work.

At the conclusion of Cycle 2, I recognized students' need for programming tasks to be simplified to allow them to focus on one aspect of programming at a time. The flowchart provided an improved visual aid illustrating steps needed to complete a programming challenge. When students ran into problems with their programming design, they could easily pinpoint where in the program changes were needed by looking at their flowchart. Watching students walk through their flowchart design showed that students understood the purpose and the benefit of the flowchart.

Developing

Based on observations, interviews, and field notes, I developed a design guide to use when creating flowcharts for students learning to computer program. This design guide suggests minimizing the variations of shapes used to two. Limiting shapes reduces the chances of confusing students who have never used a flowchart and allow them to focus on programming. The oval/ellipse block symbol should be used to indicate start/beginning and stop/end. The rectangle/square (statement) block symbol should be used to indicate one action. It is important to emphasize to students that one statement block is to represent one action or one verb. Designating the statement block symbol to only one action allows students to focus on one programming task at a time. This technique breaks the programming assignment into smaller and more manageable parts. The diamond (decision) block symbol commonly used in flowcharts should be avoided until students obtain a solid foundational understanding of the design process for flowcharts. An example of this flowchart design-guide is in Appendix X.

Action Research Cycle 3: Sensors

Planning

Preparation for Cycle 3 was similar to the planning I completed in Cycle 1 and Cycle 2. Instructional material (e.g., diagram, videos, mini-challenge instructions) were gathered, prepared, and made available for later use. I reviewed my journal notes from observations, interviews, field notes and reflection, from Cycle 1 and Cycle 2, in my planning for Cycle 3. These notes encouraged me to focus on students' prior knowledge and experience of color-coding, their decision to use or not use rulers when the distance was unknown for a programming challenge, use checkpoints to assess students' progress, and observed use of color-coded instructions. Identified patterns from Cycle 1 and Cycle 2 data suggested students benefited from the use of structured steps. I planned to implement a way to structure steps with color-coding and observe the results.

The objective for Cycle 3 was to teach students how to program the robot to use a sensor. Similar to the process in Cycle 2, my notes prioritized implementing checkpoints for students after they completed open-ended questions from the instructional video and completed challenges. Furthermore, I reminded myself to make sure that rulers were clearly placed and referenced whenever students used the robots, which would permit me to observe if students used the rulers or if they focused on the robot rather than on using the learning aids provided. Open-ended questions were printed out to administer to students at the designated time. Hardcopies were used to ensure I received all students' work. With students being in different grades, I did not want to make assumptions about students' prior experiences and knowledge. Interview questions were used to inquire about students' prior experience with the visual cue of color-coding. Pre-assessment interview questions about visual-cue color-coding were administered before any lesson in Cycle 3. A copy of the pre-assessment interview questions for color-coding can be found in Appendix T. Post-interview questions were developed to be administered immediately after the *Arm Position Challenge* and used to collect data on students' perceptions, understanding, and suggestions for using color-coding to program robots. A copy of the post-interview questions can be seen in Appendix W.

Since LEGO Mindstorms programming blocks were already color-coded, I designed instructions to be color coded to correspond to the actions required by the robots. Guided practice was used to show students examples of color-coded to correspond to the programming blocks. Font color was changed or the background color of text was colored to correspond with a specific programming block. When font color was not easily distinguishable the background color of text was changed instead.

Acting

The *Touch 1* lesson introduced students the purpose of the touch sensor and required answers to two open-end questions. The *Touch 2* lesson was only a diagram of how to construct the robot for the Touch lessons.

The objective for *Touch 3* activity was to program the robot to wait until the touch sensor was pressed before executing the remainder of the program. Students watched an instructional video on how to program the robot to wait for its sensor to be pressed before executing any other actions. Two open-ended questions were used to check understanding. There was no mini-challenge for the Touch 3 lesson.

The *Touch 4* lesson instructed students how to program the robot to move forward until its touch sensor was pressed. After detection of the touch sensor being pressed, the robot will stop moving. Students watched an instructional video and two open-ended questions to check understanding. The *Touch 4* lesson included the only mini-challenge for Cycle 3, *Vacuum*. The instructions for the *Vacuum* mini-challenge were color-coded to correspond to the corresponding LEGO Mindstorms (2017) programming blocks. The color-coded instructions were posted online for students.

The *Touch 5* lesson reviewed the previous lessons *Wait for Touch*, *Forward Until Touch*, and *Vacuum*. Students were shown the programming solution to these activities. Students were encouraged to ask questions to clarify any concerns that they had. The *Touch 5* lesson did not have an instructional video, open-ended questions, or a challenge.

The *Arm Position Challenge* was the last activity for Cycle 3. For this challenge students were required to program the robot to raise the arm to the "*up*" position, move forward to the cargo, lower the arm to secure the cargo, and return to the starting point.

Observations. Observation data was collected, in Cycle 3, in the form of field notes and notes on duration and frequency, as I completed in Cycle 1 and Cycle 2. Duration and frequency were calculated using the same method as in the previous Cycles. I observed students easily understanding how to use the color-coded instructions to program the robot. Since the color-coded instructions were posted online, I observed many students placing the color-coded instructions on one side of their computer screen and had the programming screen on the other side. This allowed students to simultaneously view the color-coded instructions and the programming screen at the same time. This technique is further discussed in the interview section.

Cycle 3 dealt with the technical issue of no sound playing on the computers. Security software is installed on the computers to allow teachers to monitor students' actions on the computer. This software controls the sound on the computers. While the software was updating, no sound was available on the computers. Some students used their headphones to listen to the videos. I was able to play the videos from my computer and show them using the promethean board in the front of the class. I replayed the video whenever students asked.

I compiled statistics in Cycle 3 similar to the manner I completed in Cycle 1 and Cycle 2. The mean score for the *Touch 1* lesson was 2.63 (SD = 0.64), with a completion time of 1.05 minutes (SD = 5.94). Activity *Touch 3*'s mean score was 2.13 (SD = .96), and its completion time was 0.99 minutes (SD = 0.77). The mean score for *Touch 4* was 2.59 (SD = .63), and its average completion time was 1.85 (SD = 0.74). The *Vacuum* mini-challenge was the only mini-challenge for Cycle 3. The mean time for completing the *Vacuum* mini-challenge was 9.41 minutes (SD = 1.79). Three groups completed the *Arm Position* challenge. The mean time for completing the *Arm Position* challenge was 29.33 minutes (SD = 10.97). The mean time for students using the color-coded chart was 4.19 minutes (SD = 1.98).

Cycle 3 high standard deviation scores for open-ended questions were similar to those of Cycle 1 and Cycle 2. The *Touch 1* instructional video was 0:42 minutes long and three groups completed the questions in minimal amount of time (0:25, 0:37, and 0:45 minutes, respectively). The *Touch 3* instructional video was 1:12 minutes long and three groups completed the open-ended questions in a shorter time than the video length (0:17, 0:30, and 0:45 minutes, respectively). The *Touch 4* instructional video was 1:23 minutes long and two groups completed the open-ended questions in a shorter time than the video length (0:37 and 0:43 minutes, respectively).

A majority of the scores for Cycle 3 were at least a 2 or higher based on Biggs and Tang's SOLO taxonomy (2011). Scores of 2 indicated that students understood or identified one concept of the objective. The average scores for Cycle 3 were as follows: *Touch 1* was 2.63, *Touch 3* was 2.38, and *Touch 4* was 2.60. I perceived the scores for the open-ended questions to be low due to of lack of effort. I observed some students answering the questions without watching the video. The mean times for color-coding instructions were high. Students found this strategy very effective.

Interviews. Interviews were used to collect information about students' views of using color-coding with computer programming and to clarify observation data. Prior to beginning *Touch 1* lesson, students were interviewed about their prior experience using the visual cue color coding, in school, the classes they were used in, and their level of expertise using them. A copy of this form can be seen in Appendix V. The interview revealed that 3 of 14 students had never used color coded visual cues before. Students who said they had used color-coding before in class indicated they had used it in their core classes (e.g., literature, science, social studies, and mathematics) and believed they had a good understanding on how to use it. Prior experience with color-coding helped students with programming the robot. Students who had never used color-coding before stated that it was easy for them to learn how to use color-coding when programming the robot.

I conducted follow-up questions with students during the last five minutes of class, similar to the questions I completed in Cycle 1 and Cycle 2. Open-ended questions were asked to help me understand students' actions and thoughts. While reviewing data during reflection period, I discovered new themes. Students stated using color-coding with computer programming as beneficial and offered the following insight:

(a) Color-coded visual cues simplified instructions by identifying key concepts;

(b) Color-coded visual cues could be used as a reference guide.

(c) Color-coded visual cues helped organize thoughts.

Disadvantages expressed were that colors can look the same and that students could be colorblind. Most students stated that no changes were needed with the color-coding techniques used with the challenges. Another notable suggestion from most of the students was to use color-coding with instructions numbered sequentially. Each numbered instruction would require only one action. Individual steps listed chronologically helped students focus on one concept at a time. One group six member said that he had looked at the color-coded instructions only one time and preferred to create the program on his own.

Reflecting

All students expressed that the numbered, color-coded instructions were helpful. I believed students' prior experience with color-coding, in other classes, greatly helped them in Cycle 3. In Cycle 1 and Cycle 2, substantial time was spent on mastering KWL charts and flowcharts. Cycle 3 required no remediation on color-coded instructions after guided practice. Students were able to concentrate on the color-coded instructions when programming the robots instead. This difference allowed me more time to focus on data collection and making modifications to my instruction based on the data collected. Since Cycle 3 had only one mini-challenge, the only chance I had to try to implement ordered instructions was with the *Arm Position challenge*. The insight gained from Cycle 1 and Cycle 2 of students benefitting from organized instructions encouraged me to order the color-coded instructions. Figures 4.3 and 4.4 displays the instructions from the *Arm Position challenge*.

Version 1

When the "Up" button on the front the EV3 is pressed, the robot must raise its arm to the "Up" position, **wait for the sensor to be pressed** then move forward to the cargo (), and bring it back to the original starting position ().

Figure 4.3. Version 1 of instructions for the *Arm Position* challenge.

I wanted to incorporate order steps with color-coding and see if the two techniques could benefit students' learning, as seen in Figure 4.4.

Version 2

1. **Robot is waiting for the "Up" button to be pressed**
2. Robot raises its arm because the "Up" button was pressed
3. **Robot waits for the sensor to be pressed**
4. Robot moves forward to the cargo
5. Robot lowers its arm
6. Robot moves backward dragging the block with it to the Start point

Figure 4.4. Version 2 of instructions for the *Arm Position* challenge.

None of the students complained about using color-coding during Cycle 3, but during Cycle 1 and Cycle 2 students repeatedly complained about the KWL chart and flowchart. Most students were candidly observed using the color-coded instructions. The ordered color-coded chart was a success.

At the end of Cycle 3, I reflected on how color-coded instruction influenced students' learning programming. Students were able to identify the correct programming block easily and quickly thanks to color-coding. Prior experience with color-coding probably encouraged students to use the technique. Most students stated that they actively used color-coding in their other classes (e.g., chemistry, literature, and social studies). I theorized that the ordered, color-coded instructions set boundaries for students to help focus their attention. Once students focused their attention on relevant concepts of the challenges, they were able to effectively work through the assignment. Even though only three groups completed the *Arm Position* challenge, I was able to pinpoint use what I learned in Cycle 1 and Cycle 2 to help address students' needs in Cycle 3.

Developing

Based on observations, interview data, and field notes, I developed a design guide to use when creating color-coded instructions for students learning to computer program with LEGO Mindstorms EV3 robots. First, instructions for the task should be divided into one requirement or criterion to be met and placed on individual lines. Next, instructions should be numbered in chronological order. Finally, the instructions should be color-coded to the corresponding color of the appropriate programming block. An example of this design guide can be seen in Appendix Y.

Findings

This section explores the themes found during data collection and analysis and how these themes related to the research questions for this action research study. The purpose of this action research study was to improve my instruction by developing custom memory/organization strategies to help my students learn computer programming with LEGO Mindstorms EV3 (LEGO, 2017). Three research questions guided me when developing custom memory/ organization strategies.

4. Did my instruction improve when using KWL charts to teach computer programming with LEGO Mindstorms?
    g. What was the influence of using KWL charts on students' performance?
    h. What were students' perceptions of using KWL charts?
    i. What were the observed patterns and frequency use of KWL charts by students?
5. Did my instruction improve when using flowcharts to teach computer programming with LEGO Mindstorms?
    a. What was the influence of using flowcharts on students' performance?
    b. What were students' perceptions of using flowcharts?
    c. What were the observed patterns and frequency use of flowcharts by students?
6. Did my instruction improve when using visual cues to teach computer programming with LEGO Mindstorms?
    g. What was the influence of visual cues on students' performance?
    h. What were students' perceptions of using visual cues?
    i. What were the observed patterns and frequency use of visual cues by students?

Table 4.2 identifies the research questions for this study and corresponding findings.

Table 4.2

Research Questions and Corresponding Findings

Research questions	Findings
1.Does my instruction improve when using the memory/organization instructional aid of KWL charts to teach computer programming using LEGO Mindstorms?	
a. What is the influence of using KWL charts on students' performance?	• All students were successful in summative assessment
b. What are students' perceptions of using KWL charts?	• Students indicated: it is useful for reference guide and note taking • Helps organize thoughts • Helps track progress • Time consuming
c. What are the observed patterns and frequency use of KWL charts by students?	• Students were hesitant to use • Minimal time spent using • Low rate of use, Mean 1.31 minutes (SD = .49) • Students preferred trial-and-error over instructional aids
2.Does my instruction improve when using the memory/organization instructional aid of flowcharts to teach computer programming with LEGO Mindstorms?	
a. What is the influence of using flowcharts on students' performance?	• None of the students completed summative assessment
b. What are students' perceptions of using flowcharts?	• Simplified programming • Useful as a reference guide • Helps organize • Time consuming
c. What are the observed patterns and frequency use of flowcharts by students?	• Students walked-out flowchart to verify • Students preferred trial-and-error over instructional aids • Low rate of use, Mean 3.41 minutes (SD = 1.79)
3.Does my instruction improve when using the memory/organization instructional aid of visual cues to teach computer programming with LEGO Mindstorms?	

a. What is the influence of visual cues on students' performance?	• Mixed results, 50% completed summative assessment
b. What are students' perceptions of using visual cues?	• Simplified programming • Useful as a reference guide • Helps organize
c. What are the observed patterns and frequency use of visual cues by students?	• High rate of use • Visual cues used as a reference guide by pairing it on-screen with programming software • High rate of use, Mean 4.19 minutes (SD = 1.98)

Research Question 1

Research Question 1 asked: What was the effect of KWL charts when teaching programming with LEGO, based on students' performances, perceptions, frequency, and observed patterns of use?

Cycle 1 demonstrated the influence of my instruction based on students' feedback, observations, and completion of the summative assessment. Using various data collection sources enabled me to verify results. Most students had no experience with computer programming, KWL chart, or LEGO Mindstorms (2017) but by the end of Cycle 1 all students could use each successfully. Students learned how to track their own progress with the KWL chart. I was me being able to design and deliver instruction and check for understanding in a consistent manner. Based on students' feedback I was able to design a custom KWL chart to help students learn computer programming.

Research Question 2

Research Question 2 asked: What was the effect of flowcharts when teaching programming with LEGO, based on students' performances, perceptions, frequency, and observed patterns of use?

None of my students completed the summative assessment for Cycle 2. However, the data I collected indicated that I was able to improve students' understanding and use of flowcharts. Only six out of 14 students indicated that they had used flowcharts before. Early evaluation of the students' use of flowcharts showed that they all had little to no understanding of flowcharts. By the end of the Cycle 2 all students could use a flowchart successfully. Students understood the benefits of flowcharts when they started to physically recreate their flowchart in person (i.e., walk through it) to check its accuracy. Based on students' feedback and observations I was able to design a flowchart that students could easily learn and use.

Research Question 3

Research Question 3 asked: What was the effect of color-coding when teaching programming with LEGO, based on students' performances, perceptions, frequency, and observed patterns of use?

Only 50% of students completed the summative assessment. However, with the group who responded, I was able to improve the influence of my instruction using color-coding. Students' feedback and observation indicated they were able to identify the correct programming block easily and quickly. Three students mentioned that they never had used color-coding before, but by the end of Cycle 3 all students were successfully using color-coding with their assignment. Students suggested using color-coding with an ordered-list of instructions. This approach helped them check their progress and simplified steps.

CHAPTER 5

SUMMARY, CONCLUSION, AND IMPLICATIONS

This chapter summarizes findings, discusses conclusions, and offers implications for practice and research learned on results of the action research study I conducted examined the impact of various custom memory/organization instructional aids on the achievement of high school students. The study was conducted in a web design class using LEGO Mindstorms (2017) robotics in Georgia. This study focused on the following research questions:

1. Did my instruction improve when using KWL charts to teach computer programming with LEGO Mindstorms?
    a. What was the influence of using KWL charts on students' performance?
    b. What were students' perceptions of using KWL charts?
    c. What were the observed patterns and frequency use of KWL charts by students?
2. Did my instruction improve when using flowcharts to teach computer programming with LEGO Mindstorms?
    a. What was the influence of using flowcharts on students' performance?
    b. What were students' perceptions of using flowcharts?
    c. What were the observed patterns and frequency use of flowcharts by students?
3. Did my instruction improve when using visual cues to teach computer programming with LEGO Mindstorms?
    a. What was the influence of visual cues on students' performance?

b. What were students' perceptions of using visual cues?

c. What were the observed patterns and frequency use of visual cues by students?

Summary

Advancements in technology have allowed for new instructional strategies that can attract and teach science, technology, engineering, and mathematics (STEM) principles to students who face increasing demands for technology skills in work settings (Bos, Ortiz, & Smith, 2015; Trivodaliev et al., 2017). Geist (2016) suggested that computer programming should be taught to all students because computers are pervasive in everyday life. Computer programming teaches students to analyze an issue, produce an algorithmic solution, and then translate this algorithm into program code (Saeli, Perrenet, Jochems, & Zwaneveld, 2011). However, learning computer programming often poses difficulties for students because of its complexity and novelty. Teachers can also experience difficulty when teaching computer programming.

Educational robots have been promoted as a way to attract and encourage students to enroll in K-12 STEM classes (Bos et al., 2015). Alimisis and Kynigos (2009) described educational robots as a learning tool that teachers can utilize to enhance and motivate learning in the classroom. Educational robots offer several educational advantages; they are interdisciplinary, engaging, open-ended, problem-based, and realistic, and they can make students' learning more concrete (visible) (Alimisis, 2009; Petre & Price, 2004). LEGO Mindstorms (2017) robots are preferred by many secondary teachers because of their tangible, practical nature and their flexibility to challenge students of various skills and ages (Pasztor et al., 2010).

Educators must be trained to understand new technology, but also new instructional and assessment practices, including an understanding of why assessment changes are warranted, and prospective benefits of these changes (Snoeyink & Ertmer, 2001; Stein, McRobbie, & Ginns, 1999). Teachers who lack specific technology-based knowledge, skills, and pedagogy are likely to avoid teaching new technologies (Ertmer, 2001). Teachers with limited knowledge of new technologies will have a hard time developing lessons, supporting students, and assessing students (Stein et al.,1999).

Teaching Strategies

Instructional strategies encompass a variety of teaching approaches, teaching tools, and materials that educators use to accomplish their objectives (Ibrahim & Omwirhiren, 2016). Instructional strategies are essential to teaching effectiveness because they can promote learning and student engagement (Ameh & Dantani, 2012; Marzano, 2003). Using proper instructional strategies can help students in organizing, remembering, monitoring, and assessing their own learning (Sangoleye, 2016). Teachers should consider allotted time, students' needs, various techniques, and the effectiveness of each concept when selecting an instructional method (VanTassel-Baska, 2003).

Instructional strategies can be categorized as either teacher-centered or student-centered. Teacher-centered strategies tend to make students passive listeners, sitting, copying, memorizing, and repeating information delivered by the instructor. Student-centered strategies involve students as active participants in learning through inquiry, problem-solving, and critical thinking through real-life experiences, collaboration, and discussion (Arseven et al., 2016; Bruner, 1996; Garrett, 2008).

Papanikolaou and Frangou (2009) recommended that a robotics curriculum should utilize the principles of constructivism, constructionism, and problem-based learning. Constructivism does not have an universal definition, but is viewed as a learning theory, an epistemology, and a theory of pedagogy (Amineh & Asl, 2015). Constructivism as a learning theory suggest that learners create new knowledge by making connections between prior knowledge and new information (Anderman & Anderman, 2009). Constructionism is a learning theory based on constructivism principles but emphasizes learning through constructing meaningful and tangible items in a social environment (Papert, 1993; Stager, 2001, 2003). Stager (2001) asserted that constructivism could be viewed as learning by doing, while constructionism could be seen as learning by making. Problem-based learning is another constructivist learning theme, in which students create meaningful artifacts to demonstrate their understanding as they try to answer a leading question. Leading questions help students understand the relevance of their task (Anderman & Anderman, 2009; Herreid, 1991).

Theoretical Framework

This action research study used Ausubel's (1968) meaningful learning theory as its framework, which view learning from two aspects; rote learning and meaningful learning. Rote learning is passive and relies on short-term memorization instead of comprehension. It is difficult to use information acquired through rote learning in meaningful ways (Ausubel, 1963; Daunoriene & Katiliute, 2011; Iqbal & Ahmad, 2015; Mayer, 2002). Meaningful learning involves understanding and retaining new information in memory by creating relevant links between prior knowledge and new knowledge (Ausubel, 1963; Mayer, 2002). Organizing and linking new information allows for long-term memory storage by creating and using association cues for accessing information (Gurlitt & Renkl, 2008). Wetzels, Kester, Merrienboer, and Broern (2011) explained how knowledge stored in memory dictates how easily and accurately it can be used. Concepts stored in memory are linked with other concepts based on associations. Learning is encouraged and strengthened by repeated use of the knowledge schemes. This approach favors students constructing their learning through activities for more meaningful and enduring knowledge instead of rote memorization (Arseven, Nakiboglu, & OzAydin 2016). Meaningful learning helps students apply what they have learned to other situations (Pellegrino & Hilton, 2012). Martinez et al. (2016) indicated that being able to apply knowledge to other situations is important because it helps learners to be flexible, creative, and innovative in future classrooms, jobs, and in the ever-changing world.

Marton and Saljo (1976) characterized students' learning it as either deep learning and surface learning. Surface learning, similar to rote learning, is teacher-centered, relies on rote memory, and emphasizes the reproduction of facts instead of analysis of information. Surface learning may appear differently in different areas, but will consistently result in lower grades compared to deep learning (Ramsden, 1992). Deep learning, similar to meaningful learning, is student-centered, promotes student engagement, and encourages conceptual development (Entwistle et al., 2001). All students are capable of both surface and deep learning, but learning intentions plays a major role in the approach (Ramsden, 1992). Entwistle (1997) asserted that the quality of teaching can influence students' learning approach. Claiming that when teachers' lessons were relevant to students' experience, they encouraged deep learning. Pellegrino and Hilton (2012) claimed deep learning allows learners to transfer knowledge to other situations. *Transferable knowledge* is the ability to answer questions and solve problems, apply knowledge in other situations, and understand how, when, and why to use the knowledge. Transferable knowledge benefits learners by enabling them to be flexible, creative, and innovative. Pellegrino and Hilton (2012) stated that transferable knowledge is supported by instruction that promotes deep learning but is not supported by rote learning.

Biggs and Collis's (1982, 1989) Structure of Observed Learning Outcome (SOLO) taxonomy was also used to provide a systematic way to measure learners' depth of knowledge of academic tasks. The SOLO taxonomy has five levels of depth of knowledge and students' understanding: (1) prestructural, (2) unistructural, (3) multistructural, (4) relational, and (5) extended abstract. Unistructural and multi-structural knowledge are viewed as products of surface learning, while relational and extended abstract knowledge are considered products of deep learning (Hattie & Brown, 2004). Biggs and Tang (2011) suggested deeper learning over surface learning to help students proceed through the levels of the SOLO taxonomy. Shuhidan et al. (2009) suggested using the SOLO taxonomy to assess student's understanding of computer programming because it provides a method to evaluate thinking. The SOLO taxonomy was used to evaluate students' understanding of the open-ended questionsimplemented to teach students how to program their robots. Ramsden (1992) suggested that students' learning intention would influence whether a deep or surface learning approach was used. The scores collected from the open-ended questions were compared with students' results from the programming challenges for any patterns or correlation. I used this information to determine if students' SOLO taxonomy rating was an indicator for the success or failure of the programming challenges. Table 5.1 shows the SOLO taxonomy's five levels of depth of knowledge and students' understanding.

Table 5.1

Levels of SOLO Taxonomy

	Depth of knowledge	Level of Understanding
1	Prestructural	Lack of knowledge or understanding; Shows minimal evidence of relevant learning.
2	Unistructural	Answer satisfies one aspect of the learning task, but misses other important factors.
3	Multistructural	Response fulfills various features, but does not address key issues.
4	Relational	Answer has multiple parts correlated and the correlation between the various aspects can be qualitatively explained (e.g., when and why).
5	Extended abstract	Answers show learning beyond the relational stage by exhibiting high-order thinking skills and applying knowledge to other context (transferrable knowledge).

Research Design

This study used an action research design to systematically investigate my instruction, analyze and interpret teaching-learning events, and plan for and implement a solution (Stringer, 2007). Lewin (1946) developed action research, which consists of four-stages, inc., planning, acting, developing, and reflection. There are many versions of action research, but they all generally have these same basic steps: problem or issue identification, data collection, reflection, and analysis action (Adleman, 1993; Creswell, 2012; Mills, 2011). Mertler (2014) explained that teacher-researchers may proceed in the steps of action research in no definite order and repeating steps several time. Action research depends on the researcher, purpose of the study, and participants (Sigler, 2009). Sigler (2009) described action research as teachers' personal endeavors to explore and implement new ideas to improve students' learning or a specific behavior in the classroom. Action research allows researchers to uncover new information that requires further analysis (Sigler, 2009). Based on the data collected and its interpretations, researchers will repeat some steps and skip over other steps to implement new solutions (Creswell, 2012; Stringer, 2007). This action research study consisted of three cycles. During each cycle, a particular instructional aid (KWL charts, flowchart, and color-coding, respectively) was used. Each cycle used a four-stage process consisting of planning (studying and preparing for upcoming cycle), acting (collecting data), developing (analyzing data), and reflecting (reviewing entire process for a better understanding).

Action research has advantages and disadvantages that should be considered before using. Results from action research are relevant to teachers because the activity occurs in their actual classrooms (Charles & Mertler, 2011). McIntyre (2005) suggested that action research can help researchers/teachers to make connections between research and educational practices by testing and verifying claims. A drawback to using action research is that personal biases can pose an issue, as researchers may have a difficult time remaining objective when the data collected does not validate their practices (Mills, 2011). Another disadvantage of action research is that it is time-consuming and teachers may need training (Creswell, 2012).

Planning. During the planning stage, I gathered and prepared instructional materials (e.g., robots, rulers, and computer paper). I also developed pre-interview questions to ask students' about their previous experience with LEGO Mindstorms, computer programming, and the instructional strategies selected for this study (i.e., KWL charts, flowcharts, and visual cues color-coding).

Acting. The acting stage consisted of collecting and analyzing data. Students watched instructional videos and answered open-ending questions. One of three instructional aids (i.e., KWL chart, flowchart, and visual cue color-coding) were introduced over three successive cycles to support the lessons pertaining to robotic programming challenges. I collected data from students' observations, interviews, and field notes, and then cross-referenced the data to verify my findings.

Reflection. The reflecting stage involved repeatedly reviewing data from observations, classwork assignments, field notes, and the processes used to collect data. Data was searched for reoccurring events and phrases from students' feedback and observed behavior. Identified patterns from collected data determined areas to further investigate to gain a better understanding of the phenomena. Reflection occurred before, during, and after all activities. I assessed the effectiveness of my teaching on the task of programming a robot and examined the influence of introducing different instructional aids during the process.

Developing. The developing stage consisted of revising my instruction and instructional aids based on insight gained from reflection on outcomes obtained from the previous cycle. Constant reflection allowed me to make timely changes and evaluate the impact of these changes.

Participants

This study was conducted in an urban high school, in Augusta, GA, with a population of approximately 1,300 students. I used a convenience sampling of my fifth-period class. The class was composed of 16 students whose ages ranged from 15 to 18; two 10th-graders, ten 11th-graders, and four 12th-graders.

These students were chosen because they were scheduled to enroll in AP Computer Science after successfully passing Web Design. Web Design is one of 17 different career areas available in the high school curriculum, where students can explore various career pathways (Georgia Department of Education, 2017). Pathways are composed of three year-long, comprehensive classes created to help students explore career areas and prepare them for employment (Scott & Sarkees-Wircenski, 2004).

Findings

The purpose of this action research study was to improve my classroom instruction by creating and administering three custom memory/organizational aids (KWL charts, flowcharts, and visual cue color-coding) to teach computer programming with LEGO Mindstorms (2017). I evaluated the effectiveness of my custom memory/organizational aids using three criteria; (a) students' performance, (b) students' perceptions, and (c)
observed patterns and frequency of use by students.

Research Question 1

Did my instruction improve when using of KWL charts to teach computer programming with LEGO Mindstorms?

a. What was the influence of using KWL charts on students' performance?

b. What were students' perceptions of using KWL charts?

c. What were the observed patterns and frequency use of KWL charts by students?

Students' initial use of the KWL chart proved troublesome because they were reluctant to complete the KWL chart and when it was completed, was done incorrectly. I observed on several occasions students not attempting to complete the KWL chart until the end of the period, when I began to collect them. Some students were focusing on irrelevant aspects of lessons. On several occasions, students wrote in the *What I Want to Know* section about concepts not pertaining to the lesson (e.g., sensors, turning, and sounds). Students' interview feedback identified that only 1 of 14 students had used a KWL chart before. Students' tentativeness in completing the KWL charts made collecting accurate data for duration and frequency difficult. Students required additional guidance and practice in constructing their KWL charts to help focus their thinking on relevant aspects of the task.

Once students understood how to use KWL charts, its use proved beneficial when programming the robots in Cycle 1. The KWL chart helped students take more control over their learning. For example, students were able to pace their learning by identifying what they needed to learn to complete the task and track their progress. Students stated various benefits of using the KWL chart, including that they: (a) were good for note-taking and as a reference guide, (b) helped organize thoughts, and (c) helped track progress. Students' views of KWL charts were similar to the assessments of Cassady et al. (2004) who found that they could help students gauge their learning as they progressed through an activity. All students in my class were able to complete the summative assessment, *Sensabot Challenge*. Three students reported negative views of the KWL chart, saying: "*I find it a waste of time,*" "*I don't need it,*" and "*It takes too long to complete*."

At the completion of Cycle 1, students had a good understanding of the purpose and proper use of KWL charts. Students offered several suggestions to improve the KWL chart I used in class, such as adding additional spaces for the *What I Learned* section. This added feature would allow students to track their learning and record notes. Students' feedback led me to create the *K(WL)^n* chart. A copy of this form can be seen in Appendix S. This chart's name is based on its design. I moved the *What I Know* section to beneath the topic but above the *What I Want to Know* and *What I Learned* sections. I also redesigned the *What I Want to Know* and *What I Learned* sections with numbered spaces to allow students to use the form multiple times.

I observed students' patterns and frequency of using KWL charts. Trying to observe and record accurate data on students using the KWL charts was difficult because they were not consistently on task. I frequently observed students on their cell phones or socializing with other groups instead of focusing on the day's assignment. I also observed that some students did not complete the KWL chart. In some instances, students did not attempt to fill out any sections of the KWL chart until the end of the period, when I started to collect them. The observed average period using the KWL chart was 1.31 minutes (SD = 0.49). I attributed the low use of the KWL chart to students' lack of interest in it. Two to three minutes probably would have been a more reasonable average time of use of KWL charts, if students took their time.

Research Question 2

Did my instruction improve when using flowcharts to teach computer programming with LEGO Mindstorms?

a. What was the influence of using flowcharts on students' performance?

b. What were students' perceptions of using flowcharts?

c. What were the observed patterns and frequency use of flowcharts by students?

Students' lack of experience using flowcharts caused substantial difficulty helping students learn the design process necessary to create a flowchart. Eight of the 14 students reported previous experience using a flowchart, but only 2 had used a flowchart in a high school class.

Students required additional guided-practice and simplification of the flowchart to help them understand the process. Students became confused trying to understand the various symbols typically used in a flowchart. A premade flowchart lesson was included with the *Introduction to Programming LEGO Mindstorms EV3*. This introduction lesson included one example of how to design a flowchart. Students did not find this tutorial lesson helpful and would have benefitted from more examples on the design process of a flowchart and additional explanation of the symbol blocks. I observed students using the diamond and rectangle block symbol incorrectly and students expressed their difficulty understanding how to use them. The diamond block symbol represented a question or a decision point in the flowchart. The answer to the diamond block symbol determined the next step in the program flow. Students used the rectangle (statement) block symbol to represent multiple actions or behaviors instead of one at a time.

To address the difficulty that students experienced, I limited the number and types of symbols to two shapes. The oval symbol was used to represent the start and end of a process, and a rectangular symbol was used to represent an action. With additional guided practice, the changes I implemented seemed to help. After making these changes, I observed various groups walking out the steps of their flowchart to check its validity. Students' feedback on the new flowchart design process was positive and included the following insights:

(a) A flowchart made programming easier or more manageable by breaking it down into smaller steps. The smaller steps allowed students to focus on one thing at a time;

(b) The diamond block symbol should be avoided because it was confusing and difficult to use;

(c) A flowchart would be helpful for first-time learners of programming and flowcharts;

(d) Flowcharts can be used as a reference; and

(e) Flowcharts help organize thoughts.

My findings from Cycle 2 were consistent with those of the current literature on flowcharts. Charntaweekhun and Wangsiripitak (2000) and Dol (2015) showed that flowcharts could be used as a reference guide because each step is visual, and it is easy to monitor the execution of the program. This clarity made programming easier for them because it allowed students to identify errors in their program and make needed changes. Even though students were not able to complete the *Orchard Challenge*, I attributed this to the extended time used to teach students about the flowchart design process. However, students were successful in completing the *90-Degree Turn* and *Dizzy Drill* mini-challenge. Although flowcharts are known to be useful when teaching beginners how to computer programming, it is worth noting that learning how to design and use a flowchart was just as important.

Research Question 3

Did my instruction improve when using visual cues to teach computer programming with LEGO Mindstorms?

a. What was the influence of visual cues on students' performance?
b. What were students' perceptions of using visual cues?
c. What were the observed patterns and frequency use of visual cues by students?

Students experienced immediate success using the color-coded visual cues during initial guided-practice. Eleven of 14 students indicated they had used color-coded visual cues before, with most experiences being with color-coded visual cues in classes (e.g., literature, science, social studies, and mathematics). The three students with no prior experience did not experience any difficulties learning to use the color-coded instructions. Students easily understood the color-coded instructions, which allowed me additional time to design various versions of instructions. I used a traditional sentence format with color-coding and ordered, color-coded instructions separated into individual steps. Steps were listed like a recipe, with each step requiring one action. Sentences could be color-coded since each sentence required only one action. Sentences' colors corresponded to one of the programming blocks' colors (i.e., green, yellow, orange, or red).

It was easy to observe students using the color-coded instructions, which were posted online. Students took advantage of this arrangement by placing the color-coded instructions on one side of their computer screen and the programming screen on the other side. Thus, students were able to simultaneously view the color-coded instructions and programming screen at the same time. I believe that students' prior experience encouraged them to use it with their programming lesson. Moreover, Cycle 2 had shown me that students benefitted from instructions being broken down into individual steps, and I wanted to try and implement this same process with color-coding. All the students chose the ordered, color-coded instructions over the sentence-formatted instructions.

Students' interview data about color-coded visual cues helping them identify key concepts while programming was consistent with the literature on (Ewoldt & Morgan, 2017; Jamet, 2013; Kalyuga et al., 1999; & Sarkar, 2015). All the students expressed positive views on using color-coded visual cues, including the following:

(a) Color-coded visual cues simplified instructions by identifying key concepts;

(b) Color-coded visual cues could be used as a reference guide; and

(c) Color-coded visual cues helped organize thoughts.

Mayer (2009) stated that signaling with visual cues can help learners identify key concepts quickly by allowing them to avoid considering all information. Ewoldt and Morgan (2017) suggested that color-coded graphic organizers helped elementary students with learning disabilities in writing by using colors to sow relationships between undeveloped ideas and complete sentences in the paragraph. Jamet's (2013) study showed that color-coding helped students reduce search time and processing time.

Limitations of Study

Programming Blocks

Programming blocks are "visual" and "component-based" concepts that allow programmers to create computer programs by assembling these blocks to visually represent programming constructs (Mihci & Donmez, 2017). Programming blocks were used in this study, so results may not be comparable to programming tasks using traditional source code. However, using programming blocks allowed beginners to gain a basic understanding of programming and to avoid the complex concepts of data structures, algorithms, and syntax that have become major issues in retaining students in programming classes (Alvarez & Larranaga, 2014; Burbaite, Damasevicius, & Stuikys, 2013). Chetty and Barlow-Jones (2012) indicated that block-based programming is effective for introducing students to computer programming, but blocked-based programming is not suitable for designing code for business applications. Therefore, if students are to be successful computer programmers, they must eventually learn text-based programming.

Time Constraints

Time was a constraining factor for me due to the multiple tasks that students had to complete on a daily basis. Students' lack of understanding of KWL charts and flowcharts required additional time. This left less time for observations, interviews, and challenges on some days. Replication of this study can avoid the issue of time constraint by extending lessons to two days instead of one day.

Curriculum Design Shortcoming

The *Introduction to Programming LEGO Mindstorms EV3* (2017) curriculum posed shortcomings that I had to address. Multiple-choice questions were used in the curriculum to check students understanding of the instructional videos. I converted these multiple-choice questions to open-ended questions to measure students' depth of knowledge from the instructional videos. Furthermore, some of the mini-challenges were easy for students to solve and did not assess students' understanding of concepts sufficiently to prepare them for the final challenge in Cycle 1 and Cycle 2. I was able to identify the short mini-challenges by observation and interviewing students after mini-challenges and the final challenge for each Cycle.

Implications for Practice

Custom Graphic Organizers

This action research study adds to the extant literature on the use of custom graphic organizers to teach computer programming to high school students. Existing research states the purpose and benefits of graphic organizers, but does not address how to evaluate custom graphic organizers (Dimitri, 2015; Mihardi, Harahap, & Sani, 2013; Ogle, 2009; Sarkar, 2015; Xinogalos 2013). This study showed how to use observation, interviews, field notes and reflection to collect relevant information on custom graphic organizers' (e.g., influence on students' performance, students' perception on the custom graphic organizers, and observed patterns and frequency of use). Insight gained from the data I collected was used to modify the graphic organizers based on students' needs and suggestions.

McIntyre (2005) stated that action research allows teachers to make connections between research and practice by testing and verifying claims. Often, there is a gap between results reported by researchers on educational topics and the practice of classroom teachers (Mertler, 2014). Kennedy (1997) argued that researchers don't always address educators' concerns about teaching and that research is not usually conducted in settings comparable to most classrooms. Furthermore, educational research is produced with generalized results that have little or no relevance to particular contexts, and practitioners may not know how to access and use educational research (Broekkamp & Hout-Wolters, 2007; Stevens, 2004; Stringer, 2007). This study shows how I created custom memory/organization instructional aids, and thus adds helpful additional instructional information to the current literature.

Personal Professional Development

This action research study was conducted to improve my educational practice and knowledge through a systematic examination of my practice (Charles & Mertler, 2011; Keegan, 2016). Students provided valuable information on how they preferred larger tasks to be reduced or chunked into smaller more manageable tasks. During Cycle 2, students stated how the custom flowchart was beneficial once it was simplified. Focusing on one action at a time helped students manage the programming task. Similar to Cycle 2, students during Cycle 3 opted for programming instructions that were divided into individual tasks. This preference is supported by research that shows making knowledge more manageable by chunking, or minimizing a bulk of information into smaller chunks, does not affect the quantity of new information being acquired (Lah, Saat, & Hassan, 2014). One example of chunking in this study was instructing students to use only one step or action for each shape for the flowchart design process, during Cycle 2. Another example occurred during Cycle 3, when instructions were numbered and divided into individual actions. Students stated during interviews that breaking down the instructions into smaller and more manageable steps made it easier to focus on one thing at a time and helped them find errors. The *cognitive load theory* states that instructional design can support or hinder learning depending on how the material is presented and the actions required by learners to process it (Sweller et al., 1998). Mayer (2017) indicated that teachers can increase instruction effectiveness by segmenting or making important information into smaller, self-paced segments.

Systematic Reflection

This study also allowed me to systematically reflect on my instructional practices to identify and better understand my strengths and weakness. This process is one that may be beneficial for other teachers as well. Thomas and Pascal (2012) argued that many practitioners claim that they do not have time for reflection. However, they argued that the busier practitioners are, the more reflection they need to ensure they are focused on what they are doing, why they are doing it, and have resources available to help them complete the task. Schon (1987) suggested that continuously reflecting on actions can enable new discoveries and more accurate designs by paying attention to a phenomenon, which allows for a better understanding. Consistently and strategically deliberating on my daily activities allowed me to identify successes and failures in my lessons. At the school at which I teach, teachers are required to administer a weekly assessment, and this often causes time constraints. With a short time to introduce a lesson, evaluate students, and submit weekly scores, this compressed time frame can force me to advance to the next lesson while focusing only on the concepts that students got wrong and not on the concepts they understood and why. This study demonstrated that considering why students were successful in Cycle 1 and Cycle 2 allowed for improved understanding of students' thinking and implementing the chunking concept with color-coding in Cycle 3. Mertler (2014) affirmed that through reflection, teachers must examine and diagnose the effectiveness of their practices because examining and responding to the events that occurred during the lesson will determine the next steps the teacher will need to take.

Recommendations for Research

Text-based Programming

I recommend replicating this study using traditional text-based programming instead of the block-based programming technique. Block-based programming is a visual programming concept that utilizes visual action blocks that are combined to represent conventional text-based programming syntax (Mihci & Donmez, 2017). Programming blocks in this study allowed students to focus on the program flow rather than actual source code. A study on the effects of using custom memory/organization instructional aids with traditional text-based programming may give insight into how to create graphic organizers for beginner programmers using text-based programming.

The results in my study with block-based programming do not guarantee that such results will be duplicated with text-based programming. Visual programming is a popular tool in motivating and teaching students about programming, but it does not necessarily mean that students will be successful when moving to traditional text-based programming. Studies by Parsons and Haden (2007) and Powers, Ecott, and Hirshfield (2004) demonstrated that students had difficulties learning traditional text-based programming's syntax after learning visual programming. Powers et al. (2007) suggested that the lack of attention to syntax errors in visual programming will only delay students' difficulty in learning correct programming syntax with text-based programming. Parsons and Haden (2007) warned that visual programming makes teaching and learning programming easier, but it hides important programming concepts so that students may not be learning those concepts at all.

Students' Willingness to Use Aids

I also recommend studying students' willingness to use instructional aids in class. Some of my students were reluctant to use the KWL chart, flowchart, and color-coded instructions. Insight into why students preferred a trial-and-error method over instructional aids may help me and other teachers develop a contingency plan to circumvent students' reluctance when using instructional aids to support programming robots. On the first day of each cycle, I explained the purpose and the benefits of each instructional aid.

During Cycle 1, I observed students not completing any sections of the KWL chart until the end of the class period. Early on in the week, some students complained about having to use the KWL chart. However, at the end of Cycle 1, students were mostly positive about using the KWL chart, and stated that it helped them. Even so, several students continued to display hesitance in using the KWL chart.

Cycle 2 produced similar results to those of Cycle 1. Students expressed unwillingness to complete the design process of flowcharts and use the color-coded instructions. I required students to submit their flowchart design before interacting with the robot. Many students rushed through the design process of the flowchart and submitted a poorly designed, ineffective flowchart.

During Cycle 3, one group stated they preferred not to use the color-coded instructions and wanted to program the robot on their own. They looked at the color-coded instructions only once to ascertain the objective of the activity and then spend the rest of the class period using the trial-and-error method. I observed this group using two to three times more programming blocks than required to complete the programming challenges. Their solution was correct, but their solution could have been more efficient if they had used a minimal number of programming blocks.

Chunking with Computer Programming

My results showed the benefits of using chunking methods. Chunking is the process of reducing large amounts of information into smaller and more manageable sizes without affecting the size of the new information being acquired (Lah et al., 2014). Examples of chunking during this study were visible in all three cycles. In Cycle 1, many students suggested adding more space in the *What I Learned* section that they could use to record notes and use for future references.

An example of chunking that occurred during Cycle 2 was visible during the design process of flowcharts. Due to students' initial difficulty I simplified the design process for flowcharts by reducing the symbols used to one action per symbol. Students stated that reducing the number of symbols used and actions per symbol allowed them to focus on one concept at a time and not become overwhelmed with the design process.

I took insight gained from redesigning flowchart in Cycle 2 and used it when designing color-coding instructions for Cycle 3. Since students stated in Cycle 2 that chunking instructions into individual actions helped them, I made sure to implement this technique with color-coding I in Cycle 3. I developed color-coded instructions that were divided into individual, numbered steps. Students stated that they liked this version of the instructions because it allowed them to focus on one action at a time, which prevented them from becoming overwhelmed when writing the program.

Further research on using chunking with programming may give insight into how to help beginner students learn programming by simplifying programming assignments. Breaking down computer programs into multiple individual-actions may allow students to focus on one aspect of programming at a time instead of a large complex program at once. Bau, Gray, Kelleher, Sheldon, and Turbak (2017) stated that first-time learners of text-based programming have difficulties understanding syntax because it has a high cognitive load. Programming blocks are easier to learn with because they reduce the cognitive load by chunking code into smaller, manageable programming blocks. Bau et al. (2017) argued that cognitive load should be lowered by chunking code because it allows new programmers to focus on the semantics of programming and not the syntax. Cognitive load is reduced because (a) programmers do not have to learn new text-based programming terms; (b) chunked code makes the program easier to read because new programmers do not have to focus on programming syntax and learning new vocabulary; and (c) visually chunked code allows new programmers to focus on the purpose of the code rather than notations.

Conclusion

This action research study was conducted to examine my instruction by creating custom memory/organization instructional aids to teach computer programming with LEGO Mindstorms. The overarching question I posed was, *Did my instruction improve when using custom memory/organization instructional aid to teach computer programming with LEGO Mindstorms?*

Connecting Students' Knowledge with Instructional Aids

Through this study, I observed the importance of connecting students' prior knowledge with the understanding and use of instructional aids. Fisher and Frey (2018) suggested that students would initially experience difficulty using graphic organizers if the lesson's content was not taught beforehand. Students may have difficulty understanding relationships between content when the lesson is first taught (Fisher & Frey, 2018). During Cycle 1, students' initial motivation was to race the robots and learn how to program the robots to turn before they even knew how to turn on the robots. This misdirected enthusiasm was evident from my observations, feedback from students' interviews, and students' KWL charts.

Since students' attention was not focused on the day's learning objective, I had to redirect and reteach students how to use KWL charts to focus their thinking. First, students had to identify their prior knowledge, then understand and identify what knowledge they needed to learn to complete the day's lesson. The KWL chart provided a logical order for progressing through the lesson by eliminating unnecessary steps that students were considering (e.g., racing the robots and programming the robots to turn). At the end of Cycle 1, students stated that the KWL chart helped them organizer their thoughts and was helpful as a reference guide. Furthermore, I observed students being more productive programming their robot by using the KWL chart to center their thinking on a relevant solution.

Cycle 2 was another example of students needing to learn how to use an instructional aid to acquire its benefits. Fisher and Frey (2018) emphasized the importance of students learning to use graphic organizers correctly and not copying a completed graphic organizer from the teacher. Learning how to correctly use the graphic organizer will help students use deep learning instead of surface learning. Students' initial design of the flowchart was very poor. They rushed through the design so they could begin work with their robot. After I observed students' lack of focus on flowcharts design, I required them to present their completed design to me and returned it if it was not completed correctly. Analyzing early flowchart designs showed that students had little understanding of how to design them. After I retaught and reemphasized the purpose of the flowchart, students' flowchart designs improved. Students now spent most of their time constructing their flowchart and walking out the steps to check its accuracy before attempting programming.

Conceptual Importance

New understanding of students' instructional needs allowed me to continue to refine my instruction. Using custom instructional aids led to the discovery and development of new instructional strategies. The initial task of Cycle 3 was to introduce students to visual-cues and give them practice using these cues. I made no assumptions about students' prior knowledge about using visual cues, even after interviewing students revealed that many had prior experience. By teaching visual cues to the class, regardless of my perception of the technique's simplicity and students' prior knowledge of it, I was able to ensure that all students obtained full understanding of visual cues in instructions. I had no remediation of visual cues with instructions which allowed for more time for students' engagement and the collection of more data. At the completion of Cycle 2, I identified that students benefitted when tasks were reduced to smaller, more individualized segments, or *chunking*. This method allowed students to concentrate on one task at a time and made it easier to correct errors. I later used this new discovery in Cycle 3 with color-coded instructions that were chunked to individualize instructions into single actions. Students preferred chunked instructions to traditional, sentence-styled instructions because the chunked instructions allowed them to focus on one aspect of programming at a time, preventing them from feeling overwhelmed when trying to program the robot to complete the challenge.

Repeated reflection allowed me to notice patterns in students' actions and behaviors that led me to further investigate these patterns using the tools I had learned. Exploring these patterns gave me more insight into students' learning strengths and needs. The reoccurring events and phrases from each cycle are identified in Table 5.2.

Table 5.2

Emergent Themes in Cycles

		Themes by data type	
Cycle	Main theme	Observation/ Fieldnotes	Interviews
Cycle 1	Students' lack of use of KWL chart	Students hesitant to use KWL chart Difficulty completing KWL chart correctly Students rushed through complete KWL chart and open-ended question to work with the robot	One of 14 students had prior experience using a KWL Chart
	Students' views of the KWL chart		Advantages • Good for note-taking and reference guide • Helps organize thoughts • Helps tracks progress Disadvantages • Time consuming • Waste of time
	Students' suggestions to improve the KWL chart		Include more space in the *What I Learned* section for note-taking
Cycle 2	Difficulty with the design process of the flowchart	Flowchart designed incorrectly • Multiple actions in individual steps • Difficulty understanding the diamond symbol	• Confused by the multiple symbols • Six of 14 students had prior experience using a flowchart • Two of 14 students had used a flowchart in high school
	Students' views of new design process for flowchart	Students understood	Advantages • Made programming easier by breaking down task into smaller steps • Students suggested avoiding the diamond symbol • Used as a reference • Helps organize thoughts

Cycle 3	Students' use of color-coded instructions	Color-coded instruction used frequently	• Three of 14 students never used color-coding before • Students liked color-coded instructions • Students stated that they had used color-coding before in other classes
	Students' views of color-coded instructions		• Simplified instructions by identifying key concepts • Can be used as a reference guide • Helps organize thoughts • Liked steps being numbered and separated

To reduce the potential for bias, I identified my personal beliefs before starting this study. I predicted that the use of flowcharts and visual cues would positively influence students' academic achievement and their perception of the interventions. I expected to observe a consistent and frequent use of flowcharts and color-coded, visual cue interventions, reasoning that students would benefit from using the custom instructional aids because they would be required to check their progress in intervals. I did not believe that the KWL chart would be a positive influence on students' academics and that, instead, students would have a negative perception of it. My expectations were that students would not like the KWL chart intervention because it does not provide the same guidance throughout a lesson as a flowchart or visual cues. Instead, I expected students to use KWL charts at the initial start of a lesson and not to look at it again.

Conrad and Serlin (2006) and Mills (2011) asserted that researchers can offset their biases by listing their hypotheses. I was incorrect in my initial prediction that the KWL chart would not positively influence students' academic performance and that students would have a negative perception of using it. I was correct in my assumption that flowcharts and visual cues would positively influence students' academic achievement and their perception of the intervention and that students would benefit from using prompts to check their learning.

Identifying my predisposed feelings helped reduce bias because it allowed me to focus on the study objectively and required me to be honest with my findings instead of trying to view my instruction strategies as more effective than they actually are. As a teacher, sometimes I can become frustrated after spending a significant amount of time developing a lesson that turns out to be ineffective with students. My immediate reaction is not always to modify my instruction based on students' needs. Instead, I might try assuming that students had understood a lesson by convincing myself that minimal understanding is equal to actual comprehension of the lesson. Failures in the classroom can be difficult to deal with because they require me to re-evaluate my beliefs, whereas successes are instant gratification. This study showed me that failures are just as important as successes. The failures help me refine my instructional strategies by giving me insight into students' instructional needs.

This study helped me understand the significance of evaluating my assessments, as well as my instruction. During Cycle 2, I was skeptical about the effectiveness of the formative assessments, *90 Degree Turn* and *Dizzy Drill* because the assessments were very short and required minimal programming. The *90 Degree Turn* solution required one programming block. The *Dizzy Drill* required students to program the robot to turn in one direction. I used formative assessments to evaluate students' understanding of programming the robot to turn. None of the students were able to complete the *Orchard Challenge*, but all were able to complete the mini-challenges *90 Degree Turn* and *Dizzy Drill* easily. The *Orchard Challenge* was the designated summative assessment for Cycle 2 and was used to assess students' overall understanding of the objective of programming the robot to turn (Mertler, 2014). Conderman and Hedin (2012) asserted that teachers can use assessments to decide what or how to reteach, assess students' errors, and ascertain students' existing knowledge. With students' difficulty completing the *Orchard Challenge*, the mini-challenges probably should have required students to program the robot to turn to the left and to the right in a single challenge.

Most Valuable Learning Experience from Study

This study provided me with three important learning aspects about designing instruction for students' needs: (a) using research to guide me in improving my instruction, (b) applying data collection techniques, and (c) analyzing students' perception.

The most valuable lesson I learned from this study was using the literature to obtain a better understanding of what is already known about a subject. This insight allowed me to make evidence-based decisions and understand what results to expect and how to analyze them. The literature explained how to use KWL charts, flowcharts, and visual cues to assist students' learning. I continued to use the literature to improve my understanding of how to customize these instructional aids based on students' needs. I referred back to the literature when I noticed the pattern of students benefitting from instructions and tasks being minimized to one action. My assumption about the benefits of this processed was verified when I came across research on the concept of "chunking." The literature identified chunking's purpose, which allowed me to implement this aspect in Cycle 3 instruction.

Using multiple data collection techniques was an important aspect of verifying the validity of the changes implemented in the custom instructional aids. In Cycle 1, I made assumptions about students being off-task, but students' feedback from interviews clarified their actions as related to excitement about using robots and not being off-task. Interviewing students allowed me to become aware of aspects that I was not able to observe directly. Students stated how they used the KWL charts and flowcharts as reference guides.

Analyzing students' perceptions allowed me to make recommended changes to KWL charts, flowcharts, and visual cue instruction. Several studies indicated that students' perceptions are a major influence on what they learned, why they learned it, and how they learned it (Aghamolaei & Fazel, 2010; Bakhshialiabad, Bakhshi, & Hassanshahi, 2015; Marton & Saljo, 1976; Mayya & Roff, 2004). Aghamolaei and Fazel (2010) and Bakhshialiabad et al. (2015) suggested that students' perceptions could be an important factor in implementing interventions because perception can have a major effect on students' academic achievement. Students' perception helped me understand students' needs when designing my custom instructional aids.

REFERENCES

Adams, D., & Hamm, M. (2011). *Shaping the future with math, science, and technology: Solutions and lesson plans to prepare tomorrows innovators*. Lanham, MA: Rowman and Littlefield.

Adelman, C. (1993). Kurt Lewin and the origins of action research. *Educational Action Research, 1*(1), 7-24. doi:10.1080/0965079930010102

Aghamolaei, T., & Fazel, I. (2010). Medical students' perceptions of the educational environment at an Iranian Medical Sciences University. *BMC Medical Education, 10*(8) 1-7. doi:10.1186/1472-6920-10-87

Alimisis, D. (2009). Teacher education on robotics-enhanced constructivist pedagogical methods. Retrieved from http://dide.ilei.sch.gr/keplinet/education/docs/book_TeacherEducationOnRobotics-ASPETE.pdf

Alimisis, D., & Kynigos, C. (2009). Robotics as learning tool. In D. Alimisis (Ed.), *Teacher education on robotics-enhanced constructivist pedagogical methods* (pp.11-26). Retrieved from https://s3.eu-central-1.amazonaws.com/robotics-edu.gr/books/book_TeacherEducationOnRobotics-ASPETE.pdf

Alvarez, A., & Larranaga, M. (2016). Experiences incorporating LEGO Mindstorms robots in the basic programming syllabus: Lessons learned. *Journal of Intelligent and Robot Systems*, *81*(1), 117-129. doi:10.1007/s10846-015-0202-6

Ameh, P., & Dantani, Y. (2012). Effects of lecture and demonstration methods on the academic achievement of students in chemistry in Nassarawa local government of Kano state. *International Journal of Modern Social Sciences*, *1*(1), 29-37. Retrieved from https://www.academia.edu/30214517/Effects_of_Lecture_and_Demonstration_Methods_on_the_Academic_Achievement_of_Students_in_Chemistry_in_Nassarawa_Local_Government_Area_of_Kano_State

Amineh, R., & Asl, H. (2015). Review of constructivism and social constructivism. *Journal of Social Sciences, Literature and Languages*, *1*(1), 9-16. Retrieved from https://pdfs.semanticscholar.org/3890/3f4a7255496f75124d639e14e9b810c17370.pdf

Arlegui, J., Fava, N., Menegatti, E., Monfalcon, S., Moro, M., & Pina, A. (2009). Robotics as learning tool. In D. Alimisis (Ed.), *Teacher education on robotics-enhanced constructivist pedagogical methods* (pp. 27-102). Retrieved from https://s3.eu-central-1.amazonaws.com/robotics-edu.gr/books/book_TeacherEducationOnRobotics-ASPETE.pdf

Arseven, Z., Kilic, A., & Sahin., S. (2016). Teacher's adaptation level of student centered education approach. *Journal of Education and Practice*, *7*(29), 133-144. Retrieved from https://www.researchgate.net/publication/322466177_Teachers'_Adaptation_Level_of_Student_Centered_Education_Approach

Ashburn, E., & Floden, R. (2006). *Meaningful learning using technology*. New York, NY: Teachers College Press.

Augusta University (2017). *Augusta metropolitan area cybersecurity workforce study*. Retrieved from www.augusta.edu/pamplin/mpa/documents/2017_augusta_metro_cybersecruity_workforce_study.pdf

Ausubel, D. (1963). *The psychology of meaningful verbal learning*. New York, NY: Grune and Stratton.

Ausubel, D. (1968). *Educational psychology: A cognitive view*. New York, NY: Holt, Rinehart, and Winston.

Ayverdi, L., Nakiboglu, C., & OzAydin, S. (2014). Usage of graphic organizers in science and technology lessons. *Social and Behavioral Sciences, 116*, 4264-4269. doi:10.1016/j.sbspro.2014.01.929

Bahr, M., Gouwens, D., & Schuh, G. (2012). Evaluation of handheld computers for direct classroom observation. *Computers in Schools*, *29*(3), 268-284. doi:10.1080/07380569.2012.702720

Bakhtiar, M., Habib, Z., & Khan, S. (2014). Reflections on student performance in mcq’s and oeq’s in Punjab. *International Journal of Social Science & Education*, *4*(2), 541-550. Retrieved from http://www.ijsse.com/sites/default/files/issues/2013/v4i2/Paper-24.pdf

Bakhshialiabad, H., Bakhshi, M., & Hassanshahi, G. (2015). Students’ perceptions of the academic learning environment in seven medical sciences courses based on DREEM. *Advances in Medical Education and Practice*, *6*, 195-203. doi:10.2147/AMEP.S60570

Beard, L., Carpenter, L., & Johnston. (2011). *Assistive technology: Access for all students* (2nd ed.). Boston, MA: Pearson.

Beelders, T., & Plessis, J. (2016). Syntax highlighting as an influence factor when reading and comprehending code. *Journal of Eye Movement Research*, *9*(1), 1-11. doi:10.16910/jemr.9.1.1

Belland, B. (2017). *Instructional scaffolding in stem education: Strategies and efficacy evidence*. Logan, UT: Springer.

Beland, C., Chan, W., Clarke, D., Park, R. & Trupiano, M. (2000). LEGO Mindstorms: The structure of an engineering revolution. Retrieved from http://web.mit.edu/6.933/www/Fall200/LEGOMindstorms.pdf

Bell, S. (2010). Project-based learning for the 21st century: Skills for the future. *The Clearing House*, *83*, 39-43. doi:10.1080/00098650903505415

Bers, M., & Kazakoff, E. (2014). Put your robot in, put your robot out: Sequencing through programming robots in early childhood. *Journal of Educational Computing Research*, *50*(4), 553-573. doi:10.2190/EC50.4.f

Biggs, J., & Collis, K. (1982). *Evaluating the quality of learning: The SOLO taxonomy (structure of the observed learning outcome)*. New York, NY: Academic Press.

Biggs, J., & Collis, K. (1989). Toward a model of school-based curriculum development and assessment using SOLO taxonomy. *Australian Journal of Education*, *33*(2), 151-163. doi:10.1177/168781408903300205

Biggs, J., & Tang, C. (2011). *Teaching for quality learning at university*. New York, NY: McGraw-Hill.

Black, P., & William, D. (1998). Inside the black box: Raising standards through classroom assessment. *Sage*, *92*(1), 81-90. doi:10.1177/003172171009200119

Blake, B., & Pope, T. (2008). Developmental psychology: Incorporating Piaget's and Vygotsky's theories in classrooms. *Journal of Cross-Disciplinary Perspectives in Education*, *1*(1), 59-67. Retrieved from https://pdfs.semanticscholar.org/6c7b/9a7b4988df15c68a14434a5f162bef984723.pdf

Bos, B., Ortiz, M., & Smith, S. (2015). The power of educational robotics as an integrated STEM learning experience in teacher preparation programs. *Journal of College Science Teaching*, *44*(5), 42-47. doi:10.2505/4/jcst15_044_05_42

Bogdan, R., & Biklen, S. (2003). *Qualitative research for education: An introduction to theories and methods*. Boston, MA: Allyn and Bacon.

Bransford, J., Brown, A., & Cocking, R. (2000). *How people learn: Brain, mind, experience, and school*. Washington, DC: National Academy Press.

Broekkamp, H., & Hout-Wolters, B. (2007). The gap between educational research and practice: A literature review, symposium, and questionnaire. *Educational Research and Evaluation*, *13*(3), 203-220. doi:10.1080/13803610701626127

Bruner, J. (1960). *The process of education*. Cambridge, MA: Harvard University Press.

Bruner, J. (1996). *The culture of ducation*. Cambridge, MA: Harvard University Press.

Burbaite, R., Damasevicius, R., & Stuikys, V. (2013). Teaching of computer science topics using meta-programming-based glos and LEGOs robots. *Informatics in Education*, *12*(1), 125-142. Retrieved from https://files.eric.ed.gov/fulltext/EJ1064312.pdf

Carnegie Mellon Robotics Academy. (2015). Retrieved from http://education.rec.ri

Craig, D. (2009). *Action research essentials*. San Francisco, CA: Wiley.

Cassady, S., Leff, J., Stanley, L., Castro, S., Thlick-Khatchadourian, L., Ukaha, S., & Ward-Roncalli, S. (2004). KWLs and anchor tasks: Assessing the dimensions of student learning through service. In *Service-learing and assessment: A field guide for teacher*s. Retrieved from http://www.communityworksinstitute.org/cwpublications/slassessguide/AssessGdChpt5.pdf

Cejka, E., Portsmore, M., & Rogers, C. (2006). Kindergarten robotics: Using robotics to motivate math, science, and engineering literacy in elementary school. *International Journal of Engagement Education*, *22*(4), 711-722. Retrieved from https://ceeo.tufts.edu/documents/journal/2006eccrmp.pdf

Charles, C., & Mertler, C. (2011). *Introduction to educational research*. Boston, MA: Pearson.

Charntaweekhun, K., & Wangsiripitak, S. (2006). Visual programming using flowchart. *Communications and Information Technologies, ISCIT International Symposium*. doi:10.1109/ISCIT.2006.339940

Chetty, J., & Barlow-Jones, G. (2012). Bridging the gap: the role of mediated transfer for computer programming. *International Association of Computer Science and Information Technology*, *43,*1-5. Retrieved from https://www.researchgate.net/publication/261725679_Bridging_the_Gap_the_Role_of_Mediated_Transfer_for_Computer_Programming

Code.org. (ND). *What is computer science and what do people do once they know it?* Retrieved from: https://code.org/files/computer_science_is_foundational.pdf

College Board (2014). Computer science A: Course description. Retrieved from https://apstudent.collegeboard.org/apcourse/ap-computer-science-a

Conderman, G., & Hedin, L. (2012). Classroom assessments that inform instruction. *Assessment Techniques*, *48*, 162-168. doi:10.1080/00228958.2012.733964

Conrad, C., & Serlin, R. (2006). *The Sage handbook for research in education: Engaging ideas and enriching inquiry*. Thousand Oaks, CA: Sage.

Craig, D. (2009). *Action research essentials.* San Francisco, CA: Jossey-Bass.

Creswell, J. (2012). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research* (4th ed.). Boston, MA: Pearson.

Creswell, J. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches* (4th ed.). Thousand Oaks, CA: Sage.

Cubukcu, Z. (2012). Teachers' evaluation of student-centered learning environments. *Education, 133*(1), 49-66. Retrieved from https://pdfs.semanticscholar.org/f57d/8a1c7e75d3ba7bfc3b95876f32c9ce349678.pdf

Deitrick, E., Sanfor, J., & Shapiro, B. (2014). *BlockyTalky: A low-cost, extensible, open source, programmable, networked toolkit for tangible creation*. Paper presented at Conference on Interaction Design for Children, at Aarhus, Denmark.

Dewey, J. (1938). *Experience and Education*. New York, NY: MacMillan.

Dimitri, G. (2015). *The impact of syntax highlighting in Sonic Pi*. Paper presented at the PPIG 26 annual workshop. Bournemouth, England.

Dol, S. (2015). *Fe.g: An animated flowchart with example to teach the algorithm based courses in engineering*. Paper presented at the 2015 IEEE Seventh International Conference on Technology for Education. Telangana, India. doi:10.1109/T4E.2015.3

Ensmenger, N. (2016). The multiple meanings of a flowchart. *Information and Culture: A Journal of History*, *51*(3), 321-351. doi:https//doi.org/10.1353/lac.2016.0013

Ertmer, P., & Simons, K. (2006). Jumping the PBL implementation hurdle: Supporting the efforts of K-12 teachers. *Interdisciplinary Journal of Problem-based Learning*, 1(1), *40-*54. doi:10.7771/1541-5015.1005

Ewoldt, K., & Morgan, J. (2017). Color-coded graphic organizers for teaching writing students with learning disabilities. *Teaching Exceptional Children*, *49*(3), 175-184. doi:10.1177/0040059916681769

Feurzeig, W., & Papert, S. (2011). Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments, 19*(5) 487-501. doi:10.1080/10494820903520040

Fry, M., Curtis, K., Considine, J., & Shaban, R. (2017). Using observation to collect data in emergency research. *Australasian Emergency Nursing Journal*, *20*(1), 25-20. doi:10.1016/j.aenj.2017.01.001

Gall, M., Gall, J., & Borg, W. (2003). *Educational research: An introduction* (7th ed.). Boston, MA: Allyn and Bacon.

Ganyaupfu, E. (2013). Teaching methods and students' academic performance. *International Journal of Humanities and Social Science Invention*, *2*(9), 29-35. Retrieved from https://www.researchgate.net/publication/264124430_Teaching_Methods_and_Students'_Academic_Performance/download

Garrett, T. (2008). Student-centered and teacher-centered classroom management: A case study of three elementary teachers. *Journal of Classroom Interactions*, *43*(1), 34-47.

Geist, E. (2016). Robots, programming and coding, oh my! *Childhood Education*, *92*(4), 298-304. doi:10.1080/00094056.2016.1208008

Georgia Department of Education. (2014). Assistive technology devices. Retrieved from http://www.gpat.org/Georgia-Project-for-Assistive-Technology/Pages/Assistive-Technology-Devices.aspx

Georgia Department of Education. (2017). Career clusters and pathway courses. Retrieved from http://www.gadoe.org/Curriculum-Instruction-and-Assessment/CTAE/Pages/cluster-pathway-courses.aspx

Gierach, J. (2009). *Assessing students' needs for assistive technology (ASNAT)*. Retrieved from http://www.wati.org/content/supports/free/pdf/ASNAT5thEditionJun09.pdf

Gill, B., Timpane, M., Ross, K., & Brewer, D. (2001). *Rhetoric versus reality*. Santa Monica, CA: RAND.

Guba, E. (1981). Criteria for assessing the trustworthiness of naturalistic inquiries. *Educational Communication and Technology*, *2*(1), 75-91. Retrieved from http://www.jstor.org/stable/30219811

Gurlitt, J., & Renkl, A. (2008). Prior knowledge activation: How different concept mapping tasks lead to substantial differences in cognitive process, learning outcomes, and perceived self-efficacy. *Instructional Science*, *38*(4), 417-433. doi:10.1007/s11251-008-9090-5

Hall. T., & Strangman, N. (2002). *Graphic organizers*. Wakefield, MA: National Center on Accessing the General Curriculum. Retrieved from http://seaver-faculty.pepperdine.edu/tflynn/EDUC564S/handouts/Graphic%20Organizers--NCAGC.pdf

Han, S. (2017). Korean students' attitudes toward STEM project-based learning and major selection. *Educational Sciences: Theory and Practice*, *17*(2), 529-548. doi:10.12738/estp.2017.20264

Han, S. & Bhattacharya, K. (2001). Constructionism, learning by design, and project-based learning. In M. Orey (Ed.), *Emerging perspectives on learning, teaching, and technology*, (pp. 1-18). Zurich:Global Text Project.

Han, S. & Bhattacharya, K. (2014). Constructionism, learning by design, and project-based learning. Retrieved from http://epltt.coe.uga.edu/index.php?title=Constructionism%2C_Learning_by_Design%2C_and_Project_Based_Learning#Project-Based_Learning_-_What_is_it.3F

Hershberger, H., Zembal-Saul, C., & Starr, M. (2006). Evidence helps the kwl get a klew. *Science and Children*, *43*(5), 50-53. Retrieved from https://www.nsta.org/publications/news/story.aspx?id=51519

Hine, G. (2013). The importance of action research in teacher education programs. *Issues in Educational Research*, *23*(2), 151-163. Retrieved from http://www.iier.org.au/iier23/hine.pdf

Hixon, E., & Buckenmeyer, J. (2009). Revisiting technology integration in schools: Implications for professional development, *Computers in the Schools*, *26*, 130-146. doi:10.1080/07380560902906070

Holt, L.C., & Kysilka, M. (2006*). Instructional patterns: Strategies for maximizing student learning*. Thousand Oaks, CA: SAGE.

Hotchkiss, J. (2018, April 19). Augusta, GA., Hopes big cybersecurity center will turn it into tech hub. *Government Technology*. Retrieved from http://www.govtech.com/security/Augusta-Ga-Hopes-Big-Cybersecurity-Center-Will-Turn-it-Into-Tech-Hub.html

Hoy, A. (2007). *Educational psychology* (10th ed.). Boston, MA: Pearson.

Ibrahim, K., & Omwirhiren, E. (2016). The effects of two teachers' instructional methods on students' learning outcomes in chemistry in selected senior secondary school in Kaduna Metropolis, Nigeria. *Journal of Education and Practice*, *7*(15), 1-9. Retrieved from https://files.eric.ed.gov/fulltext/EJ1103149.pdf

Inhelder, B., & Piaget, J. (1958). *The growth of logical thinking from childhood to adolescence*. New York, NY: Basic Books.

Iqbal, J., & Ahmad, A. (2015). Effect of extensive rote learning experience on subsequent academic achievement. *Pakistan Armed Forces Medical Journal*, *65*(4), 510-514.

Israel, M., Pearson, J., Wherfel, Q., Shehab, S., & Tapia, T. (2015). Empowering K-12 students with disabilities to learn computational thinking and computer programming. *Teaching Exceptional Children*, *48*(1) 45-53. doi:10.1177/0040059915594790

Ivic, S. (2016). Frequency of applying different teaching strategies and social teaching methods in primary schools. *Journal of Education and Practice*, *7*(33), 66-71. Retrieved from https://files.eric.ed.gov/fulltext/EJ1122875.pdf

Jamet, E. (2013). An eye-tracking study of cueing effects in multimedia learning. *Computers in Human Behavior*, *32*, 47-53. doi:10.1016/j.chb.2013.11.013

Jonassen, D., Howland, J., Marra, R., & Crismond, D. (2008). *Meaningful learning with technology* (3rd ed.). Upper Saddle River, NJ: Pearson.

Jamet, E., Gavota, M., & Quaireau, C. (2008). Attention guiding in multimedia learning. *Learning and Instruction*, *18*, 135-145. doi:10.1016/j.learninstruc.2007.01.011

Khan, S. (2013). Constructivism: An innovative inquiry-based approach to classroom teaching; With special reference to teaching of science. *The Journal of Progressive Education*, *6*(1), 60-69.

Kalyuga, S., Chandler, P., & Sweller, J. (1999). Managing split-attention and redundancy in multimedia instruction. *Applied Cognitive Psychology*, *13*(4), 351-371. doi:10.1002/(SICI)1099-0720

Karpicke, J., & Grimaldi, P. (2012). Retrieval-based learning: A perspective for enhancing meaningful learning. *Educational Psychology Review*, *24*, 401-418. doi:10.1007/s10648-012-9202-2

Katiliute, E., & Daunoriene, A. (2011). Toward meaningful learning in economics and management studies using concept maps in a quality management course. *Economics and Management*, *16*, 758-765. Retrieved from https://www.researchgate.net/publication/228411488_TOWARD_MEANINGFUL_LEA

RNING_IN_ECONOMICS_AND_MANAGEMENT_STUDIES_USING_CONCEPT_MAPS_IN_A_QUALITY_MANAGEMENT_COURSE

Keegan, R. (2016). Action research as an agent for enhancing teaching and learning in physical education: A physical education teacher's perspective. *The Physical Educator*, *73*, 255-284. doi:10.1037/0278-6133.24.2.225

Kennedy, M. (1997). The connection between research and practice. *Educational Researcher*, *26*(7), 4-12. doi:10.3102/0013189x026007004

Kito, K. (2009). *Using cooperative writing and oral presentations as peer teaching – evaluating the effectiveness of elements of inductive teaching and social constructivism on student outcomes*. Paper presented at the 39th ASSEE/IEEE Frontiers in Education Conference, San Antonio, TX. doi:10.1109/FIE.2009.5350848

Kito, K. (2010). *Understanding the effectiveness of cognitive and social constructivism, elements of inductive practice, and student learning styles on selected learning outcomes in materials engineering*. Paper presented at the 40th ASEE/IEEE Frontiers in Education Conference, Washington, DC. doi:10.1109/FIE.2010.5673507

Kopcha, T. (2011). Teachers' perceptions of the barriers to technology integration and practices with technology under situated professional development. *Computers & Education*, *59*, 1109-1121. doi:10.1016/j.compedu.2012.05.014

Koshy, V. (2005). *Action research for improving practice: A practical guide*. Thousand Oaks, CA: Sage.

Lawless, K., & Pellegrino, J. (2007). Professional development in integrating technology into teaching and learning: Knowns, unknowns, and ways to pursue better questions and answers. *Review of Educational Research*, *77*(4), 575-614. doi:10.3102/0024654307309921

Lee, A. (2015). Determining the effects of computer science education at the secondary level on STEM major choices in postsecondary institutions in the United States. *Computer & Education*, *88*, 241-288. doi:10.1016/j.compedu.2015.04.019

LEGO. (2017). *LEGO Mindstorms EV3 programming*. Retrieved from https://www.LEGO.com/en-us/mindstorms/learn-to-program

Lewis, T., Scott, T., Wehby, J., & Wills, H. (2014). Direct observation of teacher and student behavior in school settings: Trends, issues and future directions. *Behavioral Disorders*, *39*(4), 190-200. doi:10.1177/019874291303900404

Liffick, B. (2003). *Assistive technology in computer science*. Proceedings of the International Symposium on Information and Communication Technologies, Trinity College, Dublin, Ireland.

Lye, S., & Koh, J. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51-61. doi:10.1016/j.chb.2014.09.012

Manolov, R., & Losad, J. (2017). Simulation theory applied to direct systematic observation. *Frontiers in Psychology*, 8, 1-16. doi:10.3389/fpsyg.2017.00905

Marshall, C., & Rossman, G. (2010). *Designing qualitative research*. Thousand Oaks, CA: Sage.

Marzano, R., (2003). *What works in schools*. Alexandria, VA: Association for Supervision and Curriculum Development.

Matthews, M., Ritchotte, J., & McBee, M. (2013). Effects of schoolwide cluster grouping and within-class ability grouping on elementary school students' academic achievement growth. *High Ability Studies*, *24*(2), 81-97. doi:10.1080/13598139.2013.846251

Mayer, R. (2002). Rote versus meaningful learning. *Theory into Practice*, *41*(4), 226-232.

Mayer, R. (2009). *Multimedia learning* (2nd ed.). New York, NY: Cambridge University Press.

Mayer, R. (2017). Using multimedia for e-learning. *Journal of Computer Assisted Learning*, *33*, 403-423. doi:10.111/jcal.12197

Mayya, S., & Roff, S. (2004). Students' perceptions of educational environment: A comparison of academic achievers and under-achievers at Kasturba Medical College, India. *Education for Health*, *17*(3), 280-291. doi:10.1080/13576280400002445

McCoy, B. (2017, March 26). *Local school boards getting students ready for cyber careers*. WRDW. Retrieved from http://www.wrdw.com/content/news/Local-school-boards-getting-students-ready-for-cyber-careers-417131933.html

McIntyre, D. (2005). Bridging the gap between research and practice. *Cambridge Journal of Education*, *35*(3), 357-382. doi:10.1080/03057640500319065

McLeod, J., Fisher, J., & Hoover, G. (2003). *The key elements of classroom management: Managing time and space, student behavior, and instructional strategies*. Alexandria, VA: Association for Supervision and Curriculum Development.

Meece, J., & Daniels, D. (2008). *Child and adolescent development for educators* (3rd ed.). New York, NY: McGraw-Hill.

Melrose, M. (2001). Maximizing the rigor of action research: Why would you want to? How could you? *Field Methods*, *13*(2), 160-180. doi:10.1177/1525822X0101200203

Mertler, C. (2003). *Classroom Assessment: A practical guide for educators*. New York, NY: Routtledge

Mertler, C. (2014). *Action research: improving schools and empowering educators*. Thousand Oaks, CA: Sage.

Microsoft. (2017). *Creating a basic flowchart*. Retrieved from https://support.office.com/en-us/article/Create-a-basic-flowchart-e207d975-4a51-4bfa-a356-eeec314bd276

Mihardi, S., Harahap, M., & Sani, R. (2013). The effect of project-based learning model with kwl worksheet on student creative thinking process in physics problem. *Journal of Education and Practice*, *4*(25), 188-200. Retrieved from https://www.researchgate.net/publication/260064848_The_Effect_of_Project_Based_Learning_Model_with_KWL_Worksheet_on_Student_Creative_Thinking_Process_in_Physics_Problems

Mihci, C., & Ozdener, N. (2017). Teaching GUI-programming concepts to prospective K12 ICT teachers: MIT App Inventor as an alternative to text-based languages. *International Journal of Research in Education and Science*, *3*(2), 543-559. doi:10.21890/ijres.327912

Miles, M., Huberman, M., & Saldana, J. (2014). *Qualitative data analysis: A methods sourcebook* (3rd ed.). Thousand Oaks, CA: Sage.

Mills, G. (2011). *Action research: A guide for the teacher researcher*. Boston, MA: Pearson.

Milller, L., Rosendale, S., & Shanks, J. (2012). Action research in a professional development school setting to support teacher candidate self-efficacy. *SRATE Journal*, *21*(2), 26-32.

Milne, I., & Rowe, G. (2002). Difficulties in learning and teaching programming-Views of students and tutors. *Education and Information Technologies, 7*(1), 55-66. doi:10.1023/A: 1015362608943

Mohorovicic, S., & Strcic, V. (2011). An overview of computer programming teaching methods. Retrieved from https://pdfs.semanticscholar.org/f5e8/a13a42f788a0169d79551a5ff90263fa2ad3.pdf

Munyaradzi, G. (2013). *Teaching methods and students' academic performance*. International *Journal of Humanities and Social Science Invention*. *2*(9), 29-35. Retrieved from https://www.researchgate.net/publication/264124430_Teaching_Methods_and_Students'_Academic_Performance

Murbin, O., Stevens, C., Shahid, S., Al Mahmud, A., & Dong, J. (2013). A review of the applicability of robots in education. *Technology for Education and Learning,* 1-7. doi:10.2316/Journal.209.2013.1.209-0015

National Academy Press. (2000). *How people learn: Brain, mind, experience, and school (*expanded edition*)*. Washington, DC: The National Academic Press.

National Science Foundation, National Science Board. (2010). *Preparing the next generation of STEM innovators: Identifying and developing our nation's human capital* (NSB Publication No.10-33). Retrieved from https://www.nsf.gov/nsb/publications/2010/nsb1033.pdf

Noe, A. (2008). Precis of action in perception: philosophy and phenomenological research. *International Phenomenological Society, 76*(3), 660-665. Retrieved from https://www.jstor.org/stable/40041202

Ogle, D. (1986). K-W-L. A teaching model that develops active reading of expository text. *The Reading Teacher*, *39*(6), 564-570. Retrieved from http://www.jstor.org/stable/20199156

Ogle, D. (2009). Creating contents for inquiry: From kwl to prc2. *Knowledge Quest*, *38*(1), 56-61.

Ojose, B. (2008). Applying piaget's theory of cognitive development to mathematics instruction. *The Mathematics Educator*, *18*(1), 26-30. Retrieved from https://files.eric.ed.gov/fulltext/EJ841568.pdf

Onchwari, G., Onchwari, J. & Keengwe, J. (2008). Teaching the immigrant child: application of child development theories. *Early Childhood Education*, *26*, 267-273. doi:10.1007/s10643-008-0269

Ostrov, J., & Hart, E. (2013). Observational methods. In T. Little (Ed.), *The Oxford handbook of quantitative methods in psychology* (pp. 286-304). New York. NY: Oxford.

O'Sullivan, M., & Dallas, K. (2010). A collaborative approach to implementing 21st century skills in a high school senior research class. *Education Libraries*, *33*(1), 3-9.

Panadero, C., Delgaldo-Kloos, C., & Villen-Roman, J. (2010). *Impact of learning experiences using LEGO Mindstorms in engineering courses*. IEEE Xplore Conference: Education Engineering (EDUCON). doi:10.1109/EDUCON.2010.5492534

Papanikolaou, K., & Frangou, S. (2009). Robotics as learning tool. In D. Alimisis (Ed.), *Teacher education on robotics-enhanced constructivist pedagogical methods* (pp. 103-138). Heraklion, Greece: School of Pedagogical and Technological Education (ASPETE). Retrieved from https://s3.eu-central-1.amazonaws.com/robotics-edu.gr/books/book_TeacherEducationOnRobotics-ASPETE.pdf

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. NY: Basic Books.

Parsons, R., & Brown, K. (2002). *Teacher as reflective practitioner and action researcher*. Belmont, CA: Wadsworth/Thomson Learning.

Pasko, R. & Bauer, M. (2002). *Problem solving basics and computer programming*. Retrieved from http://www.cs.iit.edu/~cs100/ProblemSolving.pdf

Pasztor, A., Pap-Szigeti, R., & Torok, E. (2010). Effects of using model robots in the education of programming. *Informatics in Education*, *9*(1), 133-140. doi:10.1007/s10846-015-0202-6

Pasztor, A., Pap-Szigeti, R., & Torok, E. (2013). Mobile robots in teaching programming for it engineers and its effects. *International Journal of Advanced Computer Science and Applications*, *4*(11), 162-168. Doi:10.14569/IJACSA.2013.041123

Patton, M. Q. (2015). *Qualitative research and evaluation methods* (4th ed.). Thousand Oaks, CA: Sage.

Pea, R. (2004). The social and technological dimensions of scaffolding and related theoretical concepts for learning, education, and human activity. *The Journal of the Learning Sciences*, *13*(3), 423-451. doi:10.1207/s15327809jls1303_6

Pellgrino, J., Chudowsky, N., & Glaser, R. (2001). *Knowing what students know: The science and design of educational assessment*. Washington, DC: National Academy Press.

Petre, M., & Price, B. (2004). Using robotics to motivate ‘back door’ learning. *Education and Information Technologies*, *9*(2), 147-158. doi:10.1023/B:EAIT.0000027927.78380.60

Phillips, D.C. (2000). *Constructivism in education: Opinions and second opinions on controversial issues*. Chicago, IL: The National Society for the Study of Education.

Piaget, J. (1972). Intellectual evolution from adolescence to adulthood. *Human Development*, *51*, 40-47. doi:10.1159/000112531

Piaget, J. (1952). *The origins of intelligence in children*. NY: International Universities Press.

Piaget, J. (1969). *The psychology of the child*. NY: Basic Books.

Powell, K., & Kalina, C. (2009). Cognitive and Social Constructivism: Developing tools for an effective classroom. *Education*, *130*(2), 241-250. Retrieved from https://docdrop.org/static/drop-pdf/ConstructivismDay1-ln36v.pdf

Prince, M., & Felder, R. (2006). Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Journal of Engineering Education*, *95*(2), 123-138. doi:10.1002/j.2168-9830.2006.tb00884.x

Rosenshine, B. (2007). *Systematic instruction*. In T. L. Good (Ed.), 21st century education: A reference handbook (pp. 1-13). Thousand Oaks, CA: Sage.

Rothwell, W., & Kazanas, H. C. (2008). *Mastering the instructional design process: A systematic approach* (4th ed.). San Francisco, CA: Pfeiffer.

Saeli, M., Perrenet, J., Jochems, W., & Zwaneveled, B. (2011). Teaching programming in secondary school: A pedagogical content knowledge perspective. *Informatics in Education*, *10*(1), 73-88.

Sadler, R. (1989). Formative assessment and the design of instructional systems. *Instructional Science*, *18*, 119-144.

Saldana, J. (2009). *The coding manual for qualitative researchers*. Thousand Oaks, CA: Sage.

Sandholtz, J. (2011). Preservice teachers' conceptions of effective and ineffective teaching practices. *Teacher Education Quarterly*, *38*(3), 27-47. Retrieved from https://files.eric.ed.gov/fulltext/EJ940632.pdf

Sangoleye, Solomon. (2016). A critique of selected instructional strategies in higher institutions in Nigeria. *Journal of Education and Practice*, *7*(7), 78-84. Retrieved from https://files.eric.ed.gov/fulltext/EJ1095271.pdf

Santrock, J. (2008). *Educational psychology* (3rd ed.). New York, NY: McGraw-Hill.

Sarkar, A. (2015). The impact of syntax colouring on program comprehension. In *Proceedings of the 26th annual workshop of the Psychology of Programming Interest Group*, Bournemouth, UK (pp. 49-58). Retrieved from http://www.ppig.org/sites/ppig.org/files/2015-PPIG-26th-proceedings.pdf

Scott, J., & Sarkees-Wircenski, M. (2004). *Overview of career and technical education*. Homewood, IL: American Technical.

Schildt, H. (1999). *C++: The complete reference*. Berkeley, CA: McGraw-Hill.

Schmuck, R. (1997). *Practical action research for change*. Arlington Heights, IL: IRI/Sky Light Training and Publishing.

Schon, D. (1987). *Educating on the reflective practitioner*. San Francisco, CA: Jossey-Bassey

Schwerdt, G., & Wuppermann, A. (2011). Is traditional teaching really all that bad? A within-student between-subject approach. *Economics of Education Review*, *30*, 365-379. doi:10.1016/j.econedurev.2010.11.005

Shabani, K. (2010). Vygotsky's zone of proximal development: Instructional implications and teachers' professional development. *English Language Teaching*, *3*(4), 237-248. Retrieved from https://files.eric.ed.gov/fulltext/EJ1081990.pdf

Shaffer, D. (1996). *Developmental psychology: Childhood and adolescence* (4th ed.). Pacific Grove, CA: Brooks and Cole.

Shanks, J., Miller, L., & Rosendale, S. (2012). Action research in a professional development school setting to support teacher candidate self-efficacy. *SRATE Journal*, *21*(2), 26-32.

Shavelson, R., Young, D., Ayala, C., Brandon, P., Furtak, E., Ruiz-Primo, Maria, Tomita, M., & Yin, Y. (2008). On the impact of curriculum-embedded formative assessment on learning: A collaboration between curriculum and assessment developers. *Applied Measurement in Education*, *21*(4), 295-314. doi:10.1080/08957340802347647

Sheard, J., Simon, S., Hamilton, M., & Lonnberg, J. (2009). *Analysis of research into the teaching and learning of programming*. ICER '09 Proceedings of the Fifth International Workshop on Computing Education Research. Berkeley, CA (pp. 93-104). doi:10.1145/1584322.1584334

Shooshtari, Z. & Mir, F. (2014). ZPD, tutor; Peer scaffolding: Sociocultural theory in writing strategies application, *Social and Behavioral Sciences*, *98*, 1771-1776. doi:10.1016/j.sbspro.2014.03.605

Sigler, E. (2009). Action research in public schools: Is it research? Should it be reviewed? *Journal of Empirical Research of Human Research Ethics: An International Journal*, *4*(2), 17-25. doi:10.1525/jer.2009.4.2.17

Slavin, R. (1991). *Educational psychology* (3rd ed.). Englewood Cliffs, NJ: Prentice Hall.

Sithole, A., Chiyaka, E., McCarthy, P., Mupinga, D., Bucklein, B., & Kibirige, J. (2017). Student attraction, persistence and retention in STEM programs: Success and continuing challenges. *Higher Education Studies*, *7*(1), 46-59. doi:10.5539/hes.v7n1p46

Smelley, J. (2013). Rote learning: A revived strategy for religious instruction. *Christian Education Journal*, *3*(10), 309-321.

Snoeyink, R., & Ertmer, P. (2001). Thrust into technology: How veteran teachers respond. *Journal of Educational Technology Systems*, *30*(1), 85-111. doi:10.2190/YDL7-XH09-RLJ6-MTP1

Stager, G. (2001). *Constructionism as a high-tech intervention strategy for at-risk learners*. Paper presented at the National Educational Computing Conference Proceedings., Chicago, IL.

Stager, G. (2003). *The feeling of wonderful ideas – Implications for the future of teaching and learning*. Paper presented at the IFIP Working Groups 3.1 and 3.3 Working Conference, Melbourne, Australia.

Stankous, N.V. (2016). Constructive response vs multiple-choice tests in math: American experience and discussion (review). *European Scientific Journal*, 308-316. doi:10.19044/esj.2016.v12n10%p

Stein, S., McRobbie, C., & Ginns, I. (1999). Introducing technology education: Using teachers' questions platform for professional development. *Research in Science Education*, *29*(4), 501-414. doi:10.1007/BF02461592

STEMRobotics. (2016). *Course: STEM Robotics 101 EV3*. Retrieved from http://stemrobotics.cs.pdx.edu/node/2643

Stevens, R. (2004). Why do educational innovations come and go? What do we know? What can we do? *Teaching and Teacher Education*, *20*(4), 389-396. doi:10.1016/j.tate.2004.02.011

Stringer, E. (2014). *Action research* (4th ed.). Thousand Oaks, CA: Sage.

Stuikys, V., Burbaite, R., & Damasevicius, R. (2013). Teaching of computer science topics using meta-programming-based glos and LEGO robots. *Informatics in Education*, *12*(1), 125-142.

Sweller, J., Merrienboer, J., & Paas, F. (1998). Cognitive architecture and instructional design. *Educational Psychology Review*, *10*(3), 251-296. doi:1023/a:1022193728205

Tabbers, H., Martens, R., &Merrienboer, J. (2004). Multimedia instructions and cognitive load theory: Effects of modality and cueing. *British Journal of Educational Psychology*, *74*, 71-98. doi:10.1348/000709904322848824

Thomas, L., & Knezek, G. (2008). Information, communications, and educational technology standards for students, teachers, and school leaders. In J. Voogt & G. Knezek (Eds.), *International handbook of information technology in primary and secondary education*, (pp. 333-348). New York, NY: Springer.

Thompson, N., & Pascal, J. (2012). Developing critically reflective practice. *Reflective Practice*, *13*(2), 311-325. doi:10.1080/14623943.2012.657795

Trivodaliev, K., Stojkoska, B., Mihova, M., Jovanov, M., & Kalajdziski, S. (2017). Teaching computer programming: The Macedonian case study of functional programming. In *Global Engineering Education Conference (EDUCON)* IEEE, 1282-1289. doi:10.1109/EDUCON.2017.7943013

VanTassel-Baska, J. (2003). *Curriculum planning and instructional design for gifted learners*. Denver, CO: Love.

Vygotsky, L. (1978). *Mind in society: The development of higher psychological process*. Cambridge, MA: Harvard University Press.

Wetzels, S., Kester, L., Merrienboer, J., & Broer, N. (2011). The influence of prior knowledge on the retrieval-directed function of note taking in prior knowledge activation. *British Journal of Educational Psychology*, *81*, 274-291. doi:10.1348/000709910X517425

Wing, J. (2006). Computational thinking. *Communications of ACM*, 49(3), 33-35. Retrieved from https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf

Wood, D., Bruner, J., & Ross, G. (1976). The role of tutoring in problem solving, *Journal of Child Psychology and Psychiatry*, *17*, 89-100. doi:10.1111/j.1469-7610.1976.tb00381.x

Wuppermann, A., & Schwerdt, G. (2011). Is traditional teaching really all that bad? A within-student between-subject approach. *Economics of Education Review*, *30*, 365-379. doi:10.1016/j.econedurev.2010.11.005

Xinogalos, S. (2013). *Using flowchart-based programming environments for simplifying programming and software engineering process*. IEEE Global Engineering Education Conference (EDUCON). Berlin, Germany.

Zabala, J., Bowser, G., & Korsten, J. (2004). SETT and ReSETT: Concepts for AT implementation. *Closing the Gap*, *23*(5), 15-19.

Zeki, C. (2014). Student teachers' perceptions about their experiences in a student-centered course. *South African Journal of Education*, *34*(3), 1-11. doi:10.15700/201409161111

APPENDIX A

INTRODUCTION TO PROGRAMMING LEGO EV3 CURRICULUM CORRELATION TO EDUCATIONAL STANDARDS

Common Core Mathematics Practices Standards

Standard		Introduction to Programming the EV3
MP1	Make sense of problems and persevere in solving them	Chapters are all based around solving real-world robot problems; students must make sense of the problems to inform their solutions
MP2	Reason abstractly and quantitatively	Programming requires students to reason about physical quantities in the world to plan a solution, then calculate or estimate them for the robot
MP4	Model with mathematics	Many processes, including the process of programming itself, must be systematically modeled on both explicit and implicit levels
MP6	Attend to precision	Robots require precise (and accurate) input, or their output action will be correspondingly sloppy
MP7	Look for and make use of structure	Understanding the structure of the physical environment, the interrelated components of robot hardware and software, and commands within a program are vital to successful solutions
MP8	Look for and express regularity in repeated reasoning	Any programmed solution to a class of problems rely on the programmer recognizing and exploiting important patterns in the problem structure. There is also an emphasis throughout the module on recognizing common programmatic patterns, as well as patterns within a solution that invite the use of Loops.

Standard (CCSS.Math.Content)	Introduction to Programming the EV3

6.RP.A.1	Understand the concept of a ration and use ratio language to describe a ratio relationship between two quantities	Students use ratio language to describe and make use of the relationship between quantities such as Wheel Rotations and Distance Traveled
6.RP.A.2	Understand the concept of a unit rate a/b associated with a ratio a:b with b!=0, and use rate language in the context of a ratio relationship	The relationship between Wheel Rotations and Distance Traveled is a rate, customarily understood through a unit rate such as "# cm per rotation".
6.R.A.3	Use ratio and rate reasoning to solve real-world and mathematical problems	Students are required to apply ratios and rates when they build their prototype examples of their real-world robots.
7.RP.A.3	Use proportional relationships to solve multistep ratio and percent problems.	Comparisions between rate-derived quantities are common during robot navigations tasks

Note. Reprinted from "Introduction to Programming LEGO Mindstorms EV3", by Carnegie Mellon's Robotic Academy. Copyrighted 2014. Reprinted with Permission.

Next Generation Science Standards (NGSS)

Standard		Introduction to Programming the EV3
MS-ETS1-2	Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.	Solving challenges requires students to create and evaluate both hardware and software designs according to scenario scoring criteria. Some Reflection Questions require students to make recommendations between competing alternatives based on criteria that they define.
MS-ETS1-4	Develop a model to generate data for iterative testing and modification of a proposed object, tool, or process such that an optimal design can be achieved.	When solving more difficult and complex challenges, students are guided toward iterative testing and refinement processes. Students must optimize program parameters and design.
HS-ETS1-2	Design a solution to a complex real-world problem by breaking it down into smaller,	Problem Solving methodology for challenges directs students to break down large problems into smaller solvable ones, and build solutions up accordingly; challenges give students

	more manageable problems that can be solved through engineering.	opportunities to practice, each of which is based on a real-world robot
HS-ETS1-3	Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics as well as possible social, cultural, and environmental impacts.	Some Reflection Questions require students to make recommendations about real-world policies (e.g. requiring sensors on automobiles) based on the impact of that decision

Note. Reprinted from “Introduction to Programming LEGO Mindstorms EV3: Teacher’s Guide”, by Carnegie Mellon’s Robotic Academy. Copyright 2014.

Computational Thinking Practices (CTP)

Name of Practice		Explanation
P1	Connecting Computing	Relating computing to real-world situations
P2	Creating Computational Artifacts	Developing computational artifacts with a practical, personal, or societal intent

P3	Abstracting	Understanding and applying abstract reasoning
P4	Analyzing Problems and Artifacts	Assessing computational artifacts
P5	Communicating	Sharing or exchanging information
P6	Collaborating	Teamwork

Note. Reprinted from “Introduction to Programming LEGO Mindstorms EV3”, by Carnegie Mellon’s Robotic Academy. Copyrighted 2014. Reprinted with Permission.

Computer Science Principles Framework (CSP)

Learning Objective		Introduction to Programming the EV3
1.1.1	Use computing tools and techniques to create artifacts. [P2]	Challenge activities result in the creation of a (simple) algorithmic solution and an accompanying program that implements it.
1.1.2	Collaborate in the creation of computational artifacts. [P6]	Students work in teams to accomplish tasks.
1.1.3	Analyze computational artifacts. [P4]	Students perform debugging on their own code, as well as analyze and evaluate others’ code and suggested code in Reflection Questions.
1.3.1	Use programming as a creative tool. [P2]	Students use programming to solve model challenges based on challenges real robots face.
2.2.1	Develop an abstraction. [P2].	Robots gather information about the world through sensors, which turn physical qualities of the world into digital abstractions. Students must understand and work with this data to develop then implement their solution algorithms.
2.3.1	Use models and simulations to raise and answer questions. [P3]	Students construct and use a “program flow” model of programming itself to understand how the robot uses data to make decisions and control the flow of its own commands.

4.1.1	Develop an algorithm designed to be implemented to run on a computer. [P2]	Students develop solution algorithms to each challenge and mini-challenge problem before implementing them as code. Reflection Questions also ask students to evaluate algorithms expressed as pseudocode.
4.2.1	Express an algorithm in a language. [P5]	Students develop code to robotics challenges in the EV3 Programming Language.
5.1.1	Explain how programs implement algorithms. [P3]	Students must communicate solution ideas within groups and as part of class discussion, as well as in Reflection Questions.
5.3.1	Evaluate a program for correctness. [P4]	Students test and debug their own code and evaluate others' in the Reflection Questions.
5.3.2	Develop a correct program. [P2]	Programmed solutions to challenges must work.
5.3.3	Collaborate to solve a problem using programming. [P6]	Students develop solutions in teams.
5.4.1	Employ appropriate mathematical and logical concepts in programming. [P1]	Relationships such as "distance per wheel rotation" are important to making solutions work.
7.4.1	Connect computing within economic, social, and cultural contexts. [P1]	Reflection Questions ask students to make evaluative recommendations based on the impacts of robotic solutions in context.

Note. Reprinted from "Introduction to Programming LEGO Mindstorms EV3", by Carnegie Mellon's Robotic Academy. Copyrighted 2014. Reprinted with Permission.

APPENDIX B

OBSERVATION FORMS

Observation Form 1 of 2

Date: ____________ AR: Cycle #_____________ Time: ________________

This observation is ___ out of a set of _____ observations.

1.Participants in observation:

____

____

2. Description of the day's activity:

____

____

3. Time spent on task:

Minutes: __________ Seconds: ___________

4. Time spent using (KWL chart, flowchart, color coding) circle one.

1. Seconds: ___________ 6. Seconds: ___________

2. Seconds: ___________ 7. Seconds: ___________

3. Seconds: ___________ 8. Seconds: ___________

4. Seconds: ___________ 9. Seconds: ___________

5. Seconds: ___________ 10. Seconds: ___________

Frequency: _______________

Observation Form 2 of 2

Field notes

Descriptive notes | Reflection notes

---|---

---|---

---|---

Descriptive notes | Reflection notes

---|---

---|---

---|---

Descriptive notes | Reflection notes

---|---

---|---

---|---

Descriptive notes | Reflection notes

---|---

---|---

---|---

---|---

APPENDIX C

INTERVIEW QUESTIONS

Date: ____________ AR: Cycle #_____________ Time: ________________

This is interview ___ out of a set of _____ observations.

1. Interviewee identification: ____

2. Explain in your own words the purpose of (KWL chart, Flowchart, color coding) circle one?

3. How do you feel about using the (KWL chart, Flowchart, color coding) circle one?

4. What changes to the (KWL chart, Flowchart, color coding) circle one would you make to improve it? If any, Why?

5. Would you recommend this AT device to other students to use when learning to program? Explain why or why not?

APPENDIX D

FORMATIVE ASSESSMENT OF LEVEL OF UNDERSTANDING FOR OPEN-ENDED QUESTIONS

FORM 1 OF 4

Directions: The SOLO Taxonomy will be used to assess students understanding of the open-ended questions at the end of various lessons. Students will be evaluated from 1-5 for their answers to the questions.

Levels of SOLO Taxonomy

	Depth of knowledge	Level of Understanding
1	Pre-structural	Lack of knowledge or understanding; Shows minimal evidence of relevant learning.
2	Uni-structural	Answer satisfies one aspect of the learning task, but misses other important factors.
3	Multi-structural	Response fulfills various features, but does not address key issues.

4	Relational	Answer has multiple parts correlated and the correlation between the various aspects can be qualitatively explained (e.g., when and why).
5	Extended abstract	Answers show learning beyond the relational stage by exhibiting high-order thinking skills and applying knowledge to other context (transferrable knowledge).

Cycle 1		Move Straight 1 - Question 1	Move Straight 1 - Question 2	Move Straight 1 - Question 3		Move Straight 3 - Question 1	Move Straight 3 - Question 2	Move Straight 3 - Question 3		Move Straight 4 - Question 1	Move Straight 4 - Question 2	Move Straight 4 - Question 3	
#	Student Identification												
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													

16													

Cycle 2

#	Student Identification	Turning 1 - Question 1	Turning 1 - Question 2	Turning 1 - Question 3		Turning 3 - Question 1	Turning 3 - Question 2	Turning 3 – Question 3	Turning 3 – Question 4		Turning 4 - Question 1	Turning 4 - Question 2	
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													

Cycle 3		Touch Sensor 1 - Question 1	Touch Sensor 1 - Question 2	Touch Sensor 1 - Question 3		Touch Sensor 3 - Question 1	Touch Sensor 3 - Question 2			Touch Sensor 4 - Question 1	Touch Sensor 4 - Question 2		
#	Student Identification												
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													

APPENDIX E

SUMMATIVE ASSESSMENTS RESULTS

Students' results in each challenge will be indicated by PASS or FAIL.

	Student Identification	Sensabot Challenge KWL	Orchard Challenge Flowchart	Arm Position Challenge Visual Cue
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				

APPENDIX F

KWL CHART

Name:____ Date:_________________________

Topic:____

K What I know	W What I want to know	L What I learned

APPENDIX G

DAILY JOURNAL

Date:______________________ AR Cycle:_________ Time:______________

1. Description of the day's activity in class:

____

____

____

2.What challenges occurred during the day's actives?

____

____

____

3.What elements were a success during the day's activities?

____

____

____

4.What elements were a failure during the day's activities?

____

____

____

APPENDIX H

CHECKLIST MATRIX

Research question 1				Research question 2			Research question 3		
	Does my instruction improve when using the memory/organization instructional aid of KWL charts to teach computer programming using iteration with LEGO Mindstorms?			Does my instruction improve when using the memory/organization instructional aid of flowcharts to teach computer programming using iteration with LEGO Mindstorms?			Does my instruction improve when using the memory/organization instructional aid of visual cues to teach computer programming using iteration with LEGO Mindstorms?		
	What is the influence of KWL on students' performance?	What is the influence of KWL on students' feedback?	What is the observed influence of KWL with students?	What is the influence of flowcharts on students' performance?	What is the influence of flowcharts on	What is the observed influence of flowcharts	What is the influence of visual cues on students' performance?	What is the influence of visual cues on students' feedback?	What is the observed influence of visual cues with students?

					students' feedback?	with students?			

Student id									
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									

APPENDIX I

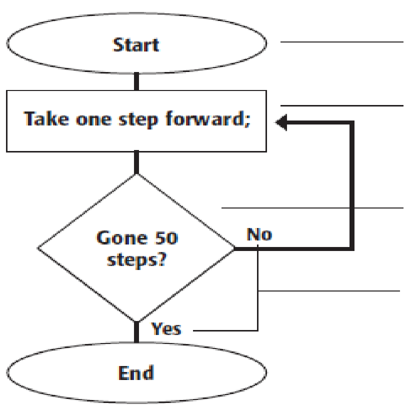
FLOWCHART INTRODUCTION

What are Flowcharts?

Robots need very detailed and organized instructions in order to perform their tasks. The programmer must break things down into simple behaviors and figure out when each behavior should run. A flowchart is a tool that can be used by programmers to determine program flow.

A flowchart provides a way of visually representing and organizing individual behaviors and decisions within a program -- it provides a diagram of the "flow" of the program. Programmers use flowcharts to lay out the steps that will be needed in their final program, and to help determine how the robot's behaviors should be broken down.

Parts of a Flowchart



Start of Program - Marks the beginning of the program, begin here. Follow the line to get to the next block.

Statement Block - A statement to execute, or a behavior to perform.

Decision Block - A decision point in your program. Ask a simple question, and do different things depending on the answer.

Yes/No - Answers to the question posed in the decision block. Follow the line labeled with the appropriate answer.

End of Program - Marks the end of the program. If you reach this point, the program is done!

Exercise

1. Make a flowchart organizing the "flow" of getting ready to go to school in the morning. Be sure to include the following steps in your chart, but don't be afraid to add other things if you need them!

Select something to wear	Look for your shoes	Put your shoes on
Take a shower	Brush your teeth	Hit snooze button
Eat breakfast	Put toast in the toaster	Get dressed
Walk or get a ride to school	Check your alarm clock	Comb your hair
Get out of bed	Turn on shower	Check the time

APPENDIX J

PSEUDOCODE MINI-CHALLENGE

1. Move robot forward
2. If (sensor is pressed) wait/pause
3. Backup from the wall 0.5 rotations
4. Turn left, 0.5 rotations (90 degrees) at 50% power
5. Repeat steps 1 – 4 three times.
6. Stop robot.

APPENDIX K

UNIVERSITY OF GEORGIA CONSENT FORM

EFFECTS OF USING CUSTOM MEMORY/ORGANIZATION INSTRUCTIONAL AID ON STUDENT LEARNING

Researcher's Statement
I am asking you to take part in a research study. Before you decide to participate in this study, it is important that you understand why the research is being done and what it will involve. This form is designed to give you the information about the study so you can decide whether to be in the study or not. Please take the time to read the following information carefully. Please ask the researcher if there is anything that is not clear or if you need more information. When all your questions have been answered, you can decide if you want to be in the study or not. This process is called "informed consent." A copy of this form will be given to you.

Principal Investigator: *Jason Turman*
Career and Information Studies
jturman@uga.edu

Purpose of the Study
I am conducting a research study to help me design custom instructional strategy to aid students learn computer programming for LEGO Mindstorms robots. I am asking you to be in the study because you are currently enrolled in the Digital Design course and will have the option to enroll in AP Computer Science Principles course in the near future. AP Computer Science Principles' curriculum requires students to be able to computer program and programming is known to be difficult for students to learn. If you agree to be in the study, you will perform tasks just as you would in a regular class. Notes will be collected and analyzed each day based on your reaction to the content and the rate of understanding assigned tasks. You may be asked questions to help clarify observations and data in an informal interview.

Study Procedures
If you agree to participate, you will be asked to work on the Introduction to Programming LEGO Mindstorms EV3 curriculum during class. I will use students' scores, daily observations made by myself, and informal interviews to help gather data to better understand how students learn. This study will take three weeks to complete. The data collected will be used to determine how to best design and implement instructional strategies for the benefit of students. The following data and information will be collected:

- Scores on questions
- Performance on computer programming with LEGO Mindstorms EV3 ROBOTS

- Questions for follow-up will consist of the following structure:
    - What is the purpose of KWL charts, flowcharts, and color-coding?
    - How do you feel about using KWL charts, flowcharts, and color-coding?
    - What changes would you make to improve the instructional strategy?
    - Would you recommend other students to use this instructional strategy?

Risks and discomforts
No risk or discomfort is anticipated as a result of participation in this study.

Benefits
This study will help determine how to design and implement custom learning strategies to aid students learning computer programming. Also, this study will allow students to influence the method with which the content is taught based on their experiences with the lesson. Furthermore, students may obtain a better understanding of computer programming.

Incentives for participation
Student involvement in this study is voluntary and you may choose to not participate or to stop at any time without penalty. Students' grades will not be helped or hindered for participation or lack thereof.

Privacy/Confidentiality
Students will be referenced by their Richmond County School issued student ID number and will not be shared externally. All identifiable data will only be used by the researcher and any documents with the student's ID number will be destroyed following the Richmond County School's procedures upon the completion of the study. The project's research may also be reviewed by a committee of professors at the University of Georgia (UGA). Identifiable results of this study will not be released to anyone other than the researcher and the UGA review committee without your written consent unless required by law.

Taking part is voluntary
Your involvement in this study is voluntary and you may choose to not participate or stop at any time without penalty or loss of benefits to which you are otherwise entitled. If you decide to withdraw from the study, the information that can be identified as yours will be kept as part of the study and may continue to be analyzed, unless you make a written request to remove, return, or destroy the information. Your participation in this study will not affect your grades or class standing.

If you have questions
The main researcher conducting this study is Jason Turman, a doctoral student at the University of Georgia. Please ask any questions you have now. If you have questions later, you may contact him at jturman@uga.edu or at 706-772-8140. If you have any questions or concerns regarding your rights as a research participant in this study, you may contact the Institutional Review Board (IRB) Chairperson at 706.542.3199 or irb@uga.edu.

Research Subject's Consent to Participate in Research:
To voluntarily agree to take part in this study, you must sign on the line below. Your signature below indicates that you have read or had read to you this entire consent form, and have had all of your questions answered.

___________________________	___________________________	
Name of Researcher	Signature	Date
___________________________	___________________________	
Name of Participant	Signature	Date

Please sign both copies, keep one and return one to the researcher.

APPENDIX L

ASSENT SCRIPT/FORM FOR PARTICIPATION IN RESEARCH:

EFFECTS OF USING CUSTOM MEMORY/ORGANIZATION INSTRUCTIONAL AID ON STUDENT LEARNING

We are doing a research study to determine what instructional strategy helps students program LEGO Mindstorms robots. We are asking you to be in the study because you are currently enrolled in the Digital Design course and will have the opportunity to enroll in AP Computer Science Principles in the near future. Computer programming is one of the skills used in AP Computer Science Principles. If you agree to be in the study, you will perform tasks just as you would in a regular class. Notes will be collected and analyzed each day based on your reaction to the content and speed of doing and understanding assigned tasks. You may be asked questions to help clarify observations and data in an informal interview.

The following data and information will be collected:

- Scores on questions
- Performances on programming with LEGO Mindstorms EV3 ROBOTS
- Questions for follow-up will consist of the following structure:
    - What is the purpose of KWL charts, flowcharts, and color-coding?
    - How do you feel about using KWL charts, flowcharts, and color-coding?
    - What changes would you make to improve the instructional strategy?
    - Would you recommend other students to use this instructional strategy?

You do not have to say "yes" if you don't want to. No one, including your parents, will be mad at you if you say "no" now or if you change your mind later. We have also asked your parent's permission to do this. Even if your parent says "yes," you can still say "no." Remember, you can ask us to stop at any time. Your grades in school will not be affected whether you say "yes" or "no."

Data gathered will be used to design custom memory/organization instructional aid to assist students in learning how to program LEGO Mindstorms robots. This study will be part of a dissertation for the University of Georgia and will be published. We will not use your name on any papers that we write about this project. We will only use a number so other people cannot tell who you are

You can ask any questions that you have about this study. If you have a question later that you didn't think of now, you can contact Jason Turman at jturman@uga.edu

---

Name of Child: ______________________________ **Parental Permission on File:** ☐ Yes ☐ No**
(If "No," do not proceed with assent or research procedures.)

(For Written Assent) **Signing here means that you have read this paper or had it read to you and that you are willing to be in this study. If you don't want to be in the study, don't sign.**

Signature of Child: ________________________________ **Date:** ___________

(For Verbal Assent) **Indicate Child's Voluntary Response to Participation:** ☐ Yes ☐ No

Signature of Researcher: ___________________________ **Date:** ___________

APPENDIX M

DAILY SCHEDULE CHECKLIST

Cycle	Lesson	Week	Day M	T	W	R	F	Completed Initials
1	Moving Straight 1	1	X					
1	Moving Straight 2	1	X					
1	Moving Straight 3	1		X				
1	Moving Straight 4	1			X			
1	Moving Straight 5	1				X		
1	Sensabot Challenge	1				X	X	
1	Reflection	2	X	X	X	X	X	
2	Turning 1	3	X					
2	Turning 2	3	X					
2	Turning 3	3		X				
2	Turning 4	3			X			
2	Turning 5	3				X		
2	Orchard Challenge	3				X	X	
2	Reflection	4	X	X	X	X	X	
3	Touch Sensor 1	5	X					
3	Touch Sensor 2	5	X					
3	Touch Sensor 3	5		X				
3	Touch Sensor 4	5			X			
3	Touch Sensor 5	5				X	X	
3	Arm Position Challenge	5				X	X	
3	Reflection	6	X	X	X	X	X	

APPENDIX N

OPEN-ENDED QUESTIONS CONVERSION

Lesson	Multiple Choice Questions	Open-Ended Questions
Move Straight 1	1. Why is it important to inspect industrial facilities often? • Frequent inspections keep facilities safe • Problems are easier • Industrial facilities require inspects by law • All of the above	Why is it important to inspect industrial facilities often?
	2. What is the advantage of Sensabot over human inspectors? • Sensabot can inspect much faster • Sensabot can enter hazard areas • Sensabot has wheels instead of legs • There is no big advantage	What is the advantage of Sensabot over human inspectors?
	3. In addition to basic movement, what specific skill will you need to complete this challenge? • Operate the small motor (arm) • Move a specific distance • Control speed of the motors • All of the above	In addition to basic movement, what specific skill will you need to complete this challenge?
Move Straight 3	1. What does the robot do when the program is run? • Move forward until its wheels have turned 1 rotation • Move forward until its wheels have turned 3 rotations • Move backward • Turn to the left	What does the robot do when the program is run?

	2. How do you run a program that has been downloaded to the EV3? • File>Run on the EV3 brick • My Files>Software Files>Run on the EV3 brick • “File Navigation” tab>Project name> Program on the EV3 brick • Program can only be run from the PC or Mac	How do you run a program that has been downloaded to the EV3?
Move Straight 4	1. What does the Medium Motor block do? • Controls both of the large motors • Controls the medium motor like the one on the arm • Controls the ultrasonic sensor • Controls all the motors	What does the Medium Motor block do?
	2. What happens when you use a negative power level? • The motor runs backwards • The robot moves faster • The motor does nothing • The program crashes	What happens when you use a negative power level?
	3. What happens when you put more than one block in a program? • The program runs backward • The program runs in sequence • The program doesn’t run at all • The program runs the largest motor block first	What happens when you put more than one block in a program?
	4. Does the program ever end? • The program keeps running • The program ends immediately	Does the program ever end?

	5. Does the second medium motor block ever get to run? • The second motor block never runs • The second motor block is skipped over	Does the second medium motor block ever get to run?
	6. What happens if a block cannot complete its action? • The program will immediately skip to the next block • The program will try for a while, then display an error and quit • The program will try for a while, then skip the "stuck" block and move • The program will get "stuck" trying to complete the action, and later blocks will never be run	What happens if a block cannot complete its action?
Turning 1	1. Why is it important to be able to drive through an orchard? • To perform specialized tasks to different types of crops • GPS is required while navigating around the orchard • To perform tasks like inspection and spraying which cannot be done as effectively through other means • All of the above	Why is it important to be able to drive through an orchard?
	2. What is the advantage of the Autonomous Tractor over a human driver? • Reduces the need for humans to perform the repetitive task of driving through the orchard over and over • Reduces exposing human to hazard areas while performing inspections • Autonomous Tractor can travel through an area where a human driver may get lost • There is no big disadvantage	What is the advantage of the Autonomous Tractor over a human driver?
	3. In addition to basic turning, what additional, new knowledge will help you complete this challenge? • How a robot moves straight	In addition to basic turning, what additional, new knowledge will help you complete this challenge?

	• How a robot moves back and forth • How a robot turns and different types of turns • All of the above	
Turning 3	1. What does the robot do when the TurnRight program is run? • Move straight forward • Spin to the robot's right without moving forward at all • Spin to the robot's left without moving forward at all • Spin for 360 degrees	What does the robot do when the TurnRight program is run?
	2. TRUE or FALSE: With "Rotations" on the Move Steering Block to 1, the whole robot rotates 1 time. • TRUE: the robot will turn around 1 time • FALSE: the wheels will turn 1 time, not the body	What happens when "Rotations" on the Move Steering is set to 1?
	3. How much did the robot's wheel turn during this movement? • 1 rotation • 1 degree • Enough to make the robot spin completely around one time • One lap around the table	How much did the robot's wheel turn during this movement?
	4. What does the "1 rotation" refer to in the Move Steering Block's controls? • 1 full rotation of the robot's body during a turn • 1 rotation of the robot's wheels • 1 time that the robot is picked up and turned around • 1 rotation of the Earth and its axis	What does the "1 rotation" refer to in the Move Steering Block's controls?
Turning 4	1. In the movement you programmed, the left motor was told to move forward at 50% power, and the right motor was told to…? • Move forward at 50% power	In the movement you programmed, the left motor was told to move forward at 50% power, and the right motor was told to…?

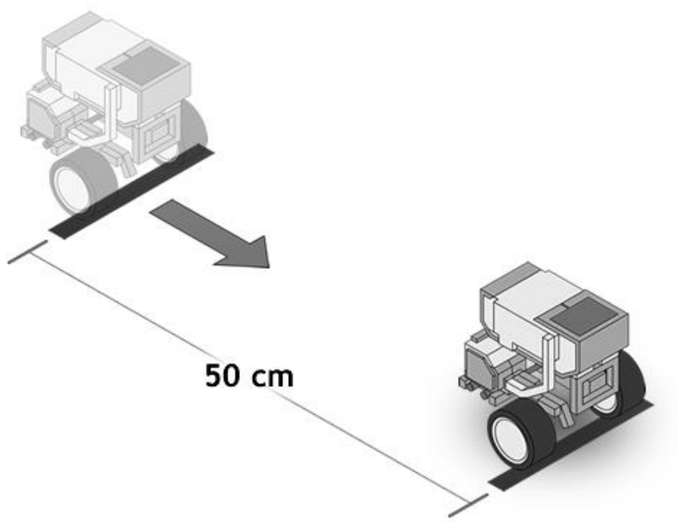
	• Move backwards at 50% • Stay in place • Spin freely	
	2. What kind of turn did the robot produce with one motor running and one motor stopped? • Goes straight • Turns in place • Turns “wide” • Backs up	What kind of turn did the robot produce with one motor running and one motor stopped?
Touch Sensor 1	1. Why are sensors important to robots? • They allow multiple commands to run in order • They give the robot information about its surrounds • They allow robots to repeat similar tasks • All of the above	Why are sensors important to robots?
	2. What is the advantage of Sensor Control over Sequential Commands? • The robot can remember hazard areas • The robot can perform actions a lot faster • The robot can react to its environment • There is no big advantage	What is the advantage of Sensor Control over Sequential Commands?
Touch 3	1. What does the robot do when the WaitTouch program runs? • Runs continuously until the Touch Sensor is pressed in • Waits for 1 second, then moves 1 rotation • Waits for the Touch Sensor to be pressed in, then moves 1 rotation • Runs for 1 rotation	What does the robot do when the WaitTouch program runs?
	2. The program waits BEFORE it moves because…?	Why does the program wait BEFORE it moves?

	• The Wait Block comes first in the program • The Wait Block always takes priority over Move Blocks	
Touch 4	1. What does a Move command do when its Mode is set to "On"? • Turn the motors on • Turn the motors on for a certain number rotations • Turn the motors on until Touch Sensor is triggered • Combines with the next block to make a special command What does a Move command do when its Modes is set of "Off"? • Turns the motors off • Waits for the Touch Sensor to be pressed • Wait for the Touch Sensor to be pressed, then turn the motors off • End the program	What does a Move command do when its Mode is set to "On"? What does a Move command do when its Modes is set of "Off"?

APPENDIX O

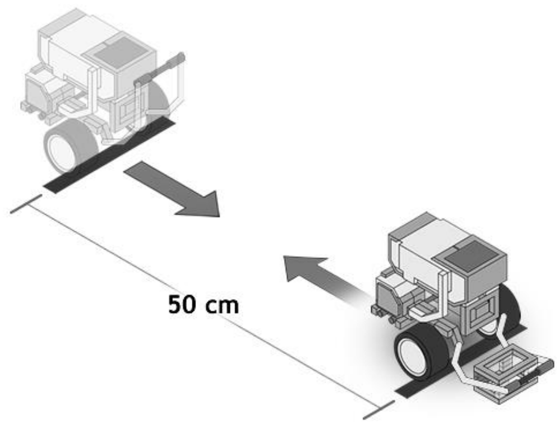
MINI-CHALLENGES

Cycle	Lesson	Instructions
1	Move Straight 3 – 50 cm challenge	Program robot to move 50 cm

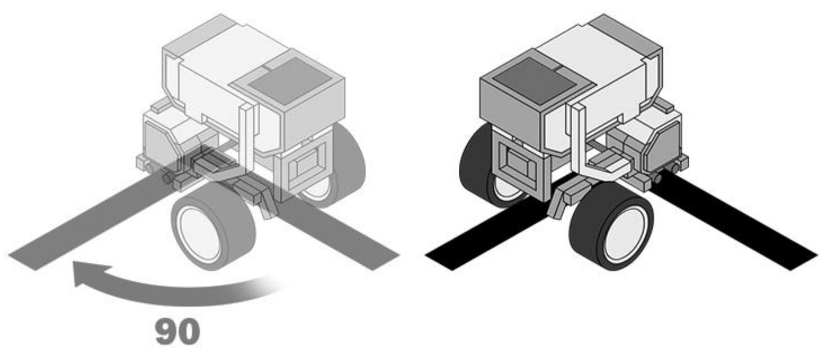


Cycle	Lesson	Instructions
1	Move Straight 4 – Cargo Retrieval	Program robot to: • Raise its arm • Move 50 cm to the box • Drop the arm down • Back up to robot's starting position [1]

[1] Note. Reprinted from "Introduction to Programming LEGO Mindstorms EV3", by Carnegie Mellon's Robotic Academy. Copyrighted 2014. Reprinted with Permission.



Cycle	Lesson	Instructions
3	Turning 3 – 90 degrees turn	Program you robot to turn exactly 90 degrees to its right

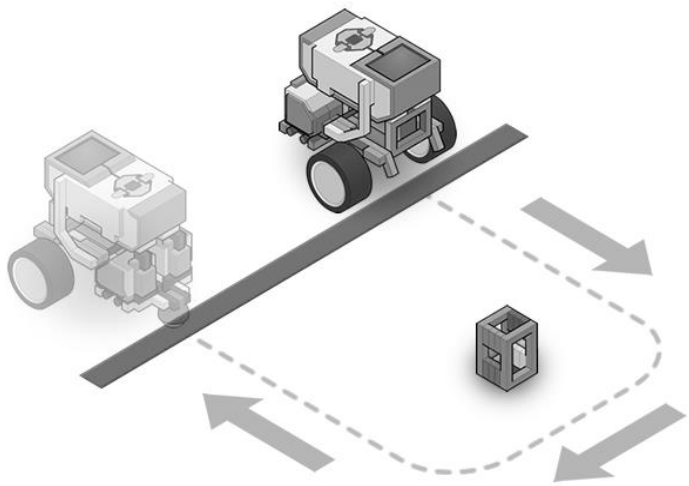


[2]

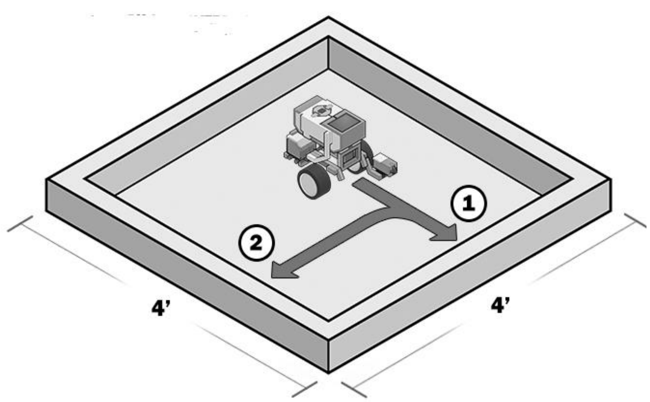
Cycle	Lesson	Instructions

[2] Note. Reprinted from "Introduction to Programming LEGO Mindstorms EV3", by Carnegie Mellon's Robotic Academy. Copyrighted 2014. Reprinted with Permission.

3	Turning 4 – Dizzy Drills	Program your robot to run out to an obstacle, go around it, then come back



Cycle	Lesson	Instructions
5	Touch 4 – Mini vacuum	Program the robot to touch all four walls of a room, using its Touch Sensor to know when it has reached each one [3]

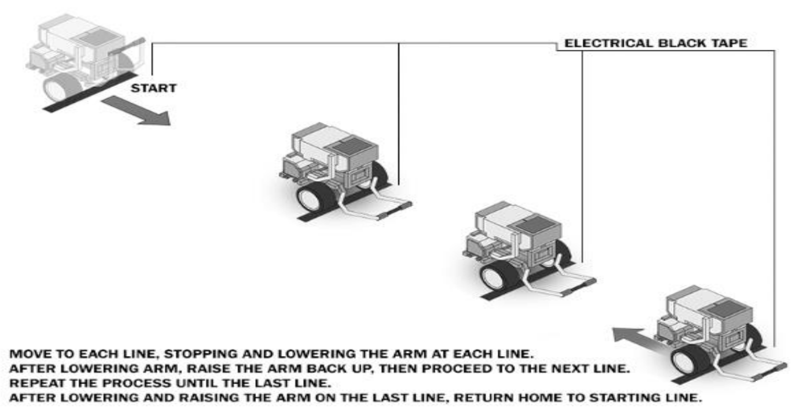


[3] Note. Reprinted from “Introduction to Programming LEGO Mindstorms EV3”, by Carnegie Mellon’s Robotic Academy. Copyrighted 2014. Reprinted with Permission.

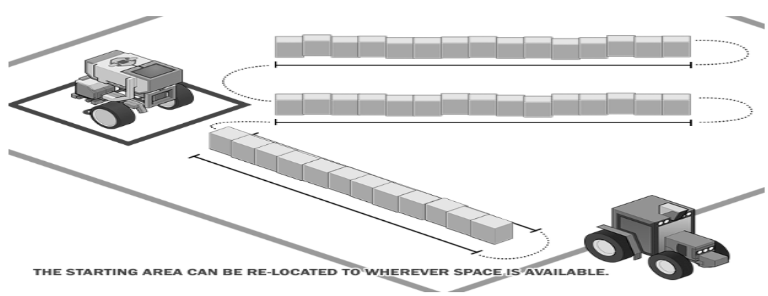
APPENDIX P

SUMMATIVE CHALLENGES

Cycle	Lesson	Instructions
1	Sensabot	In this challenge, you will program your EV3 robot to move from its starting box to three different lines on a game board, stopping at each one to perform an inspection, represented by lowering and raising the robot's arm. When the robot is done inspecting all three locations, it should back up and return home to its starting box to recharge.

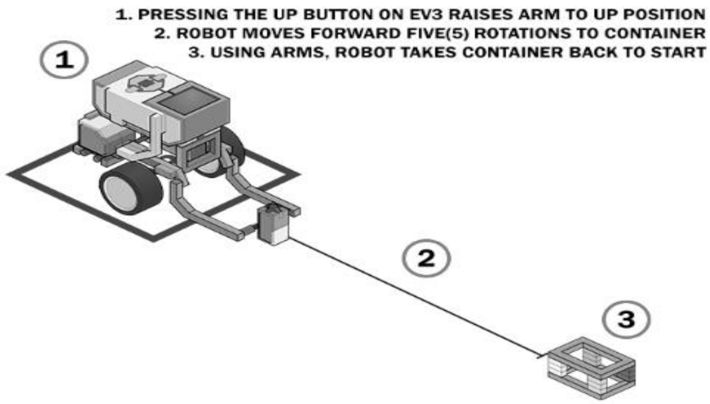


Cycle	Lesson	Instructions
3	Orchard	In this challenge, you will program your EV3 robot to move from its starting area through three rows of fruit trees. You may choose your own path through the orchard, but the robot must pass along both sides of each row.



Cycle	Lesson	Instructions

5	Arm Position	In this challenge, you will program your EV3 robot to raise its arm when pressing the Up button on the EV3, then retrieving a container and bringing it back to the station location.



APPENDIX Q

STUDENTS' PRIOR EXPERIENCE AND KNOWLEDGE

1. Have you ever used a KWL chart before? **If yes, go to Question 2, if no go to Question 6.**
2. When did you use a KWL chart before?
3. What would you say is your level of expertise of using KWL charts? (circle one)
   (a) beginner
   (b) average
   (c) expert.
4. Have you used a KWL chart in high school? If so, what subject
5. Did you find the KWL chart helpful?
6. Have your ever-used LEGO Mindstorms robot before? **If yes go to Questions 7, if no go to Questions 10.**
7. When did you use LEGO Mindstorms?
8. What was the purpose of using the LEGO Mindstorms?
9. What would you say is your level of expertise of using LEGO Mindstorms (circle one)?
   (a) beginner
   (b) average
   (c) expert
10. Have you ever done programming at school before? **If yes, go to Question 11.**
11. When did you do programming at school?
12. What would you say is your level of expertise of programming? (circle one)
    (a) beginner
    (b) average
    (c) expert

APPENDIX R

CYCLE 1 POST INTERVIEW

1. Explain the objective of today's challenge.
2. Did you successfully complete the challenge? (circle one) Yes or No
3. Estimate how many times you referred back to the KWL chart after you started the challenge?
4. What does KWL stand for?
5. In your own words, what is the purpose of the KWL chart?
6. What are the benefits of using the KWL chart for this challenge, if any?
7. What were the disadvantages of using the KWL chart for this challenge, if any?
8. Explain if the KWL chart was optional if you would use it or not for this challenge?
9. What subjects do you think the KWL would be useful in?
10. Do you think you will use the KWL chart again (circle one)? Yes or No
11. Explain: If a future challenge required several steps would that influence your decision to use a KWL chart? For instance, if you had to program the robot to do 10 different things.

APPENDIX S

K(WL)^{N} CHART

Topic____			
What I know____ ____			
What I want to know		**What I learned**	
1		1	
2		2	
3		3	
4		4	
5		5	
6		6	
7		7	
8		8	
9		9	

APPENDIX T

FLOWCHART PREASSESSMENT

1. Have you ever used a flowchart?
2. When did you use a flowchart?
3. What would you say is your level of expertise of using a flowchart?

APPENDIX U

CYCLE 2 POST INTERVIEW

1. Explain the objective of today's challenge.
2. Did you successfully complete the challenge (circle one)? Yes or No
3. What is the purpose of a flowchart?
4. Estimate how many times you referred back to the flowchart after the challenge started.
5. What do you think are the benefits of using a flowchart?
6. What do you think are some disadvantages of using a flowchart?
7. How could you improve your flowchart?

APPENDIX V

COLOR-CODED VISUAL CUES PREASSESSMENT

1. Have you used visual cues (color coding) in class before?
2. If you have used visual cues (color coding) in class before, in what subjects did you use them?
3. What would you say is your level of understanding of visual cues (color coding)?

APPENDIX W

CYCLE 3 POST INTERVIEW

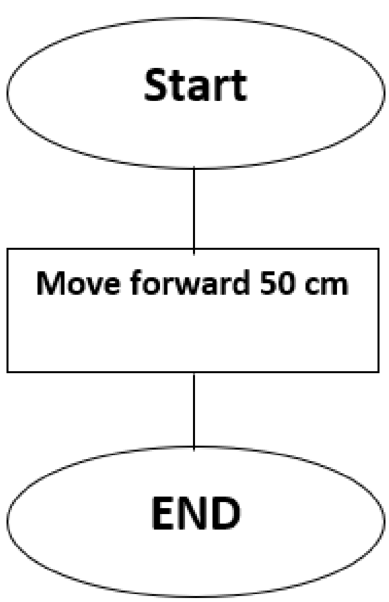
1. Explain the objective of today's challenge.
2. Did you successfully complete the challenge (circle one)? Yes or No
3. What is the purpose of the color-coding visual cue?
4. Estimate how many times you referred back to the color-coded instructions after the challenge started.
5. What do you think are the benefits of using color-coded instructions?
6. What do you think are some disadvantages of using color-coded instructions?
7. How could you improve using color-coded instructions?

APPENDIX X

COMPUTER PROGRAMMING FLOWCHART DESIGN GUIDE

Directions: Use this design guide to construct flowcharts for beginners learning computer programming. The shapes used in this guide is limited to three symbols. Each flowchart will begin with the **START OF PROGRAM** block symbol and finish with the **END OF PROGRAM** block symbol. The **STATEMENT BLOCK** symbol is used to represent one action and can be used multiple times to demonstrate various actions. The example below shows a flowchart designed to show program to move 50 cm.

Parts of a Flowchart | Description



START OF PROGRAM – The beginning of the program. Follow the line to the next block.

Statement Block – One action to perform.

END OF PROGRAM – The end of the program.

APPENDIX Y

COLOR-CODING GUIDE WITH LEGO MINDSTORMS EV3

Directions: This guide is designed to assist beginners in learning how to program LEGO Mindstorms EV3 robots. Instructions are color-coded to help the programmer identify the type of programming block that is needed. Additionally, instructions are separated into individual robot actions and are numbered to help the programmer focus on one task at a time. See the example below.

1. **Robot is waiting for the "Up" button to be pressed**
2. Robot raises its arm because the "Up" button was pressed
3. **Robot waits for the sensor to be pressed**
4. Robot moves forward to the cargo
5. Robot lowers its arm
6. Robot moves backward dragging the block with it to the Start point